



Notations for Reactive Systems

Venkatesh (Venky)

July 31, 2013

Motivation

- Wider acceptance of formal methods requires good usable notations

What good are our formal specs if our engineers and certification authorities cannot decipher them?

- Lee Pike

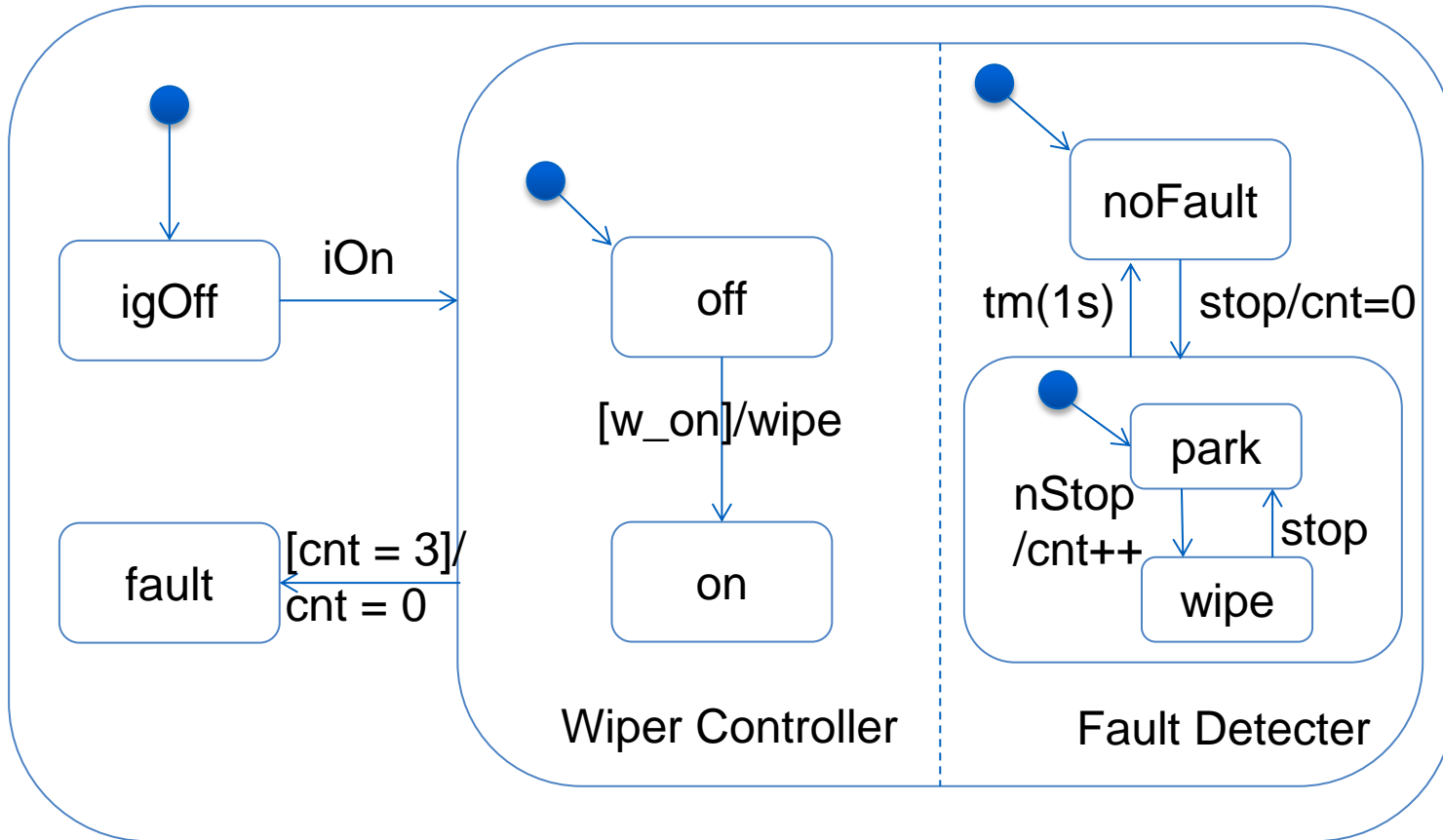
- Very few researchers
 - Harel an exception
- Try and highlight *usable* aspect of notations

Sample Automotive Requirements

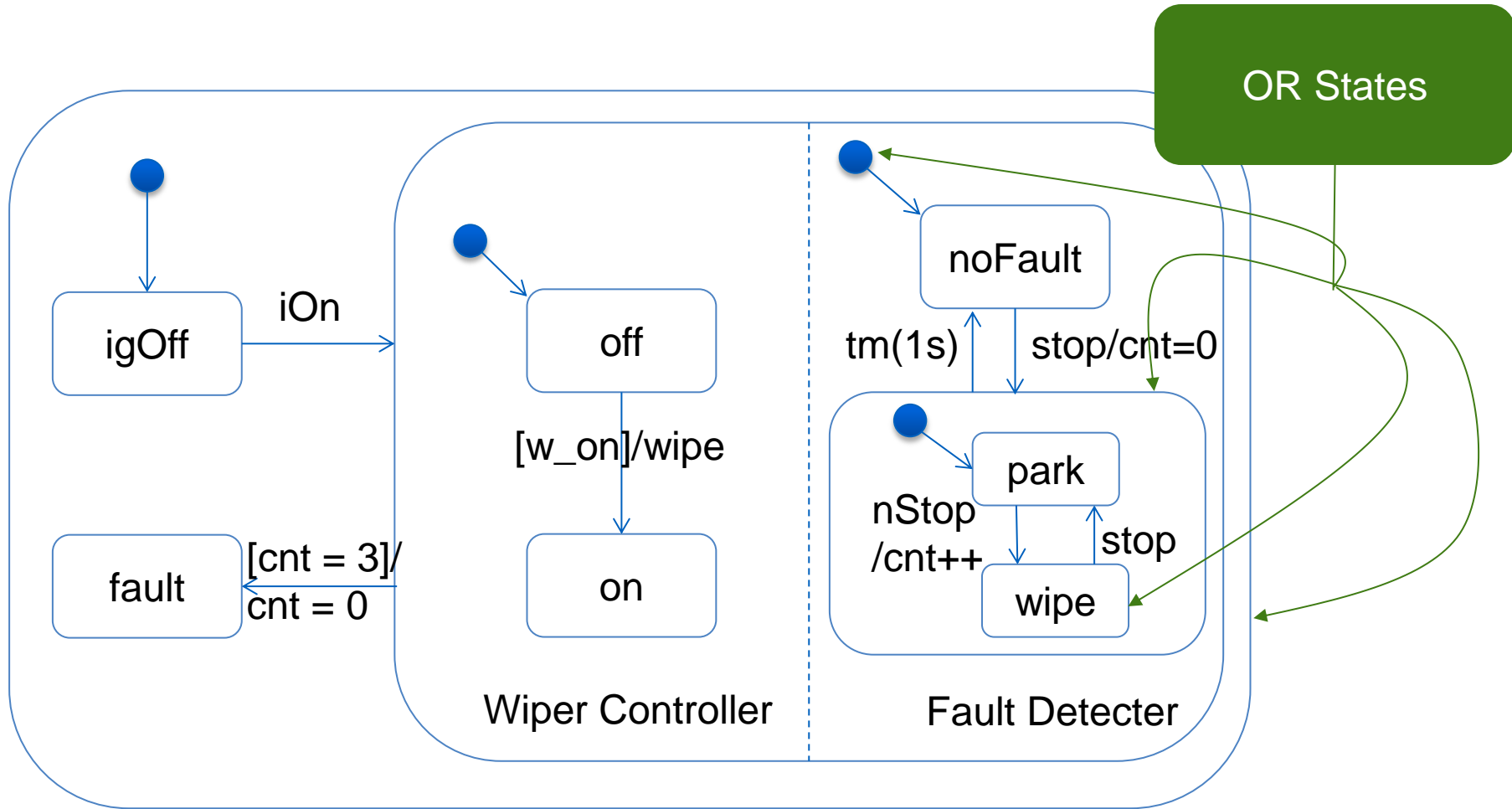
- Wiper
 - If *ignition* is *on* and *wiper* switch is *on* and there is no fault send *wipe* message
 - If the wiper is stuck, that is it switches between *stop* and *notstop* state thrice in 1s, then there is a fault

- Terminology
 - Events : transient
 - Ignition turning on
 - State : persistent
 - Ignition switch is on

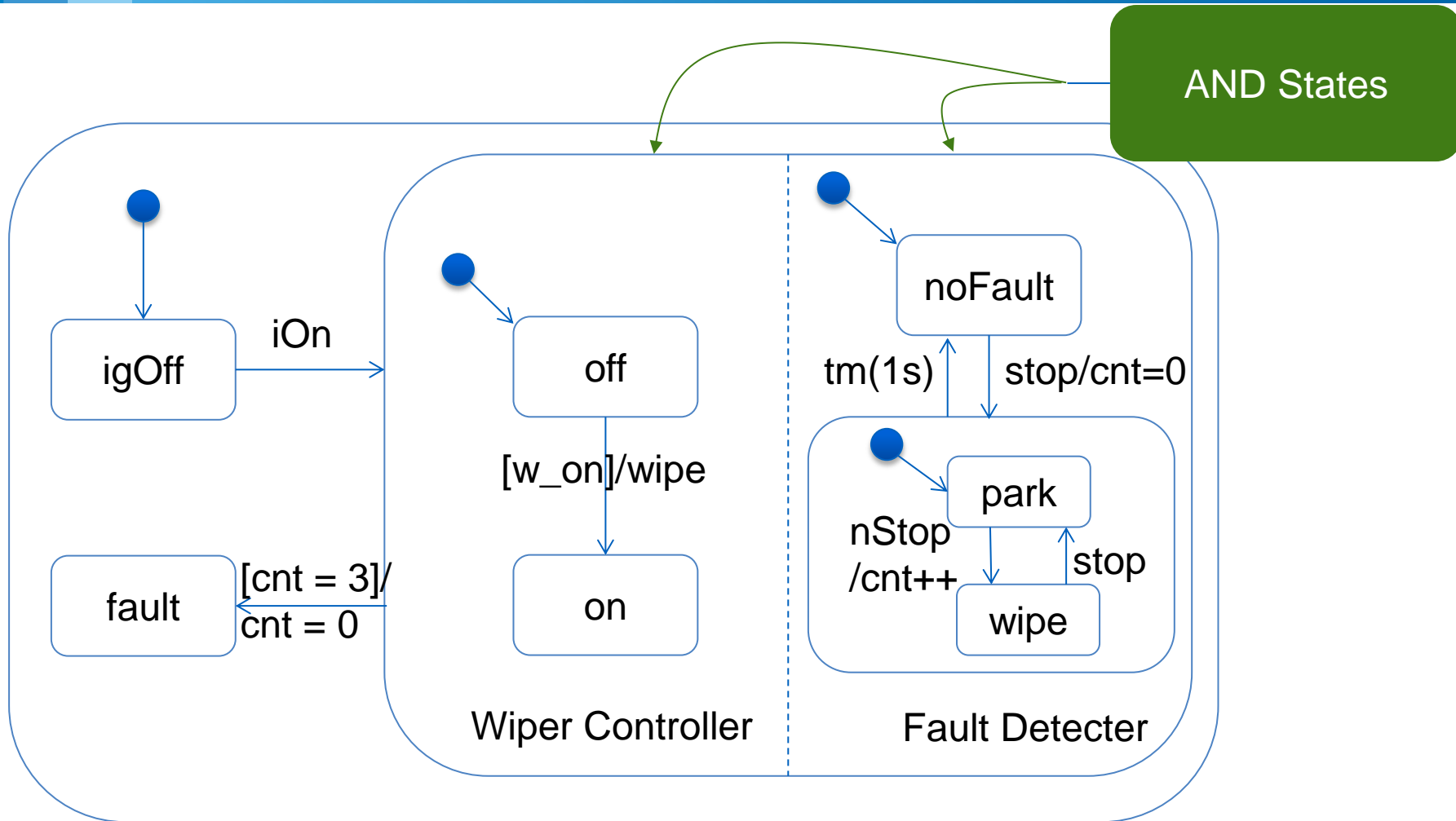
Wiper - Statecharts



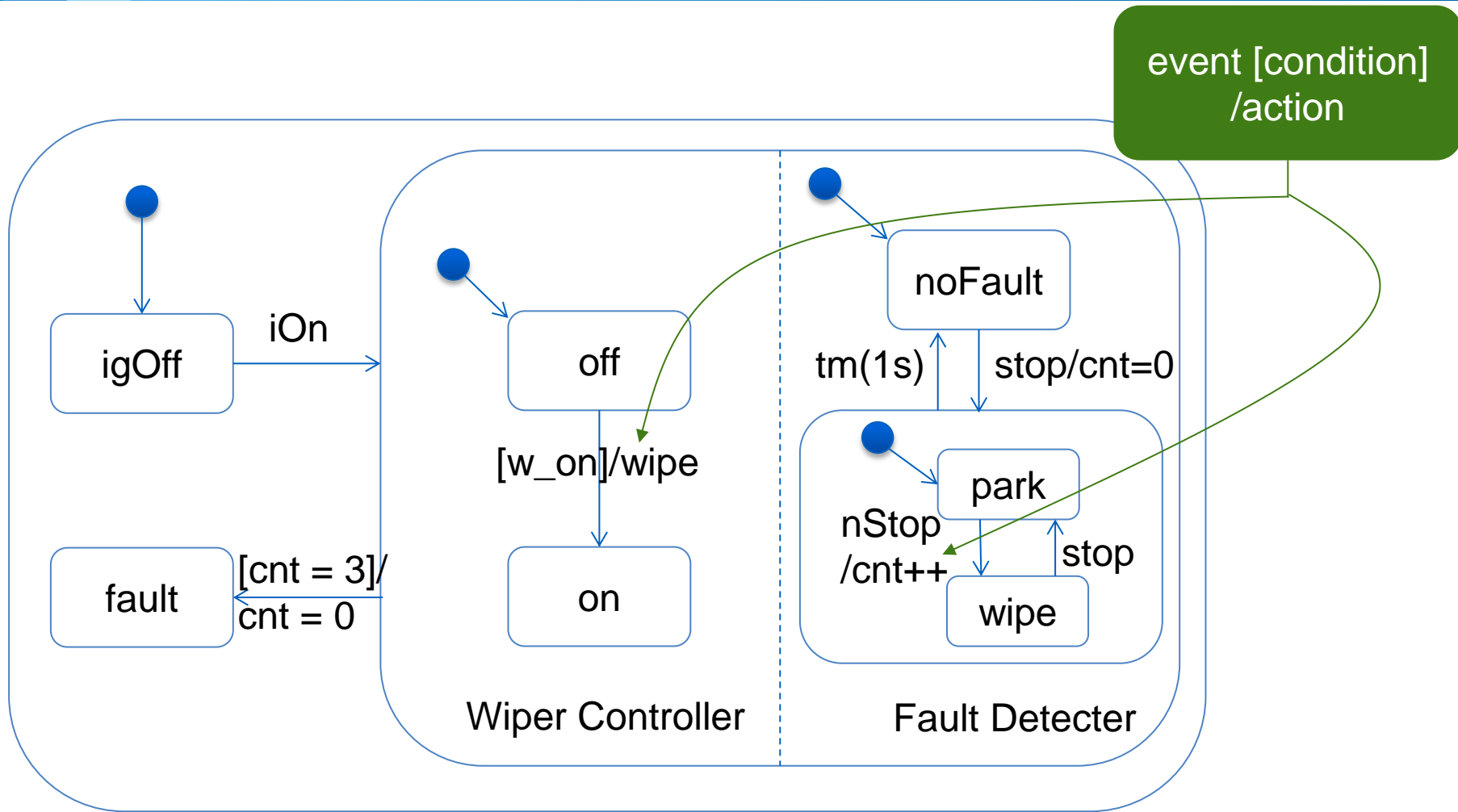
Wiper - Statecharts



Wiper - Statecharts

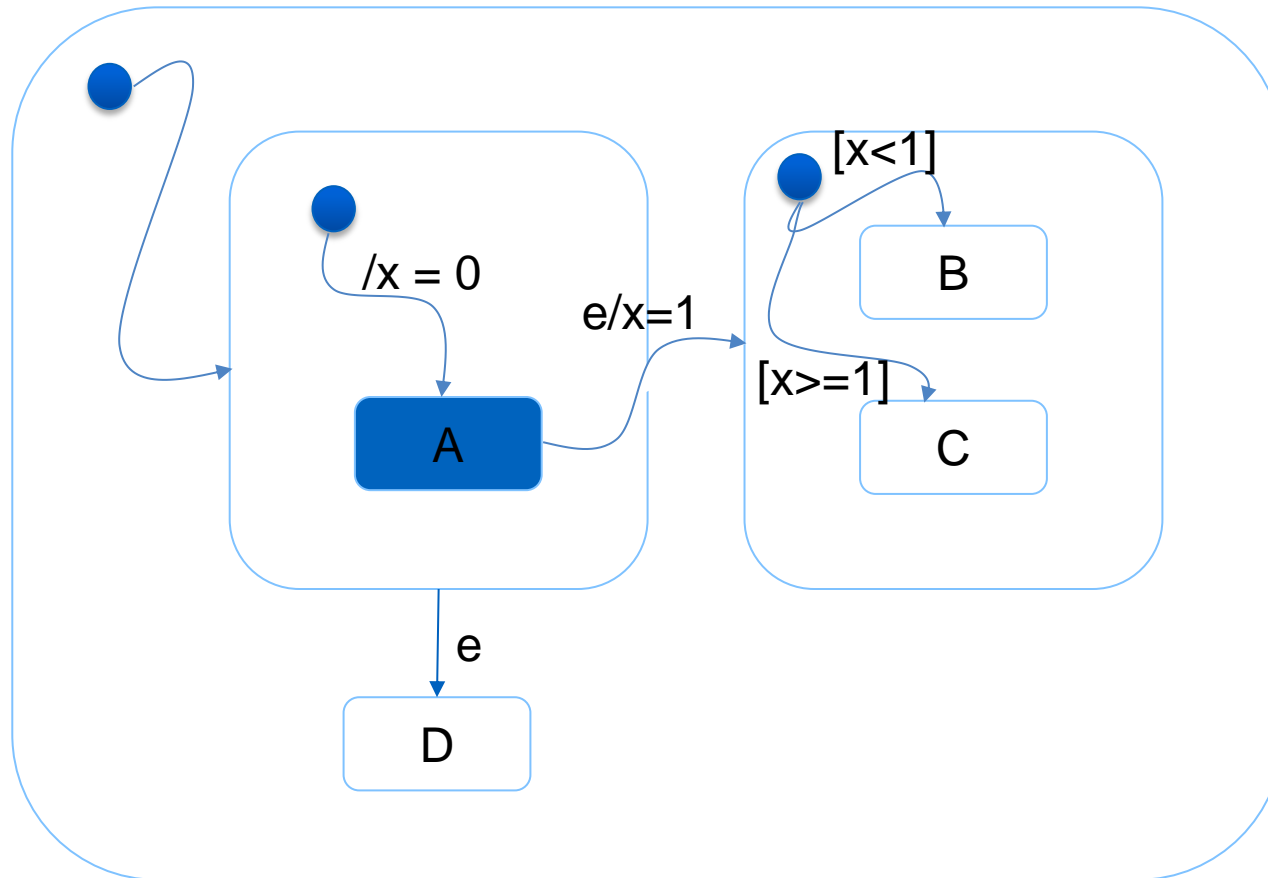


Wiper - Statecharts



Statecharts : Semantic Variations

System is in State A and event e occurs



Rhapsody

- Event Queues, Step, Micro step
 - Each Statechart is associated with an object
 - Each *active* class has an event queue
 - Only one event from the queue is processed at a time : Step
 - A series of null transitions may fire as a result : micro steps

Read next event from queue

T = maximal set of enabled transitions exiting lowest states

for each t in T

perform actions sequentially

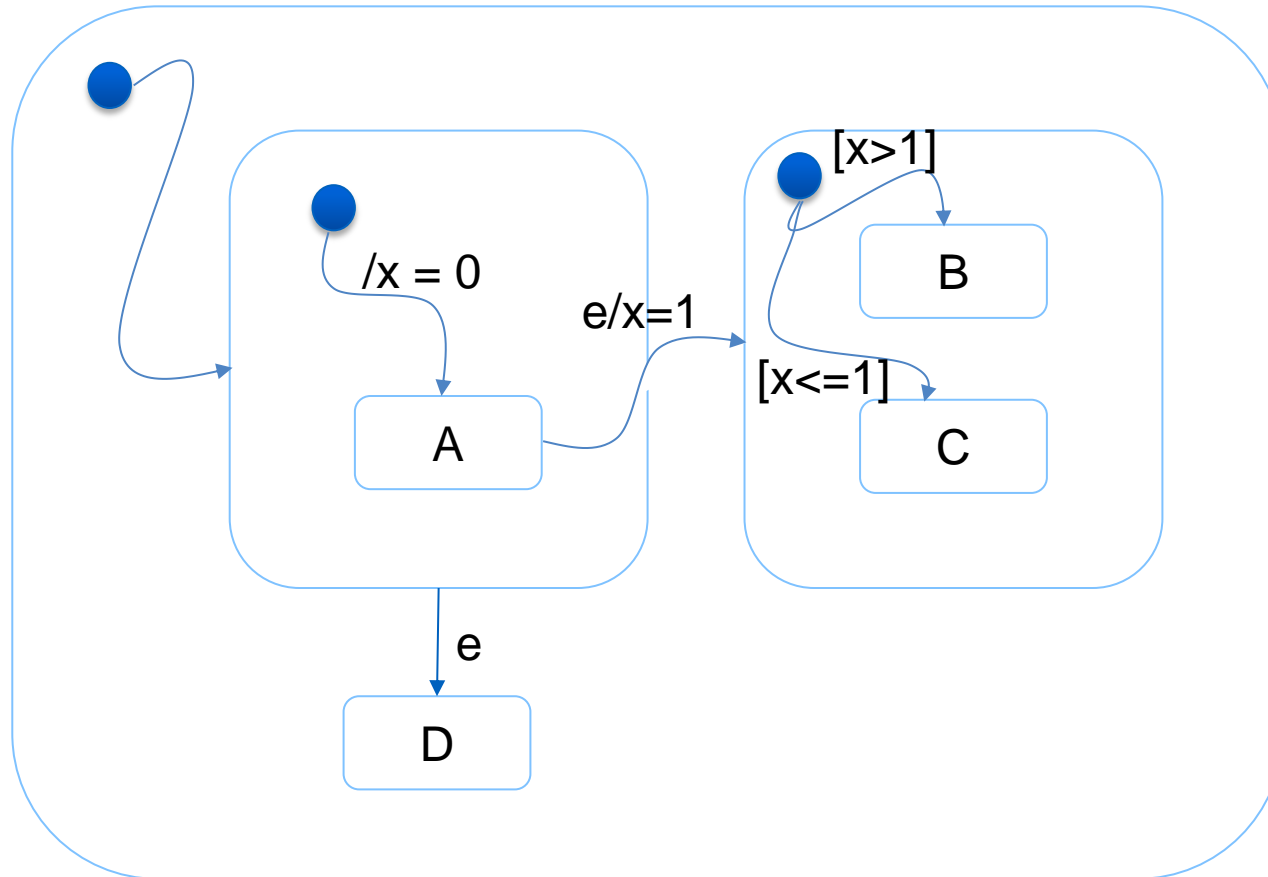
perform default transitions

update active configuration

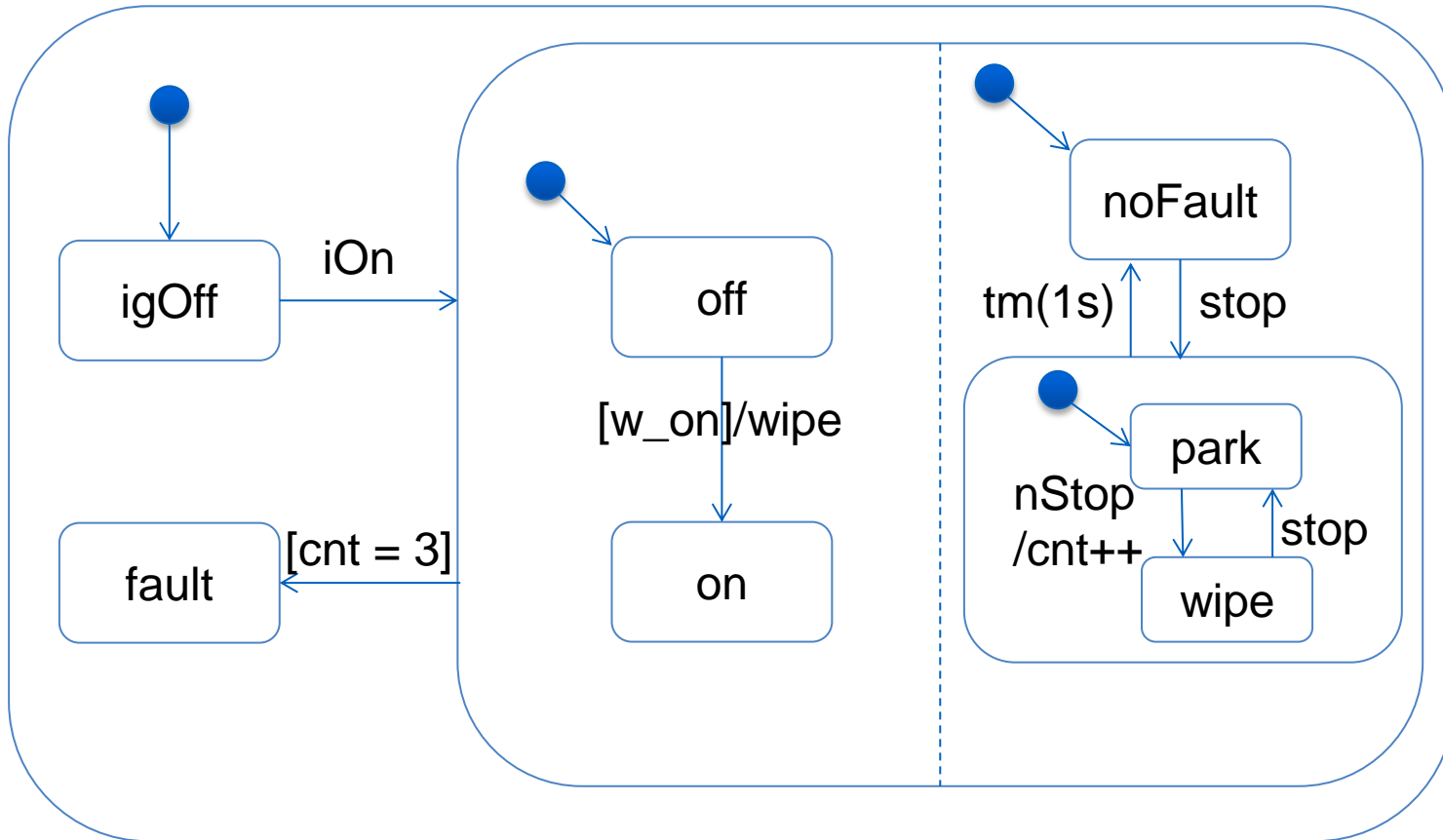
despatch null events till the system reaches a stable state

Statecharts : Semantic Variations

System is in State A and event e occurs



Wiper – a Critical Look



References

- Semantics
 - Rhapsody : Harel & Kugler, LNCS 3147 '01
 - Stateflow : Hamon & Rushby, FASE '04
 - UML : Borger, Caverra, Riccobene, ASM '00
 - Dialects : Crane & Dingel, MoDELS '05
- Harel
 - Statecharts : Science of computer programming '87
 - LSCs : Formal methods in systems design '01
 - Behavioral programming : CACM 55(7) '12

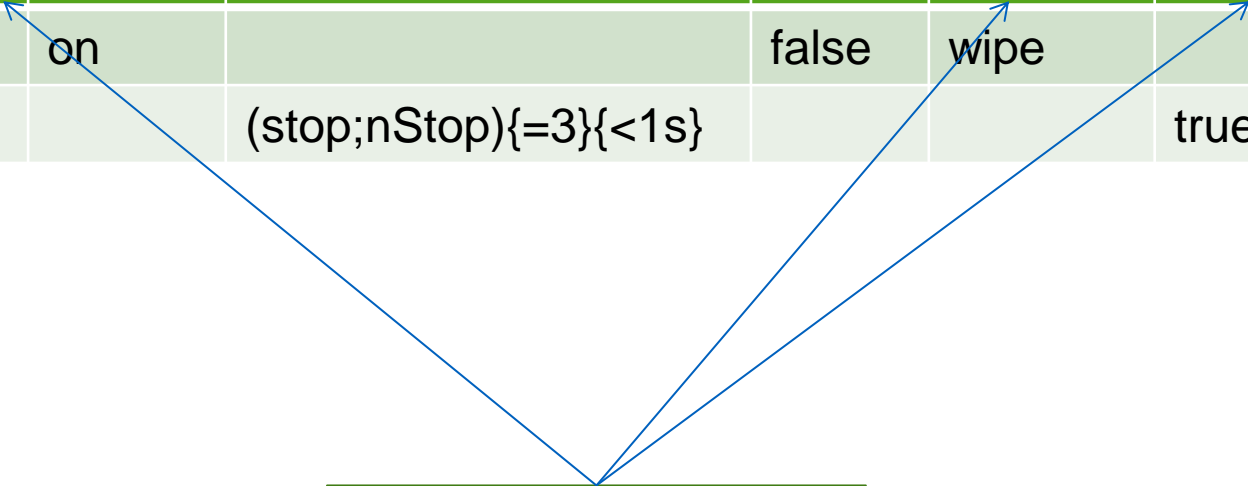
Expressive Decision Tables : EDT

Sr. No.	In Ignition	Wiper Switch	WiperPosition	Fault	Out W Cmd	Fault
1	on	on		false	wipe	
2			(stop;nStop){=3}{<1s}			true

Expressive Decision Tables : EDT

Sr. No.	In Ignition	Wiper Switch	WiperPosition	Fault	Out W Cmd	Fault
1	on	on		false	wipe	
2			(stop;nStop){=3}{<1s}			true


Ports



Expressive Decision Tables : EDT

Sr. No.	In Ignition	Wiper Switch	WiperPosition	Fault	Out W Cmd	Fault
1	on	on		false	wipe	
2			(stop;nStop){=3}{<1s}			true

Rows



Expressive Decision Tables : EDT

Sr. No.	In Ignition	Wiper Switch	WiperPosition	Fault	Out W Cmd	Fault
1	on	on		false	wipe	
2			(stop;nStop){=3}{<1s}			true

Cell Patterns

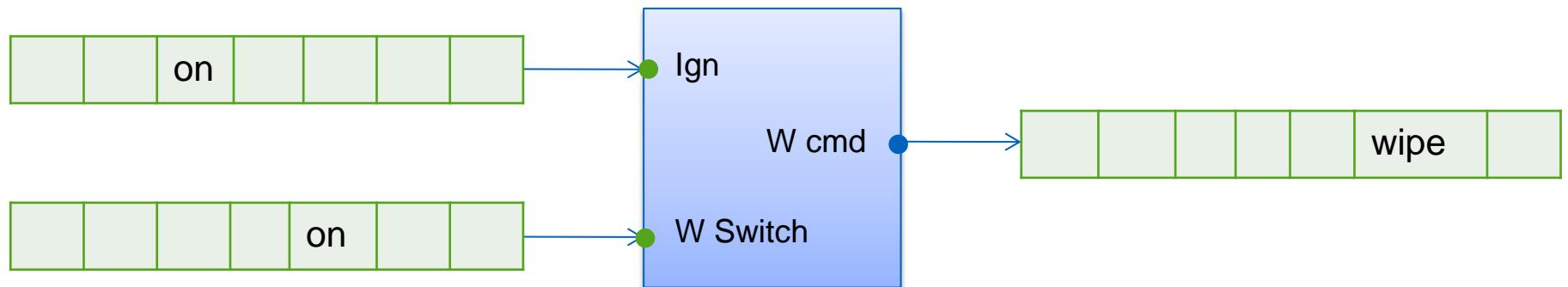


EDT : Semantics

- System as input and output ports
 - Port has a name and type
- A 'string' or 'sequence' of events for each port
- Discrete time

E.g: When both ignition and wiper request are on, start wiper

S.No	in Ign	W Switch	out W cmd
1	on	on	wipe



Formalizing the Semantics

- Pad patterns with ϵ
 - on becomes $on. \epsilon^*$
- Match predicates for cell and row
 - $match_c^{t \rightarrow u}$, $match_r^{t \rightarrow u}$
 - matches from input t to input u
 - each pattern extended only by ϵ^*
- If a row starts matching at t output is enabled at t+1
- Time is translated to length of strings

Research Challenges

- Better notations
 - Turn indicator specifications too complex
 - Non reactive domains, data structures/types
- Analysis
 - Scaling up
 - Highly parallel
 - Proving
- Testing
 - Generation
 - Oracles