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

- Apartement purchase application
- Payment
- Issue certificate
- Document archive
- Insurance
- Public notification
Outline

• Background

• BPMN Model
  • Formal semantic model
  • Compute the degree of parallelism
  • Conclusion and perspectives
BPMN Gateways

Parallel gateway

T1: Decide: normal post or special carrier

G1: E1: goods to ship

G2: T2: Package goods

T3: Check extra insurance

G3: T5: Take out extra insurance

G4: T4: Request quotes from carriers

G5: T6: Fill in a post label

G6: T7: Assign a carrier

T8: Move package to pick area

E2: goods available for pick
BPMN Gateways

Parallel gateway

Inclusive gateway

Exclusive gateway

T1: Decide: normal post or special carrier

G1

T2: Package goods

G2

T3: Check extra insurance

G3

T4: Request quotes from carriers

G4

T5: Take out extra insurance

G5

T6: Fill in a post label

G6

T7: Assign a carrier

T8: Move package to pick area

E1: goods to ship

E2: goods available for pick
BPMN Gateways

1. Parallel gateway
   - T1: Decide: normal post or special carrier
   - G1: E1: goods to ship
   - T2: Package goods
   - G2

2. Inclusive gateway
   - T3: Check extra insurance
   - G3
   - T4: Request quotes from carriers
   - G4: T5: Take out extra insurance
   - G5
   - T6: Fill in a post label
   - G6
   - T7: Assign a carrier
   - G7
   - T8: Move package to pick area
   - G8: E2: goods available for pick

3. Exclusive gateway
BPMN Gateways

Parallel gateway

Inclusive gateway

Exclusive gateway

T1: Decide: normal post or special carrier

G1: E1: goods to ship

G2: T2: Package goods

T3: Check extra insurance

G3: G4: T5: Take out extra insurance

T6: Fill in a post label

T7: Assign a carrier

G5: G6: T8: Move package to pick area

E2: goods available for pick
Degree of Parallelism

To compute the degree of parallelism: maximum number of executable tasks at the same time.
Degree of Parallelism

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

Degree: 3
Contributions
Contributions

Process algebra

LNT

Information about parallelism

Degree of parallelism

Model-checking
Outline

• Background

• BPMN Model

• Formal semantic model

• Compute the degree of parallelism

• Conclusion and perspectives
From BPMN to LTS

Decision: normal post or special carrier

Move package to pick area
From BPMN to LTS

[Diagram showing a process flow from E1: goods to ship, through T1: Decide: normal post or special carrier, T2: Package goods, T3: Check extra insurance, G1: Decide normal post or special carrier, G2: Check extra insurance, T4: Request quotes from carriers, G3: T5: Take out extra insurance, G4: T6: Fill in a post label, G5: T7: Assign a carrier, G6: T8: Move package to pick area, E2: goods available for pick, t1, and t3.]
From BPMN to LTS
From BPMN to LTS
From BPMN to LTS
From BPMN to LTS
Extended LTS

Parallel gateways

Inclusive gateways

Exclusive gateways (the same for event-based gateways)
Extended LTS
Extended LTS

E1: goods to ship

Decision: normal post or special carrier

G1

Package goods

T2

T4

Request quotes from carriers

T3

Check extra insurance

G2

T5

Take out extra insurance

G3

T6

Fill in a post label

G4

T7

Assign a carrier

G5

T8

Move package to pick area

G6

E2: goods available for pick

Split !2

t1

t2

Split !2

t3

t4

Split !2

t5

Merge !2

t6

 Merge !2

t7

t8
Outline

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

Formula 1.

\[ \text{SPLIT\_MERGE}(N) = \]
\[ \mu X \ (c : \text{Nat} \ := \ 1). \]  
\[ ((c \geq N) \ or \]
\[ < \text{SPLIT} \ ? \ \text{fan\_out} : \text{Nat} > X(c + (\text{fan\_out} - 1)) \ or \]
\[ < \text{MERGE} \ ? \ \text{fan\_in} : \text{Nat} > X(c - (\text{fan\_in} - 1)) \ or \]
\[ < \text{not(SPLIT... or MERGE...)} > X(c) \]
Model Checking

Formula 1.

\[ SPLIT\_MERGE(N) = \]
\[ \mu X (c : \text{Nat} := 1). \]
\[ ((c \geq N) \text{ or} \]
\[ < SPLIT ? \text{fan}_\text{out} : \text{Nat} > X(c + (\text{fan}_\text{out} - 1)) \text{ or} \]
\[ < \text{MERGE} ? \text{fan}_\text{in} : \text{Nat} > X(c - (\text{fan}_\text{in} - 1)) \text{ or} \]
\[ < \text{not}(SPLIT... \text{ or MERGE...}) > X(c) \]

If \( N=2, \ c=1 \)
Model Checking

Formula 1.

\[
\text{SPLIT\_MERGE}(N) =
\begin{align*}
\text{mu } X & (c : \text{Nat} := 1). \\
((c \geq N) \text{ or } \text{< SPLIT} \ ? \ \text{fan\_out : Nat} > X(c + (\text{fan\_out} - 1)) \text{ or } \text{< MERGE} \ ? \ \text{fan\_in : Nat} > X(c - (\text{fan\_in} - 1)) \text{ or } \text{< not(SPLIT... or MERGE...) > X(c)})
\end{align*}
\]  

If N=2, c=2 return true
Formula 1.

\[
\begin{align*}
\text{SPLIT\_MERGE}(N) & = \\
\mu X \ (c : \text{Nat} := 1). & \quad (1) \\
((c \geq N) \ or & \quad (3) \\
< \text{SPLIT} \ ? \ fan\_out : \text{Nat} > X(c + (fan\_out - 1)) \ or & \quad (4) \\
< \text{MERGE} \ ? \ fan\_in : \text{Nat} > X(c - (fan\_in - 1)) \ or & \quad (5) \\
< \text{not}(\text{SPLIT}... \ or \ \text{MERGE}...) > X(c) & \quad (6) \\
\end{align*}
\]

If $N=5$, $c=2$
Model Checking

Formula 1.

\[ SPL\_\text{MERGE}(N) = \]
\[ \mu X \ (c : \text{Nat} := 1). \]
\[ ((c \geq N) \text{ or} \]
\[ < \text{SPLIT} \ \text{fan\_out} : \text{Nat} > X(c + (\text{fan\_out} - 1)) \text{ or} \]
\[ < \text{MERGE} \ \text{fan\_in} : \text{Nat} > X(c - (\text{fan\_in} - 1)) \text{ or} \]
\[ < \text{not(SPLIT... or MERGE...)} > X(c) \]

If \( N=5, \ c=3 \)
Model Checking

Formula 1.

\[ \text{SPLIT\_MERGE}(N) = \]
\[ \mu X \ (c : \text{Nat} := 1). \]
\[ ((c \geq N) \ or \]
\[ < \text{SPLIT} \ ? \ \text{fan\_out} : \text{Nat} > X(c + (\text{fan\_out} - 1)) \ or \]
\[ < \text{MERGE} \ ? \ \text{fan\_in} : \text{Nat} > X(c - (\text{fan\_in} - 1)) \ or \]
\[ < \not(\text{SPLIT... or MERGE...}) > X(c) \]

If \( N=5, \ c=2 \)
Formula 1.

\[
\begin{align*}
SPLIT\_MERGE(N) = & \quad \mu X \ (c : Nat := 1). \\
\ (c \geq N) \ or & \quad \langle SPLIT \ ? \ fan\_out : Nat > X(c + (fan\_out - 1)) \ or & \quad \langle MERGE \ ? \ fan\_in : Nat > X(c - (fan\_in - 1)) \ or & \quad \langle not(SPLIT... \ or \ MERGE...) > X(c) \\
\end{align*}
\]  

If \(N=5,\ c=1\)
Model Checking

Formula 1.

\[ SPLIT\_MERGE(N) = \]
\[ \mu X (c : Nat := 1). \]
\[ ((c >= N) \ or \]
\[ < SPLIT ? fan\_out : Nat > X(c + (fan\_out - 1)) \ or \]
\[ < MERGE ? fan\_in : Nat > X(c - (fan\_in - 1)) \ or \]
\[ < not(SPLIT... or MERGE...) > X(c) \]

If N=5, c=1, return false
Model Checking

Formula 1.

\[ \text{SPLIT\_MERGE}(N) = \]
\[ \mu X (c : \text{Nat} := 1). \]
\[ ((c >= N) \text{ or} \]
\[ < \text{SPLIT} ? \text{fan\_out : Nat} > X(c + (\text{fan\_out} - 1)) \text{ or} \]
\[ < \text{MERGE} ? \text{fan\_in : Nat} > X(c - (\text{fan\_in} - 1)) \text{ or} \]
\[ < \text{not(SPLIT... or MERGE...)} > X(c) \]

Binary search for N to compute final degree

<table>
<thead>
<tr>
<th>true</th>
<th>true</th>
<th>true</th>
<th>false</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^1$</td>
<td>$2^2$</td>
<td>$2^3$</td>
<td>...</td>
</tr>
<tr>
<td>$2^{n-1}$</td>
<td>$2^n$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

- BPMN 2.0 Specification
- Translation (python scripts)
- LNT
- VerChor Extension
- CADP compilers
- Infinite?
- NO
- Finite BPMN ELTS
- CADP
- Update degree
- NO
- Final degree?
- YES
- Evaluator
## Experimental Results

| BPMN              | |S| | | | | 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     | 6     | 4     | 2     | 134/275       | 30/63          | 3     | 3s05 |
| P050              | 5     | 6     | 6     | 2     | 95/220        | 8/7            | 4     | 2s21 |
| P060              | 8     | 4     | 2     | 2     | 576/1596      | 40/68          | 5     | 3s31 |
| P070              | 30    | 2     | 4     | 4     | 20,745/100,234| 3/2            | 6     | 3s27 |
| P080              | 40    | 2     | 2     | 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                  | |S| |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 | 2 | 134/275 | 30/63 | 3               | 3s05 |
| P050                 | 5 | 6 | 95/220 | 8/7 | 4               | 2s21 |
| P060                 | 8 | 4 | 576/1596 | 40/68 | 5               | 3s31 |
| P070                 | 30 | 2 | 20,745/100,234 | 3/2 | 6               | 3s27 |
| P080                 | 40 | 2 | 746,505/4,852,234 | 3/2 | 8               | 3s58 |
| P110                 | 25 | 4 | 32,849/245,932 | 24/36 | 15              | 8s15 |
| P111                 | 45 | 4 | 33,554,513/335,544,492 | 24/36 | 15              | 8s18 |
| P120                 | 31 | 3 | 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