Process Discovery and Conformance Checking Using Passages

2nd Belgian Process Mining Research Day
Ghent University, September 24th, 2012

Wil van der Aalst
www.processmining.org
Big Data: Even Dilbert and the "pointy-haired boss" know about it ...
Distributing/Decomposing Process Mining Problems
Big Data: Opportunities and Challenges
What if?

- There are more than 100,000,000 events?
- There are more than 1000 different activities?
- There are more than 1,000,000 cases?
Distributed Computing

- multicore CPU
- manycore GPU
- cluster computing
- grid computing
- cloud computing
- ...

\[2^{40/2} \approx 1.000.0000 \text{ times faster, bigger, etc.}\]
How to distribute/decompose process discovery?
How to distribute/decompose process discovery?
How to distribute/decompose conformance checking?

?
How to distribute/decompose conformance checking?

b is often skipped

f occurs too often
Replication: Same event log on all computing nodes

Only makes sense if random elements, e.g., genetic process mining.
Classification based on partitioning of event log: vertical and horizontal
Vertical distribution I: Split cases arbitrarily

sets of cases

sets of cases
Vertical distribution II: Split cases based on a specific feature
Horizontal distribution

sets of activities

\[ \text{abcdeg} = \text{abeg} + \text{bcde} \]
Horizontal distribution: The key idea

Projected on a, b, e, f, g

Projected on b, c, d, e
Passage $P=(X, Y)$

$\emptyset \neq X \subseteq N$

$\emptyset \neq Y \subseteq N$

causal dependency: may trigger or enable

$X \overset{G}{\bullet} = Y$

$X = \overset{G}{\bullet} Y$
Minimal passages

A passage is minimal if it does not contain smaller passages.
Passages define an equivalence relation on the edges in the graph.
Minimal passage 1: \((X,Y) = (\{a\},\{b,c\})\)
Minimal passage 2: \((X,Y) = (\{b,c,d\}, \{d,e,f\})\)

\[ X = \{b, c, d\} \quad Y = \{d, e, f\} \]
Minimal passage 3: \((X,Y) = (\{e\},\{g\})\)
Minimal passage 4: $(X,Y) = (\{f\},\{h\})$
Minimal passage 5: \((X,Y) = (\{g,h\},\{i\})\)
So What?

- Any process model can be partitioned in minimal passages.
- Claim: *Discovery and conformance checking can be done per passage!*

Clouds may contain arbitrary subprocesses not explicitly recorded in the event log (invisible activities or small networks used for routing, e.g. XOR/AND/OR-split/joins)
Let Us Get A Bit More Technical …
Union of two passages

\[ P_1 \cup P_2 = (X_1 \cup X_2, Y_1 \cup Y_2) \]
Difference of two passages

\[ P_1 \setminus P_2 = (X_1 \setminus X_2, Y_1 \setminus Y_2) \]
Properties of passages

If

\[ P_1 = (X_1, Y_1) \in \text{pas}(G) \]
\[ P_2 = (X_2, Y_2) \in \text{pas}(G) \]

then:

\[ P_3 = P_1 \setminus P_2 \text{ is a passage if } P_3 \neq (\emptyset, \emptyset), \]
\[ P_4 = P_2 \setminus P_1 \text{ is a passage if } P_4 \neq (\emptyset, \emptyset), \]
\[ P_5 = P_1 \cap P_2 \text{ is a passage if } P_5 \neq (\emptyset, \emptyset), \] and
\[ P_6 = P_1 \cup P_2 \text{ is a passage.} \]
Passage partitioning

- $P_1, P_2, \ldots, P_n$ is a passage partitioning if and only if
  - $P_1, P_2, \ldots, P_n$ are passages,
  - the passages are disjoint, and
  - the passages cover all edges.

5 passages
More examples of passage partitionings

2 passages
({{a},{b,c}})

4 passages
({{a},{b,c}}) → ({b,c,d},{d,e,f})
({{e,g,h},{g,i}}) → ({f},{h})

2 passages
({{b,c,d},{b,c,d,e,f}})

4 passages
({{b,c,d},{d,e,f}})

({{a,e,f,g,h},{b,c,g,h,i}})

({{e,f,g,h},{g,h,i}})

({{a,b,c,d},{b,c,d,e,f}})
Minimal passages

• A passage is minimal if it does not contain smaller passages.
• Each edge is involved in precisely one minimal passage.
• Passages are composed of minimal passages.
• The minimal passages form a passage partitioning.
Decomposing Petri nets using passages

start

register request

examine thoroughly

c1

examine casually

c3

d

check ticket

c2

do

decide

c5

a

request

b

examine thoroughly

c

examine casually

d

check ticket

c

decide

c

pay compensation

g

reject request

h

pay compensation

e

reinitiate request

f

register request

c

c

c

c

c

c

c

c

c

c

c

Decomposing Petri nets using passages

- a: register request
- b: examine thoroughly
- c1, c3: examine casually
- d: check ticket
- e: decide
- f: initiate request, reinitiate request
- g: pay compensation
- h: reject request

Connections:
- a → e: register request
- e → g: decide → pay compensation
- e → h: decide → reject request
- d → c: check ticket
- c → e: examine casually
- e → f: decide → initiate request
- f → e: reinitiate request
- e → a: decide
- a → e: register request
- e → g: pay compensation
- g → h: pay compensation
- h → e: reject request
- e → f: reinitiate request
- f → e: reinitiate request
Conformance checking can be decomposed

Theorem 5.4. (Conformance Checking Can be Decomposed)
Let \( L \in \mathcal{B}(A^*) \) be an event log and let \( SN = (PN, M_i, M_o) \) be a connected system net. For any passage partitioning \( \{P_1, P_2, \ldots, P_n\} \) of \( \text{ske}(PN) \): \( L \) is perfectly fitting system net \( SN \) if and only if for all \( 1 \leq i \leq n \): \( L|_{P_i} \) is perfectly fitting \( SN^i = (PN_{P_i}, [], []) \).

“The event log fits all passages if and only if the event log fits the whole model.”

Key insight: interface transitions controlled by event log
Conformance checking

acefl
acdddefl
abdefl
acdddefl
acefl
abefl
...
Create Skeleton

a: book car
b: skip extra insurance
c: add extra insurance
d: change booking
e: confirm
f: initiate check-in
g: add extra insurance
h: skip extra insurance
i: check driver's license
j: charge credit card
k: supply car
l: supply car

Diagram:
- Blue: book car
- Red: change booking
- Green: confirm
- Purple: initiate check-in
- Black: supply car

Additional node: conformance checking
Suppose d is executed too late
Decomposed/distributed quantification of conformance: Two examples

Theorem 5.6. (Lower Bound for Misalignment Costs)
Let $L \in \mathcal{B}(A^*)$ be an event log and let $SN = (PN, M_i, M_o)$ be a connected system net. For any passage partitioning $Q = \{P_1, P_2, \ldots, P_n\}$ of $skel(PN)$:

$$\text{costs}(L, SN, \delta) \geq \sum_{1 \leq i \leq n} \text{costs}(\overline{L} \upharpoonright P_i, SN^i, \delta_Q)$$

where $SN^i = (PN^{P_i}, [,], [])$.

“bound on misalignment costs”

Theorem 5.7. (Fraction of Perfectly Fitting Traces)
Let $L \in \mathcal{B}(A^*)$ be an event log and let $SN = (PN, M_i, M_o)$ be a connected system net. For any passage partitioning $Q = \{P_1, P_2, \ldots, P_n\}$ of $skel(PN)$:

$$\frac{||\{\sigma \in L \mid \sigma \in \tau(SN)\}||}{|L|} = \frac{||\{\sigma \in \overline{L} \mid \forall 1 \leq i \leq n \sigma \upharpoonright P_i \in \tau(SN^i)\}||}{|L|}$$

“exact value for percentage of fitting traces”
Discovery example
Limitations

- Need to discover causal dependencies first (only issue for discovery, use fuzzy/heuristic rules).
- “Interface transitions” need to have a unique label.
- Minimal passages may be large in dense graphs.
“Almost passages”

- balance between size and quality
- discard same arcs in all passages (use an iterative procedure)
- also adding arcs? (don't think so)

Goal: small passages with only low-frequent arcs violating the rule.
Extended passages

- abstraction yields passage
- internal transitions may be silent or unique
Tool Support in ProM
Tool Support in ProM (implemented by Eric Verbeek)

- discovery
- conformance
Passage-Based Conformance Checking
Result

11 passages

passage graph
Example Passage
Process Model has 11 Passages
Diagnostics per Passage

no problems
Problematic Passage
Discovery (no initial model, just events)

algorithm used to derive causal graph (to determine passages)

discovery algorithm (applied per passage)
Result (alpha + ILP with proper completion)

sound WF-net perfectly fitting the event log
Welcome to BPM 2013

Posted by bpm on July 4, 2012

BPM 2013 is the eleventh edition of the reference conference for researchers and practitioners in the field of Business Process Management (BPM). The conference covers all aspects of BPM, including theory, models, techniques, architectures, systems, and empirical studies, and engages the most renowned representatives of the BPM community worldwide in talks, tutorials, and scientific discussions.

BPM 2013 will take place in Beijing, the capital of China, and it will be the first edition of the BPM conference series in Asia. Next to the world-famous sites of the Great Wall, the Summer Palace, the Forbidden City, and Tiantan – four world cultural heritage sites recognized by UNESCO – Beijing hosts a constantly growing concentration of cutting-edge, industrial research labs and academic institutions from all over the world, making it a melting pot of creativity and innovation.

http://bpm2013.tsinghua.edu.cn/

Call for Workshop Proposals

Posted by bpm on August 17, 2012