SECPI: Searching for Explanations for Clustered Process Instances

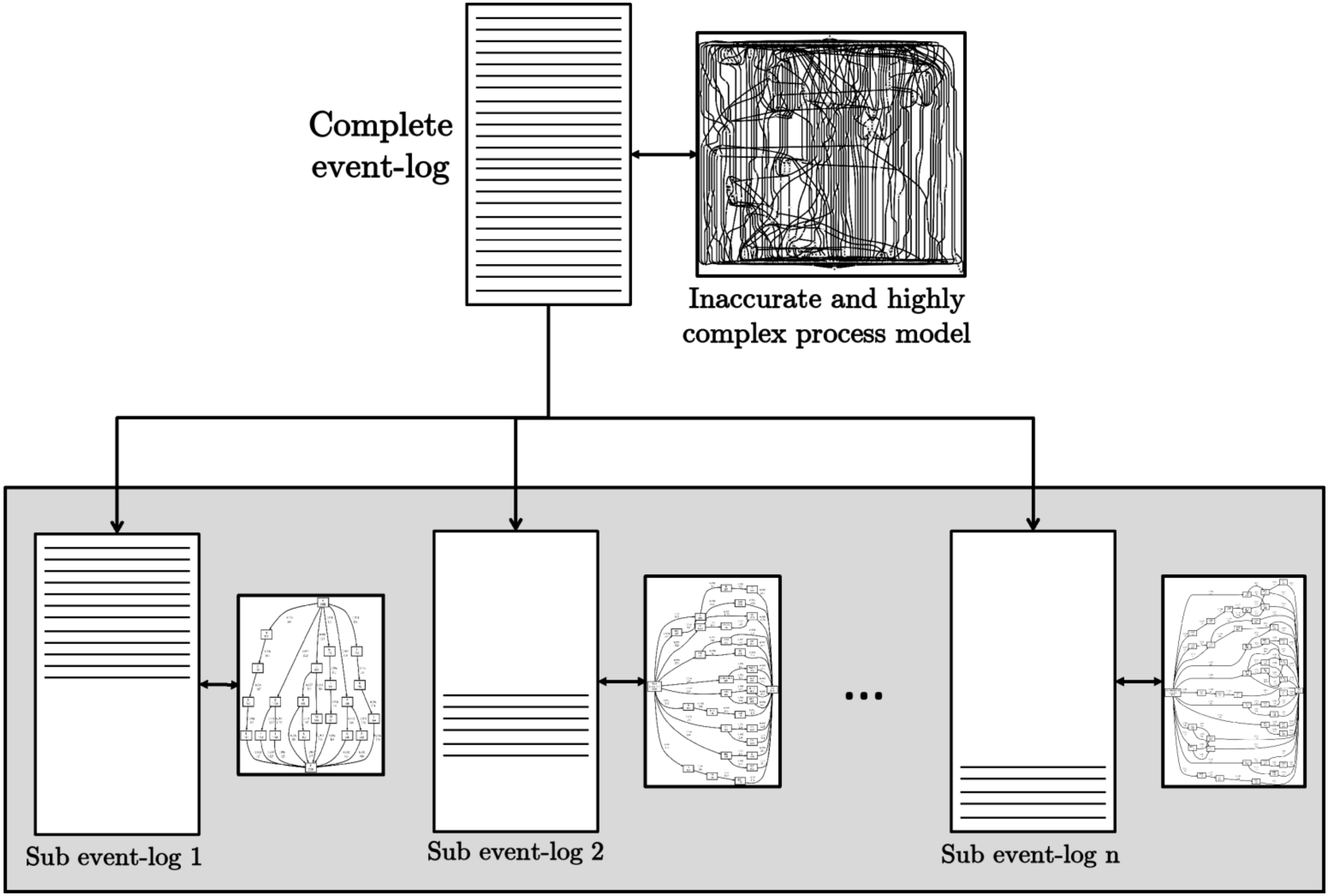
Jochen De Weerdt & Seppe vanden Broucke
KU Leuven, Research Centre for Management Informatics

Jochen.DeWeerdt@kuleuven.be (@jochendw)
Seppe.vandenBroucke@kuleuven.be (@macuyiko)

Outline

- Introduction
  - Trace clustering
  - Problem
  - Potential alternative solutions
- Our solution: SECPI
- How does it work?
- Implementation
- Evaluation
- Ideas for future work

Trace clustering



KU LEUVEN

Trace clustering algorithms

Reference	Data Representation	Clustering Bias
Greco, G., Guzzo, A., Pontieri, L., Saccà, D.: Discovering expressive process models by clustering log traces. IEEE Trans. Knowl. Data Eng. **18**(8) (2006) 1010–1027	propositional	instance similarity
Song, M., Günther, C.W., van der Aalst, W.M.P.: Trace clustering in process mining. In Ardagna et al., eds.: BPM Workshops. Vol. 17 of LNBIP, Springer (2008) 109–120	propositional	instance similarity
Ferreira, D.R., Zacarias, M., Malheiros, M., Ferreira, P.: Approaching process mining with sequence clustering: Experiments and findings. In Alonso et al. eds.: BPM. Vol. 4714 of LNCS, Springer (2007) 360–374	event log	maximum likelihood
Bose, R.P.J.C., van der Aalst, W.M.P.: Context aware trace clustering: Towards improving process mining results. In: SDM, SIAM (2009) 401–412	bag of strings	instance similarity
Bose, R.P.J.C., van der Aalst, W.M.P.: Trace clustering based on conserved patterns: Towards achieving better process models. In Rinderle-Ma, S. et al., ed.: BPM Workshops. Vol. 43 of LNBIP, Springer (2009) 170–181	propositional	instance similarity
Folino, F., Greco, G., Guzzo, A., Pontieri, L.: Mining usage scenarios in business processes: Outlier-aware discovery and run-time prediction. Data Knowl. Eng. **70**(12) (2011) 1005–1029	event log	maximum likelihood
De Weerdt, J., vanden Broucke, S.K.L.M., Vanthienen, J., Baesens, B.: Active trace clustering for improved process discovery. IEEE Trans. Knowl. Data Eng. **25**(12) (2013) 2708–2720	event log	fitness
Ekanayake, C.C., Dumas, M., García-Bañuelos, L., La Rosa, M.: Slice, mine and dice: Complexity-aware automated discovery of business process models. In Daniel et al., eds.: BPM. Vol. 8094 of LNCS, Springer (2013) 49–64	propositional	complexity

Problem

- Evaluation of trace clustering results
  - Compute intra/inter cluster similarity/dissimilarity (e.g. with a distance measure)
  - Compute fitness, precision, generalization, and simplicity of discovered process models for the clusters

- However

1) What are the driving elements that determine a clustering solution?
2) How can a clustering solution be understood by end-users, thus explained from a *domain* perspective?

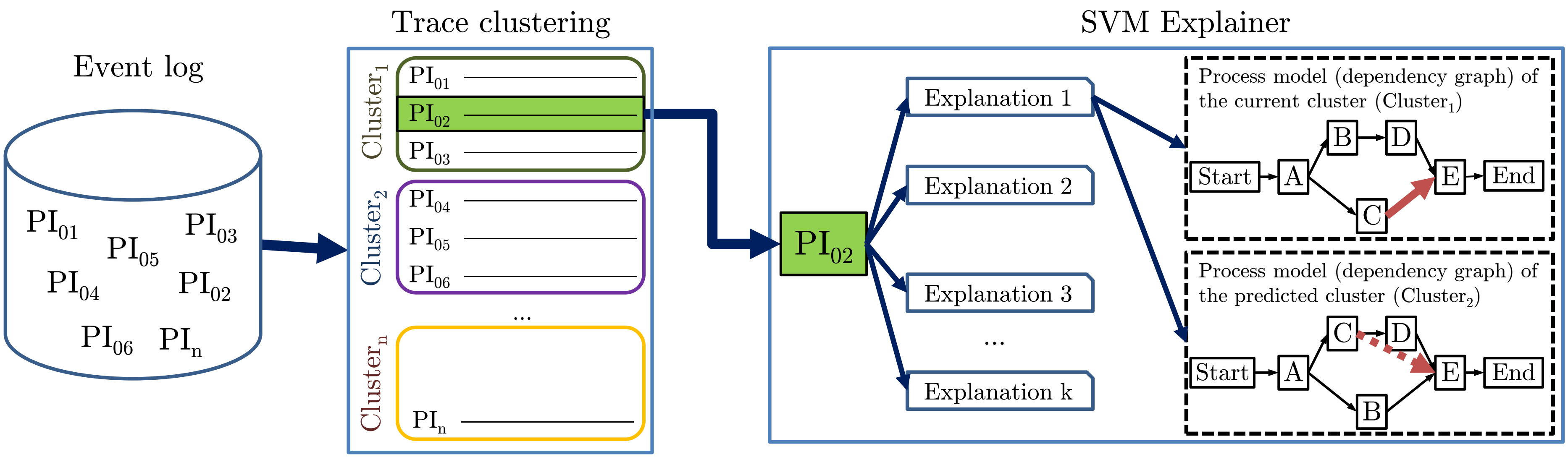
Potential alternative solutions

- Visual analysis of the underlying process models
- Process model similarity
  - Metrics (e.g. Alves de Medeiros et al., 2008; Dijkman et al., 2011)
  - Visualization (e.g. Dijkman 2007; 2008)
- Footprints and behavioural profiles
- White box classification model (e.g. decision tree)
- Cross-cluster conformance checking

➔ All these techniques are valuable, but also present disadvantages to solve the problem at hand

Our solution: SECPI

- Learn a minimal set of control-flow characteristics for each process instance individually *whose absence would prevent the process instance to be in its current cluster*
- Control-flow characteristics: SometimesDirectlyFollows(A,B)



Explanation 1 for PI_{02}: **IF** SometimesDirectlyFollows(C,E) $= 0$ **THEN** $Cluster_2$

SECPI: Steps

1. Construct the data set
   - Propositional data set consisting of SometimesDirectlyFollows(A,B)-attributes (binary variables)
   - Cluster label for each instance
2. Derive explanations from a Support Vector Machine (SVM) classifier → SECPI algorithm
   - Inspired by: Martens, D. & Provost, F. Explaining data-driven document classifications. MISQ Vol. 38, Issue 1, pp. 73-99, 2014.
   - SVM-*liblinear* because of scalability (dimensionality explosion)
   - Key modifications to Martens & Provost
     - Multi-class classification
     - Explanations are restricted to characteristics that are present in traces
     - Performance optimisations

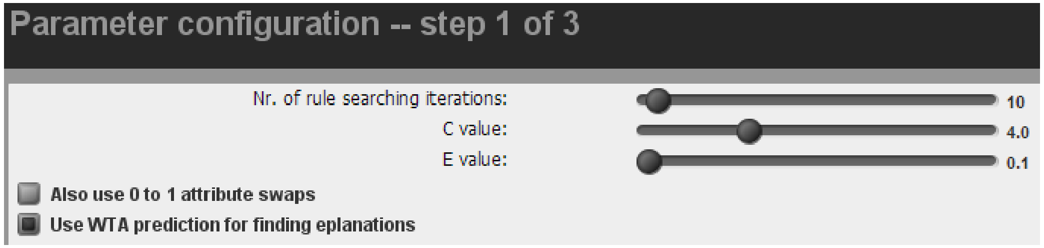
SECPI algorithm

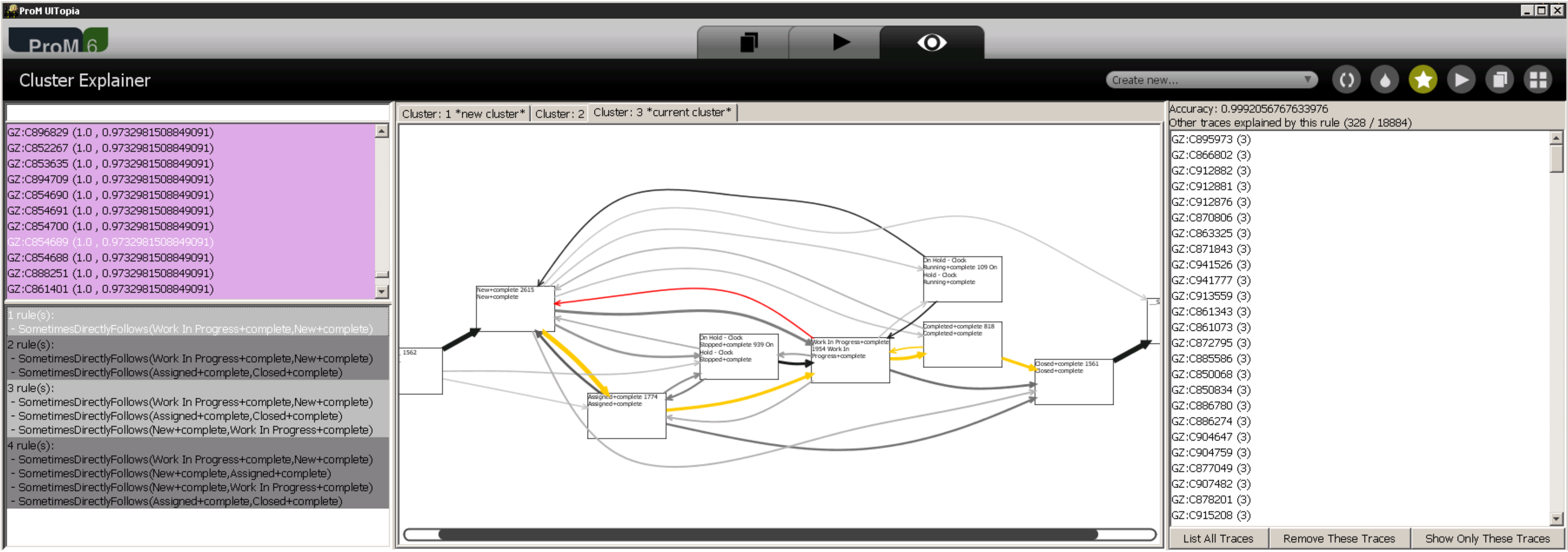
- Inputs
  - Process instance (sequence of binary attributes)
  - Classifier (SVM)
  - Three configuration parameters
    - Nr. of *iterations:* determines the length of the explanations
    - *zero_to_one*: boolean that determines whether 0-to-1 swaps are allowed
    - *require_support*: boolean that determines whether swaps of invariable attributes are allowed
- Output
  - A set of explanatory rules: set of sets of attribute indices
  - If (¬zero_to_one) → “This process instance would leave its current cluster when it would not exhibit the behaviour as represented by these attributes”

SECPI algorithm

- Step 1: Find single-attribute rules
- Step 2: Best-first search procedure with pruning
    - ○ Expand on currently available combinations
    - ○ Using the classifier’s scoring function
        - Idea: find attribute swaps that move the instance farthest away from current cluster
    - ○ Check whether any of the expanded combinations leads to a class change

ProM 6 – implementation: SVMExplainer





http://processmining.be/svmexplainer/

Evaluation

- We have compared our approach to global, white-box classification techniques
  - Decision trees: C4.5
  - Rule learner: RIPPER
- We found across 5 real-life data sets
  - Much shorter explanations
  - On par or better accuracy of the classification/explanation model

Ideas for future work

- Aggregating individual explanations into a global model
  - Finding similar explanations
  - Clustering explanations
  - Network representation
- Opportunities
  - Studying how trace clustering techniques actually work from a domain perspective
  - Other or better characteristics to be used beyond SometimesDirectlyFollows(A,B)
  - Relate exogenously defined clusters to process-specific control characteristics

thank you
danke
рахмат
謝謝
ngiyabonga
teşekkür ederim
spas
Баярлалаа
mersi
kia ora
barka
welalin
tack
mahalo
tapadh leat
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
gracias
хвала
kiitos
dankie
nanni
nandri
dhanyavad
maururu
köszönöm
ačiū
asante
manana
bedankt
bayarlalaa
gràçie
hvala
dankon
obrigada
tenki
enkosi
akun
djiere dieuf
tau
mochchakkeram
murakoze
dziękuję
дякую
mamnun
chokrane
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
go raibh maith agat
sobodi
děkuji
sagolun
sukriya
kop khun krap
obrigado
mèsi
arigatô
takk
dakujem
trugarez
ありがとう
tanemirt
grazie
didi madloba
najis tuke
terima kasih
rahmet
ευχαριστώ
diolch
dhanyavadagalu
shukriya
mercé
мерси
kam sah hamnida
rahmat
তোমাকে ধন্যবাদ
감사합니다
xièxie
merci

KU LEUVEN