CBSE 2014

Quantifying the Parallelism in BPMN Processes using Model Checking

Radu Mateescu, Gwen Salaün, Lina Ye

Presented by Lina Ye, 03 July 2014

CONVECS, Inria Grenoble

Outline

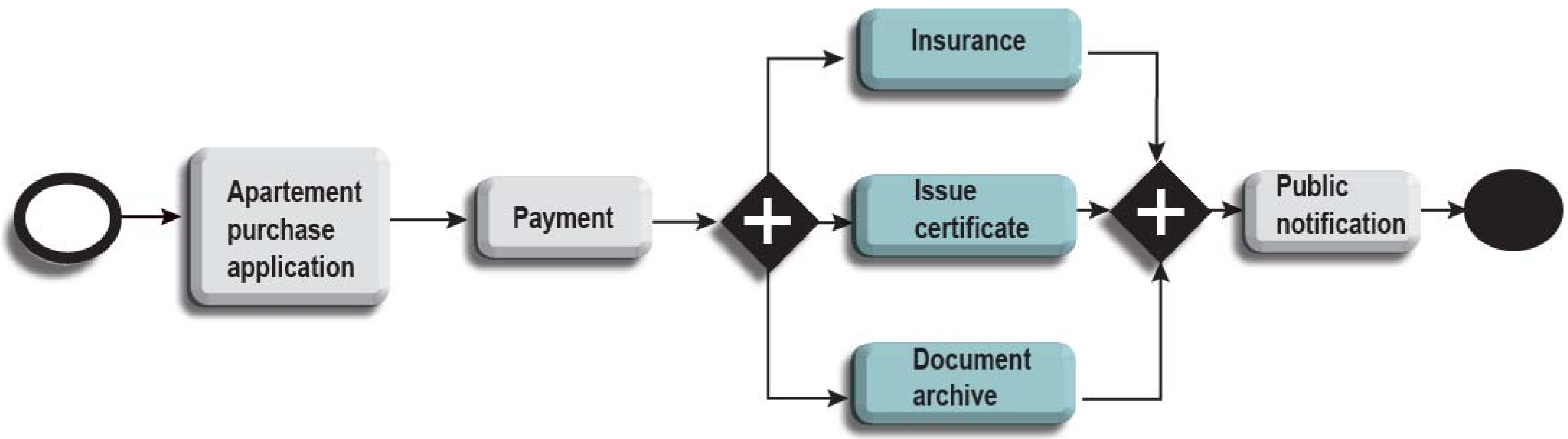
- Background
- BPMN Model
- Formal semantic model
- Compute the degree of parallelism
- Conclusion and perspectives

Background

- BPMN: published as ISO/IEC standard in 2013
- BPMN process: specific ordering of a set of structured, related tasks
- Degree of parallelism: maximum number of tasks executable in parallel
- Degree ↑ efficiency ↑ complexity ↓
- Valuable guide for resource allocation

Peak Workload Demands

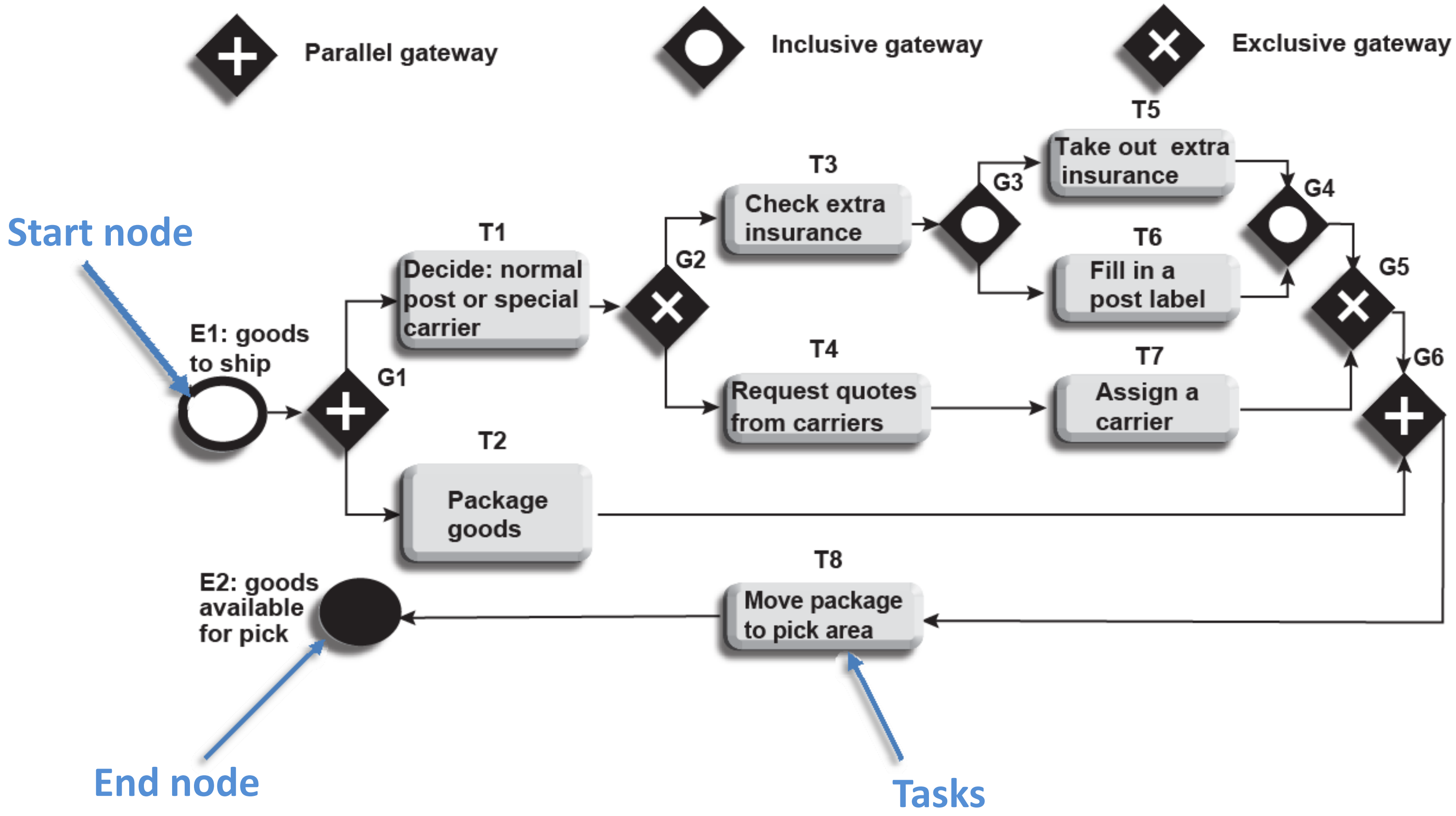
Apartement purchase



Outline

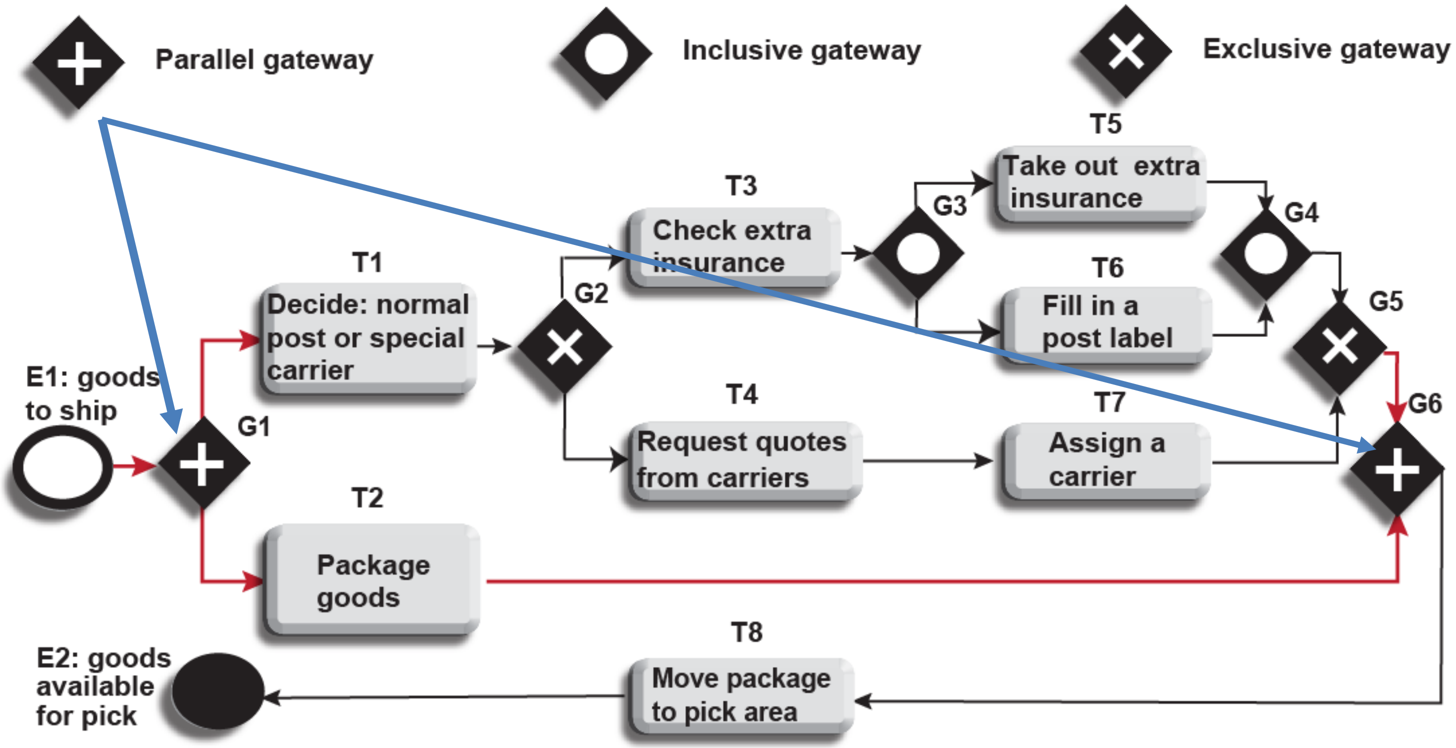
- Background
- BPMN Model
- Formal semantic model
- Compute the degree of parallelism
- Conclusion and perspectives

BPMN Model

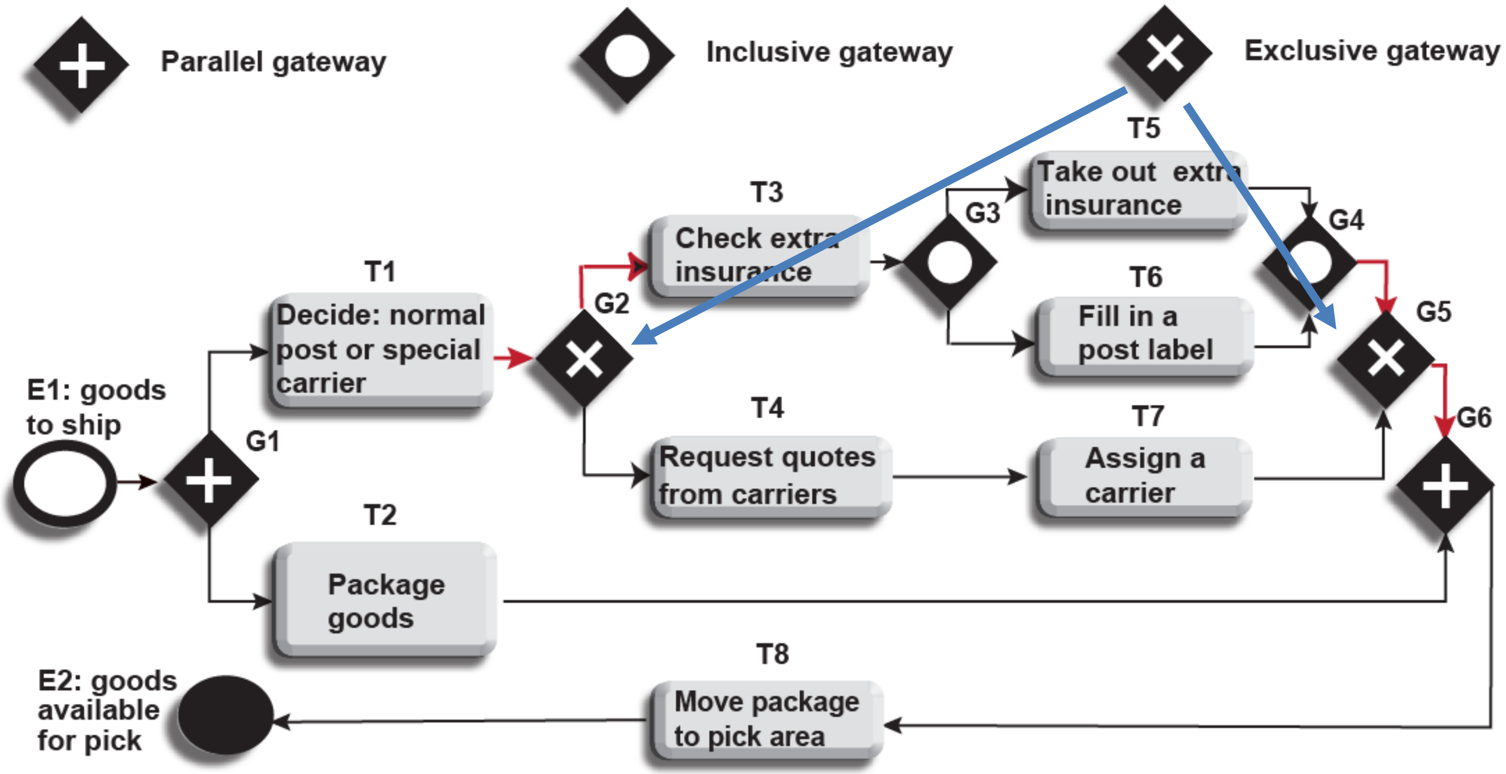


Parallel gateway
Inclusive gateway
Exclusive gateway
Start node
E1: goods to ship
G1
T1
Decide: normal post or special carrier
T2
Package goods
G2
T3
Check extra insurance
T4
Request quotes from carriers
G3
T5
Take out extra insurance
T6
Fill in a post label
T7
Assign a carrier
G4
G5
G6
T8
Move package to pick area
E2: goods available for pick
End node
Tasks

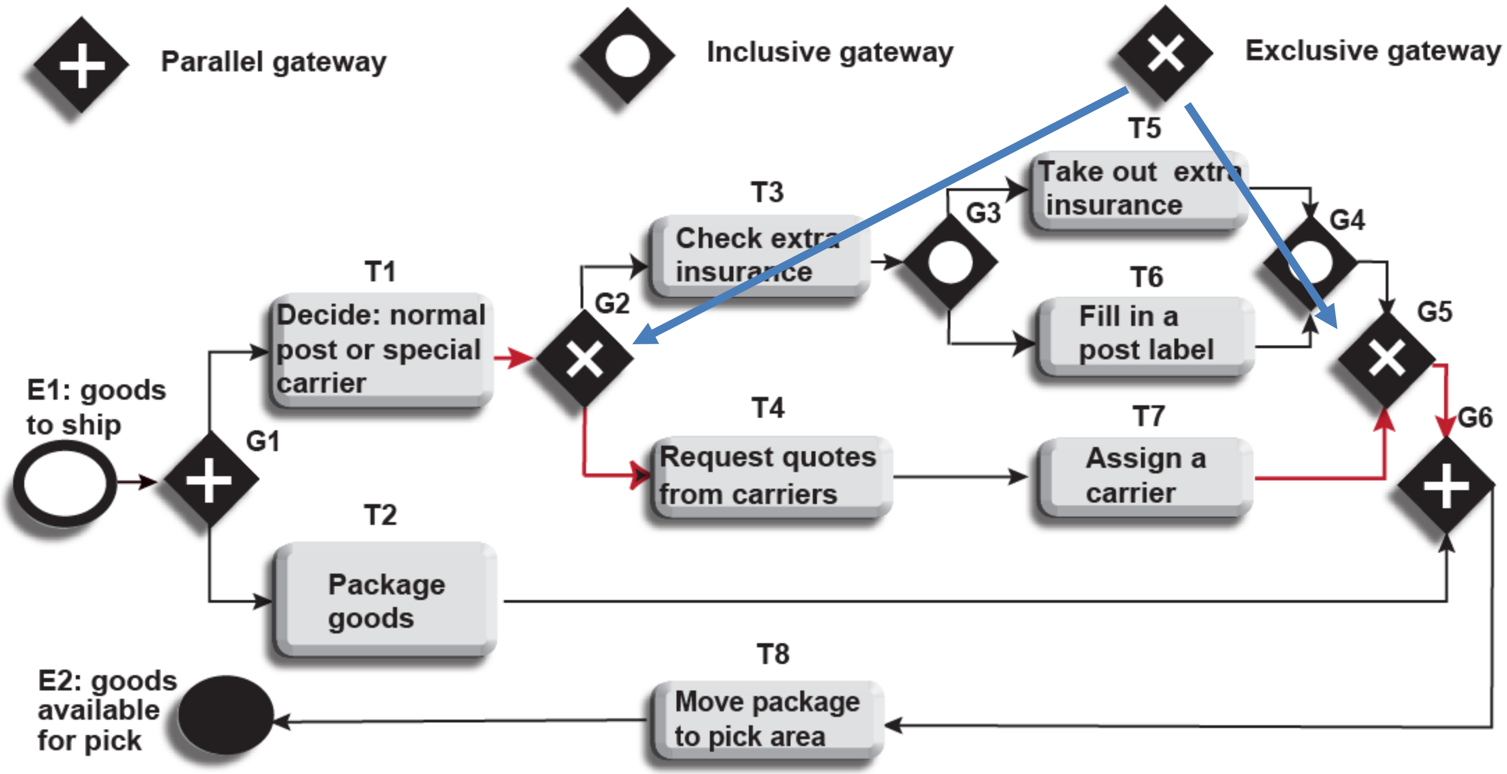
BPMN Gateways



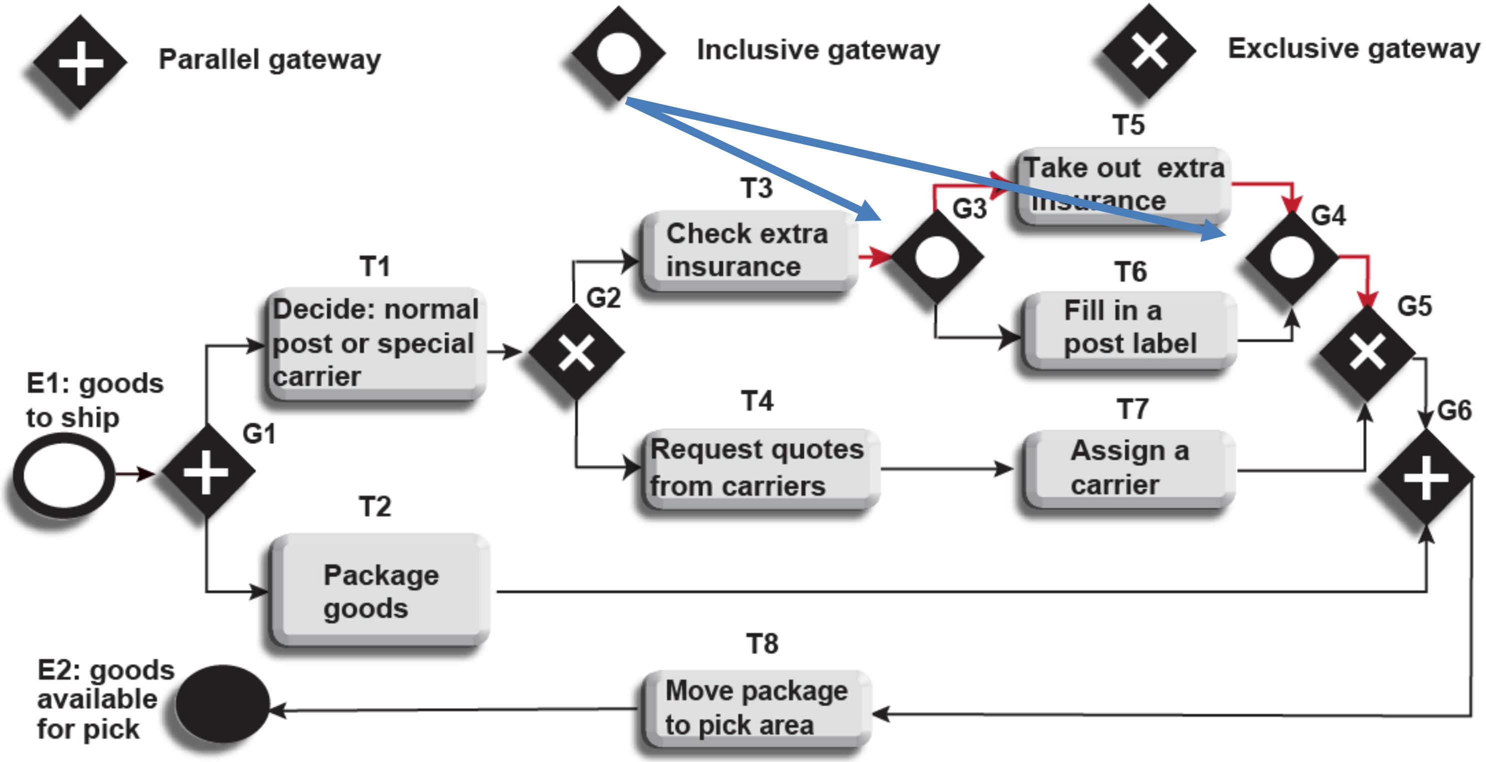
BPMN Gateways



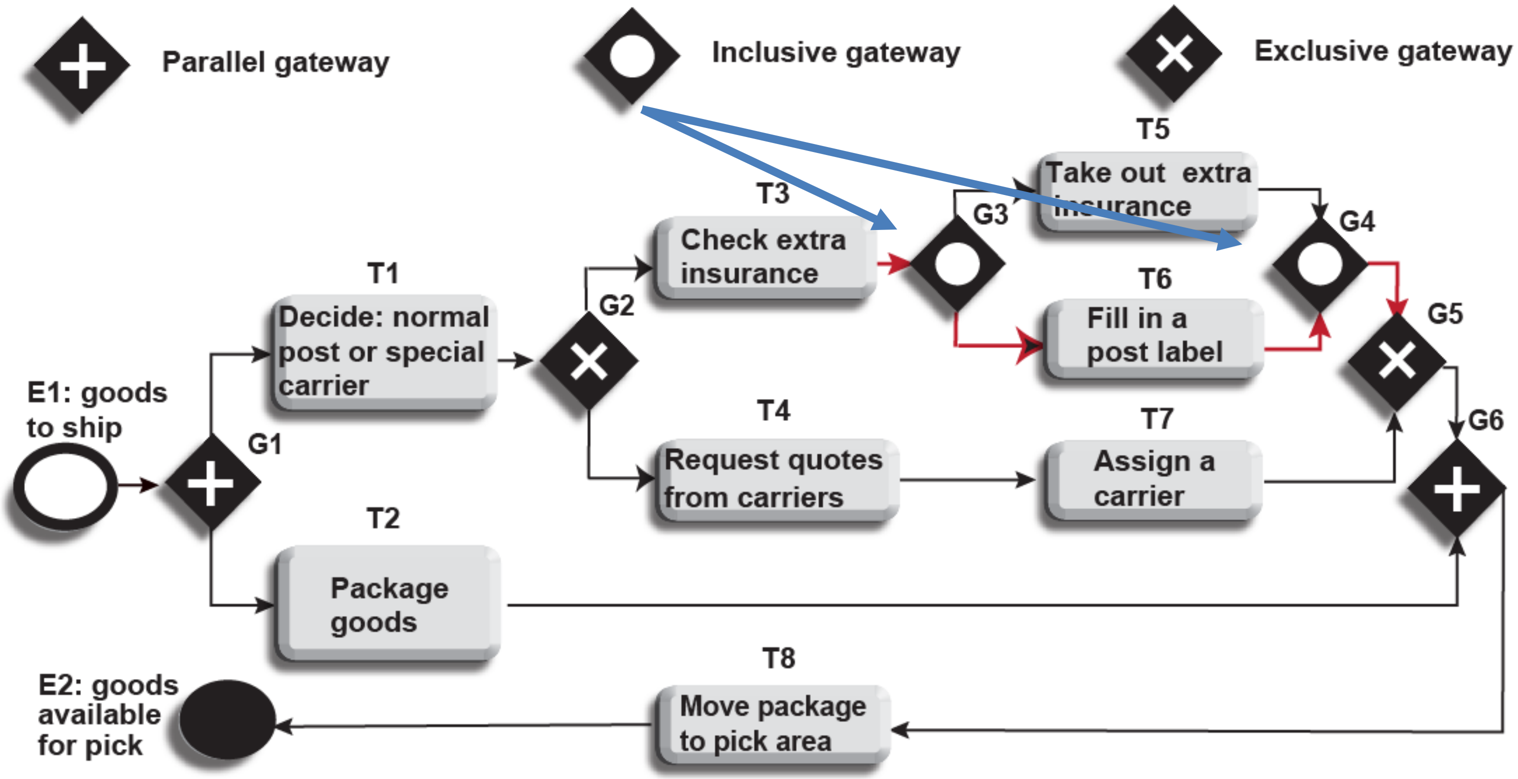
BPMN Gateways



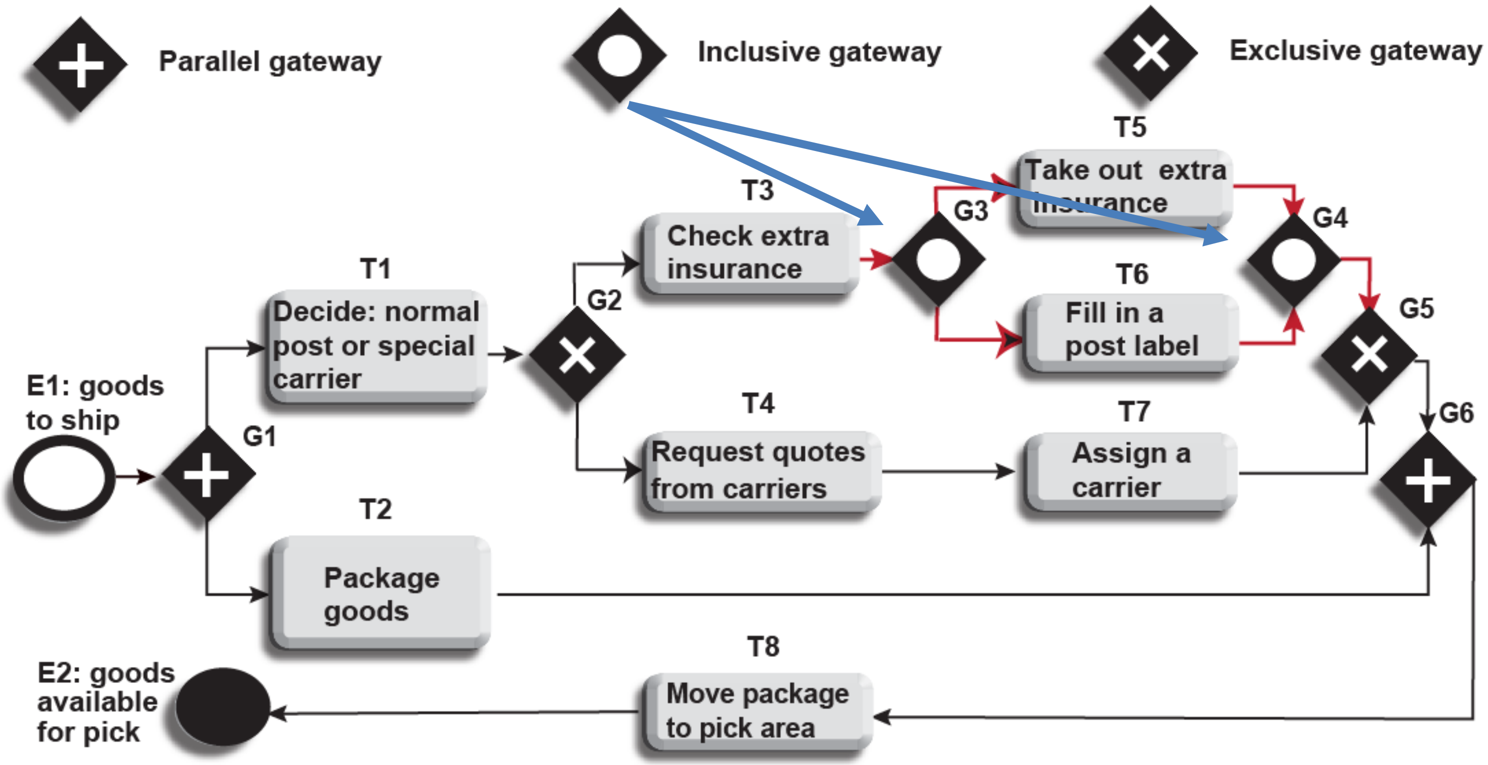
BPMN Gateways



BPMN Gateways

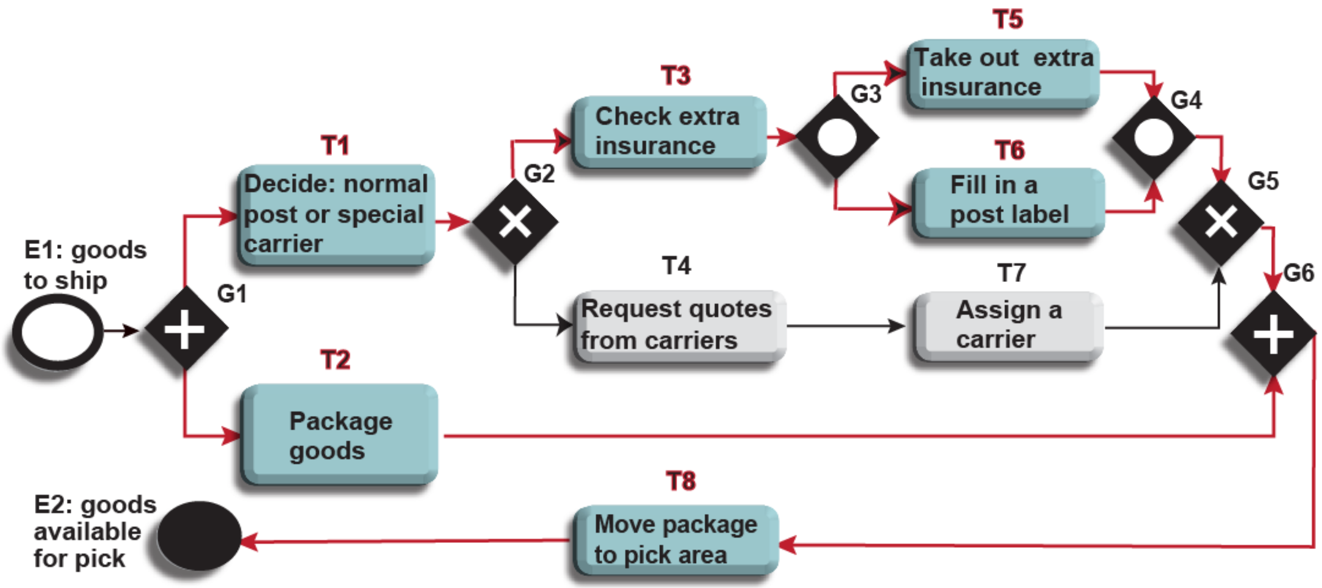


BPMN Gateways



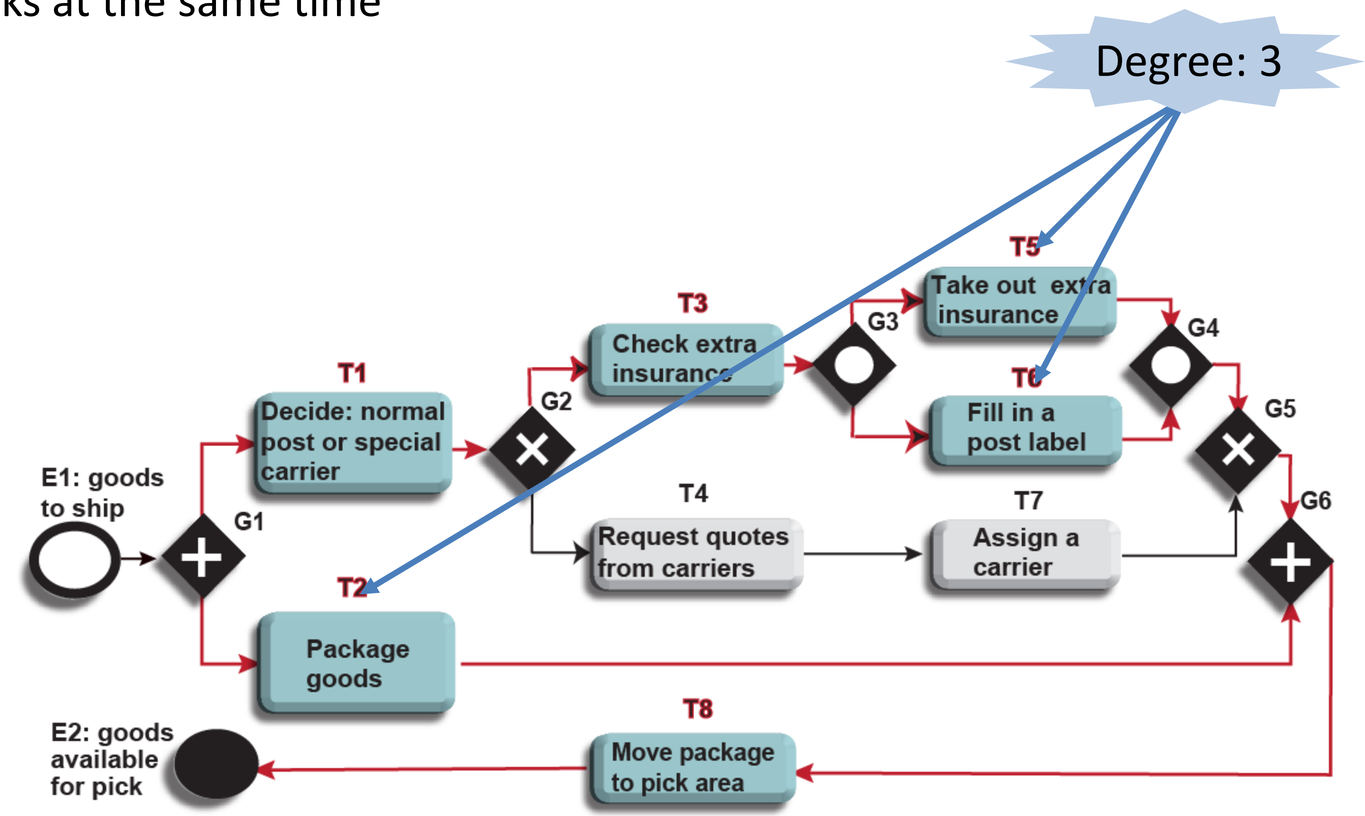
Degree of Parallelism

To comput the degree of parallelism: maximum number of executable tasks at the same time

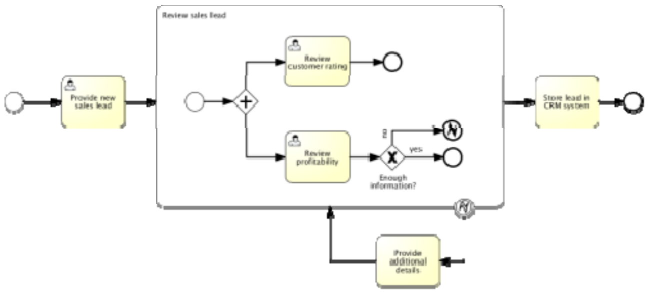


Degree of Parallelism

To comput the degree of parallelism: maximum number of executable tasks at the same time



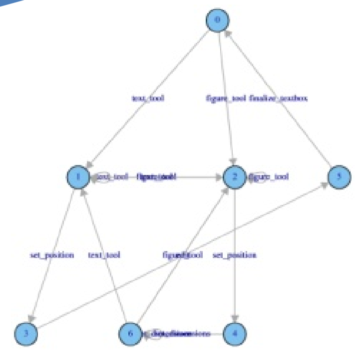
Contributions



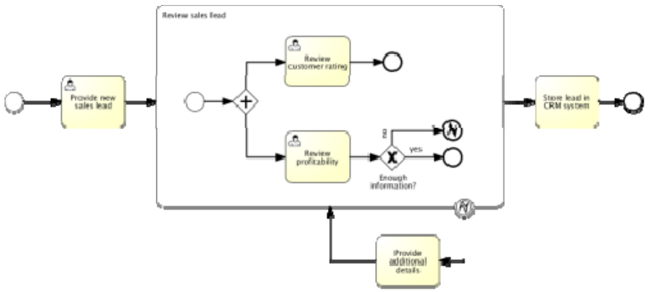
LNT

Information about parallelism

Process algebra



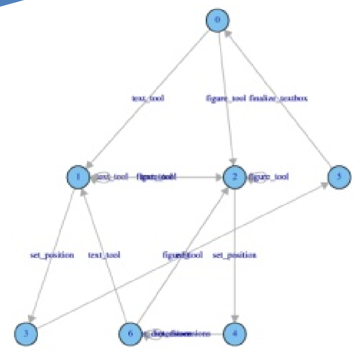
Contributions



Process algebra

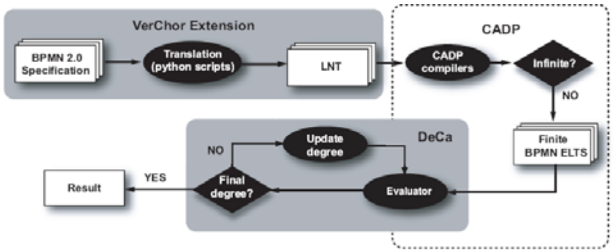
LNT

Information about parallelism



Model-checking

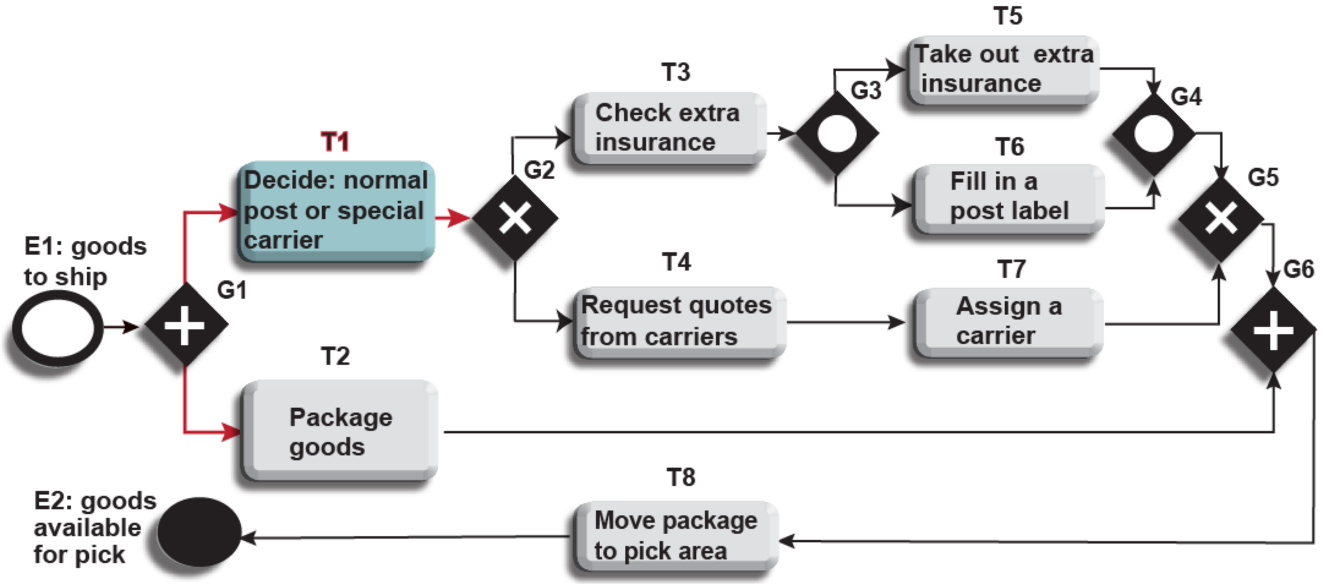
Degree of parallelism

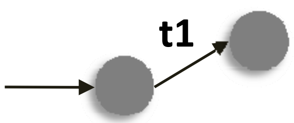


Outline

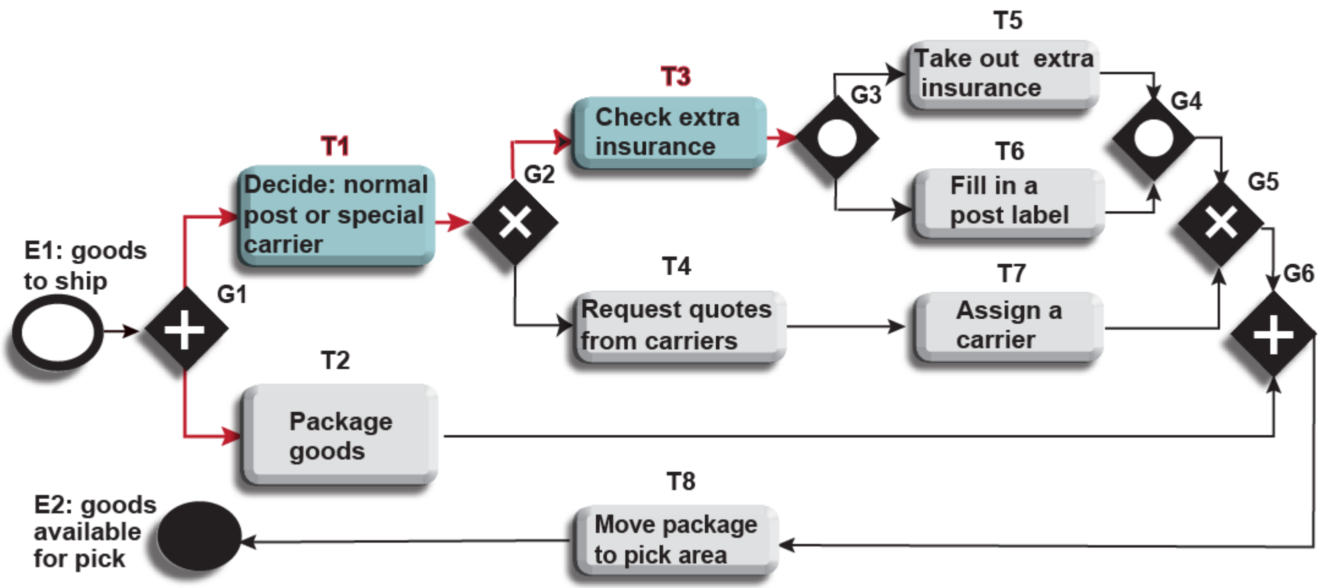
- Background
- BPMN Model
- Formal semantic model
- Compute the degree of parallelism
- Conclusion and perspectives

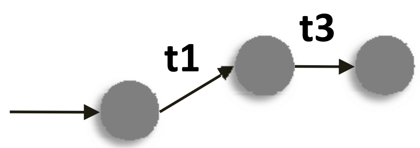
From BPMN to LTS



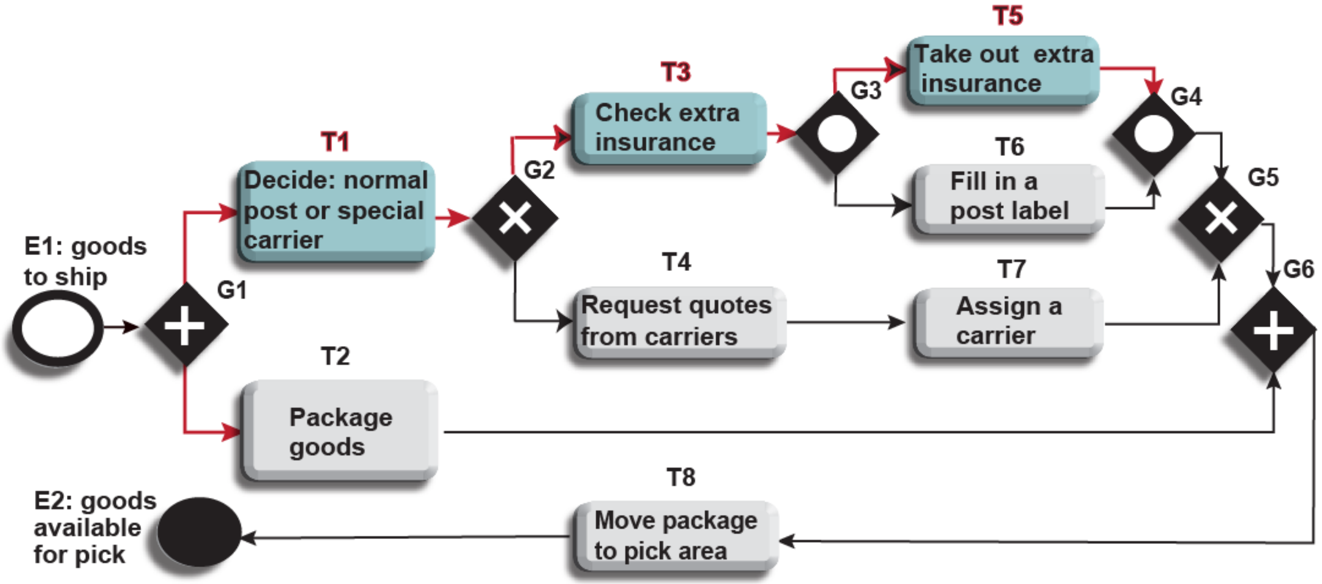


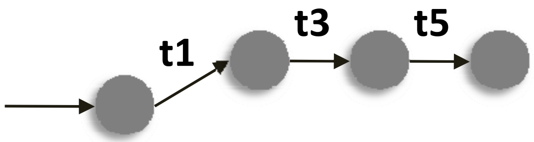
From BPMN to LTS



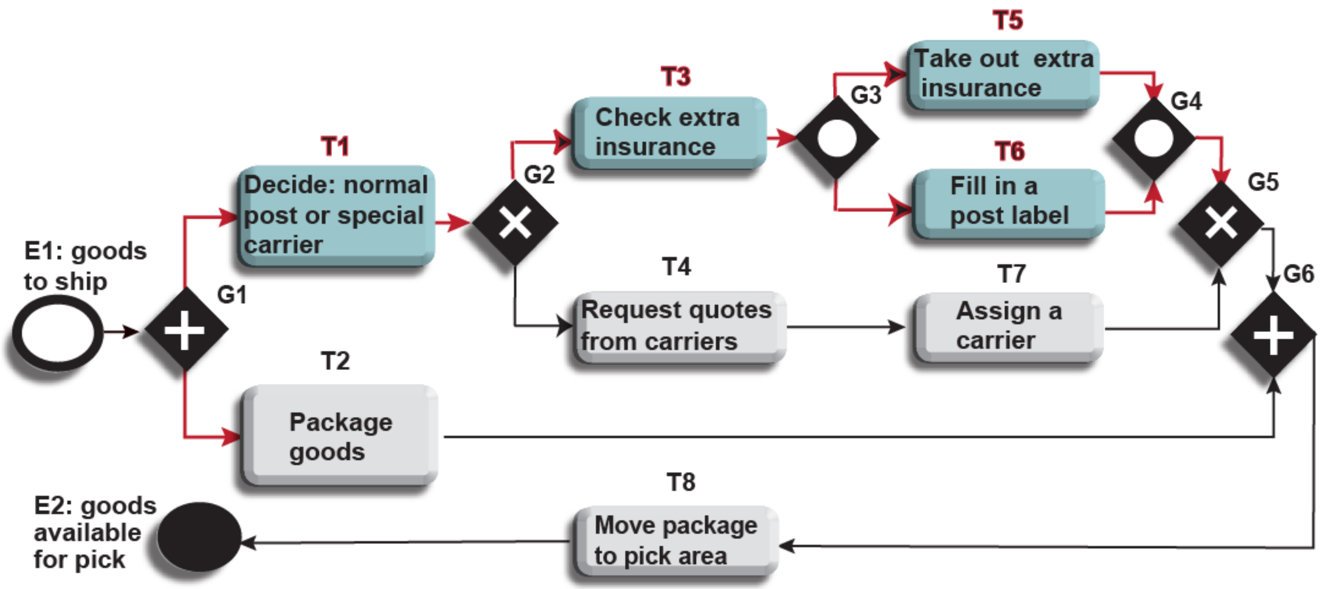


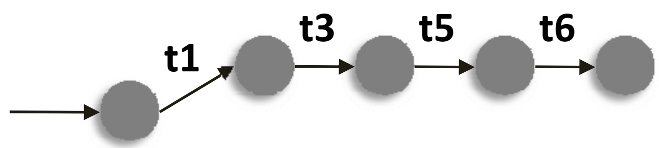
From BPMN to LTS



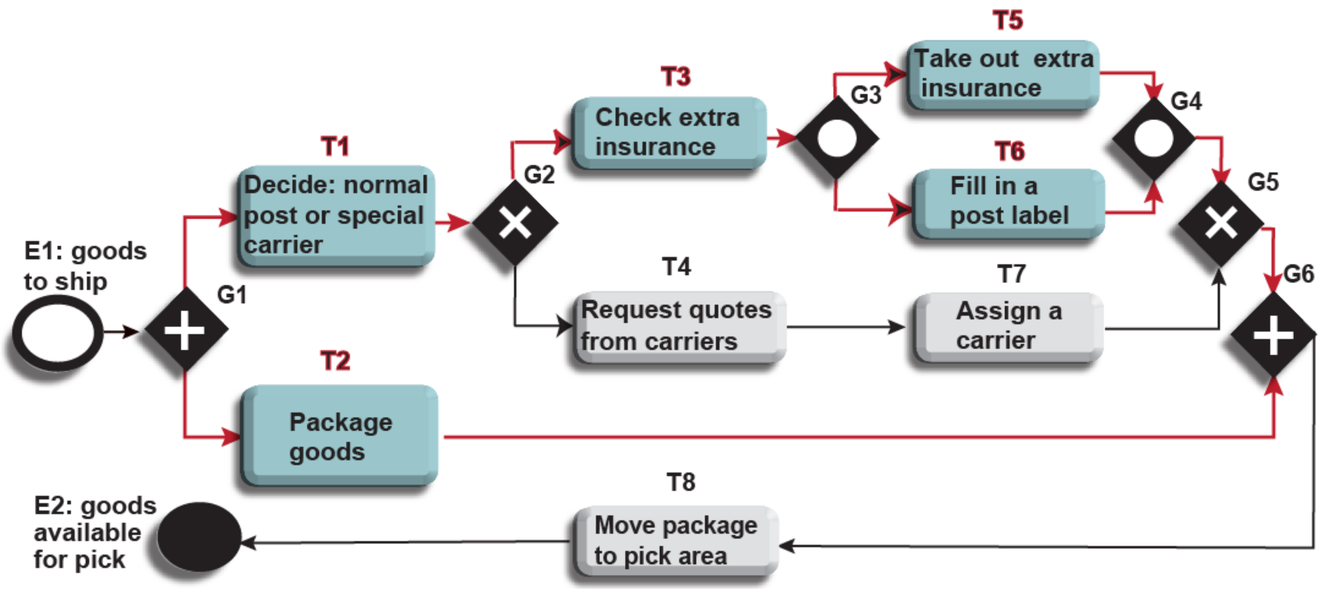


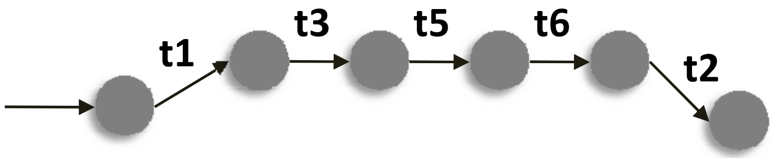
From BPMN to LTS



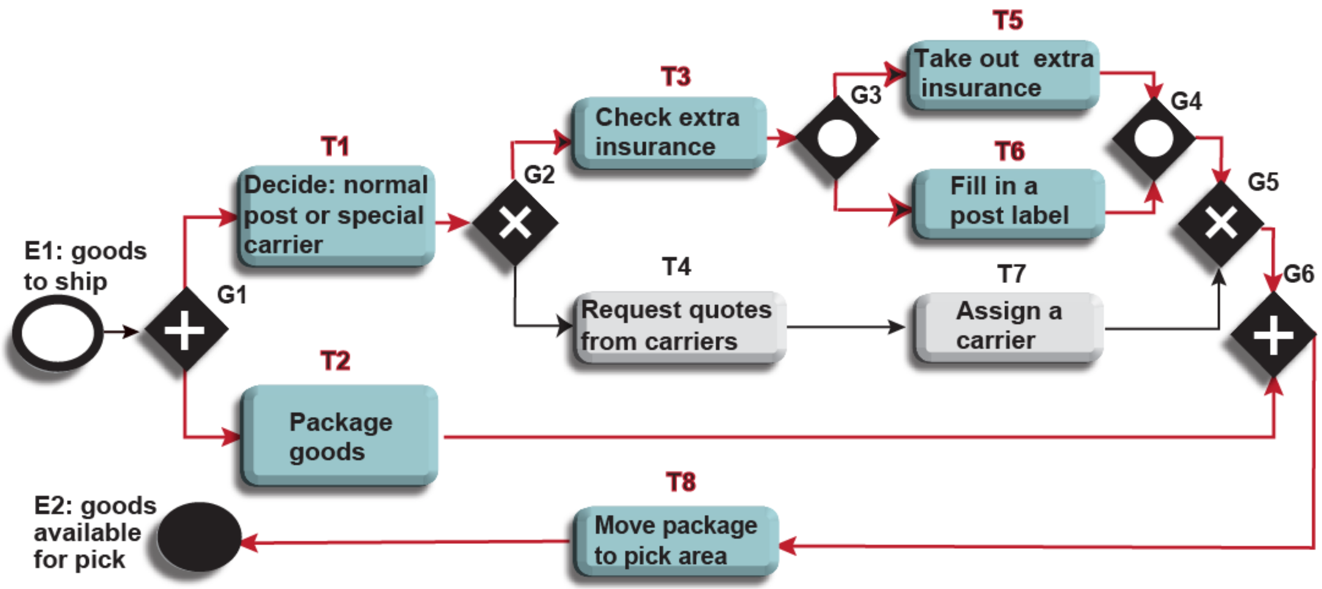


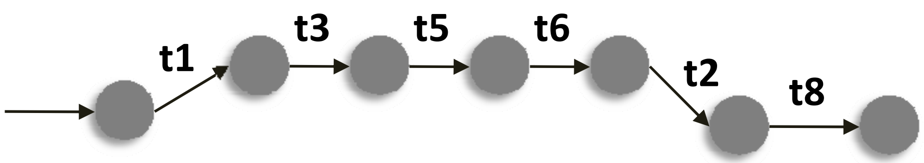
From BPMN to LTS



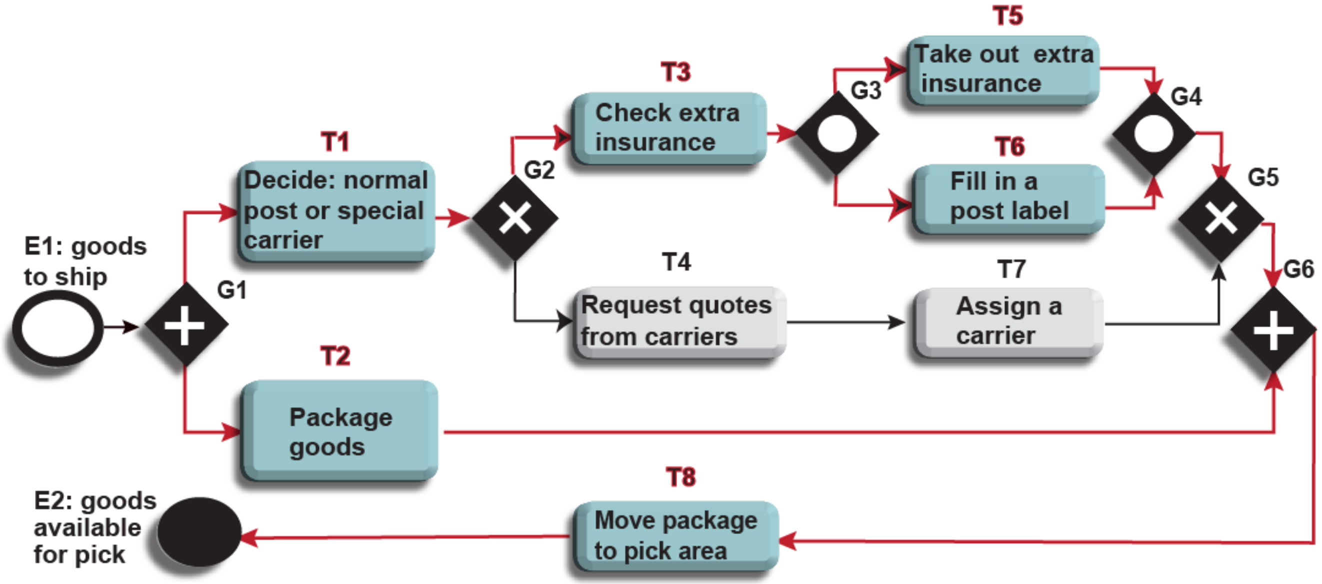


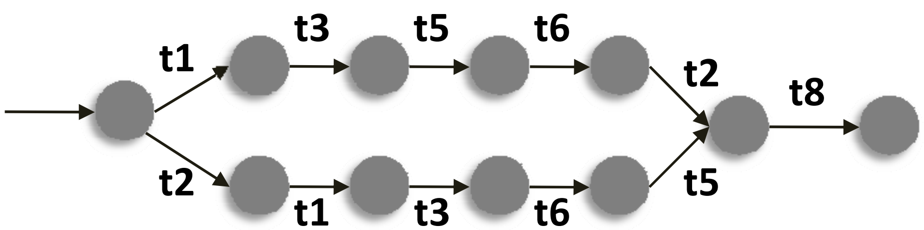
From BPMN to LTS



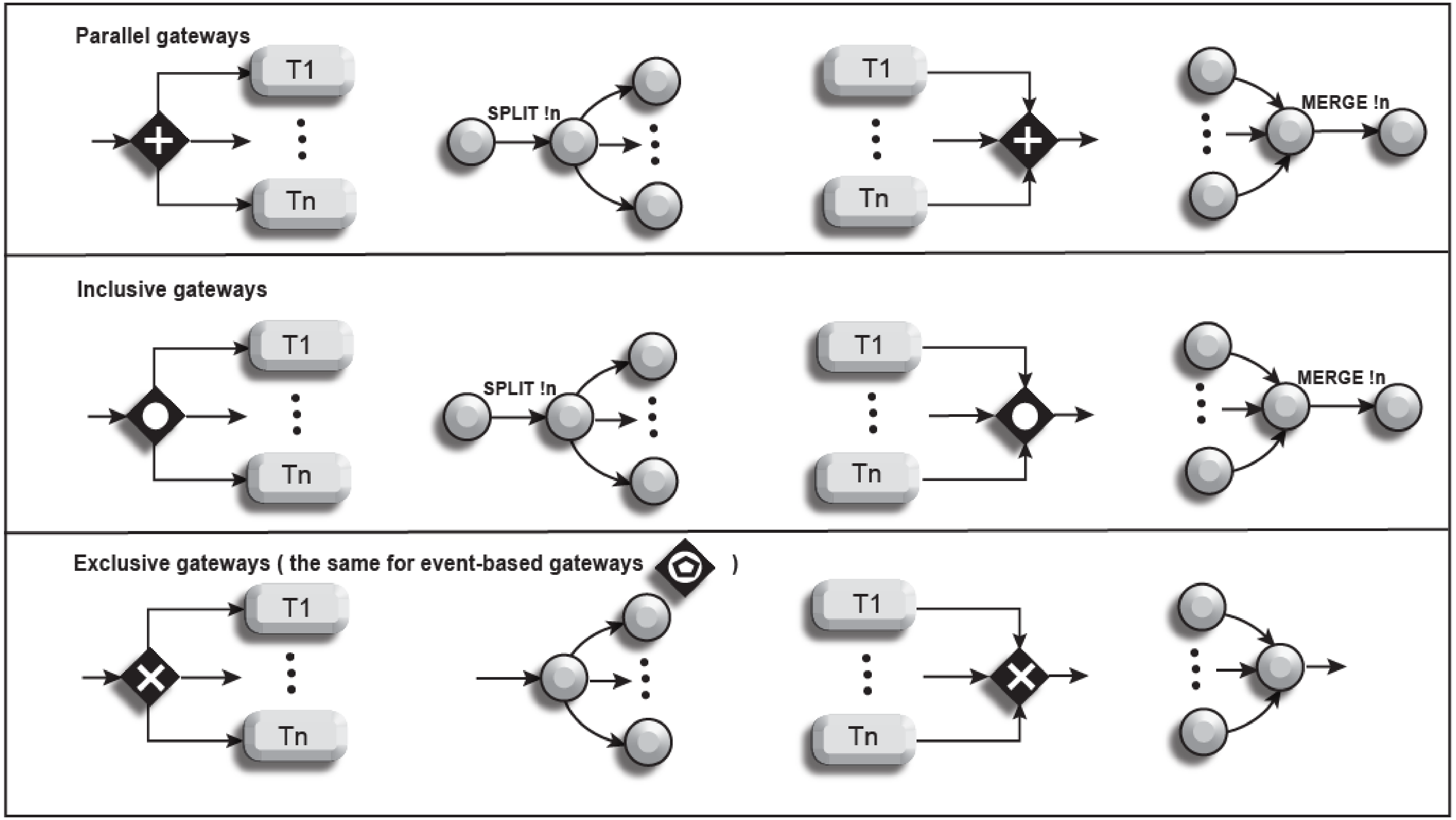


From BPMN to LTS

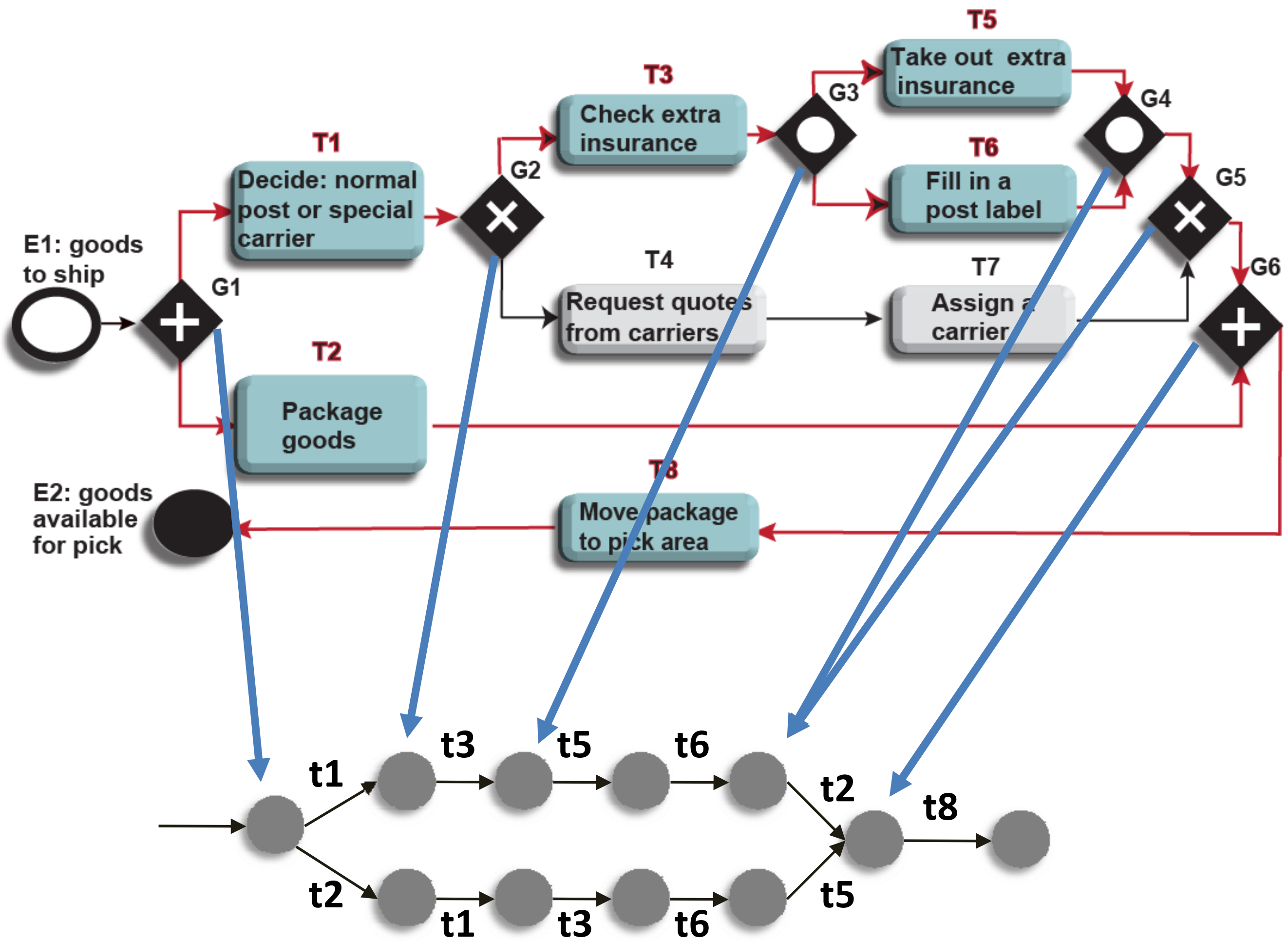




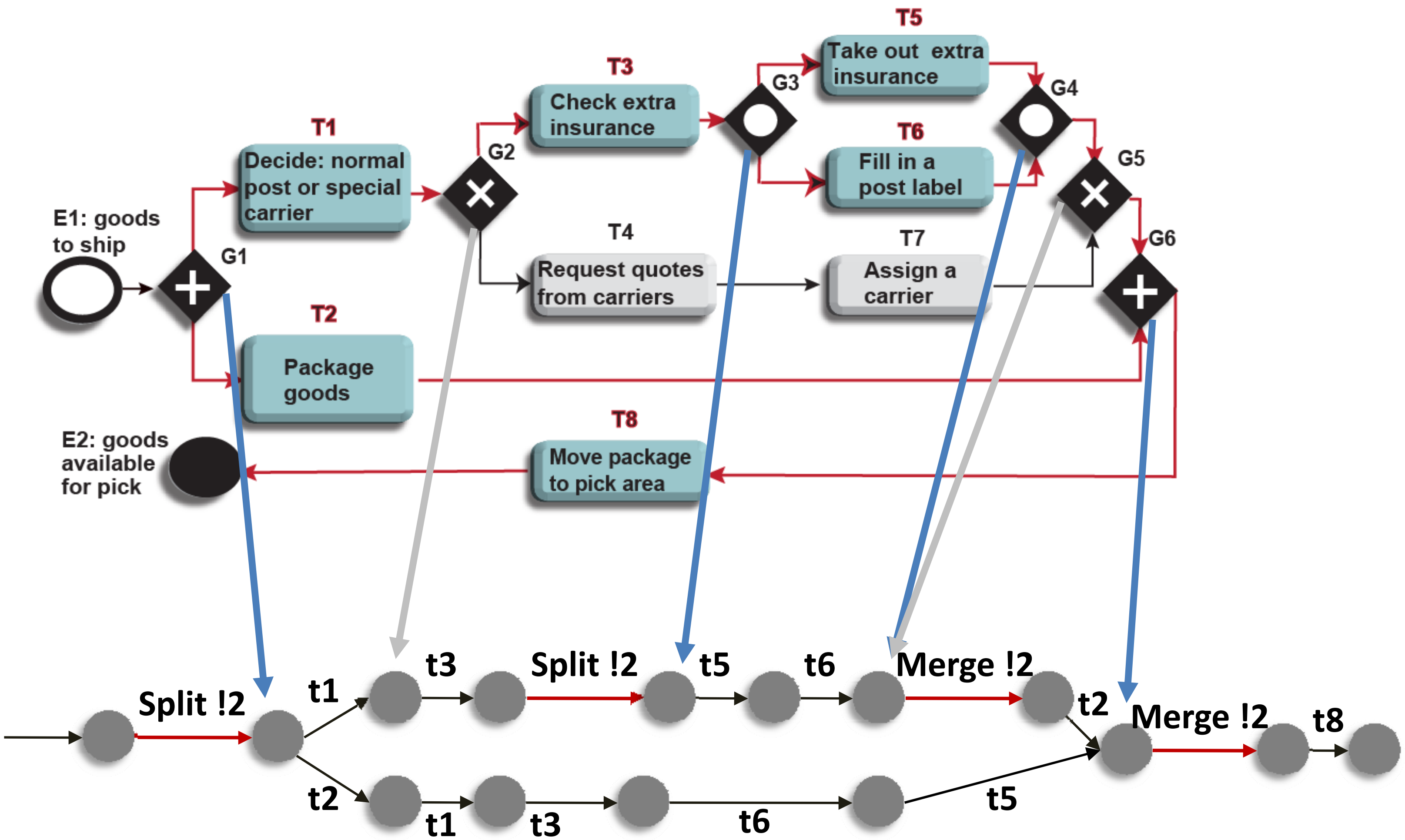
Extended LTS



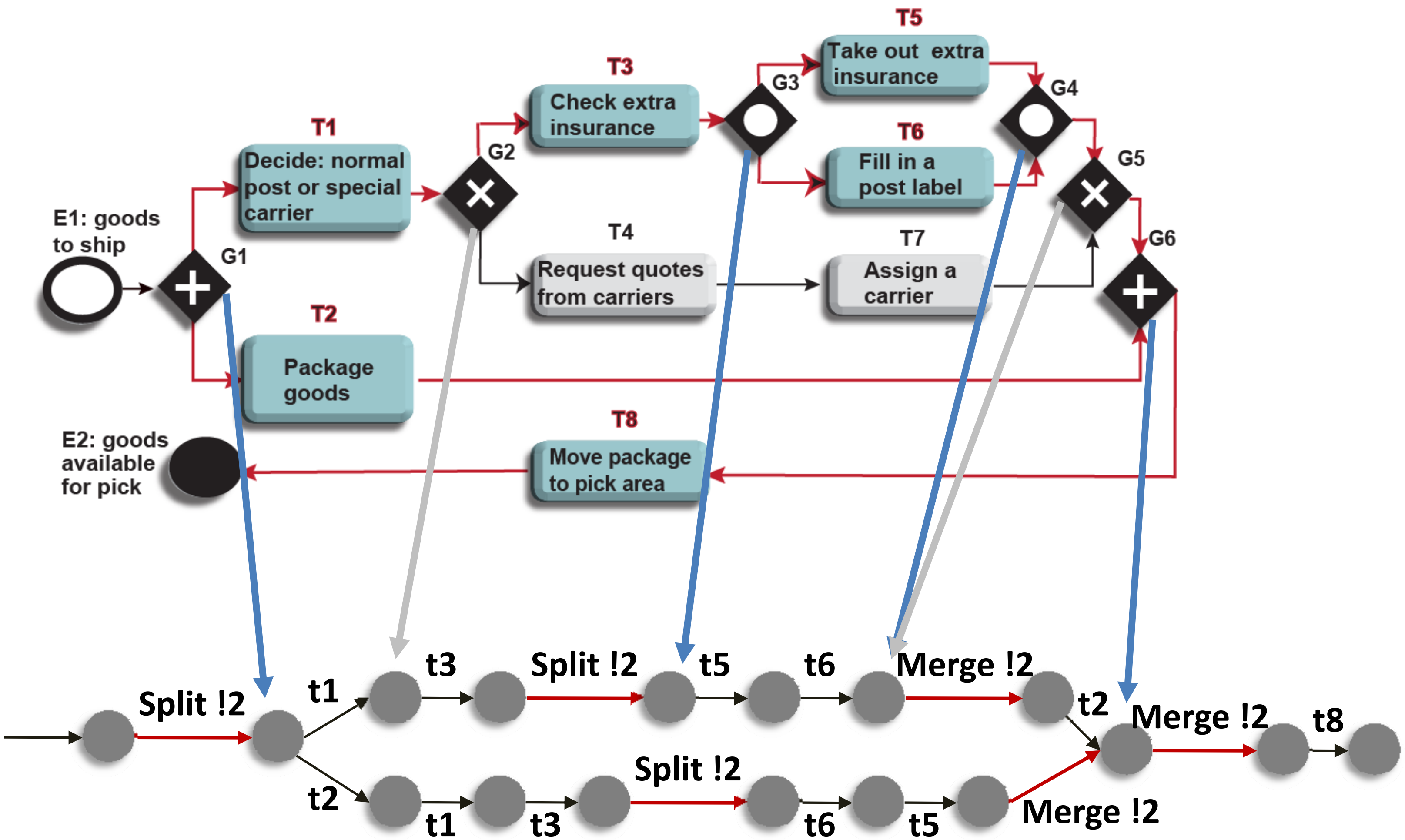
Extended LTS



Extended LTS



Extended LTS



Outline

- Background
- BPMN Model
- Formal semantic model
- Compute the degree of parallelism
- Conclusion and perspectives

Model Checking

Formula 1.

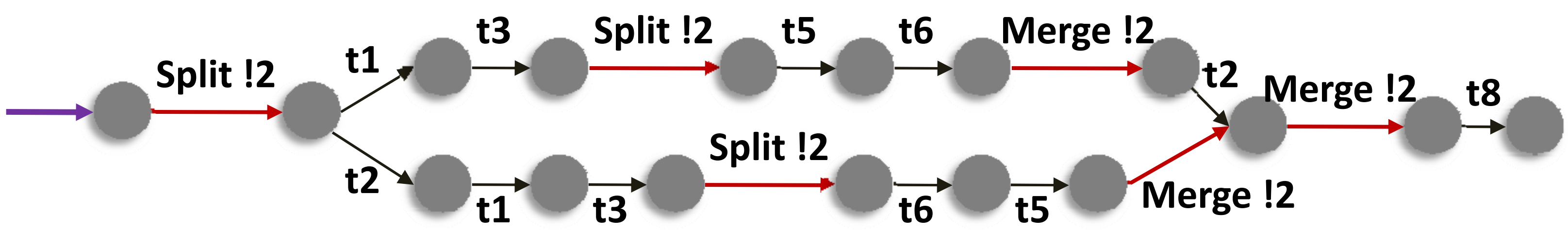
$$SPLIT\_MERGE(N) = \tag{1}$$
$$mu\ X\ (c : Nat := 1). \tag{2}$$
$$((c >= N)\ or \tag{3}$$
$$< SPLIT\ ?\ fan\_out : Nat > X(c + (fan\_out - 1))\ or \tag{4}$$
$$< MERGE\ ?\ fan\_in : Nat > X(c - (fan\_in - 1))\ or \tag{5}$$
$$< not(SPLIT...\ or\ MERGE...) > X(c)) \tag{6}$$

Model Checking

Formula 1.

$$SPLIT\_MERGE(N) = \quad (1)$$
$$mu\ X\ (c : Nat := 1). \quad (2)$$
$$((c >= N)\ or \quad (3)$$
$$< SPLIT\ ?\ fan\_out : Nat > X(c + (fan\_out - 1))\ or \quad (4)$$
$$< MERGE\ ?\ fan\_in : Nat > X(c - (fan\_in - 1))\ or \quad (5)$$
$$< not(SPLIT...\ or\ MERGE...) > X(c)) \quad (6)$$

If N=2, c=1

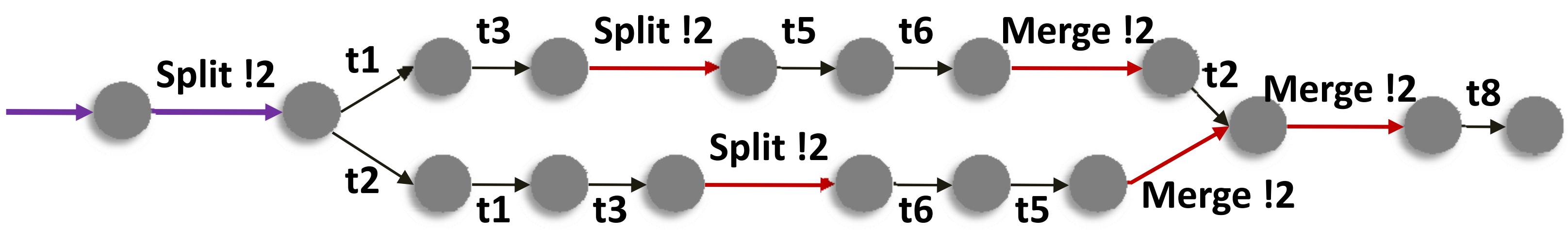


Model Checking

Formula 1.

$$SPLIT\_MERGE(N) = \tag{1}$$
$$mu\ X\ (c : Nat := 1). \tag{2}$$
$$((c >= N)\ or \tag{3}$$
$$< SPLIT\ ?\ fan\_out : Nat > X(c + (fan\_out - 1))\ or \tag{4}$$
$$< MERGE\ ?\ fan\_in : Nat > X(c - (fan\_in - 1))\ or \tag{5}$$
$$< not(SPLIT...\ or\ MERGE...) > X(c)) \tag{6}$$

If N=2, c=2 return true

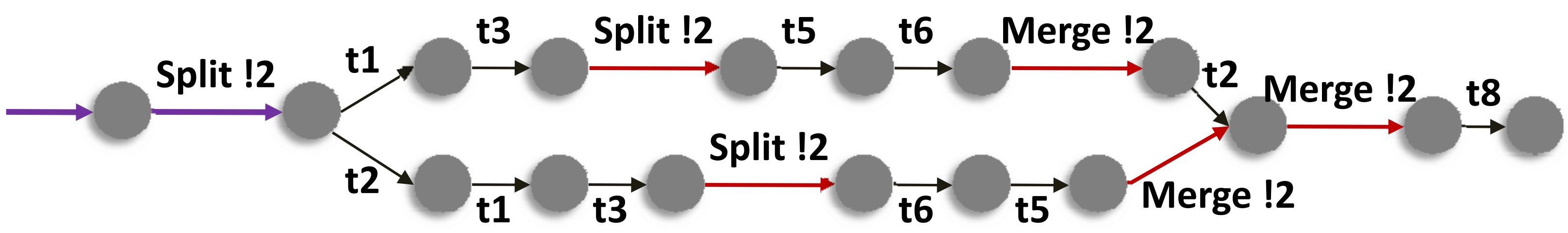


Model Checking

Formula 1.

$$SPLIT\_MERGE(N) = \quad (1)$$
$$mu\ X\ (c : Nat := 1). \quad (2)$$
$$((c >= N)\ or \quad (3)$$
$$< SPLIT\ ?\ fan\_out : Nat > X(c + (fan\_out - 1))\ or \quad (4)$$
$$< MERGE\ ?\ fan\_in : Nat > X(c - (fan\_in - 1))\ or \quad (5)$$
$$< not(SPLIT...\ or\ MERGE...) > X(c)) \quad (6)$$

If N=5, c=2

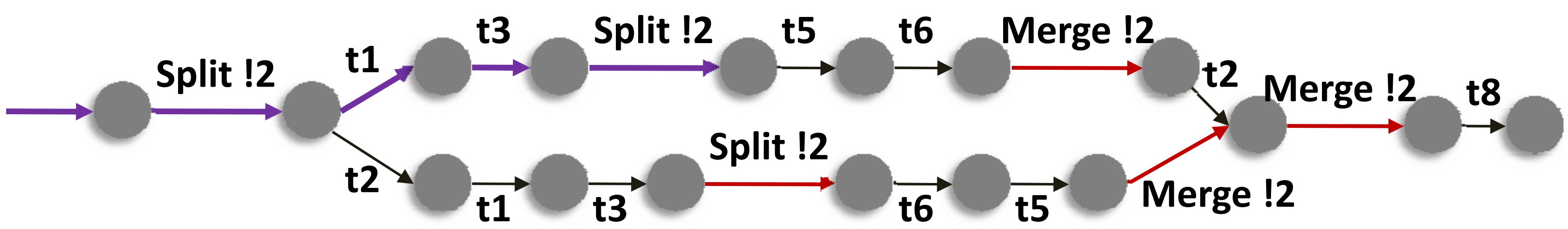


Model Checking

Formula 1.

$$SPLIT\_MERGE(N) = \quad (1)$$
$$mu\ X\ (c : Nat := 1). \quad (2)$$
$$((c >= N)\ or \quad (3)$$
$$< SPLIT\ ?\ fan\_out : Nat > X(c + (fan\_out - 1))\ or \quad (4)$$
$$< MERGE\ ?\ fan\_in : Nat > X(c - (fan\_in - 1))\ or \quad (5)$$
$$< not(SPLIT ...\ or\ MERGE ...) > X(c)) \quad (6)$$

If N=5, c=3

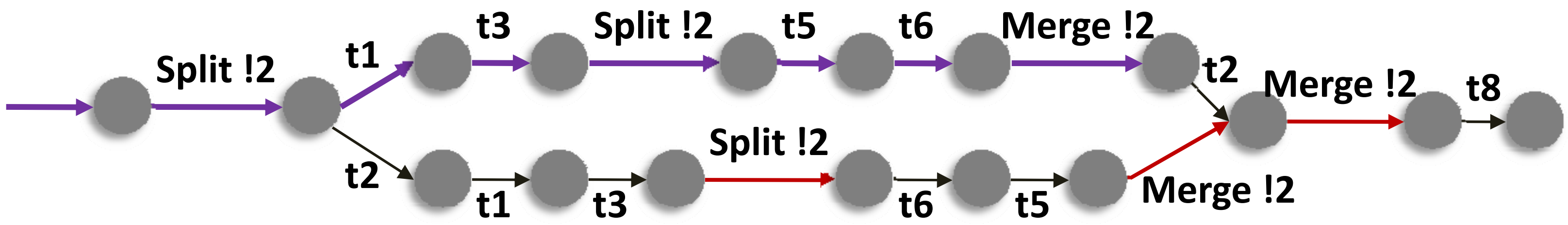


Model Checking

Formula 1.

$$SPLIT\_MERGE(N) = \tag{1}$$
$$mu\ X\ (c : Nat := 1). \tag{2}$$
$$((c >= N)\ or \tag{3}$$
$$< SPLIT\ ?\ fan\_out : Nat > X(c + (fan\_out - 1))\ or \tag{4}$$
$$< MERGE\ ?\ fan\_in : Nat > X(c - (fan\_in - 1))\ or \tag{5}$$
$$< not(SPLIT ...\ or\ MERGE ...) > X(c)) \tag{6}$$

If N=5, c=2

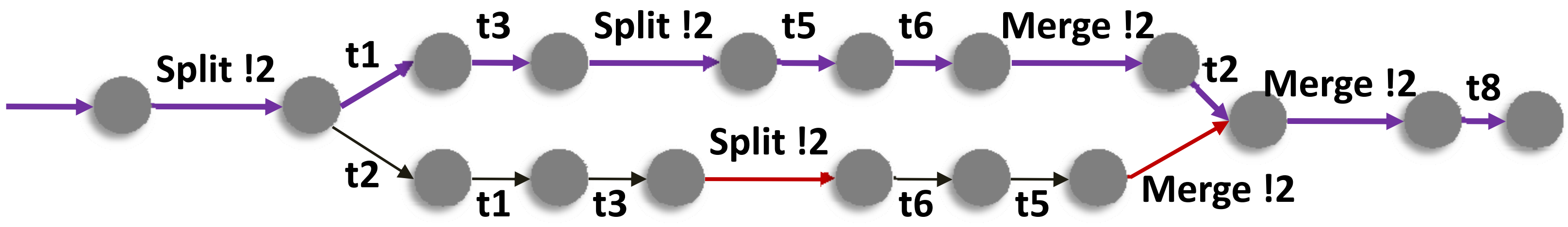


Model Checking

Formula 1.

$$SPLIT\_MERGE(N) = \quad (1)$$
$$mu\ X\ (c : Nat := 1). \quad (2)$$
$$((c >= N)\ or \quad (3)$$
$$< SPLIT\ ?\ fan\_out : Nat > X(c + (fan\_out - 1))\ or \quad (4)$$
$$< MERGE\ ?\ fan\_in : Nat > X(c - (fan\_in - 1))\ or \quad (5)$$
$$< not(SPLIT... \ or\ MERGE...) > X(c)) \quad (6)$$

If N=5, c=1

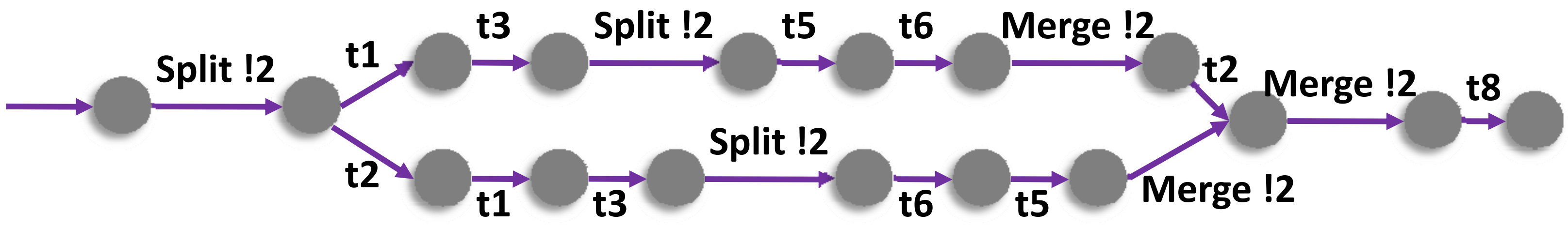


Model Checking

Formula 1.

$$
\begin{aligned}
&SPLIT\_MERGE(N) = && (1)\\
&mu\ X\ (c : Nat := 1). && (2)\\
&((c >= N)\ or && (3)\\
&< SPLIT\ ?\ fan\_out : Nat > X(c + (fan\_out - 1))\ or && (4)\\
&< MERGE\ ?\ fan\_in : Nat > X(c - (fan\_in - 1))\ or && (5)\\
&< not(SPLIT...\ or\ MERGE...) > X(c)) && (6)
\end{aligned}
$$

If N=5, c=1, return false



Model Checking

Formula 1.

$$SPLIT\_MERGE(N) = \quad (1)$$
$$mu\ X\ (c : Nat := 1). \quad (2)$$
$$((c >= N)\ or \quad (3)$$
$$< SPLIT\ ?\ fan\_out : Nat > X(c + (fan\_out - 1))\ or \quad (4)$$
$$< MERGE\ ?\ fan\_in : Nat > X(c - (fan\_in - 1))\ or \quad (5)$$
$$< not(SPLIT \ldots\ or\ MERGE \ldots) > X(c)) \quad (6)$$

Binary search for N to compute final degree

true true true false

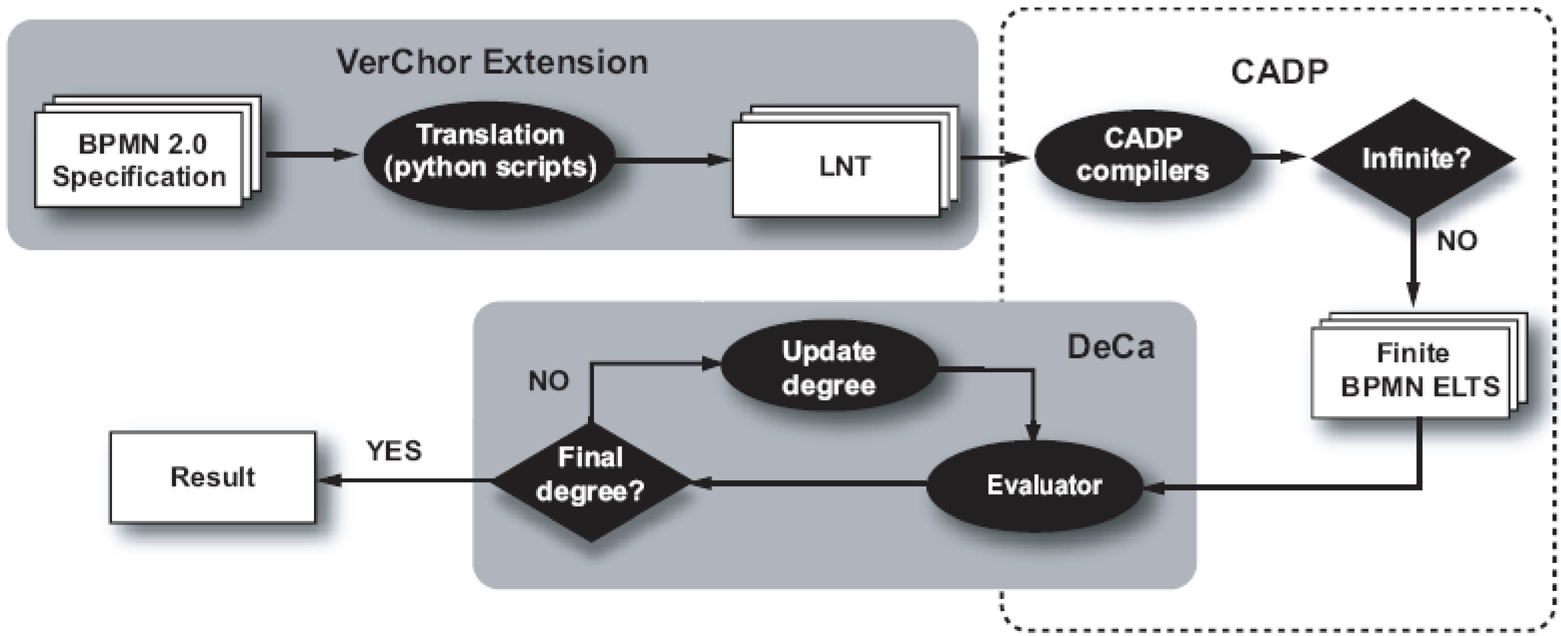
$2^1 \longrightarrow 2^2 \longrightarrow 2^3, \ldots, 2^{n-1} \longrightarrow 2^n \longrightarrow$

if $2^n - 2^{n-1} = 1$: 2^{n-1} is the degree

else $(2^n - 2^{n-1})/2$ ......

until the difference is 1

Tool Support



Experimental Results

BPMN	|TS|	|P|	|I|	|E|	raw ELTS($\|S\|/\|T\|$)	abs. ELTS($\|S\|/\|T\|$)	|D|	time
Shipment [21]	8	2	2	2	56/96	9/11	3	2s01
PizzaOrder [21]	9	2	0	0	30/51	3/2	3	1s53
ChoreOs1 [1]	6	4	0	0	20/26	6/5	3	1s56
ChoreOs2	6	0	4	0	22/37	7/8	3	2s03
BookingSystem [22]	6	1	0	1	12/12	2/1	2	1s23
P010	4		4	2	134/275	30/63	3	3s05
P050	5	6			95/220	8/7	4	2s21
P060	8	4	2		576/1596	40/68	5	3s31
P070	30	2		4	20,745/100,234	3/2	6	3s27
P080	40	2			746,505/4,852,234	3/2	8	3s58
P110	25	4	2	2	32,849/245,932	24/36	15	8s15
P111	45	4	2	2	33,554,513/335,544,492	24/36	15	8s18
P120	31	3	2	4	983,188/9,847,132	14/19	17	7s72

Experimental Results

BPMN	|TS|	|P|	|I|	|E|	raw ELTS($\|S\|/\|T\|$)	abs. ELTS($\|S\|/\|T\|$)	|D|	time
Shipment [21]	8	2	2	2	56/96	9/11	3	2s01
PizzaOrder [21]	9	2	0	0	30/51	3/2	3	1s53
ChoreOs1 [1]	6	4	0	0	20/26	6/5	3	1s56
ChoreOs2	6	0	4	0	22/37	7/8	3	2s03
BookingSystem [22]	6	1	0	1	12/12	2/1	2	1s23
P010	4		4	2	134/275	30/63	3	3s05
P050	5	6			95/220	8/7	4	2s21
P060	8	4	2		576/1596	40/68	5	3s31
P070	30	2		4	20,745/100,234	3/2	6	3s27
P080	40	2			746,505/4,852,234	3/2	8	3s58
P110	25	4	2	2	32,849/245,932	24/36	15	8s15
P111	45	4	2	2	33,554,513/335,544,492	24/36	15	8s18
P120	31	3	2	4	983,188/9,847,132	14/19	17	7s72

Reduction: abstract ELTS with null replacing tasks in raw ELTS

Results: raw ELTS for P111 requires 36m and abs ELTS for P111 requires only 8s

Outline

- Background
- BPMN Model
- Formal semantic model
- Compute the degree of parallelism
- Conclusion and perspectives

Conclusion and Perspectives

Conclusion

- Represent BPMN in terms of ELTSs through LNT
- General enough to handle nested gateways and infinite loops
- Replace tasks with empty statements while keeping control flow labels

Perspectives

- Consider a larger subset of BPMN
- Degree more precise when consider data objects