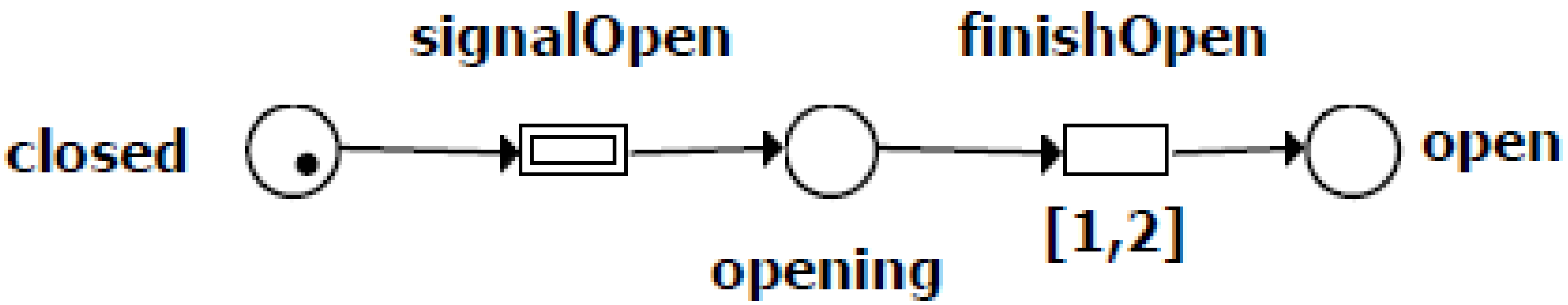
Compositional Analysis of Discrete Time Petri nets

Y. Thierry-Mieg, B. Berard, F. Kordon, D. Lime & O. H. Roux

June 2011 - Compo'Net
1st workshop on Petri Nets Compositions

Modelling time constrained systems

- Modeling time constraints :
  - The gate takes one to two time units to open



  - signalOpen should be « triggered »
- Modeling using integers is natural

Dense time semantics

- **Discrete events are instantaneous**

  (closed=1,x=0) -signalOpen-> (opening=1, x=0)

- **Time elapses by an arbitrary duration**

  (opening=1,x=0) -1.312 -> (opening=1,x=1.312)

  (opening=1,x=1.312) -finishOpen-> (open=1,x=0)

- **Infinite state space => use time zones**
  - *(closed=1,x in [0,+oo[)*
  - *(opening=1,x in [0,2])*
  - *...*

- **With several clocks=> linear inequation system**
  - **Difference Bounded Matrix (DBM)**

Discrete time semantics

- Discrete events are instantaneous

  (closed=1,x=0) -signalOpen-> (opening=1, x=0)

- Time elapses by one time unit

  (opening=1,x=0) -1-> (opening=1,x=1)

  (opening=1,x=1) -finishOpen-> (open=1,x=0)

- Finite state space => integer clocks as additional variables

  (opening=1,x =0) -1-> (opening=1,x =1)

  -1->(opening=1,x =2) ...

- Very large state space
  - Decision Diagrams

Dense vs Discrete

- For closed bounds e.g. [0,3] but not [0,3[semantics are equivalent (Popova'91-HMP'92)
- Dense time : use DBM
  - *++ Efficient representation of time regions*
  - *+ Scales to large absolute values [0,1000] ⇔ [0,10]*
  - *-- Poor scale up in number of locations*
  - *-- Limited scale up in number of concurrent clocks*
- Discrete time : use DD
  - *++ Strong scale up in number of locations*
  - *+ Good scale up in number of concurrent clocks*
  - *- Poor scale to large absolute clock values*
  - *++ Back to a (large) finite transition system*

Instantiable Transition Systems [Tacas'09]

- **Instantiable Transition Systems**
  - Notion of type and instance to capture similarity
  - Simple labeled Kripke structure semantics
  - Efficient solution engine using Hierarchical Set Decision Diagrams (SDD)
- **ITS type definition = ITS Semantics**
  - Elementary types based on Labeled Transition System or any finite state model
  - Composite types contain nested instances
  - Hierarchical composition mechanism using event-based label synchronization

ITS Type Semantics

- An ITS Type must define :
  - S : a set of states
  - A : a set of action labels (Interface)
  - Locals : S -> 2^S
    the local successors function
  - Succs : S x A* -> 2^S
    the synchronization function

SDD Encoding

Homomorphism Encoding

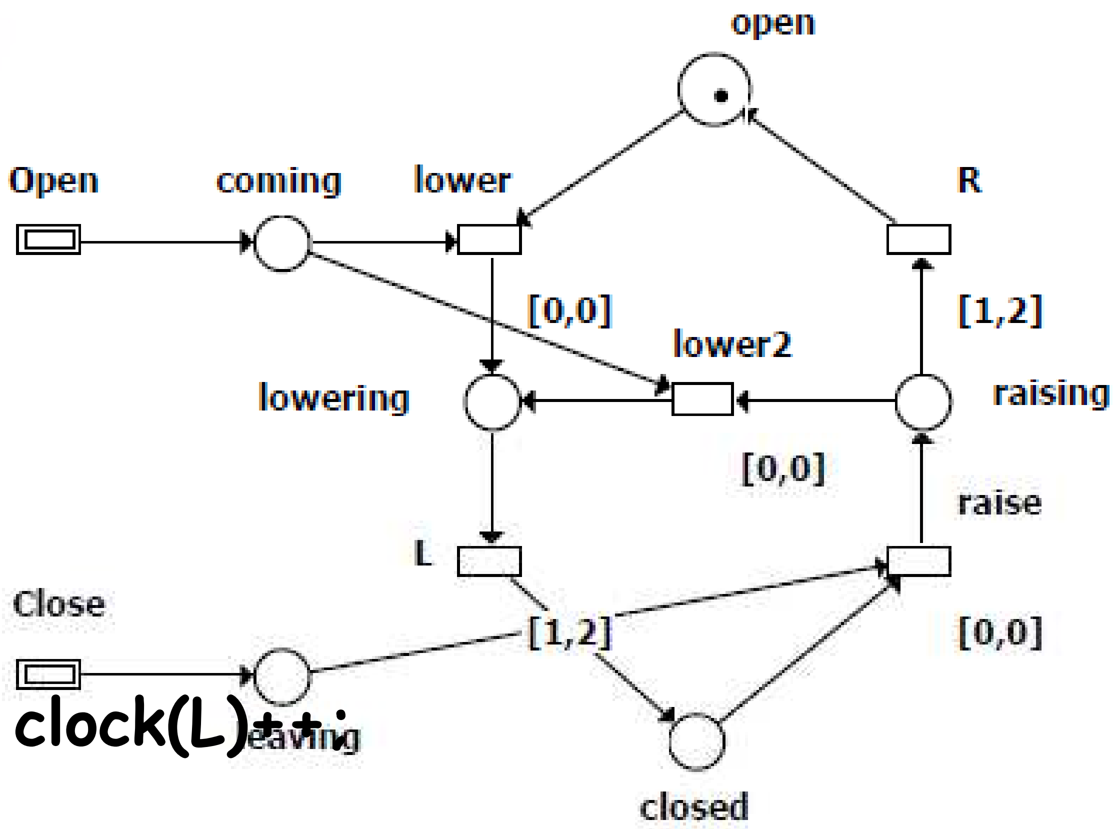
- An ITS instance i has a type noted type(i)
- Reachability of a state by an instance I in a given initial state is defined using Locals.

Time Petri nets as elementary ITS

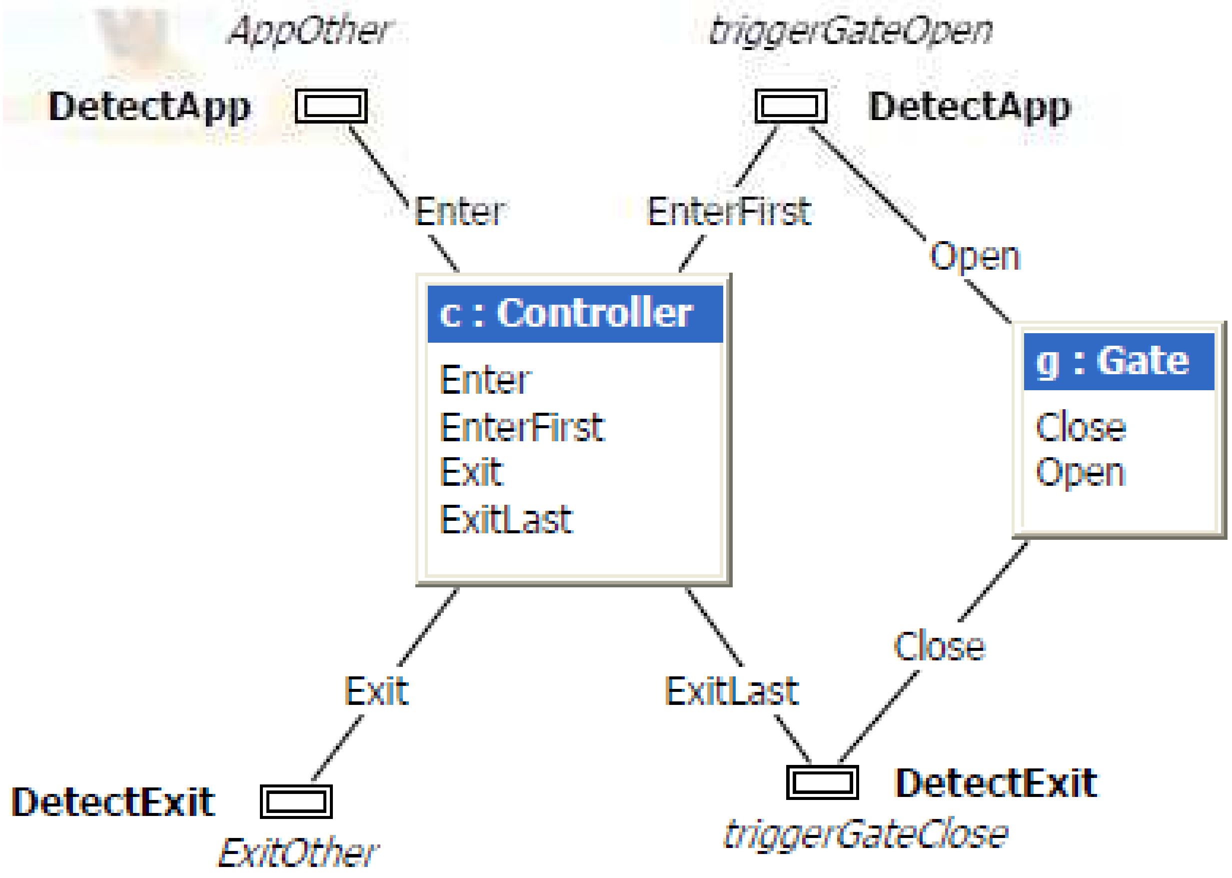
- States
  - Place markings
  - Transition clocks
- Interface
  - Open, Close
  - 1

If (enabled(L)) {
  if (clock(L) < 2) clock(L)++;
  else empty set; }

- Locals
  - All other "private" transitions



- Compose instances of arbitrary ITS type
- Synchronize on labels



A composite type : tabular view

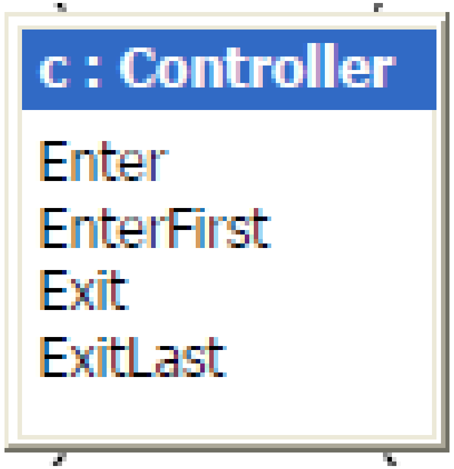
- ITS Composite semantics use a partial synchronization function

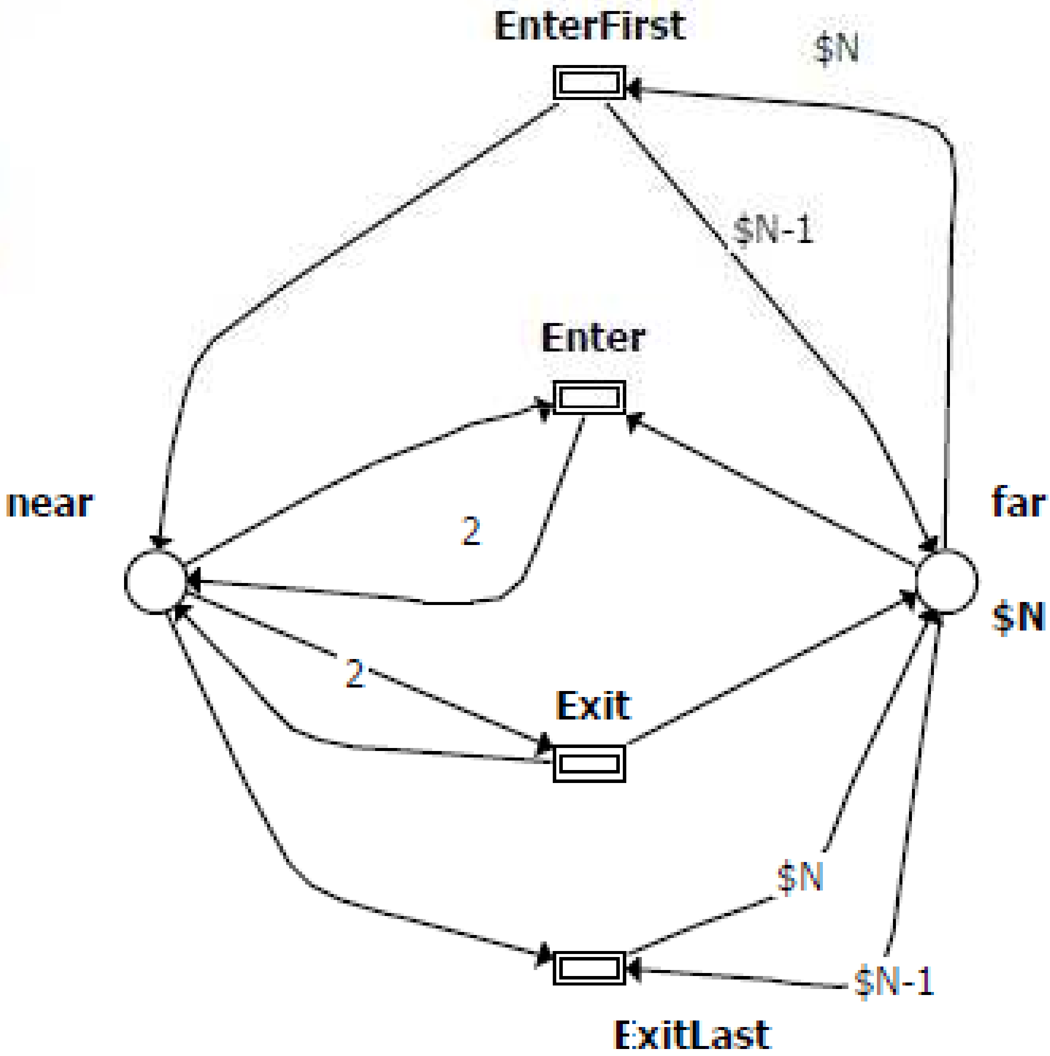
c:Controller	g:Gate	label
EnterFirst	Open	DetectApp
Enter	-	DetectApp
Exit	-	DetectExit
ExitLast	Close	DetectExit
1	1	1

- State is a Cartesian product of instance states

A controller to count trains

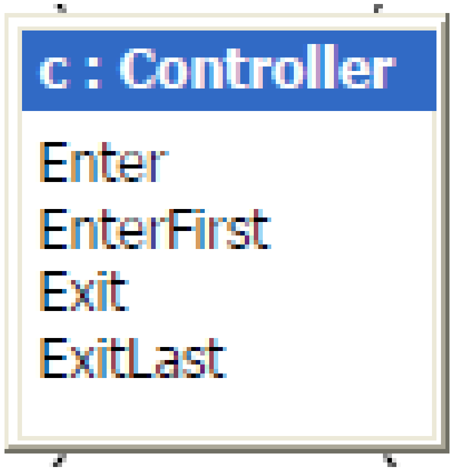
- $N is a constant
- Interface :
  - Enter, EnterFirst
  - Exit, ExitLast
  - 1 (does nothing)

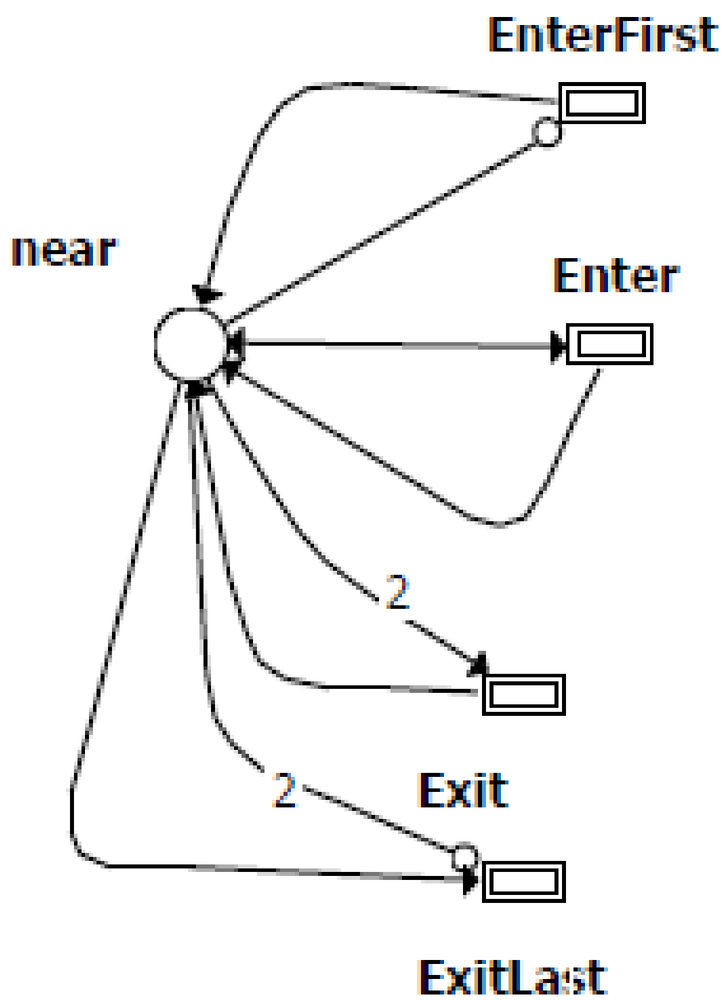




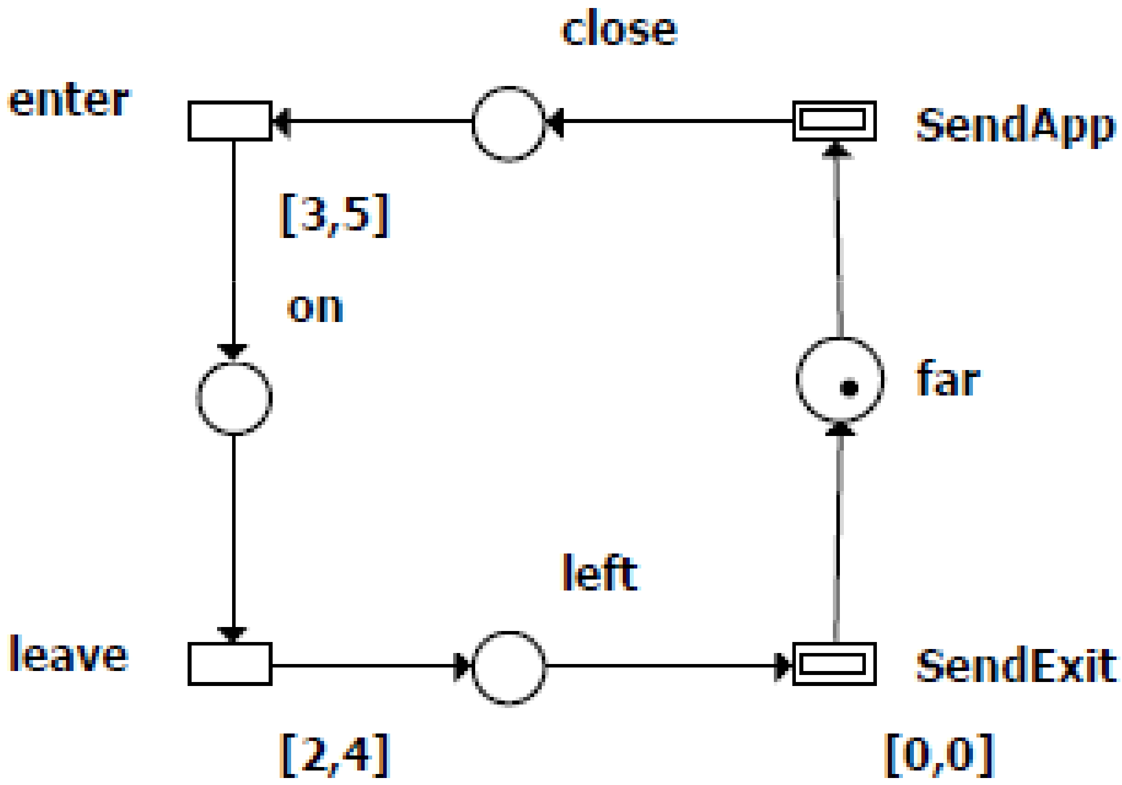
Another controller to count trains

- Interface :
  - Enter, EnterFirst
  - Exit, ExitLast
  - 1 (does nothing)
- Use inhibitor arcs, test arcs, reset arcs

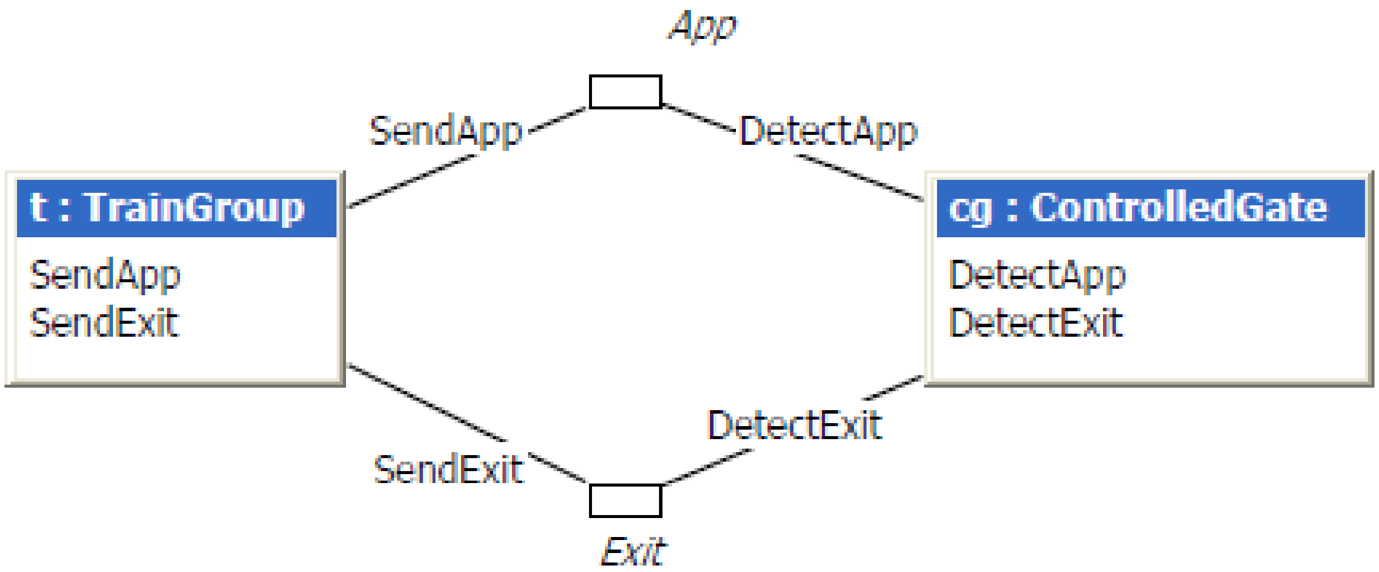




TPN model of a train



Composing Train and Controlled gate

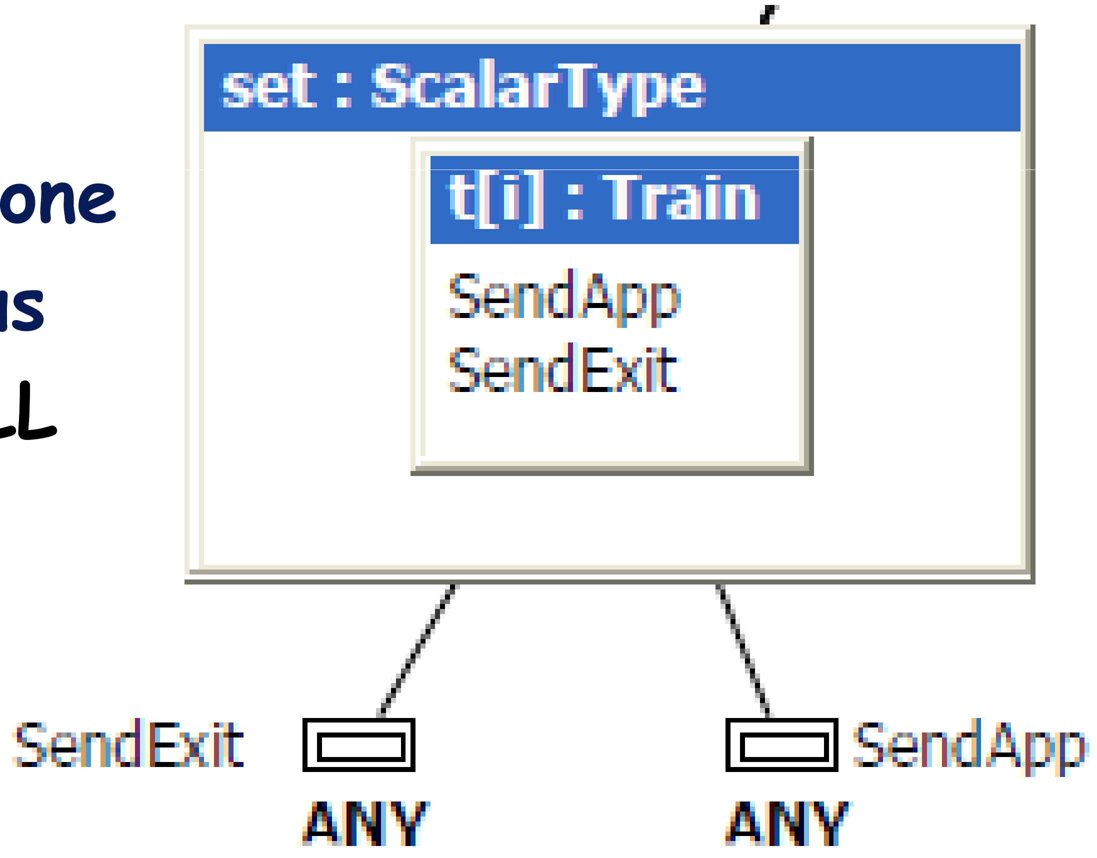


- T is the "local" label
- 1 is considered local at topmost level

t:Trains	g:Gate	label
SendApp	DetectApp	T
SendExit	DetectExit	T
1	1	T

Modeling Symmetric Systems

- Scalar Set
  - $SIZE constant
- Only two types of "delegates"
  - ANY : choice of one
  - ALL : rendez-vous
- 1 is implicitly an ALL delegate



- Scalar Set : $SIZE=3

t[0]:Train	t[1]:train	t[2]:Train	label
SendApp	-	-	SendApp
-	SendApp	-	SendApp
-	-	SendApp	SendApp
SendExit	-	-	SendExit
-	SendExit	-	SendExit
-	-	SendExit	SendExit
1	1	1	1

ANY

ANY

ALL

Performance: Reachability

Fischer (N is the number of processes)

N	Roméo			RED		UPPAAL/sym			Roméo/SDD		
	tm	*mm*	*sm*	*tm*	*mm*	*tm*	*mm*	*sm*	*tm*	*mm*	*sm*
8	1 051	282 108	740 633	11	278 028	0.01	160	137	0.1	2 020	$1.17\ 10^6$
9	73 071	$1.77\ 10^6$	$3.72\ 10^6$	67	785 108	0.03	160	172	0.1	2 156	$6.20\ 10^6$
10	DNF	OOM	OOM	652	$2.35\ 10^6$	0.1	160	211	0.1	2 332	$3.26\ 10^7$
170	-	-	-	-	OOM	7 783	47 956	57 971	23	101 896	$2.27\ 10^{120}$
700	-	-	-	-	-	DNF	-	-	1391	$1.82\ 10^6$	$2.66\ 10^{491}$
730	-	-	-	-	-	-	-	-	1803	$2.33\ 10^6$	$2.58\ 10^{512}$

Train (N is the number of trains)

N	Roméo			RED		UPPAAL/sym			Roméo/SDD		
	tm	*mm*	*sm*	*tm*	*mm*	*tm*	*mm*	*sm*	*tm*	*mm*	*sm*
6	43.1	36 948	29 640	7	202 412	0.14	908	432	1.5	7 360	$4.83\ 10^6$
7	6 115	377 452	131 517	66	723 428	0.23	3 200	957	2.5	10 304	$6.28\ 10^7$
8	DNF	-	-	-	OOM	1	3 336	2 078	4	14 188	$8.16\ 10^8$
13	-	-	-	-	-	2 634	13 188	79 598	26	56 660	$3.02\ 10^{14}$
15	-	-	-	-	-	60 860	61 256		42	86 360	$5.11\ 10^{16}$
16	-	-	-	-	-	DNF	-	-	52	104 848	$6.65\ 10^{17}$
44	-	-	-	-	-	-	-	-	1143	$2.13\ 10^6$	$1.03\ 10^{49}$

Table 1. Performances measured for the *Fischer* and *train* models.

Conclusion

- Discrete time modeling
  - A natural model for many systems (hardware)
  - Allows to revert to discrete state space algorithms
  - SDD based solution empirically effective
- ITS for compositional modeling
  - Extensible framework to exploit SDD
  - Efficient support for symmetric models
- Perspective:
  - TCTL model checking with discrete semantics

Thank you for your attention !

SDD and ITS-tools are distributed as an open-source LGPL/GPL C++ source and pre-compiled tools :

http://ddd.lip6.fr

Coloane « incubation » plugin for ITS manipulation and CTL model-checking

http://coloane.lip6.fr/night-updates

Come to the tool demo session tomorrow

See more about ITS tools performance (for untimed systems) at SuMO MCC session this afternoon