

Meronymy-based Aggregation of Activities in Business Process Models

Sergey Smirnov¹, Remco Dijkman²,
Jan Mendling³, and Mathias Weske¹

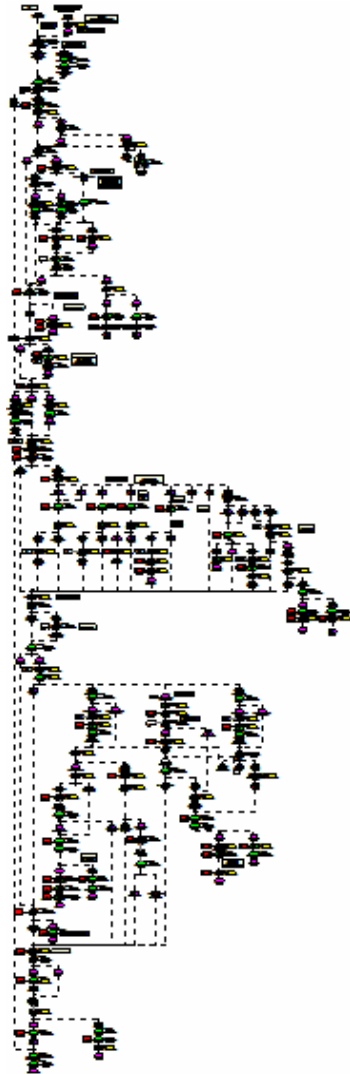
¹ Hasso Plattner Institute, Germany

² Eindhoven University of Technology, The Netherlands

³ Humboldt-Universität zu Berlin, Germany

Motivation

2



- > 300 nodes
- > 150 activities

Business Process Model Abstraction



3

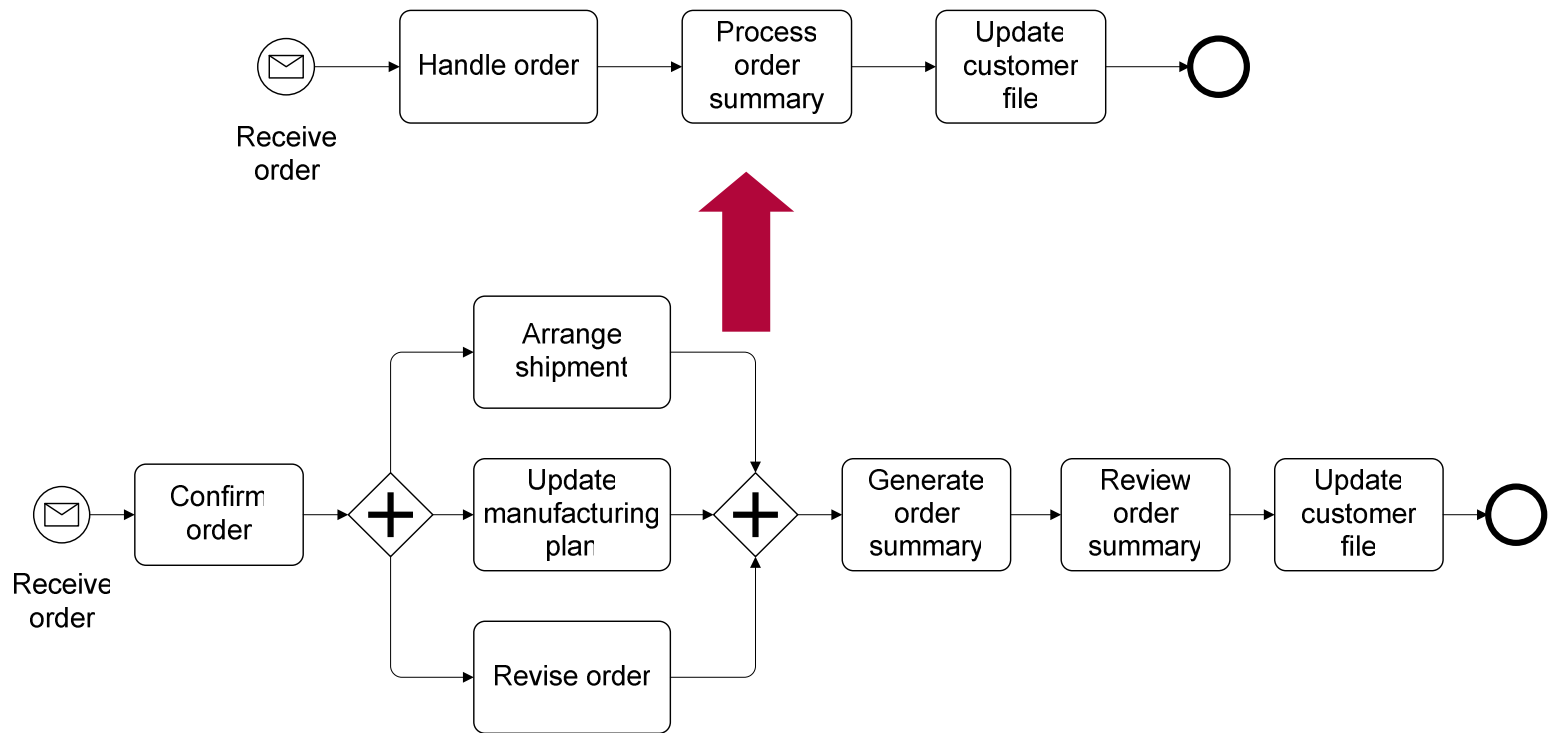
... is an operation on a business process model preserving essential process properties and leaving out insignificant process details in order to retain information relevant for a particular purpose

BPMA Scenario

4

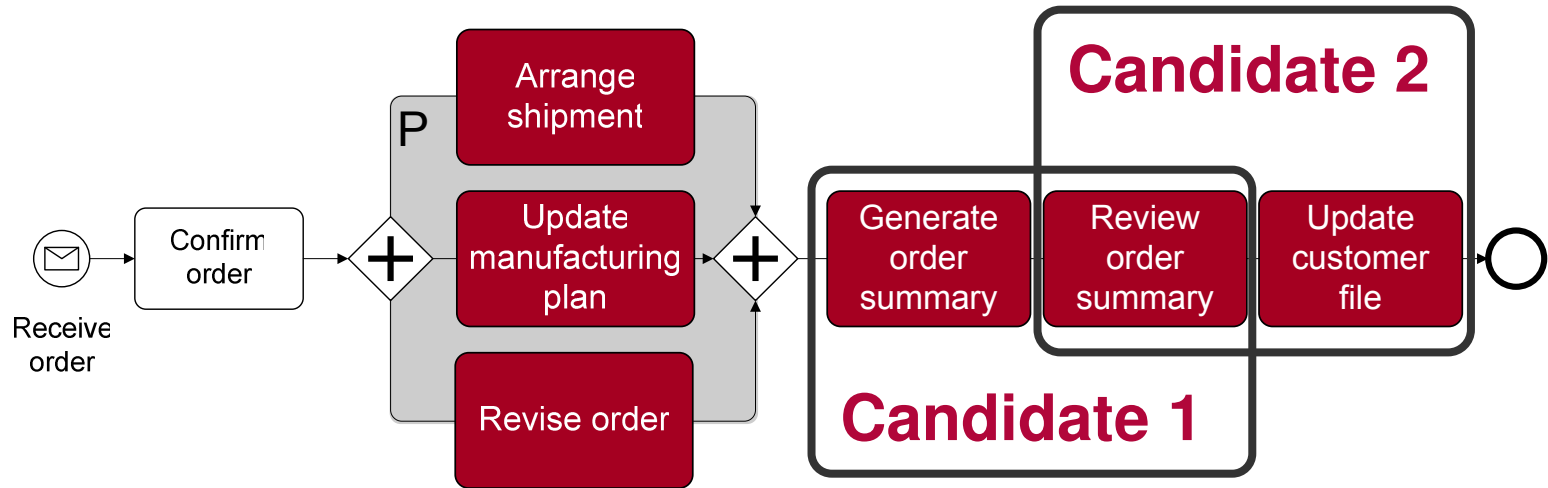
Abstraction objects = Activities

Abstraction operation = Aggregation



Structural BPMA Challenges

5



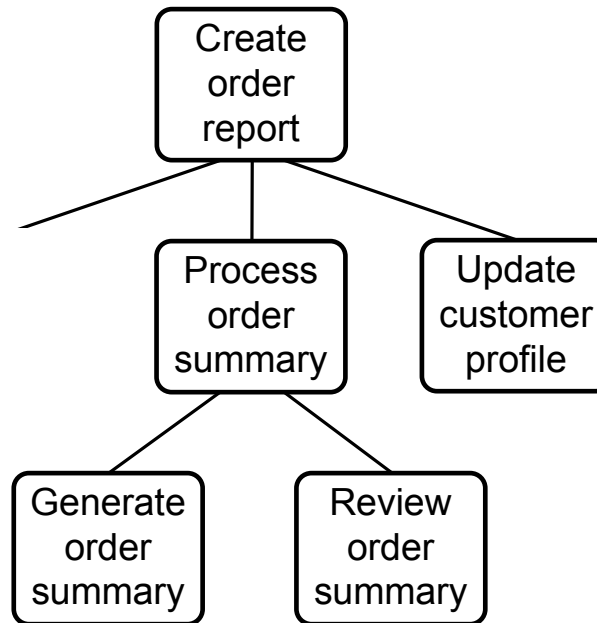
Candidate 1



Candidate 2

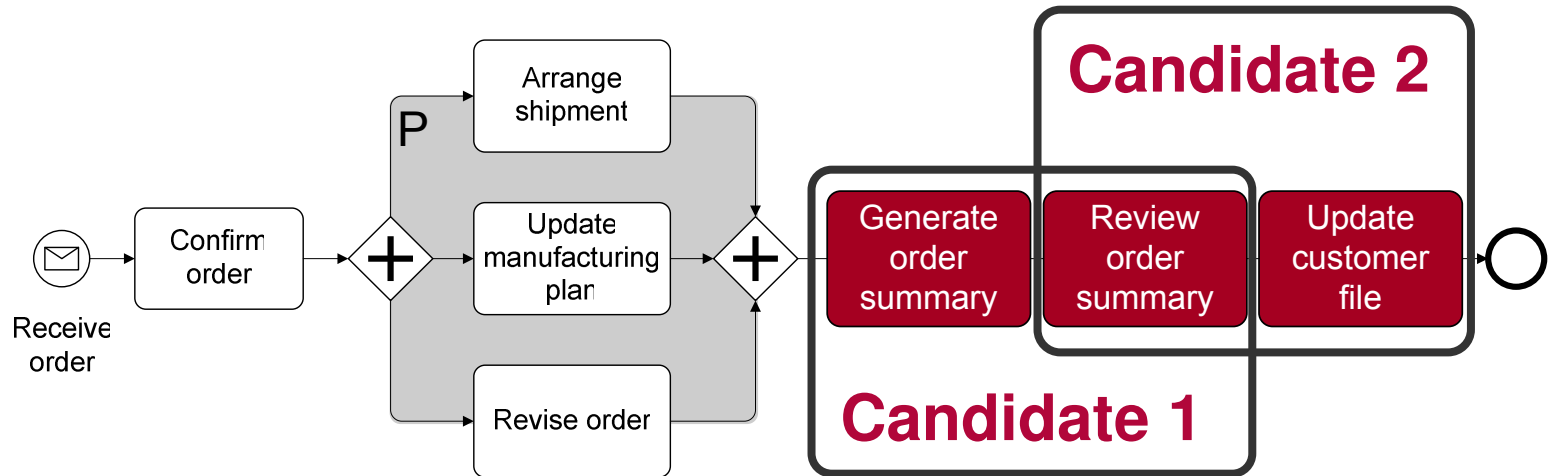
Activity Ontology

6

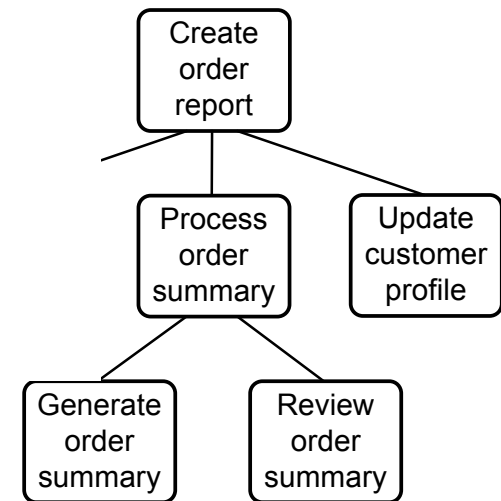


Role of Activity Ontology in BPMA

7



Candidate 1 ? **Candidate 2**



Aggregation Mining Idea

8

Input: Process model + Ontology

Output: Aggregations

Algorithm Sketch:

FOR each *aggregation candidate*

map each *aggregation candidate* activity to
an ontology activity

IF (*ontology activities* are strongly related)

aggregation candidate is an aggregation

How to find an aggregation candidate efficiently?

How to judge on ontology activity relatedness?

Activity Alphabet

9

A

Create
order
report

Process
order
summary

Update
customer
profile

Generate
order
summary

Review
order
summary

Process Model

10

$PM = (A, G, E)$ is a process model, where:

- $A \subseteq \mathcal{A}$ finite non-empty set of activities
- G finite set of gateways
- $A \cap G = \emptyset$
- $N = A \cup G$ finite set of nodes
- $E \subseteq N \times N$ the flow relation
- (N, E) a connected graph

Aggregation Candidate

11

In process model $PM = (A, G, E)$

$C \subseteq A$ is an aggregation candidate.

Meronymy Tree

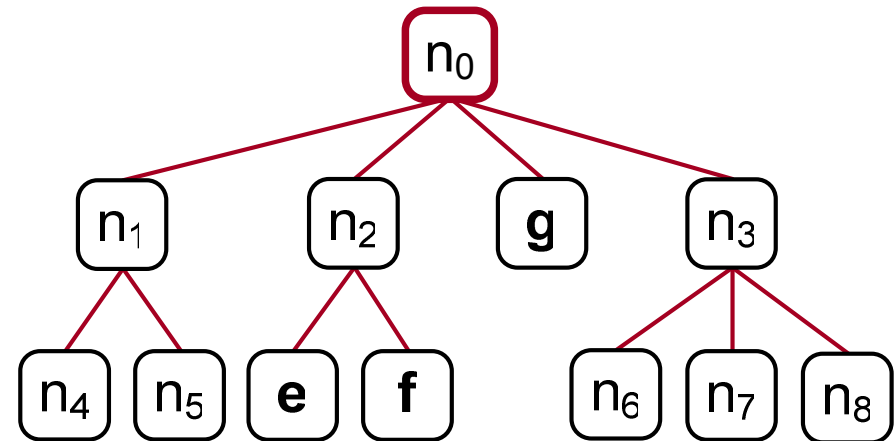
12

Meronymy tree is a tuple $t = (A_t, r_t, M_t)$

$$A_t \subseteq \mathcal{A}$$

$$r_t \in A_t$$

$$M_t \subseteq A_t \times (A_t \setminus r_t)$$

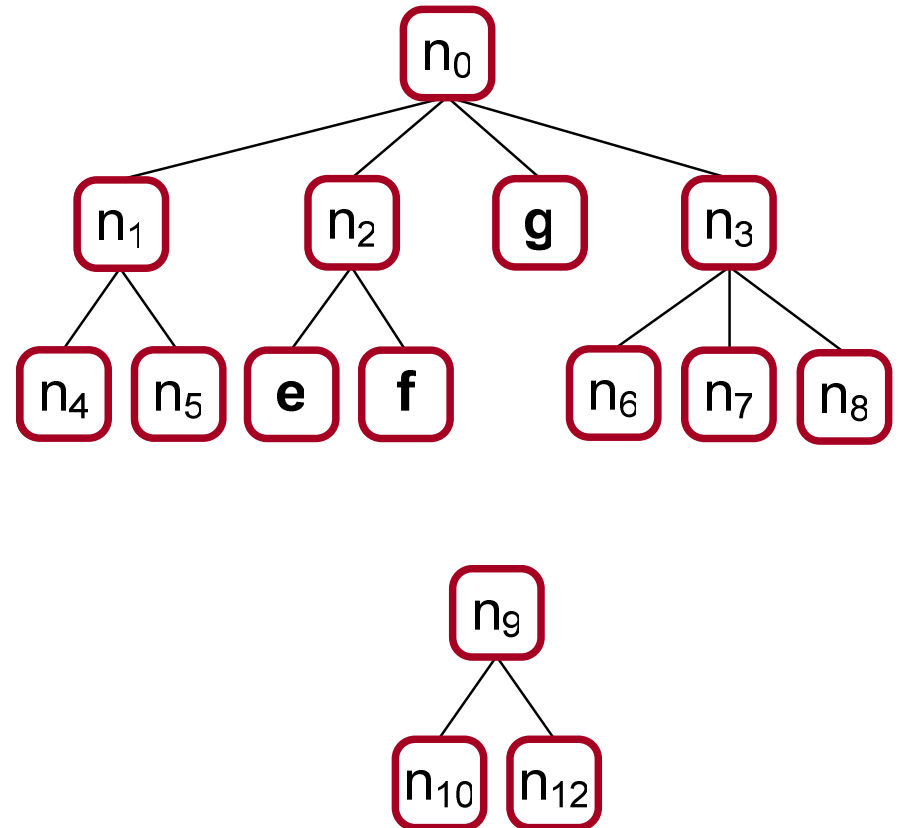
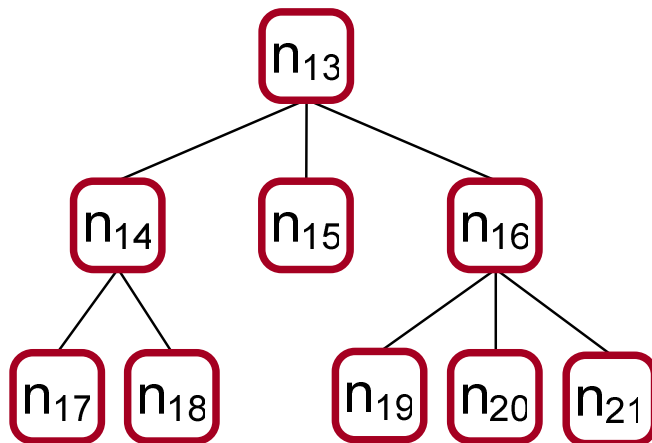


Meronymy Forest

13

Meronymy forest F is a disjoint union of meronymy trees

$$A_F = \bigcup_{t \in F} A_t$$



Aggregation Candidate Construction

14

Construction of aggregation candidates aggregation through aggregation candidate size increment

start: $k = 2$

***i* iteration:** construct *i*-size aggregation candidates from (*i-1*) aggregation candidates

prune insignificant candidates

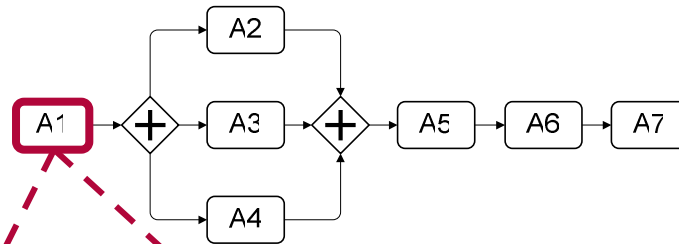
prune candidates with large distance

stop: $k = |A|$, $PM = (A, G, E)$ **OR** all the aggregation candidates of size *k* are pruned

Activity Match (1)

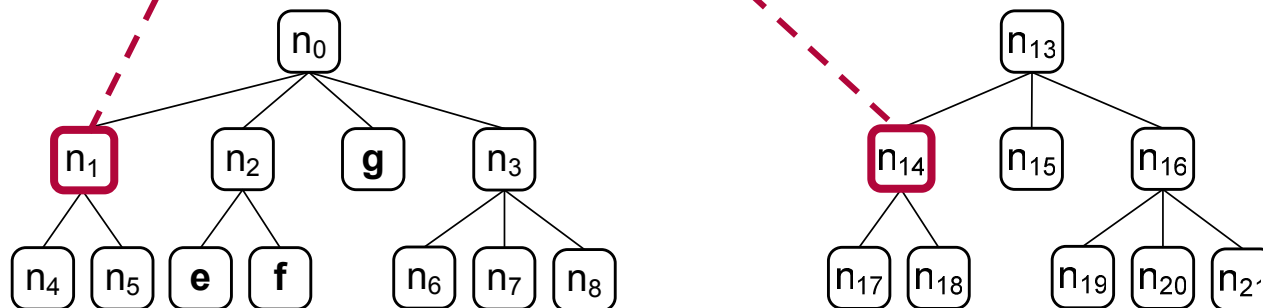
15

Process model $PM = (A, G, E)$



$$match_{PM} : A \rightarrow \mathcal{P}(A_F)$$

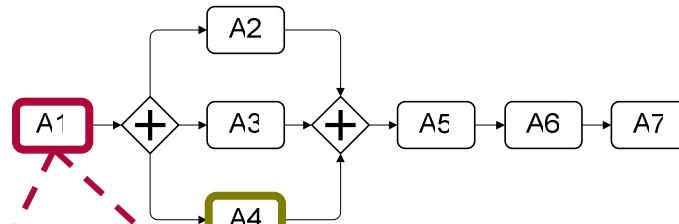
Meronymy forest F



Activity Match (2)

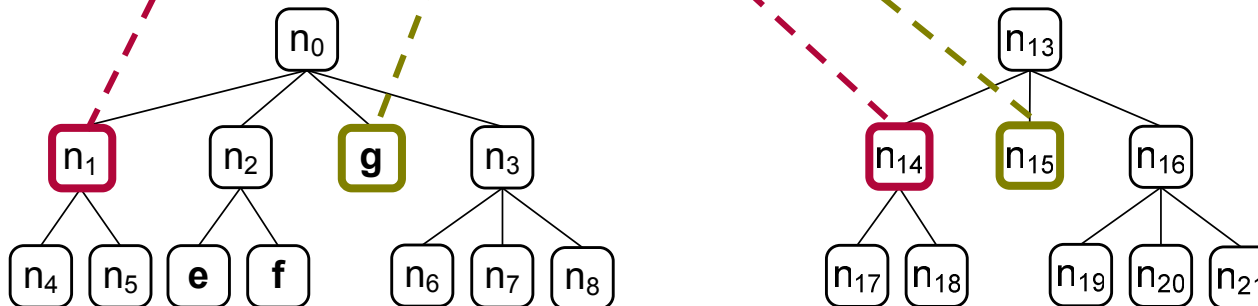
16

Process model $PM = (A, G, E)$



$match_{PM} : \mathcal{P}(A) \rightarrow \mathcal{P}(\mathcal{P}(A_F))$

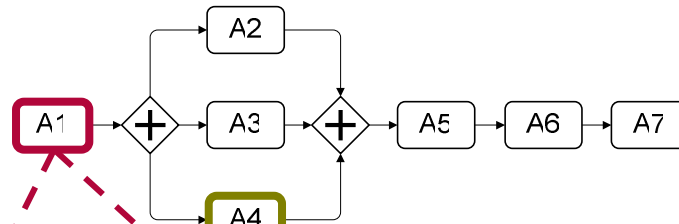
Meronymy forest F



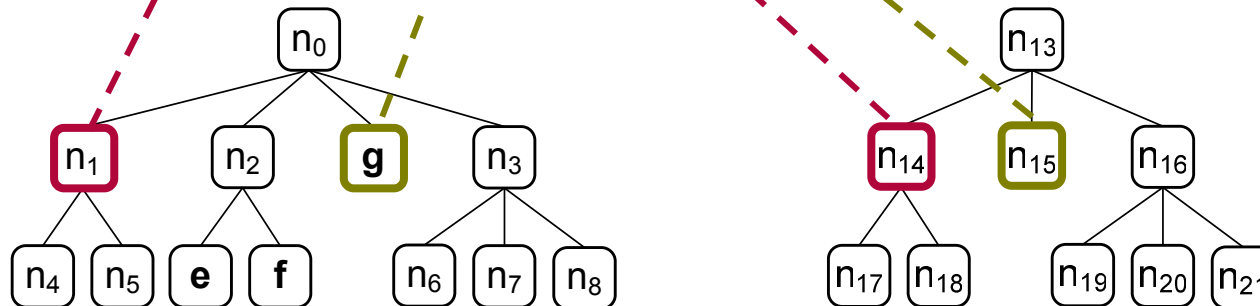
Activity MixMatch

17

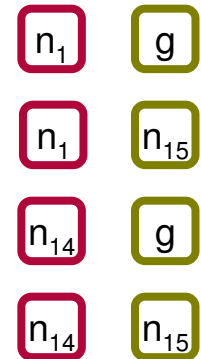
Process model $PM = (A, G, E)$



Meronymy forest F



$mixmatch_{PM} : \mathcal{P}(A) \rightarrow \mathcal{P}(A_F)$



Lowest Common Ancestor

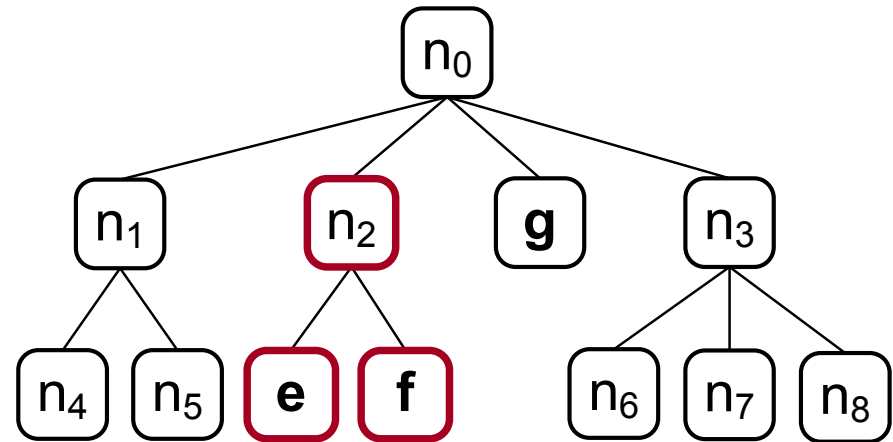
18

Lowest common ancestor is a function

$$lca_t : \mathcal{P}(A_t) \rightarrow A_t$$

maps a tree node set to its
lowest common ancestor

$$lca_{t_1}(\{e, f\}) = n_2$$



$$t_1 = (A_{t_1}, r_{t_1}, M_{t_1})$$

Meronymy Leaves

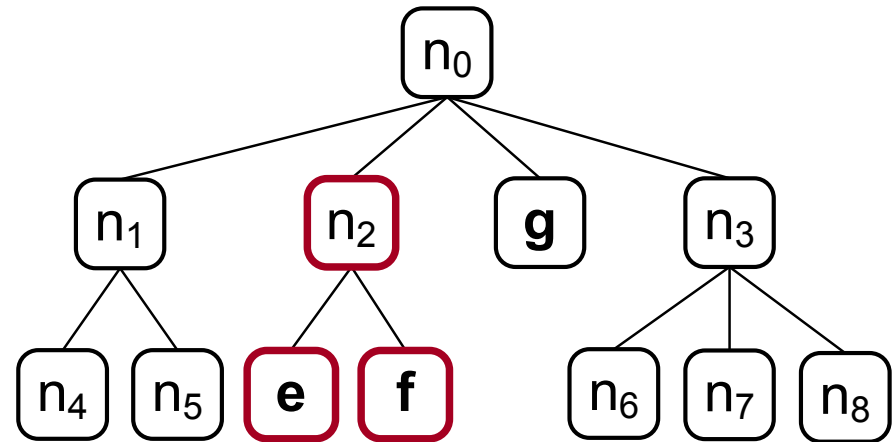
19

Meronymy leaves is a function

$$w_t : A_t \rightarrow \mathcal{P}(A_t)$$

maps an activity to the leaves of the subtree rooted to this activity

$$w_{t_1}(n_2) = \{e, f\}$$



$$t_1 = (A_{t_1}, r_{t_1}, M_{t_1})$$

Degree of Aggregation Coverage (1)

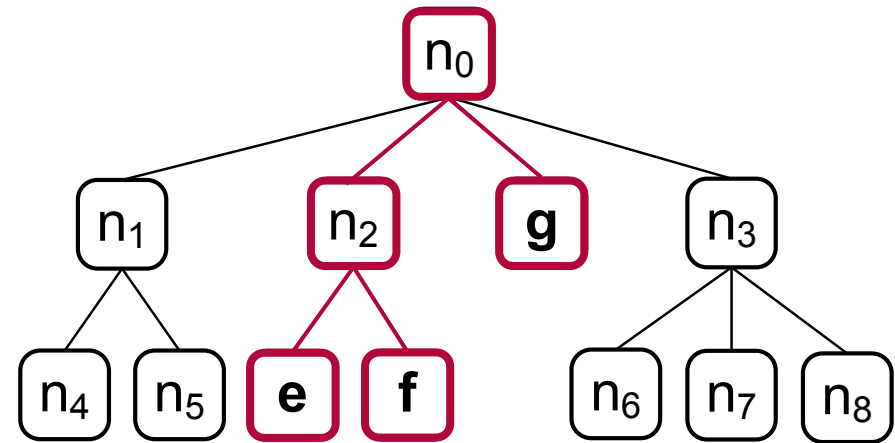
20

Degree of aggregation coverage is a function

$$cover : \mathcal{P}(A_t) \rightarrow (0, 1]$$

$$cover(C) = \frac{\left| \bigcup_{\forall a \in C} w_t(a) \right|}{|w_t(lcat_t(C))|}$$

$$cover(\{e, g\}) = 0.25$$



$$t_1 = (A_{t_1}, r_{t_1}, M_{t_1})$$

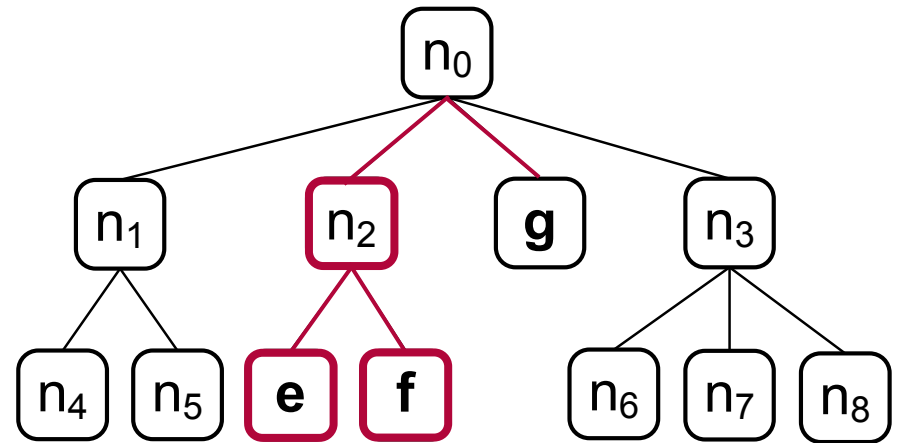
Degree of Aggregation Coverage (2)

21

Degree of aggregation coverage is a function

$$cover : \mathcal{P}(A_t) \rightarrow (0, 1]$$

$$cover(C) = \frac{\left| \bigcup_{\forall a \in C} w_t(a) \right|}{|w_t(lcat_t(C))|}$$



$$cover(\{e, f\}) = 1$$

$$t_1 = (A_{t_1}, r_{t_1}, M_{t_1})$$

Degree of Aggregation Coverage Properties

22

Shows

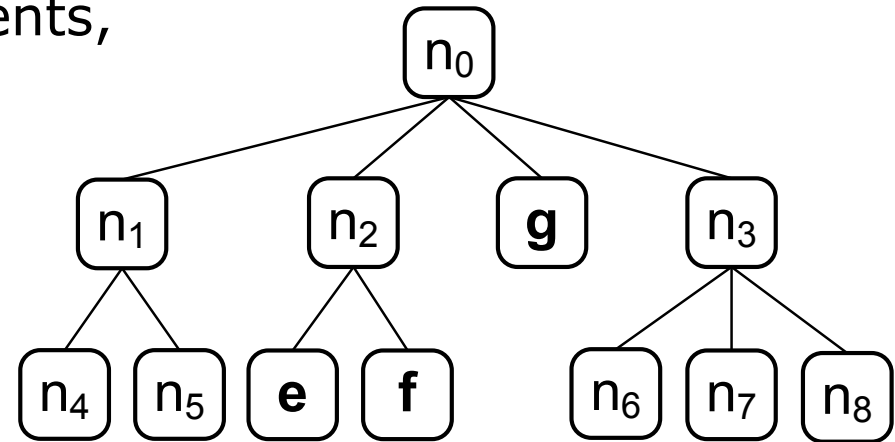
- if the LCA has other descendents, except aggregation candidate

Ignores

- aggregation candidate size
- the aggregation candidate depth in the LCA subtree
- ignore the LCA depth

Possesses

- value between 0 and 1



Model collection

- 6 process models (42 activities on average)

Meronymy forest

- MIT Process Handbook
- processes elicited in the interviews with process experts
- \approx 5000 activities
- specifies meronymy and hyponymy
- spans several business domains

Evaluation Approach

24

- Each activity aggregation is decomposed into a set of subsets of size 2, e.g.:

$$\{a, b, c\} \rightarrow \{a, b\}, \{a, c\}, \{b, c\}$$

- Modeling expert evaluates pair relevance
- Experiments varying *node distance*, *cover*
- Observe the precision value

Evaluation Results

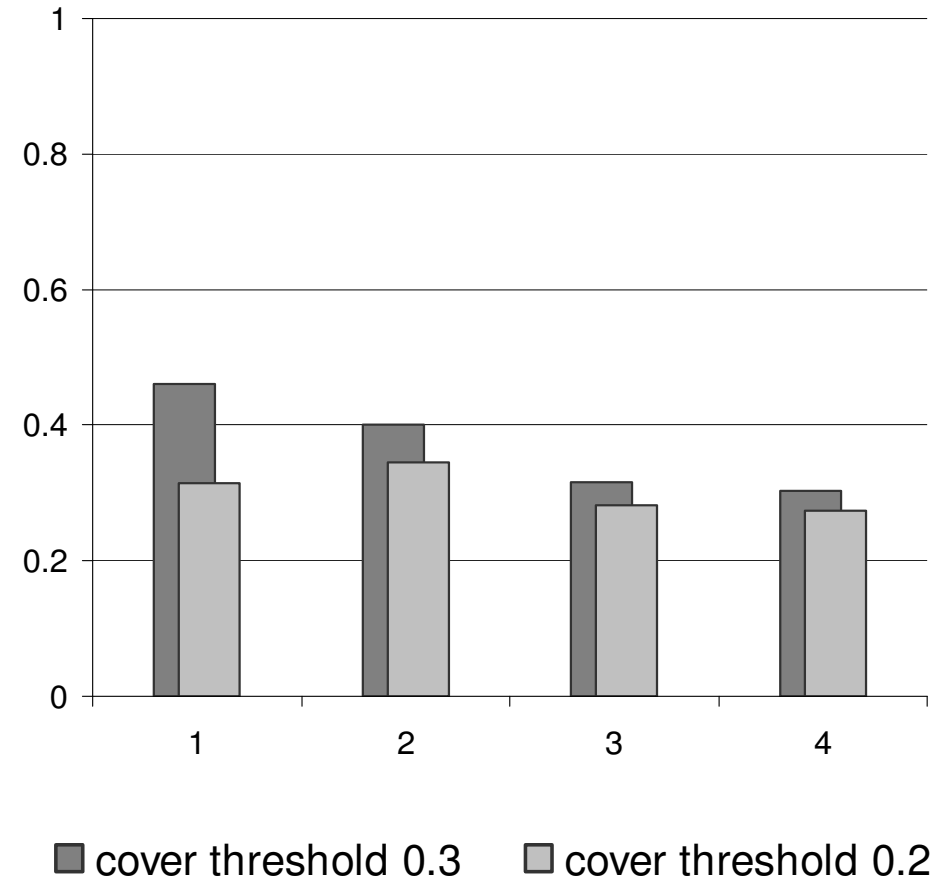
25

Observations

$0.27 \leq \textit{precision} \leq 0.46$

$\uparrow \textit{cover} \rightarrow \uparrow \textit{precision}$

$\uparrow \textit{node distance} \rightarrow \downarrow \textit{precision}$



Contributions

- Metric for relatedness of activity sets
- Activity aggregation mining algorithm

Future work

- Improve activity matching technique
- Precise aggregation mining technique evaluation
- Investigate other information enabling activity aggregation

Thank you!

Contact Details



28

Sergey Smirnov

PhD Student

Business Process Technology Group

Hasso Plattner Institute

Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany

Email: sergey.smirnov@hpi.uni-potsdam.de

Phone: +49 (0) 331 5509 194

Fax: +49(0) 331 5509 189