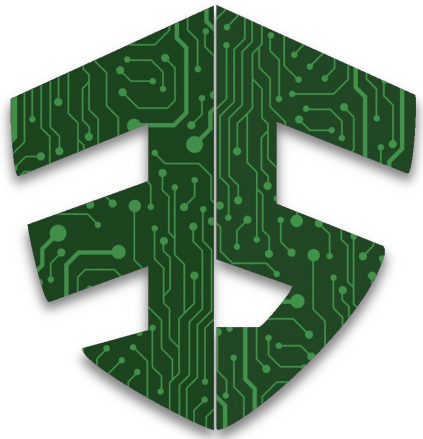


VALU3S

Verification of multiple models of a safety-critical motor controller in railway systems

RSSRail 2022, Paris

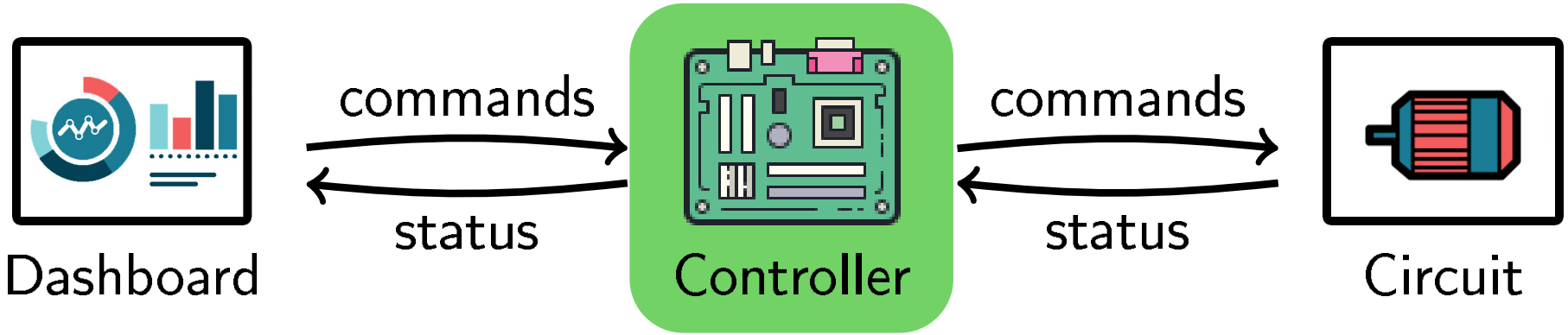
José Proença (ISEP), Sina Borrami (Alstom), Jorge Sanchez de Nova (Alstom), David Pereira (ISEP), Giann Nandi (ISEP)

1 June 2022

Public

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876852. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Czech Republic, Germany, Ireland, Italy, Portugal, Spain, Sweden, Turkey.

Verification of a motor controller in signalling systems



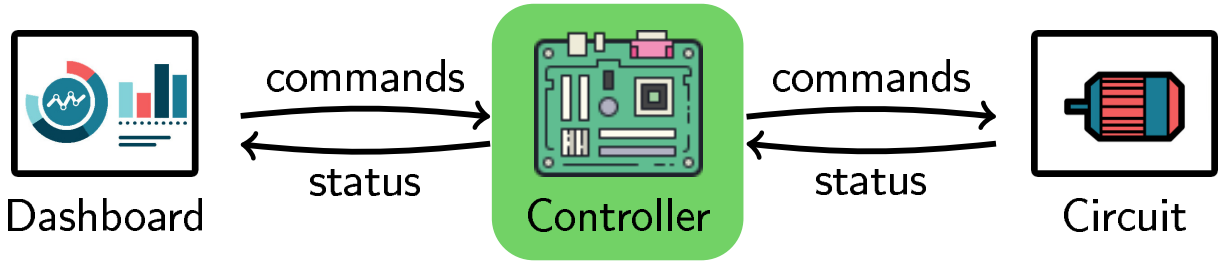
Development team

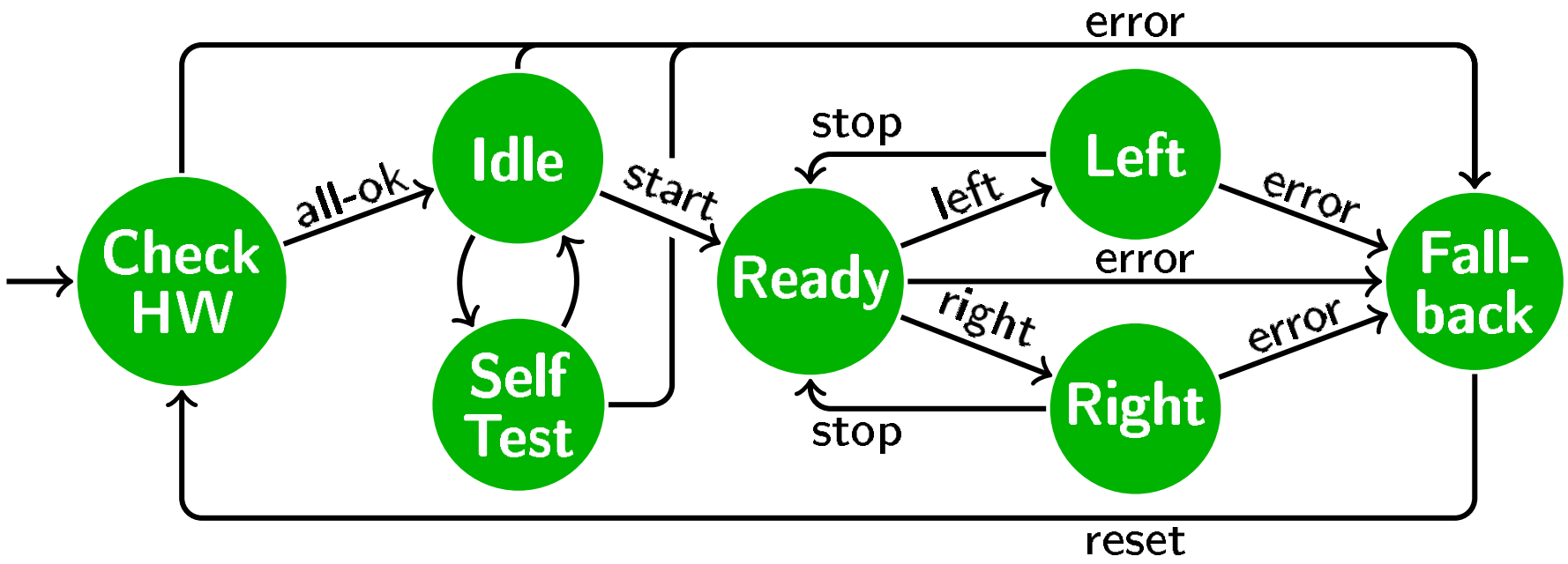
Verification team

ALSTOM

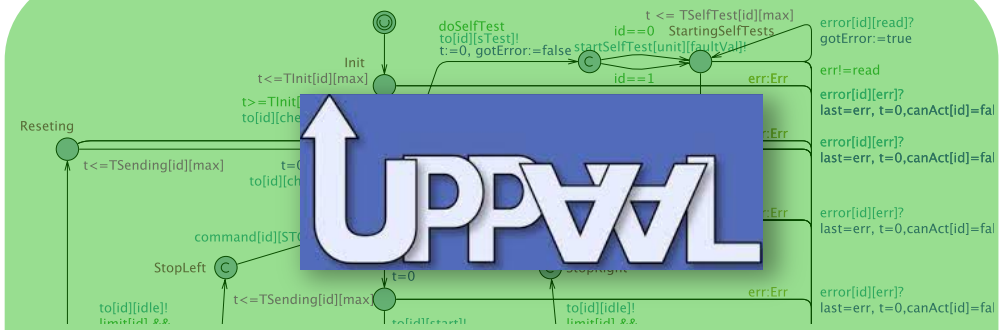
isep

Verification of a motor controller in signalling systems





Overview of this talk



1. Model **behaviour** in UPPAAL model checker

State	Trigger	Comp.	Expected
controller$_1$ is ready	decoder receives a `left` command	controller$_1$	send a `left` command within 100ms
	monitor$_1$ or reader$_1$ fail	controller$_2$	go to a fallback state within 100ms

2. Specify **requirements** (temporal formula)

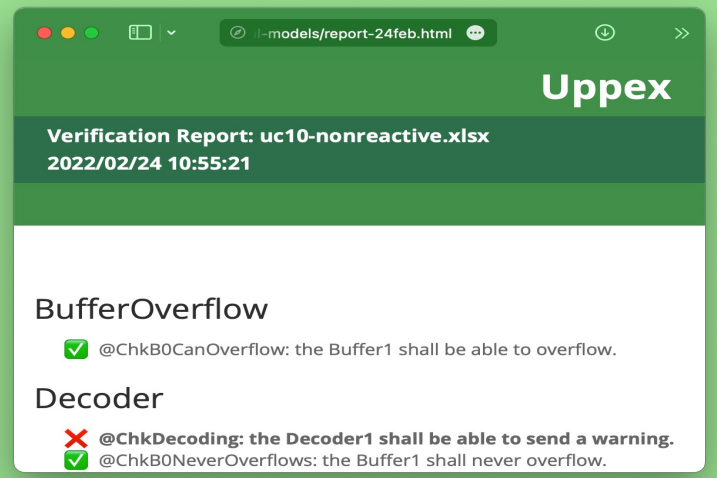
3. Configure **instances** of the models and requirements in Excel

1	Configuration	Heartbeats	SyncMon	SyncDec	ReadCircuit	SelfTesting	StartWithSel	ShortInj	StopA
3	Monitor		x						
4	Decoder			x					
5	JustHeartBeat	x							
6	SelfTest				x	x			

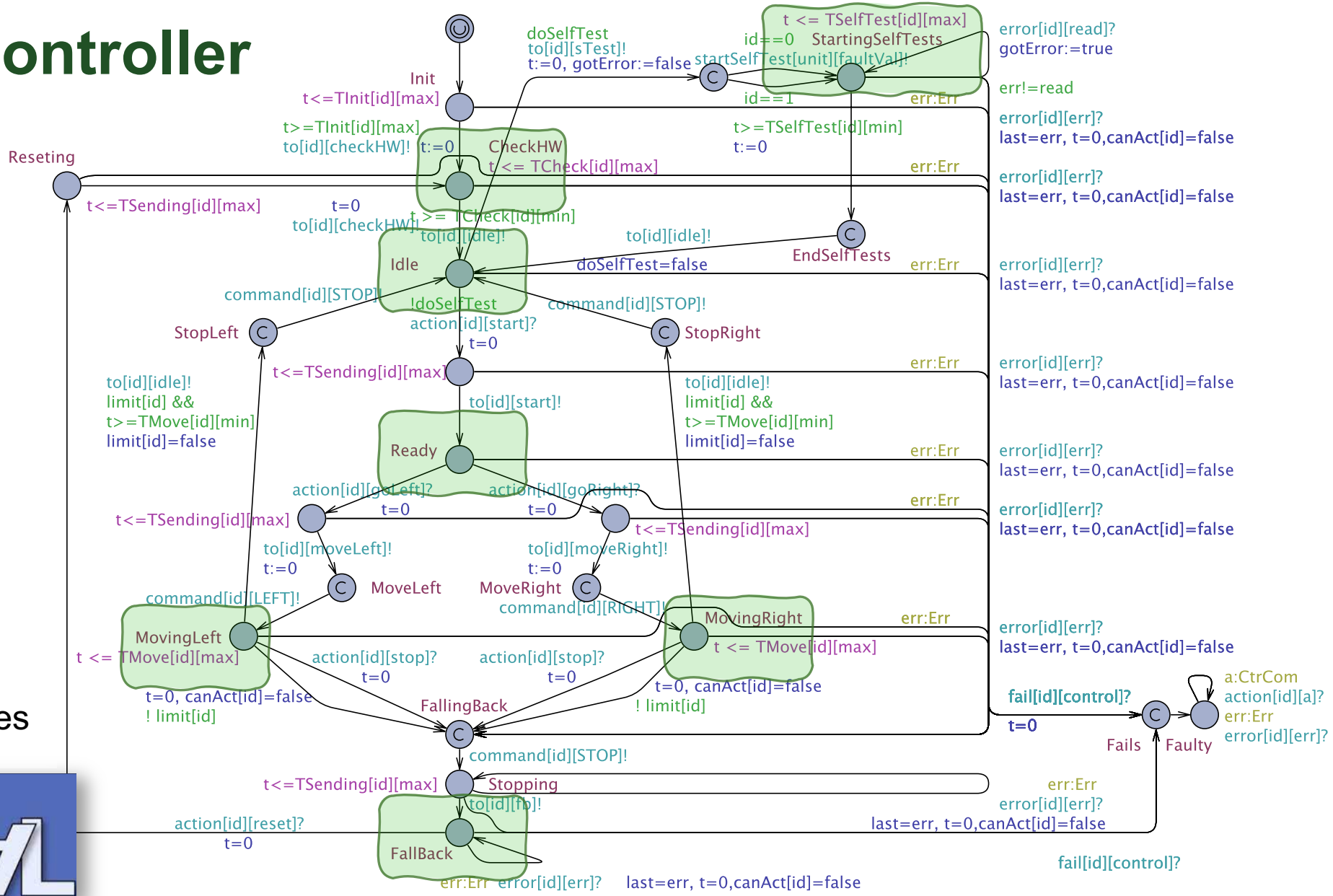
@Configurations @Scenarios

4. Verify **all** instances and **all** requirements in one go

Uppex



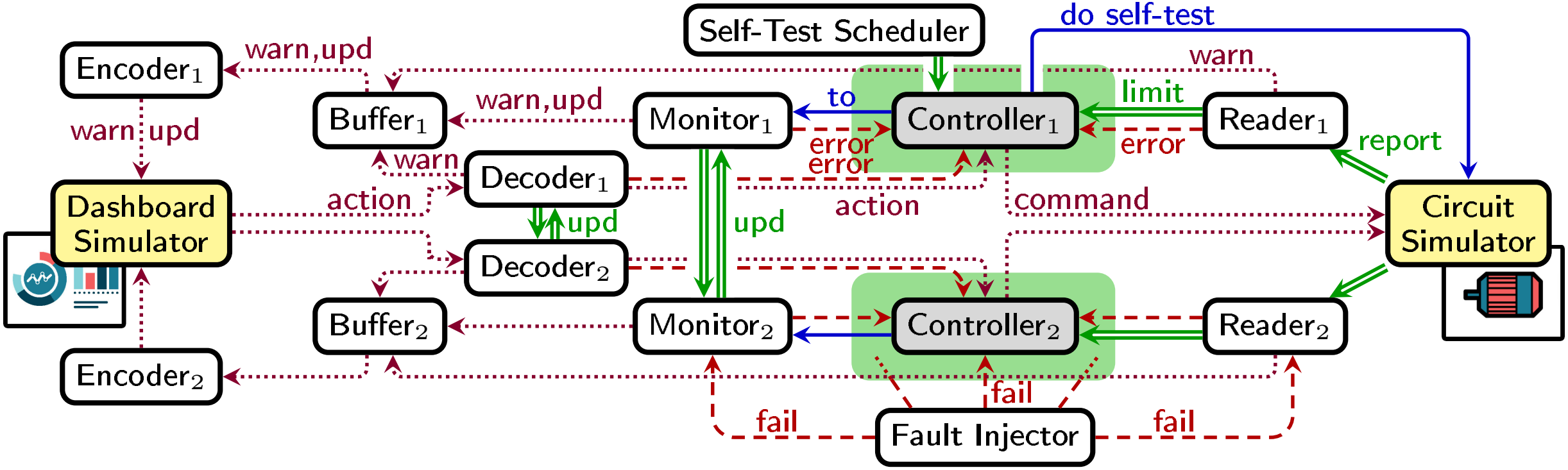
The Controller



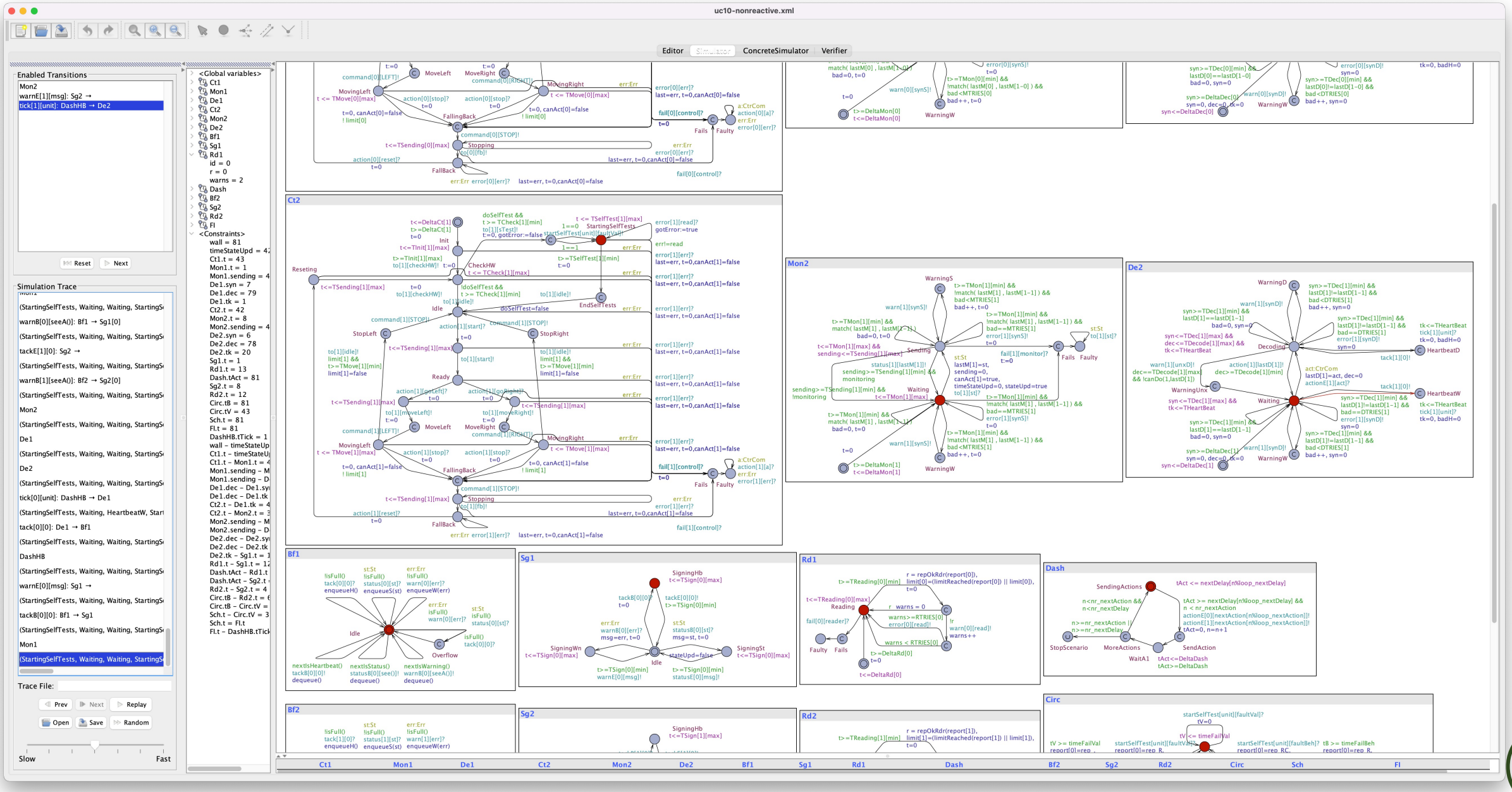
Model-checker of

Real-time properties

Component architecture



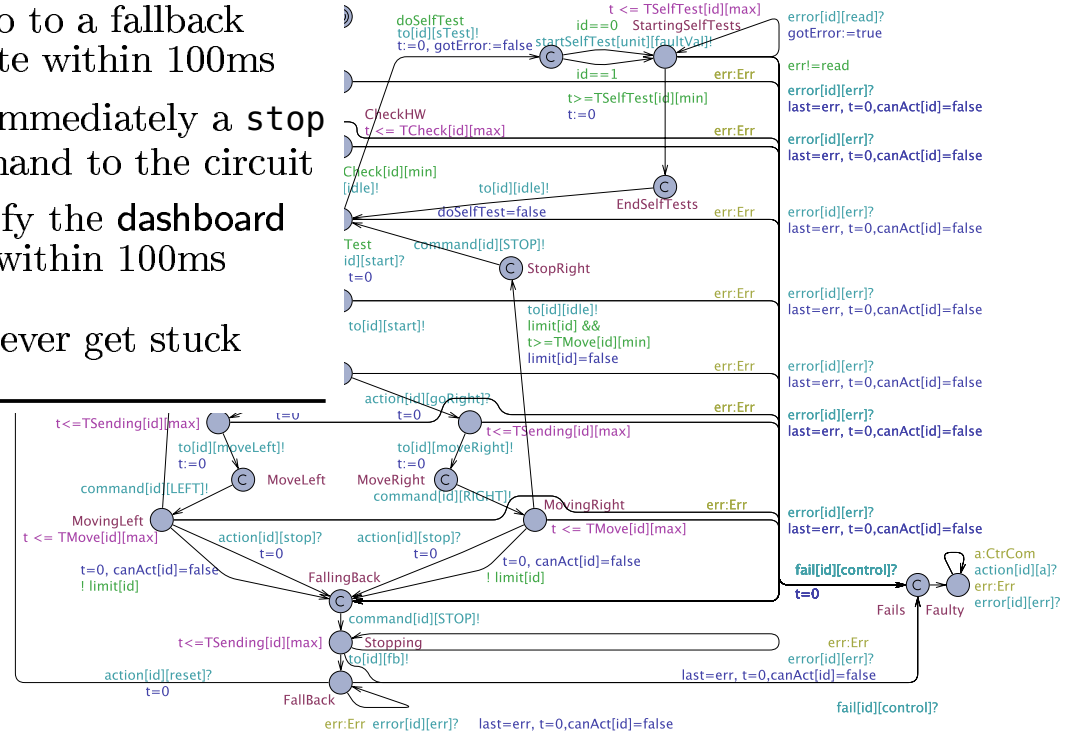
16x Automata


uc10-nonreactive.xml
Editor
Simulator
ConcreteSimulator
Verifier
Enabled Transitions
Mon2
warnE[1][msg]: Sg2 →
tick[1][unit]: DashHB → De2
Reset
Next
Simulation Trace
(StartingSelfTests, Waiting, Waiting, StartingS
warnB[0][seeA()]: Bf1 → Sg1[0]
(StartingSelfTests, Waiting, Waiting, StartingS
tackE[1][0]: Sg2 →
(StartingSelfTests, Waiting, Waiting, StartingS
warnB[1][seeA()]: Bf2 → Sg2[0]
(StartingSelfTests, Waiting, Waiting, StartingS
Mon2
(StartingSelfTests, Waiting, Waiting, StartingS
De1
(StartingSelfTests, Waiting, Waiting, StartingS
De2
(StartingSelfTests, Waiting, Waiting, StartingS
tick[0][unit]: DashHB → De1
(StartingSelfTests, Waiting, HeartbeatW, Start
tack[0][0]: De1 → Bf1
(StartingSelfTests, Waiting, Waiting, StartingS
DashHB
(StartingSelfTests, Waiting, Waiting, StartingS
warnE[0][msg]: Sg1 →
(StartingSelfTests, Waiting, Waiting, StartingS
tackB[0][0]: Bf1 → Sg1
(StartingSelfTests, Waiting, Waiting, StartingS
Mon1
(StartingSelfTests, Waiting, Waiting, StartingS
Trace File:
Prev
Next
Replay
Open
Save
Random
Slow
Fast
<Global variables>
Ct1
Mon1
De1
Ct2
Mon2
De2
Bf1
Sg1
Rd1
id = 0
r = 0
warns = 2
Dash
Bf2
Sg2
Rd2
Fl
<Constraints>
wall = 81
timeStateUpd = 42
Ct1.t = 43
Mon1.t = 1
Mon1.sending = 4
De1.syn = 7
De1.dec = 79
De1.tk = 1
Ct2.t = 42
Mon2.t = 8
Mon2.sending = 4
De2.syn = 6
De2.dec = 78
De2.tk = 20
Sg1.t = 1
Rd1.t = 13
Dash.tAct = 81
Sg2.t = 8
Rd2.t = 12
Circ.tB = 81
Circ.tV = 43
Sch.t = 81
Fl.t = 81
DashHB.tTick = 1
wall – timeStateUp
Ct1.t – timeStateU
Ct1.t – Mon1.t = 4
Mon1.sending – M
Mon1.sending – D
De1.dec – De1.sy
De1.dec – De1.tk
Ct2.t – De1.tk = 4
Ct2.t – Mon2.t = 3
Mon2.sending – M
Mon2.sending – D
De2.dec – De2.sy
De2.dec – De2.tk
De2.tk – Sg1.t = 1
Rd1.t – Sg1.t = 12
Dash.tAct – Rd1.t
Dash.tAct – Sg2.t
Rd2.t – Sg2.t = 4
Circ.tB – Rd2.t = 6
Circ.tB – Circ.tV =
Sch.t – Circ.tV = 3
Sch.t = Fl.t
Fl.t – DashHB.tTick
Ct2
Mon2
De2
Bf1
Sg1
Rd1
Dash
Circ
Ct1
Mon1
De1
Bf2
Sg2
Rd2
Sch
Fl

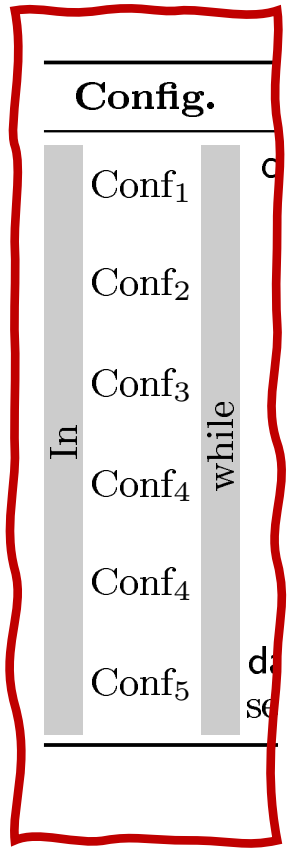
Model = Requirements + Network of Automata

Config.		State	Trigger	Comp.	Expected
In	$Conf_1$	controller$_1$ is ready	decoder receives a `left` command	controller$_1$	send a `left` command within 100ms
	$Conf_2$		monitor$_1$ or reader$_1$ fail	controller$_2$	go to a fallback state within 100ms
	$Conf_3$		controller$_1$ fails	controller$_2$	go to a fallback state within 100ms
	$Conf_4$		controller$_1$ receives an error message	controller$_1$	send immediately a `stop` command to the circuit
	$Conf_4$		controller$_1$ receives an error message	encoder$_1$	notify the dashboard within 100ms
	$Conf_5$	dashboard can send messages		full system	never get stuck

while | when | the | shall



Examples of Configurations



Configuration 1

- The motor takes exactly 4.5s to move left or right (OK)
- The dashboard starts at 2s, asks to move left at 5s, and asks to move right at 10s
- No fault is injected

Configuration 2

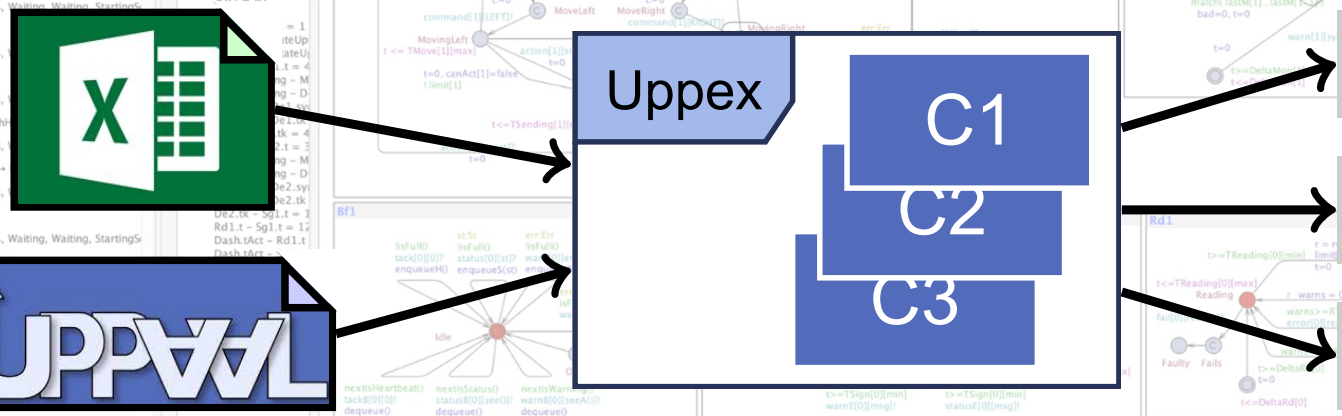
- The motor takes 6s to move left (not OK)
- (rest as Conf. 1)

Configuration 3

- The monitor1 components becomes faulty after 5s
- Buffer is smaller
- Heartbeats are off
- (rest as Conf 1.)

Uppex: Challenges and Workflow

- Large model that can be refined
- Variability (unfixed parameters)
- Understanding & Maintenaince
- Developers + Modellers



Apply a configuration

Verify 1 configuration

Verify all configurations

https://cister-labs.github.io/uppex

Demo: A look into the configurations

const int T$Name[Ids][Intrv] = {{$Min-1,$Max-1},{$Min-2,$Max-2}};

Name	Min-1	Max-1	Min-2	Max-2	Comment	Features
Init	50	50	70	70	control: tim	
Check	100	100	100	100	control: max	
SelfTest	0	0	0	0	time to run	
SelfTest	200	200	200	200	time to run	Sel

▶ @Global | @Local | @TimeBound

<query> <formula>$Formula</formula> <comment>$Comment</comment></query>

Formula	Features	While	When	Who
A[] (not deadlock) || Dash.StopScer	ChckDeadlock	Dashboard can send		full system
(Ct1.Ready && De1.dec==0 && lastI	Scn1	Controller1 is ready	Decoder receives a GOLEFT	Circuit
Mon1.Fails --> (Ct2.FallBack && Mo	FailMon10		Monitor1 fails	Controller2

▶ @Configurations | @Scenarios | <queries> | @Global | +

1	Configuration	Heartbeats	SyncMon	SyncDec	ReadCircuit	SelfTesting	StartWithSel	ShortInj	StopAtMon	SmallBuffer	Scn1	Scn2	Scn3	Scn4	ChckDeadlock	ChkDecoding	ChkC0CanErr	ChkB0CanOve	ChkB0NeverC	ChkRd
3	Monitor		x									x			x		x		x	
4	Decoder			x								x			x	x	x		x	
5	JustHeartBeat	x												x		x	x	x		
6	SelfTest				x	x	x		x					x	x					x

◀ ▶ @Configurations | @Scenarios | <queries> | @Global | @Local | @TimeBounds | @DataT

Wrap up

1. **Annotate** Uppaal model

2. **Configure** annotations in Excel

Uppex

3. **Intantiate** & **Verify** many configurations

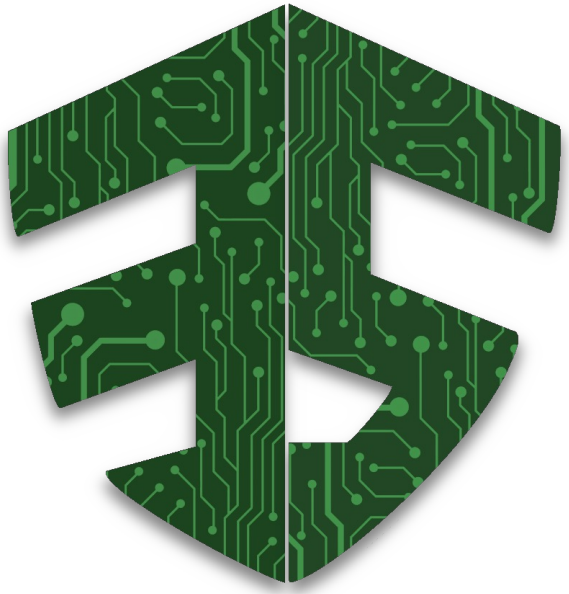
Development team

Verification team

ALSTOM

isep

VALU3S



VALU3S

Verification and Validation of Automated Systems' Safety and Security

www.valu3s.eu

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876852. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Czech Republic, Germany, Ireland, Italy, Portugal, Spain, Sweden, Turkey.