INFINT'09

Bertinoro, 20 Marzo 2009

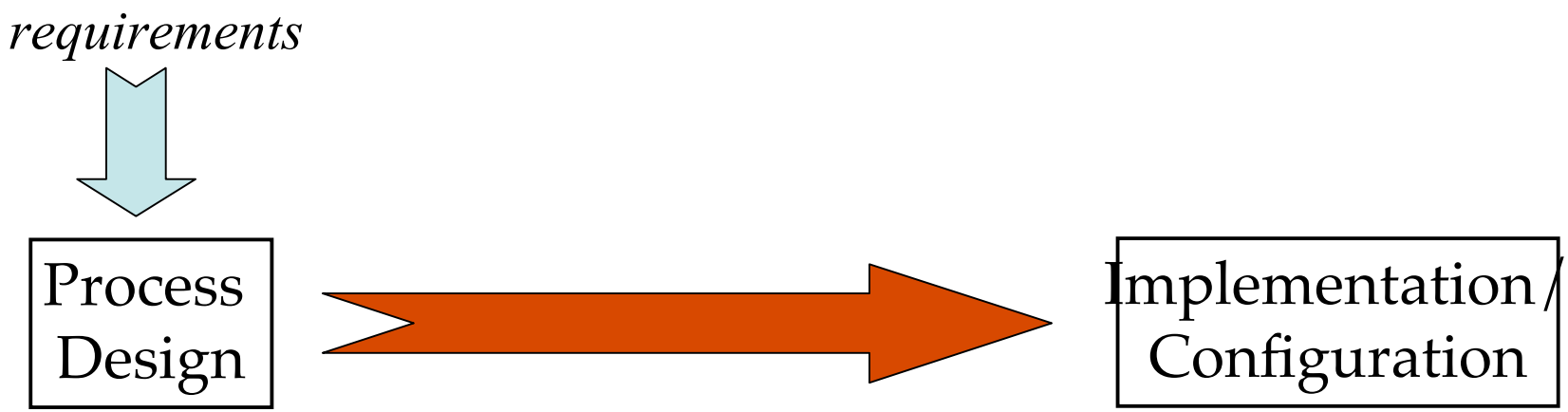
Pushing Intelligence into Workflow Systems

Gianluigi Greco, Domenico Saccà

UNIVERSITÀ DELLA CALABRIA
CAMPUS DI ARCAVACATA

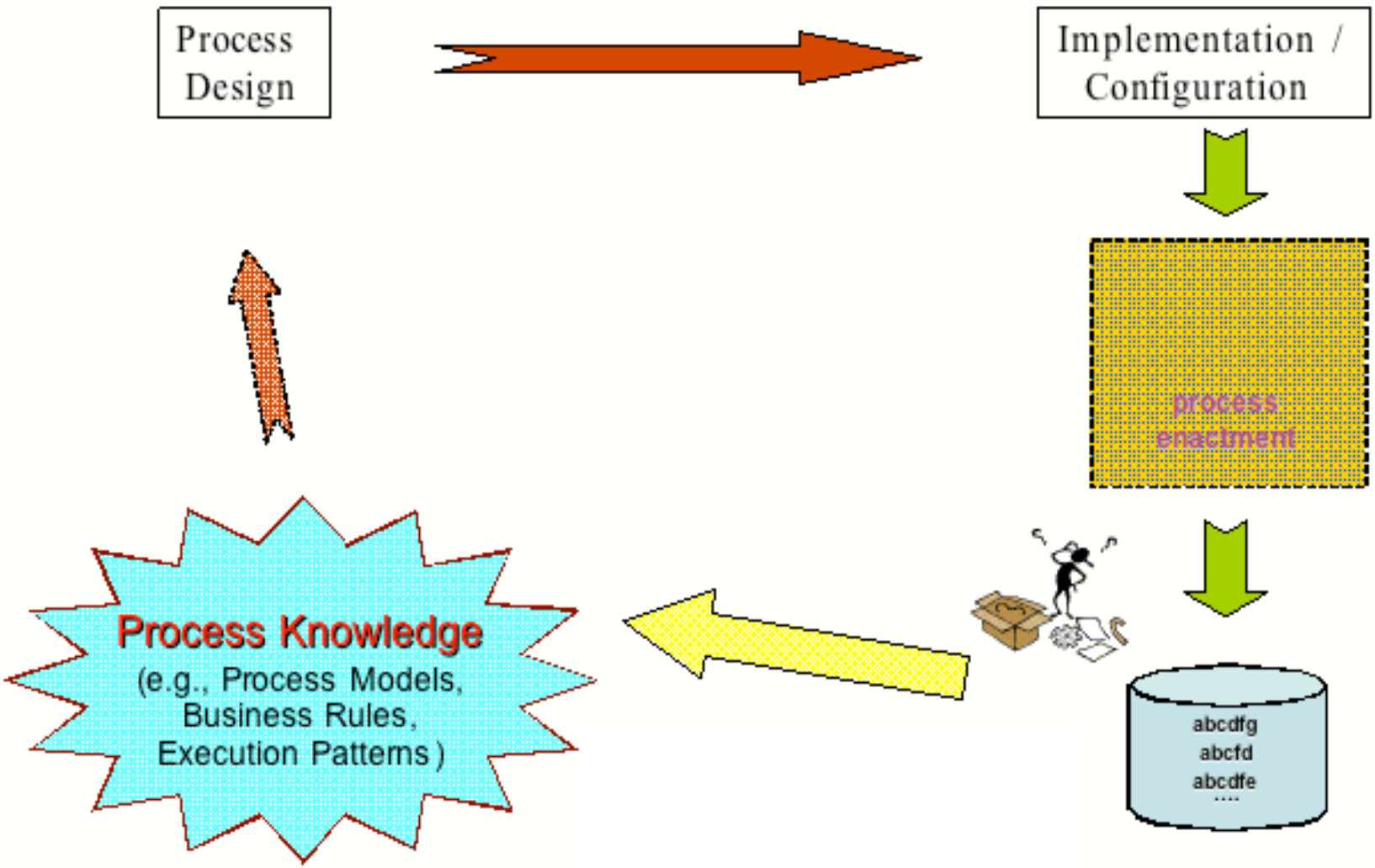
ICAR CNR
Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Workflow Design



- Key Problems
  - Designing process models is complex and time consuming
  - Organizational/behavioral rules may be used in practice, even if not formalized in a model
  - Practical enactments may differ from what is prescribed by the abstract model

Business Process Intelligence



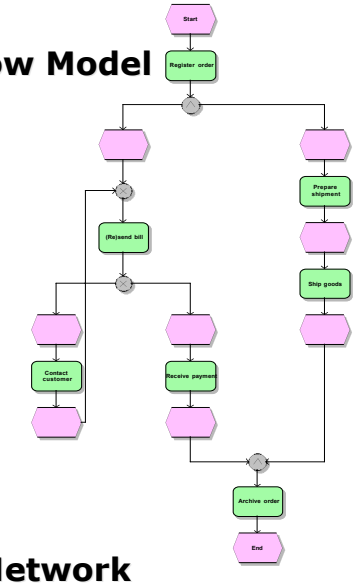
Process Mining

Historical logs used as a reflection of reality

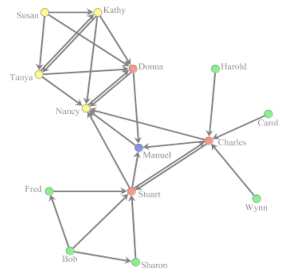
- behavioral models are linked to real events
- reduces the abstraction gap between model and reality

- Different analysis perspectives (and models to be discovered)
  - Control flow perspective:
    - *What is the typical flow of work for the handling of orders?*
  - Organizational perspective:
    - *Which people appear to closely work together?*
    - *What are people roles?*
  - Case-oriented perspective:
    - *How process attributes impact on the execution of tasks*

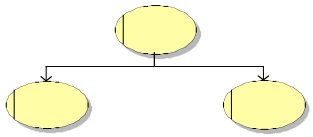
Workflow Model



Social Network

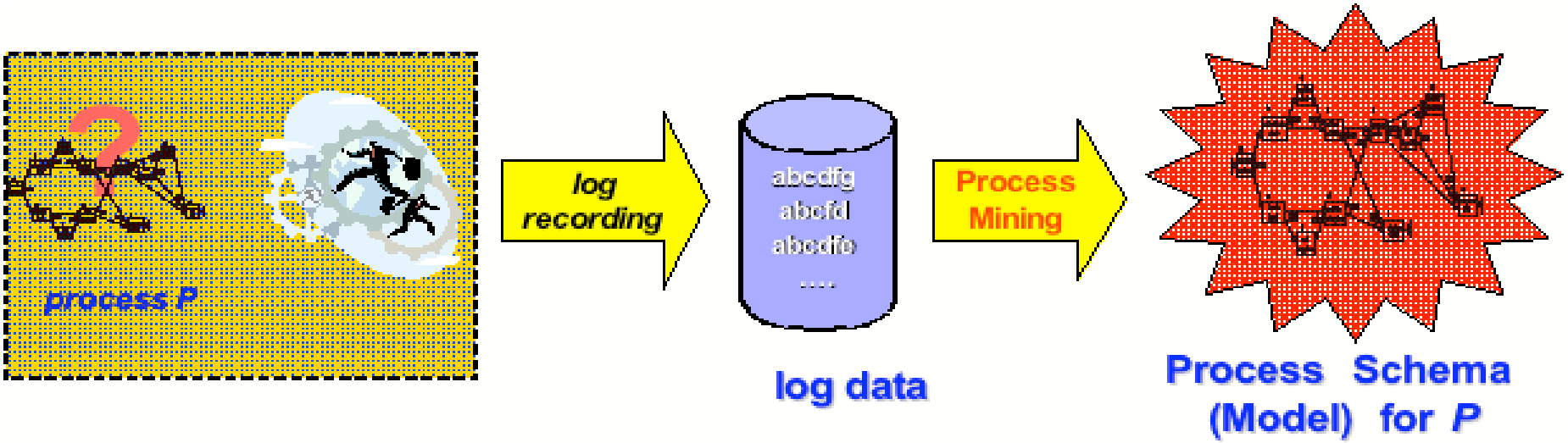


Organizational Model



Process Mining: Control Flow Perspective

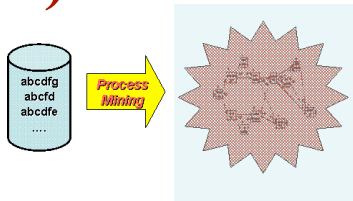
- The Mined Model…
  - …helps to better comprehend the process behavior, and to (re-) design/optimize concrete (i.e. executable) process models
  - …provides the user with a summarized view on process operations that will efficiently support the enactment of the process



Brief Overview of Standards in Process Mining

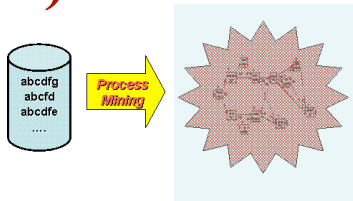
- **Models**
- **Logs**
- **Tools**

Output of PM Techniques(1/2)

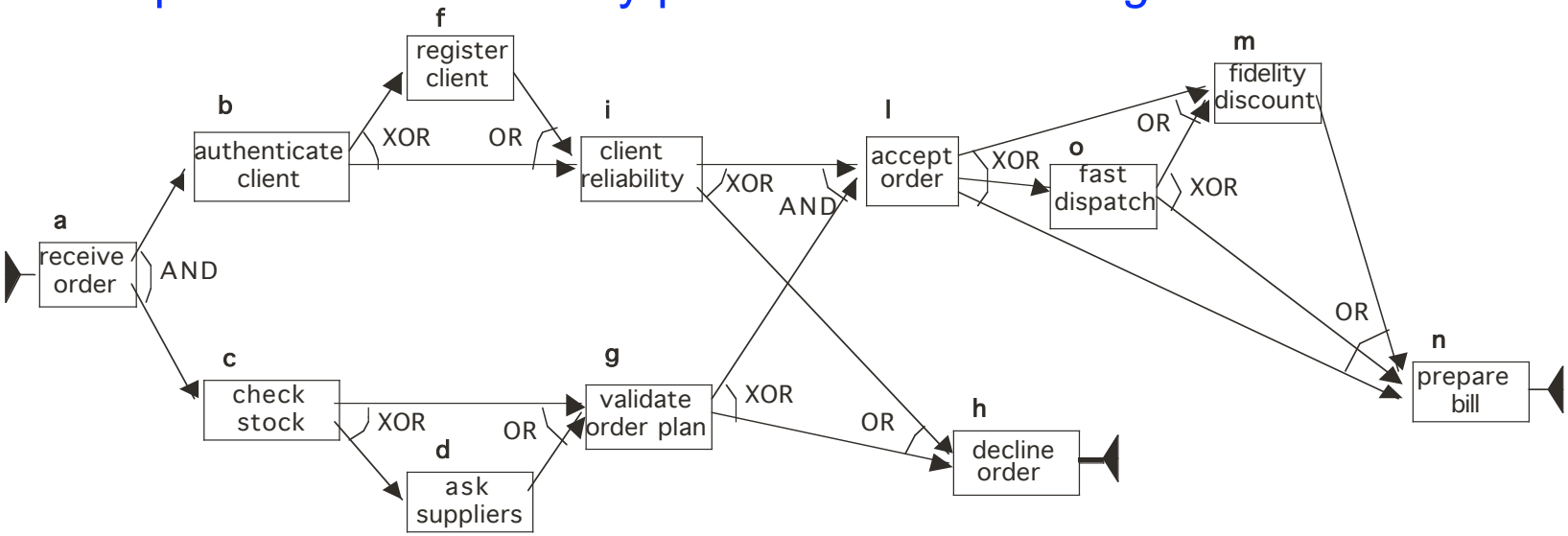


- A plethora of meta-models introduced to represent process models (and, in particular, workflow schemata)
  - Block-structured languages, Petri Nets, Logics, Process Algebra,…
  - Graph-based languages are a reasonable choice w.r.t. expressiveness, complexity and comprehensibility
    - Most process mining approaches produce a graph as the core of the mined model or simply exploit it to derive a schema in a specific representation language
    - (a few) exceptions, e.g.: block-structured model induction only use suitable term rewriting techniques to generalize the log traces
- Control Flow Graph
  - An intuitive, and broadly used, mean for specifying which execution flows are allowed through the tasks
    - A labeled, directed graph
    - Each node corresponds to a task (and vice-versa)
    - Each arc represents a (temporal) precedence between two tasks
  - Local constraints may further restrict the possible execution flows

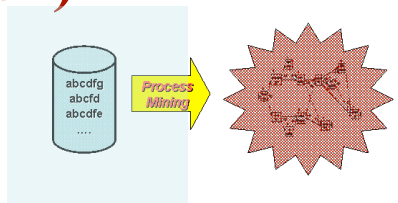
Output of PM Techniques(2/2)

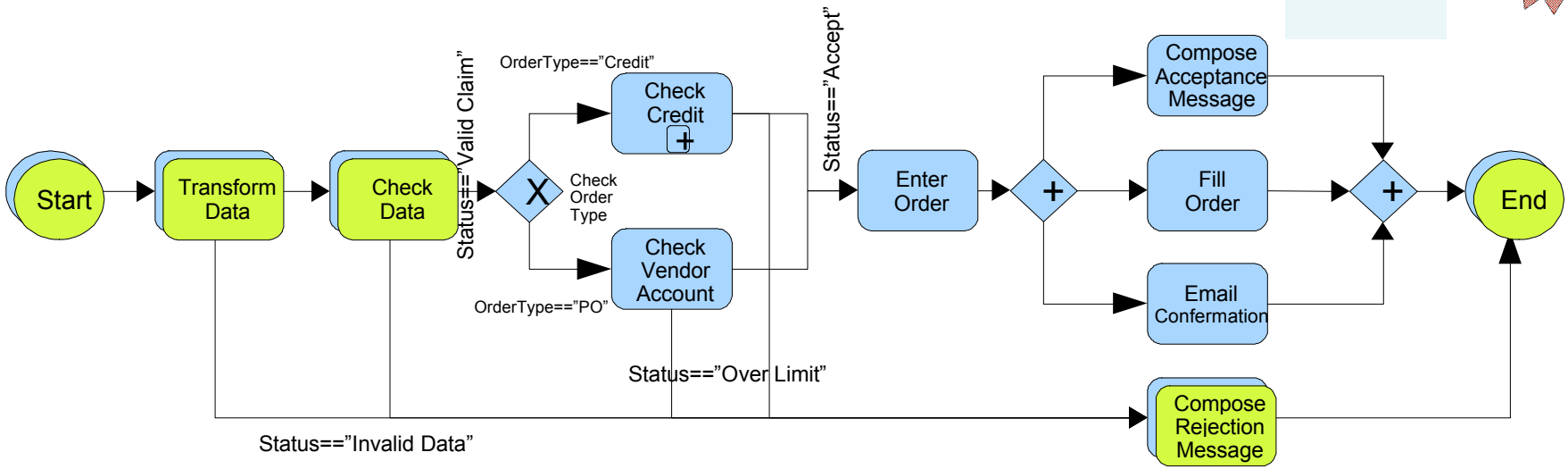


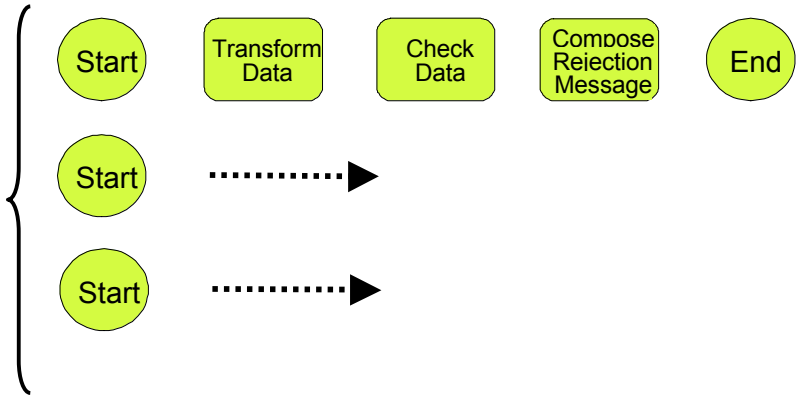
- Local constraints:
  - local: involving close nodes only
  - typical constructs: AND-split, AND-join, OR-split, XOR-split
  - can be expressed in terms of nr. of in/out-going arcs
    - (min) # of in-going arcs to be activated before a task can be executed
    - (min/max) # of out-going arcs which a task activates after it is executed
- Example: a CFG for the toy process *Order Management*



Input of PM Techniques(1/2)

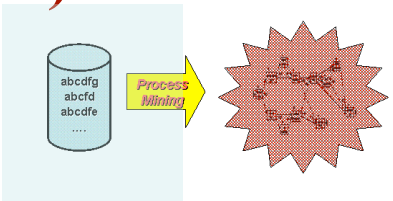




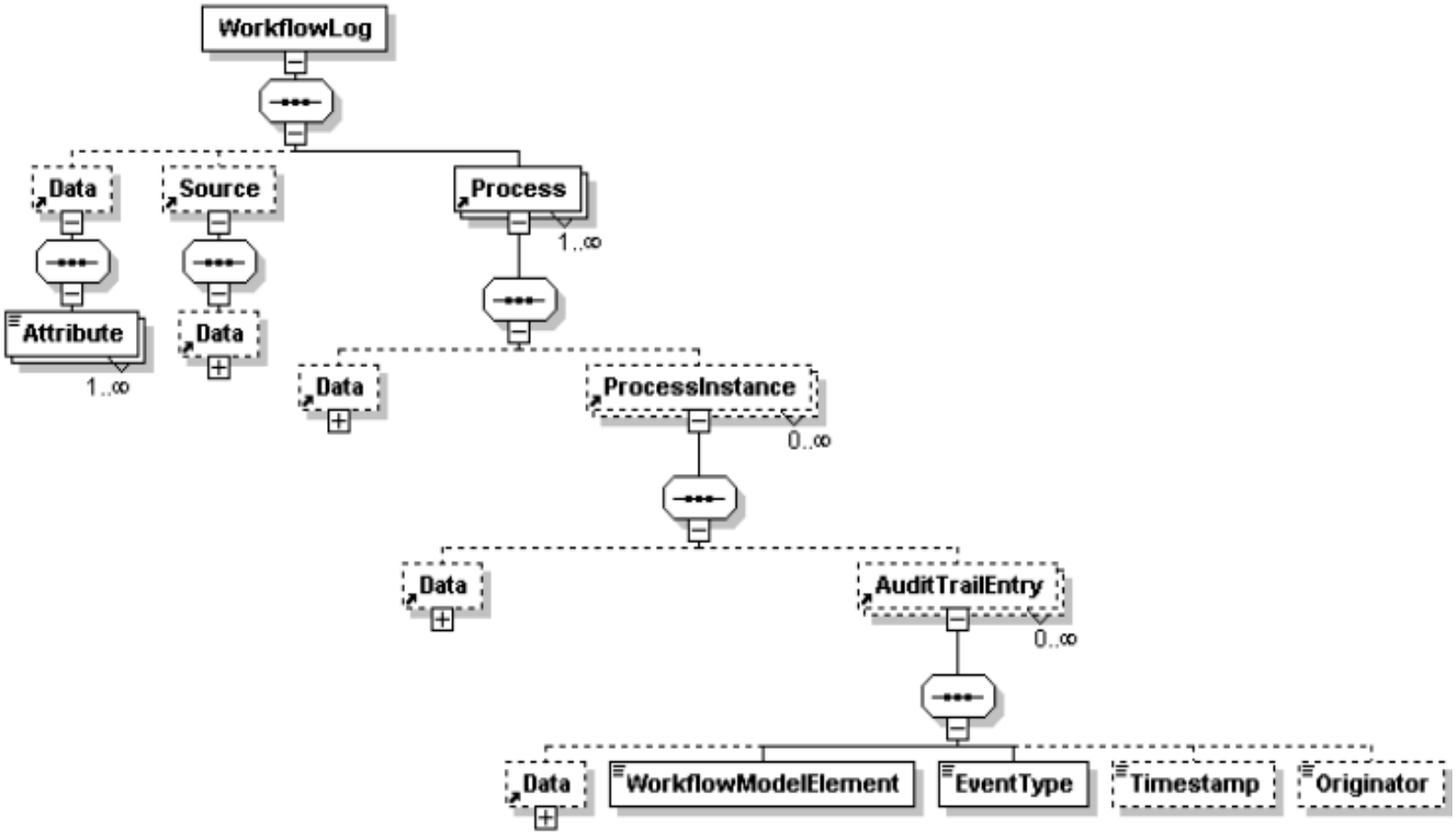


- **Input**: execution data
  - log: a list of traces
  - each trace reports the tasks performed during one execution
  - a trace can just be a sequence of task IDs

Input of PM Techniques(2/2)



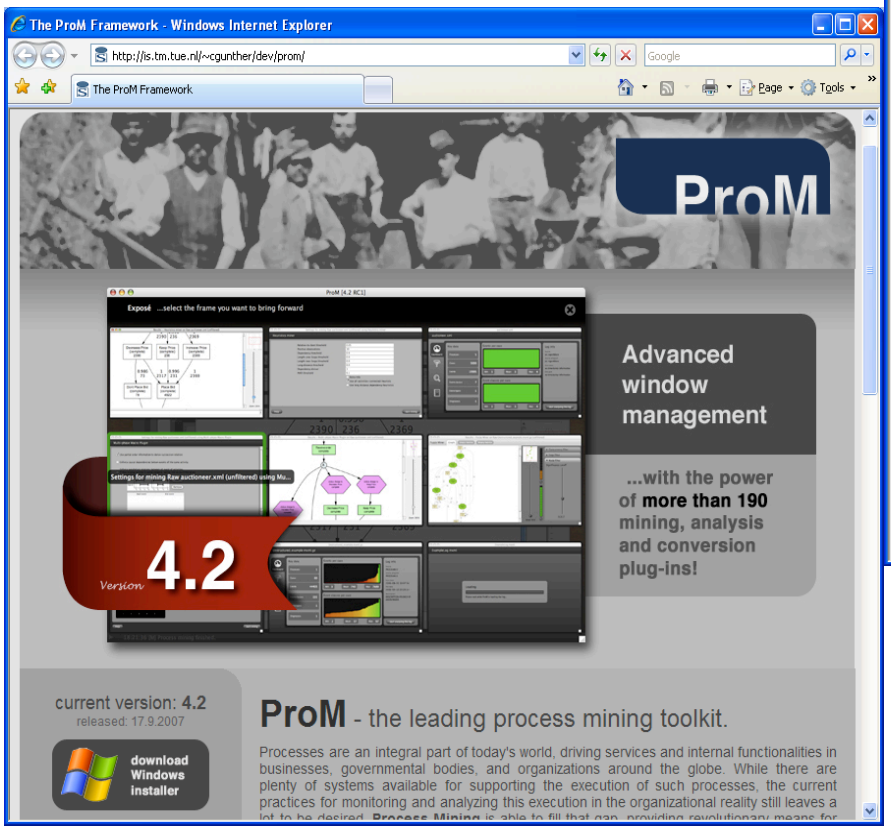
Standard: MXML



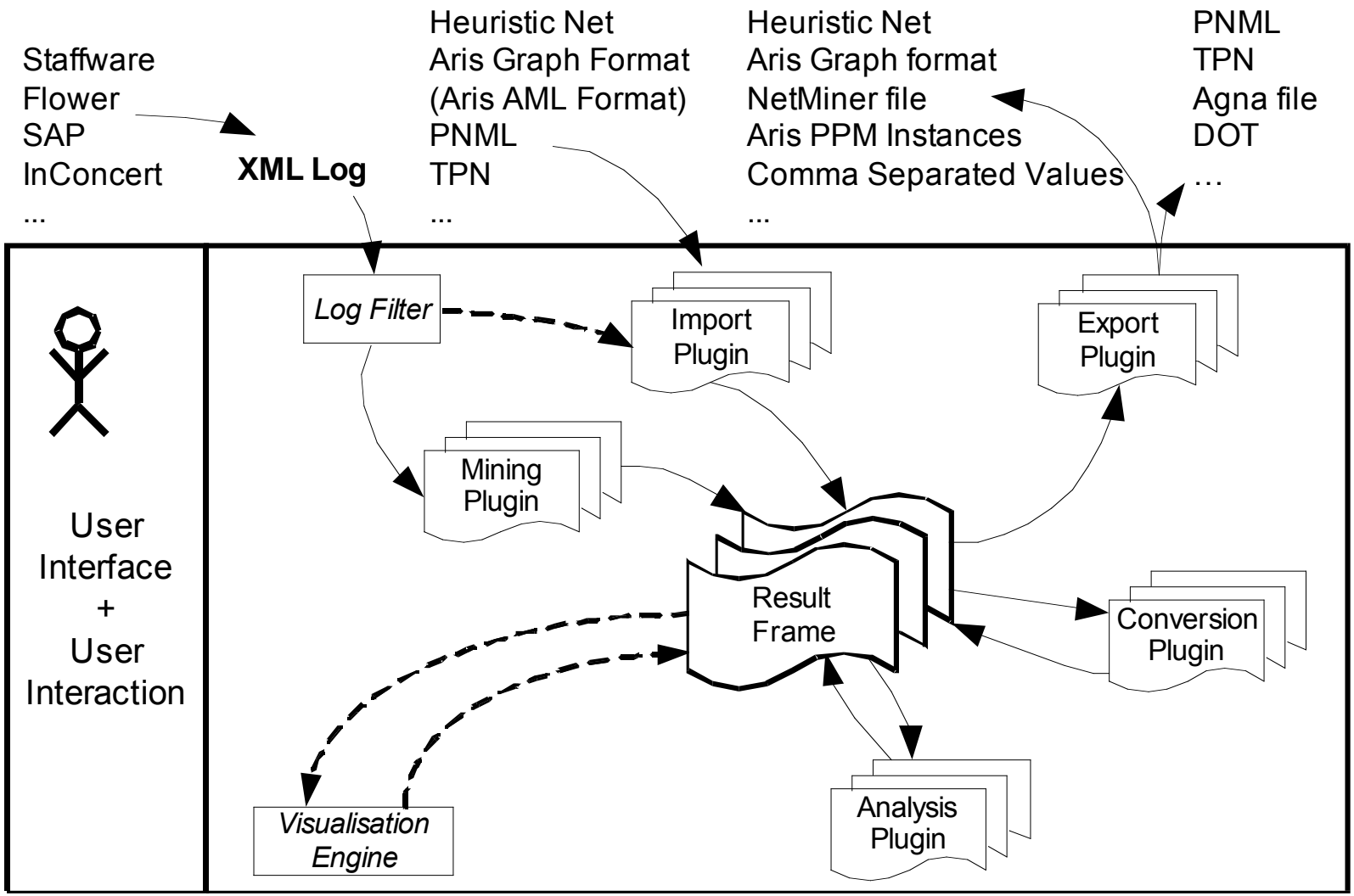
Process Mining Tools

Open-source frameworks

- ProMimport
- ProM



ProM Architecture



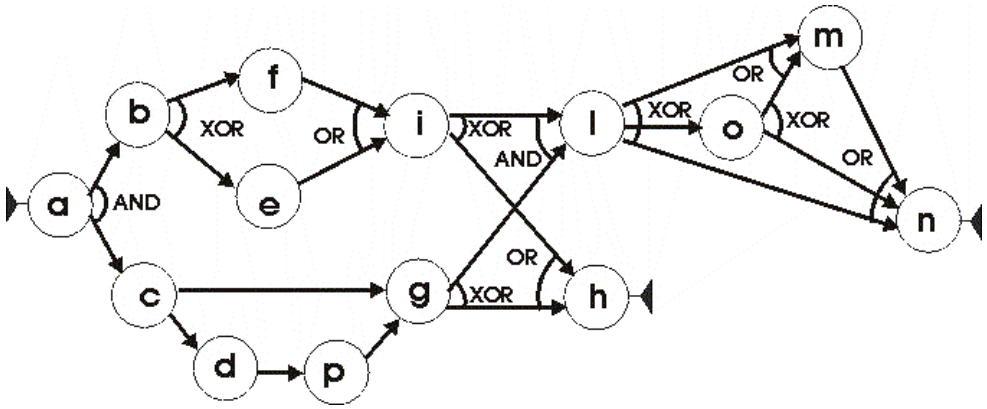
Mining Techiniques

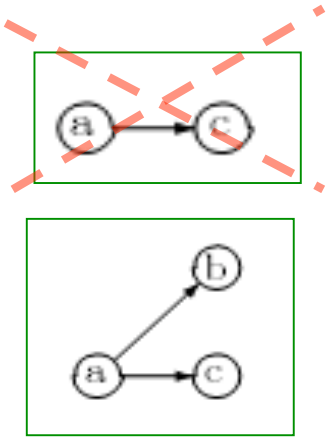
- Frequent Patterns
- Workflow Discovery
- Discovery of more refined process models
  - Co-clustering
  - Abstraction
  - Decision Trees
  - Outlier Detection

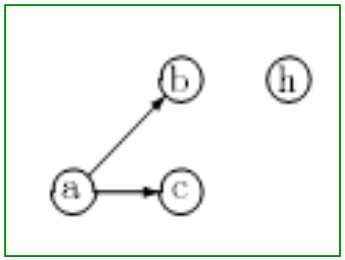
Frequent Patterns

Patterns as Sub-Graphs

- Given a workflow model (constrained graph), find frequent subgraphs
  - Ad-hoc strategies are used to prune the search space and outperform classical methods, *e.g., Frequent Pattern (basket analysis), graph-mining (molecules)*
- A key concept: weak-pattern (w-pattern)
  - A sub-graph *P* of the workflow schema is a weak-pattern if it is "closed" w.r.t. the constraints:
    - every arc coming into an "AND-join" node of *P* belongs to *P* too
    - every [only one] arc coming out of an "AND-split" [XOR-split] node of *P* belongs to *P* too



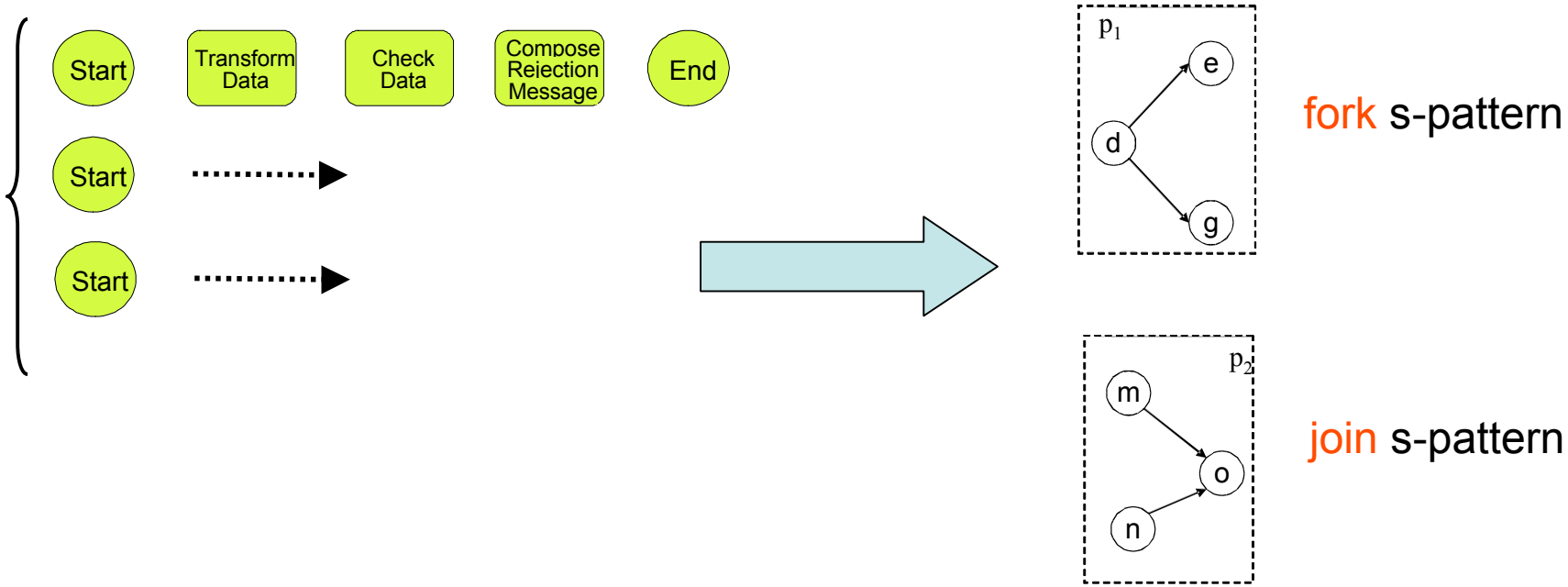




(unconnected)

Patterns as (Structured) Sequences

- *s-patterns* (structural patterns) are graphs over process tasks

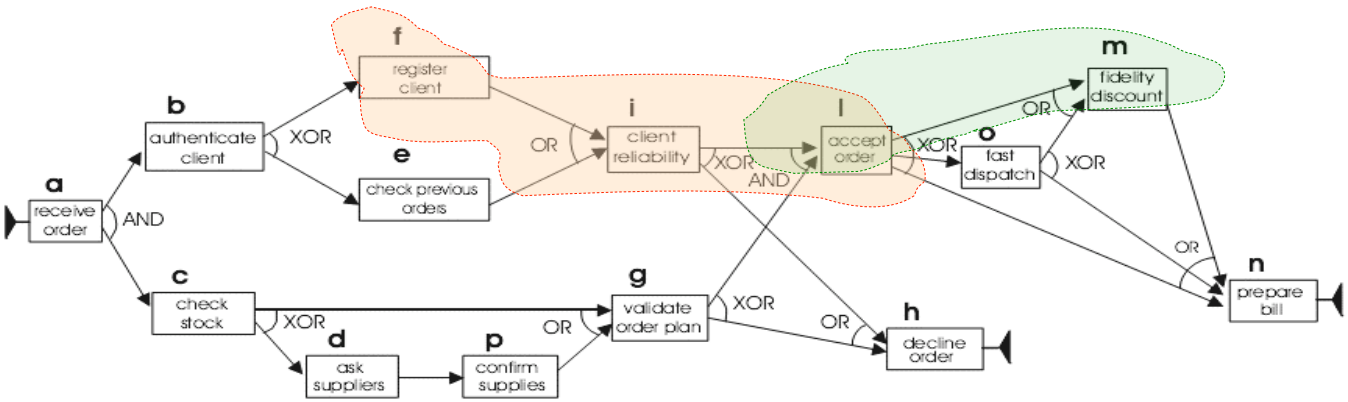


- Identify causalities

Patterns as Global Constraints

- An important kind of models are association rules:
  - $Px \rightarrow Py$ or $P\not{x} \rightarrow Py$
- A *discriminating rule* is a special kind of negative association rule built out of sequence patterns: $[a_1 \ldots a_h] \not\rightarrow a$
  - $[a_1 \ldots a_h]$ and $[a_h a]$ are both frequent, whereas $[a_1 \ldots a_h\, a]$ is not
    (w.r.t. the traces in L, and some given threshold)
  - it models a frequent behavior, unexpected w.r.t. the WF schema
- Example:

$$\boxed{fil \not\rightarrow m}$$



  - *fil* and *lm* are frequent, but their combination *film* never occurs
  - It captures the fact that *m* does not occur whenever *f* does

Patterns as Sequences+Time

- Sequential pattern = sequence of itemsets (no time info)
  - E.g.: < {a}, {d}> (or {a} → {d})
  - Support count based on set containment and sequentiality
- Annotate Sequences
  - Time used to sort itemsets (+ some gap constraints)

$$TAS = \{a\} \xrightarrow{3} \{b\} \xrightarrow{2} \{c\} \qquad \tau = 0.5$$

$$\{a, x\}_{t=0} \xrightarrow{3} \{b\}_{t=3} \xrightarrow{2.4} \{c, x\}_{t=5.4}$$ OK

$$\{a\}_{t=0} \xrightarrow{3.1} \{x\}_{t=2} \xrightarrow{} \{b, c\}_{t=3.1} \xrightarrow{1.8} \{c, x\}_{t=4.9}$$ OK

$$\{a\}_{t=0} \xrightarrow{3} \{b, x\}_{t=3} \xrightarrow{2.6} \{c, x\}_{t=5.6}$$ NO! | 2.6 – 2 | > 0.5

Mining Techiniques

- Frequent Patterns
- Workflow Discovery
- Discovery of more refined process models
  - Abstraction
  - Co-clustering
  - Decision Trees
  - Outlier Detection

Workflow Discovery

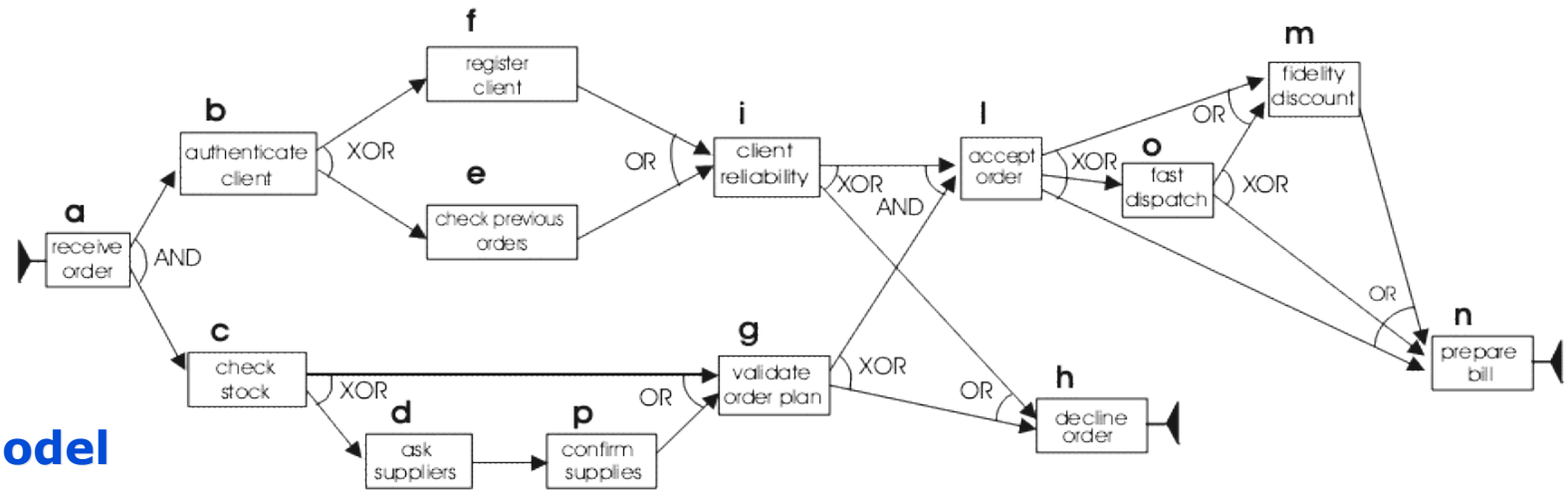
Overview of Basic Approaches

s_1: acdbfgih	s_5: abicglmn	s_9: abficgln	s_{13}: abcidglmn
s_2: abficdgh	s_6: acbiglon	s_{10}: acgbfilon	s_{14}: acdbiglmn
s_3: acgbfih	s_7: acbgilomn	s_{11}: abcfdigln	s_{15}: abcdgilmn
s_4: abcgiln	s_8: abcfgilon	s_{12}: acdbfigln	s_{16}: acbidgln

Event Log

Basic induction scheme

- Mine a **Precedence Graph** encoding a minimal set of precedence links
- Mine a set of cardinality (local) constraints, based on simple statistics
- Introduce support thresholds to handle noisy data



Mined Model

α-algorithm: Ordering Relations >,→,||,#

- **Direct succession**:

  x>y iff for some case, x is directly followed by y

- **Causality**:

  x→y iff x>y and not y>x

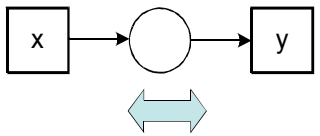
- **Parallel**:

  x||y iff x>y and y>x

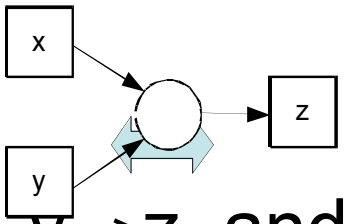
- **Unrelated**:

  x#y iff not x>y and not y>x

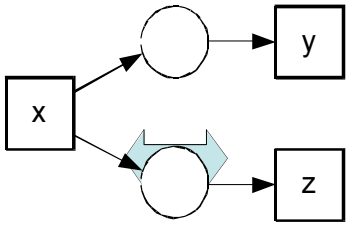
From the Orderings to Petri Nets



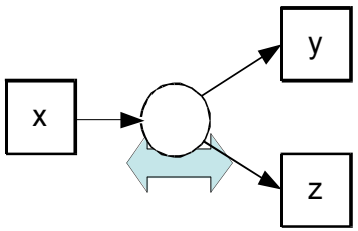
$x \rightarrow y$



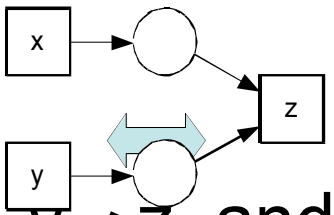
$x \rightarrow z$, $y \rightarrow z$, and $x \# y$



$x \rightarrow y$, $x \rightarrow z$, and $y || z$



$x \rightarrow y$, $x \rightarrow z$, and $y \# z$



$x \rightarrow z$, $y \rightarrow z$, and $x || y$

Handling noise: Heuristics Miner

Let W be an event log over T, and $a, b \in T$:

- $|a >_W b|$ is the number of times $a >_W b$ occurs in W,
- $a \Rightarrow_W b = \left(\dfrac{|a >_W b| - |b >_W a|}{|a >_W b| + |b >_W a| + 1} \right)$

Insight:

The more frequently task A directly follows task B, and the less frequently the converse occurs, the higher the probability that A causally follows B.

Challenges in Process Mining

- *Problem*: complex processes may involve lots of activities, and complex behavioral rules for combining them
  - the discovered model may fail in representing the process with enough accuracy
  - … and may be too complex for business users who want to monitor and analyze process executions at an appropriate abstraction level

Desiderata

Execution Classification

This allows to gain in accuracy,modularity, and understandability, w.r.t. a single workflow schema mixing all executions

Abstraction

BPA platforms (e.g, iBOM by HP) allow to manually define abstract views over a workflow, by mainly aggregating groups of activities

Quality of a Mined Schema

The quality of a schema *W* can be measured w.r.t. the log *L* it was extracted from

- *Soundness*: % of traces of *W* that occur in *L*
- *Completeness*: % of traces in *L* that comply with *W*

High soundness is difficult to be achieved...

- Real-life processes may follow complex behavioral rules, which are beyond precedence links and local constraints
  - e.g., in the process *Order Management*, task *m* (*fidelity discount*) cannot be executed whenever task *f* (*register new client*) is
  - The mined graph-based schema will have low soundness: many modeled executions never occur in the actual log

How to Mine Good Models?

Alternatives

~~**Use more expressive languages / meta-models**~~

- ~~e.g., control flow graphs could be enriched with additional "global" constraints, relating nodes that are not adjacent to each other~~
- ~~but this can lead to knotty models and makes Process Mining harder~~

Mine more schemas

- The model consists of a number of workflow schemas
  - more expressive and precise than a single one
  - but still intuitive and easy to mine
- Complex behavioral rules are caught indirectly, by recognizing different usage scenarios

Various Process Discovery Problems

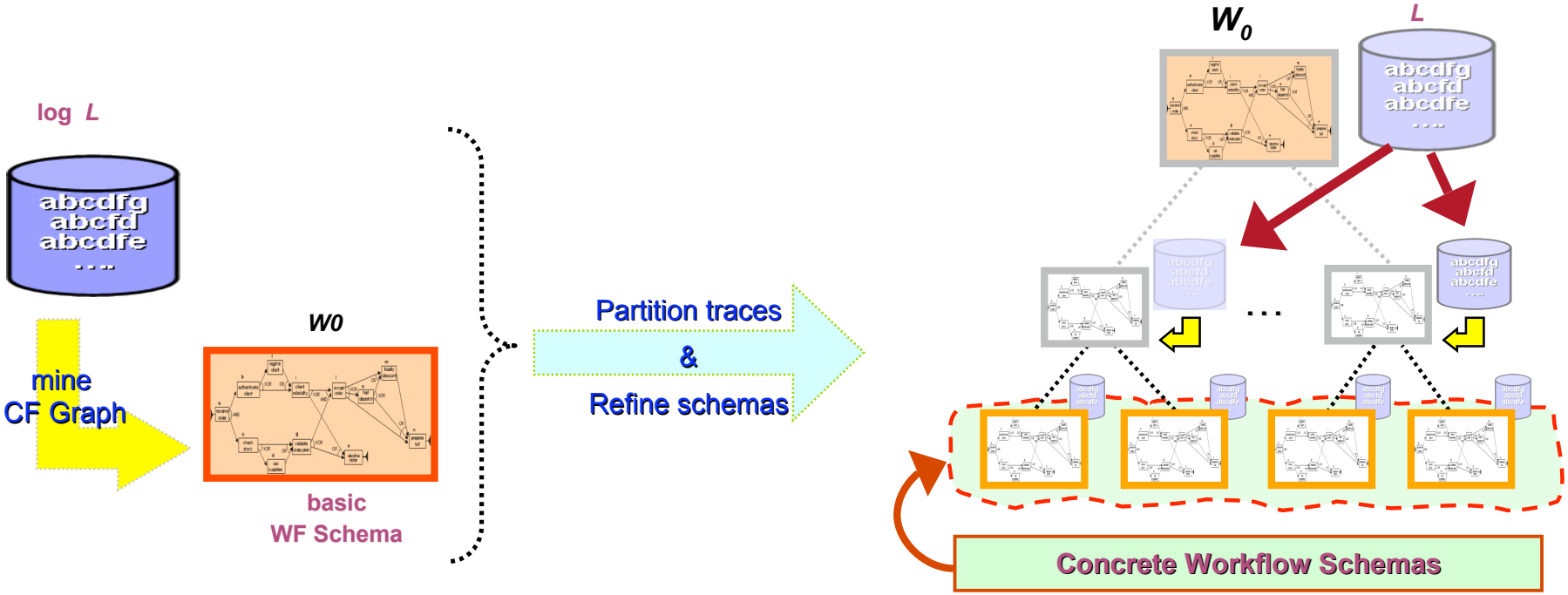
- **Exact Process Discovery Problem**
  - *Given*: a log *L,* two numbers *S*, *C* and an integer *M*
  - *Find*: a disjunctive schema *W** such that *soundness(W*,L)*≥ *S*, *completeness(W*,L)*≥*C* and *size(W*)*≤ *M*

  ➔The problem is NP-hard in the general case
- **Maximal Process Discovery**
  - *Given*: a log *L,* a numbers *C* and an integer *M*
  - *Find*: a disjunctive schema *W** s.t. *size*(*W**)≤ *M, completeness*(*W**,*L*)≥*C,* and *soundness*(*W**,*L*) is maximal

  ➔ An intractable problem again
- **Process Discovery Problem**
  - *Given*: a log *L,* a numbers *C* and an integer *M*
  - Find a disjunctive schema *W** s.t. *size(W*)*≤*M completeness(W*,L)*≥*C*, and *W** is as sound as possible

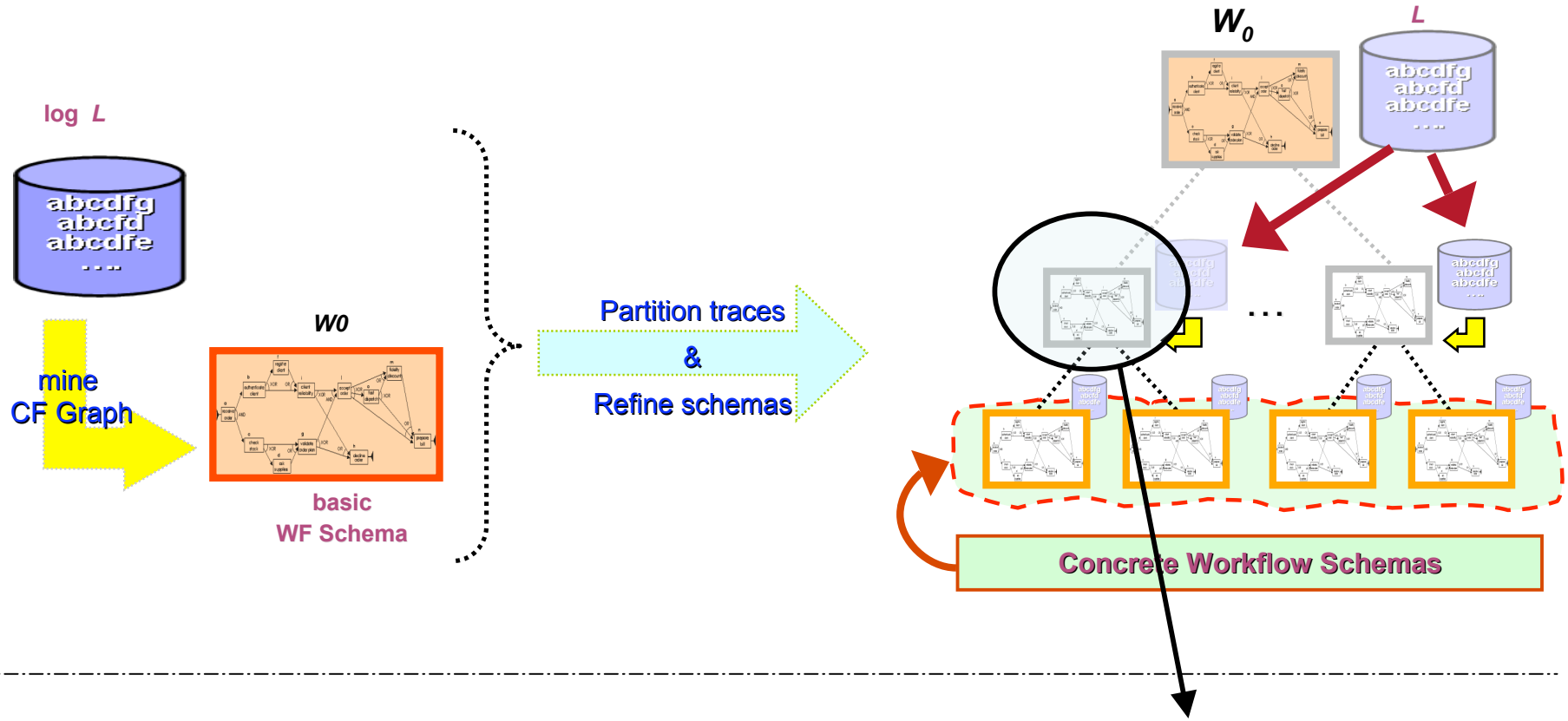
  ➔ Greedy, heuristic-based, solving algorithm

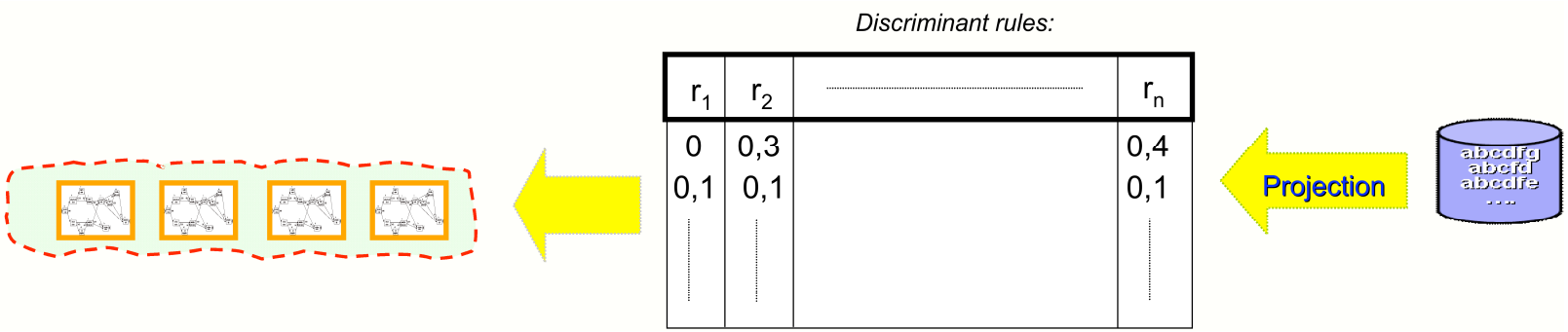
A Clustering-Based Approach to Process Discovery



- The basic schema W_0 is a first attempt to model all the log traces
- Iteratively, a leaf schema is refined to get higher soundness
  - the associated traces are split into (more homogenous) clusters
  - a new schema is derived for each cluster
- The schemas in the tree (specifically its leaves) are a model sounder than W_0
  - Soundness = % of traces of the leaf schemas that occur in the log

A Clustering-Based Approach to Process Discovery





Mining Techiniques

- Frequent Patterns
- Workflow Discovery
- Discovery of more refined process models
  - Co-clustering
  - Abstraction
  - Outlier Detection
  - Classification

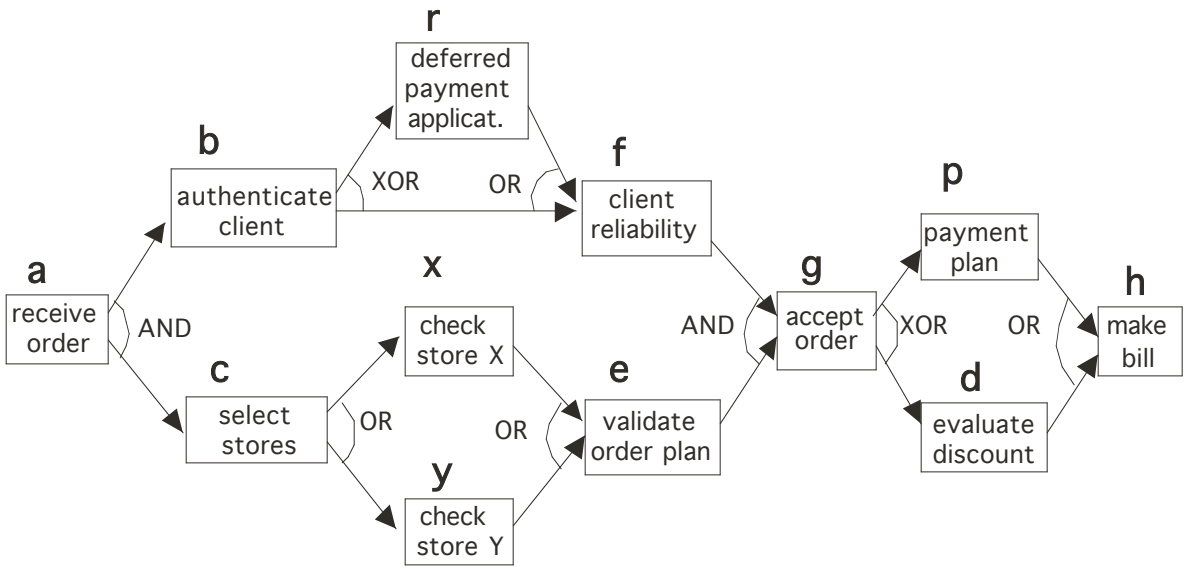
Co-Clustering (1/4)

Co-Clustering

- A Toy Example: **HANDLEORDER** process

ID	task sequence
t_1	abcrxefgph
t_2	abcrxyfegph
t_3	acbrxefgph
t_4	abcrxyefgph
t_5	acybrefgph
t_6	abcryefgph
t_7	abrcyefgph
t_8	abcyrfegph
t_9	abcyefgdh
t_{10}	abcxfyegdh
t_{11}	acbfyegdh
t_{12}	abfcxegdh

ID	time
t_1	8
t_2	9
t_3	9
t_4	8
t_5	2
t_6	4
t_7	4
t_8	5
t_9	4
t_{10}	2
t_{11}	1
t_{12}	1



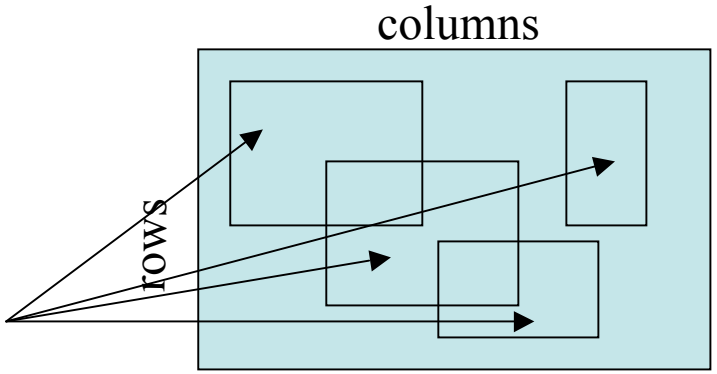
- NOTE:
  - a, b, c, e, f, g and h are executed in all the instances, while the other ones appear in a subset of them only.
  - Traces containing r also includes p and does not contain d, conversely d appears any time does not.

Two *(structurally)* homogeneous clusters:

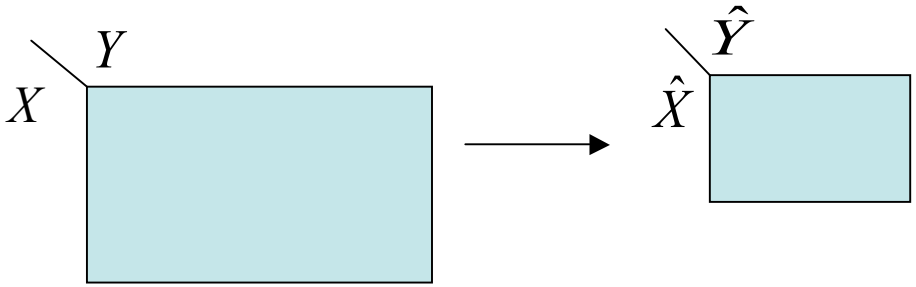
Orders with deferred payment & Order with immediate payment

Information-Theoretic Framework (1/2)

Co-clustering: Simultaneous row/column clustering, possibly overlapping ..

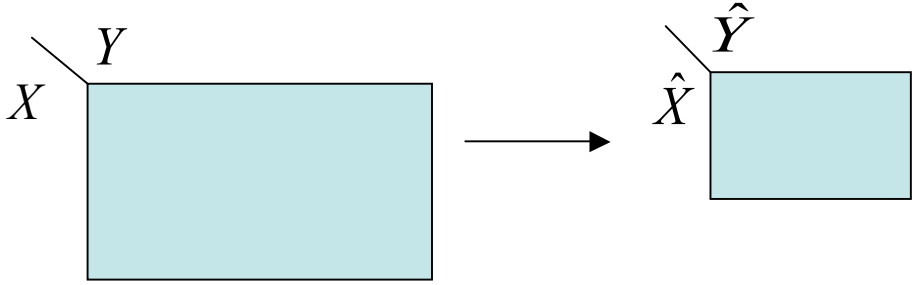


Information Theoretic: views the contingency table as an empirical join probability distribution of two discrete variable…



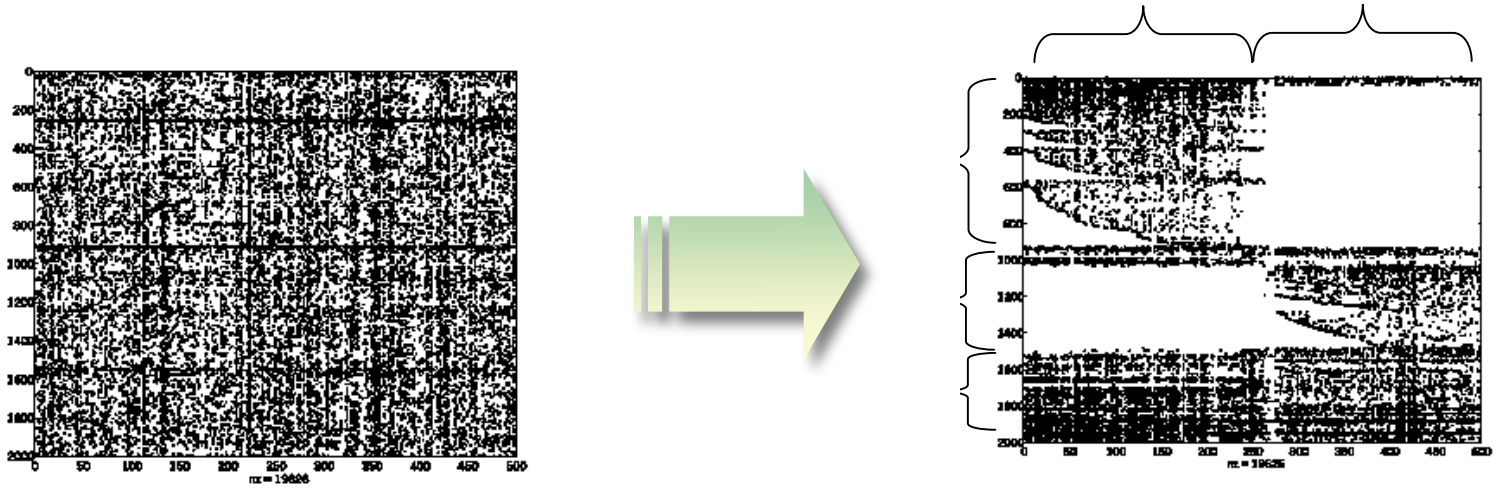
….and poses the co-clustering problem as an optimization problem

Information-Theoretic Framework (2/2)

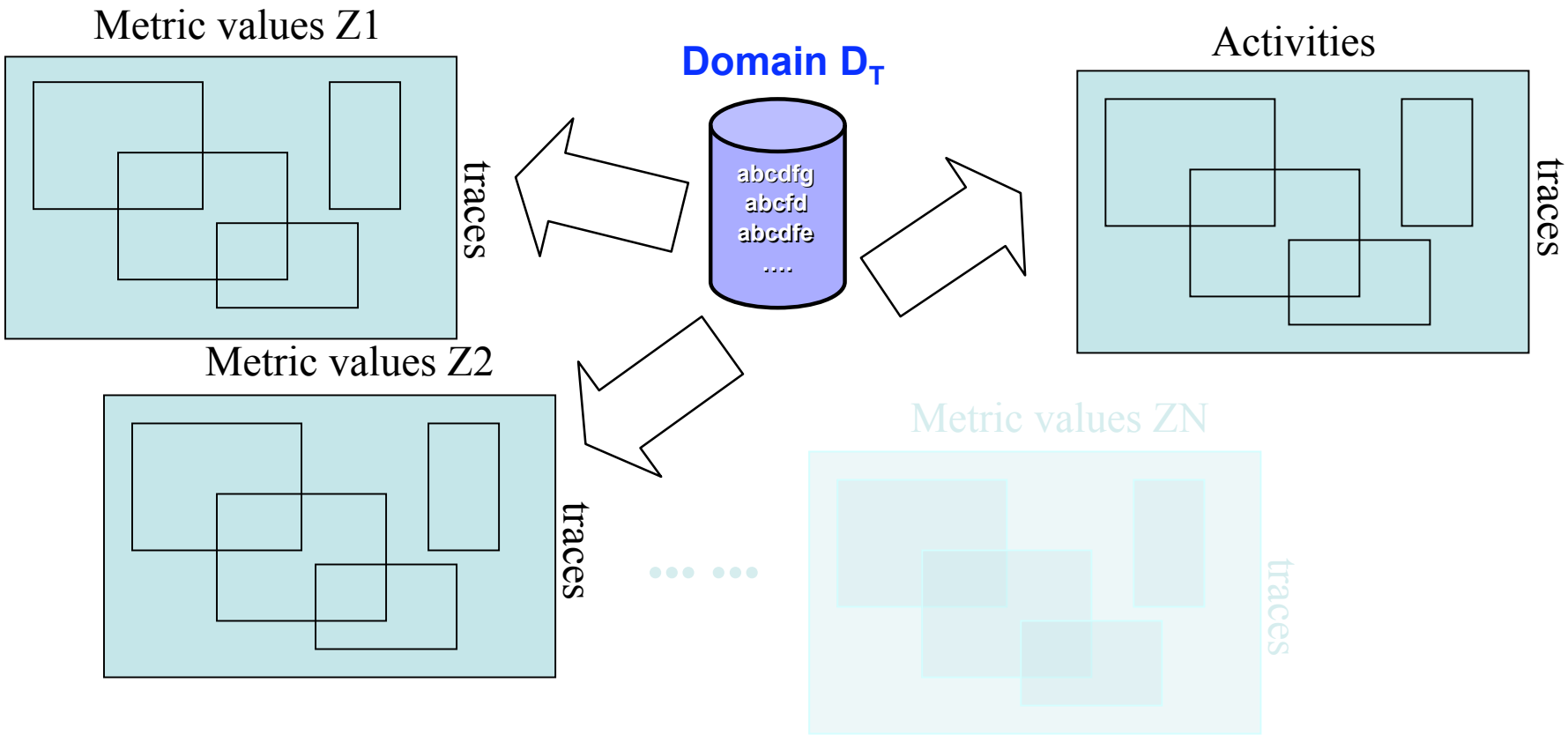


Optimal clustering: minimizes the loss in mutual information:

$$\Delta I = I(X;Y) - I(X'=C_X(X);Y'=C_Y(Y)) = I(X;Y) - I(X';Y')$$



Multidimensional Co-clustering (1/2)



For each auxiliary domain, one Information Loss : ΔI_A, ΔI_{z1}, …, ΔI_{zN}

Multidimensional Co-clustering (2/2)

Consider the linear combination $\beta_A \Delta I_A + \Sigma\, \beta_{Zi} \Delta I_{Zi}$

β_A; β_1;.. β_N; are meant to quantify the relevance that the corresponding auxiliary domain should have in determining the co-clustering of the whole log data, with $\beta_A + \Sigma\, \beta_j = 1$ (j =1…N).

Performance measures (*Metric Domains)* are ordered!

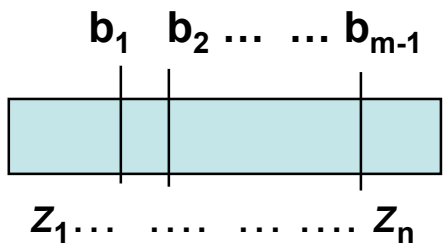
- the ordering must still holds among the clusters…
- …but, traditional approaches to IT co-clustering are defined for (unordered) sets.

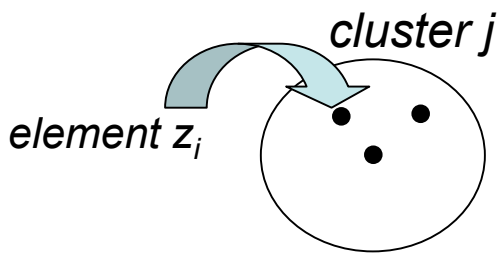
Partitioning Numerical Domains

Problem Formulation:

- Given **n** numerical values $\mathbf{D_z=\{z_1,...,z_n\}}$ and **m** cluster labels
- **Find** a clustering function $\mathbf{C_Z}$ that:
  - *"preserves the ordering"* over $\mathbf{D_z}$.
  - *minimizes* the information loss $\Delta\mathbf{I_Z}$

✓ **C_Z can be defined through m-1 boundary indexes**





iff $b_j-1 \leq i < b_j$

✓ **ΔI_Z can expressed in terms of *point-wise* Information Losses**

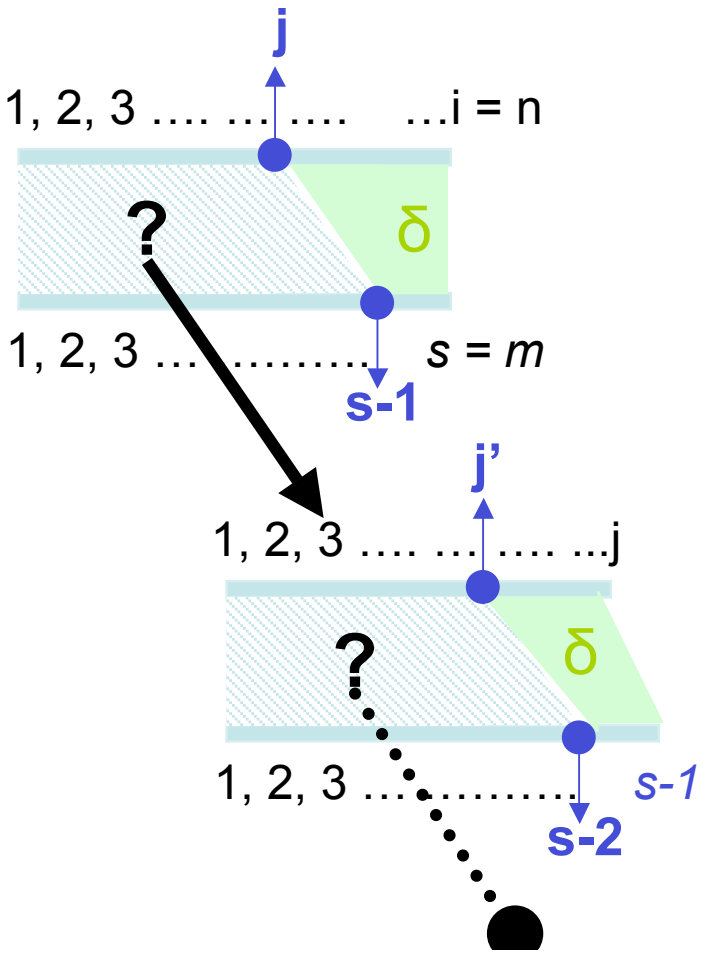
$$\sum_{\widehat{z}_l \in \widehat{D}_Z} \sum_{i=b_{l-1}}^{b_l-1} \delta(z_i, \widehat{z}_l)$$

A dynamic programming approach

Objective function:

$$\Delta I_Z = \sum_{\hat{z}_l \in \hat{D}_Z} \sum_{i=b_{l-1}}^{b_l - 1} \delta(z_i, \hat{z}_l)$$

$$\Delta I_Z = g\,(m, n)$$



Let ***g(s,i)*** be the minimum cost of splitting the first ***i*** values in D_Z among the first ***s*** clusters

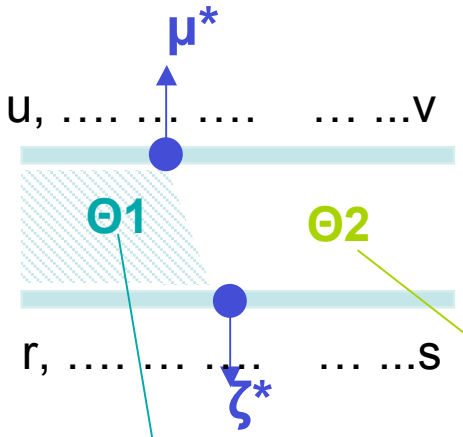
Recursive formulation:

$$g(s, i) = \min_{\{1 \le j \le i\}} [\, g(s-1, j) + \sum_{j<k\le n} \delta(k, s)]$$

- Cost of assigning every element k following j to the s-th cluster :
- *Trivial computation…can be improved by precomputing cumulative sums for each element and for each cluster*

A greedy approach

Iteratively performs a binary split on both D_Z and $D^{\wedge}_{Z,}$
…until there are only singleton cluster ranges



Split point μ* guarantees the lowest value the (global) loss in mutual information Θ1 + Θ2, based on the estimation **heuristic function Θ**

Cluster range [r,s] is divided into two equi-numerous sub-ranges,

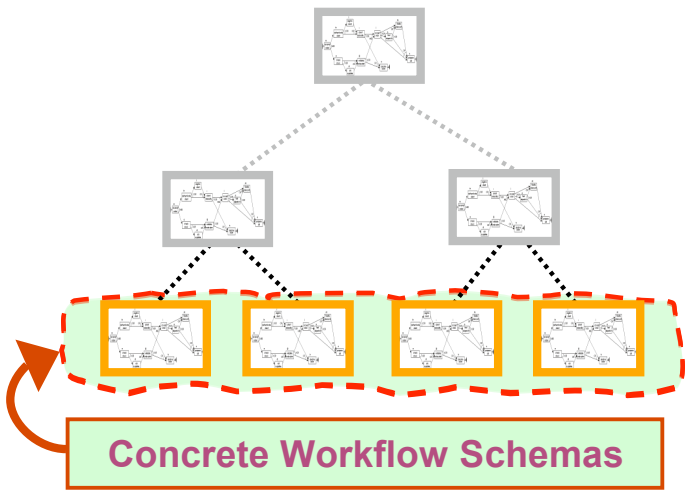
Estimates the mutual information assigning $z_{\mu^*+1}, \ldots, z_v$ to some clusters in $z^{\wedge}_{\zeta^*+1}, \ldots z^{\wedge}_s$

Estimates the mutual information assigning $z_u, \ldots, z_{\mu^*}$ to some clusters in $z^{\wedge}_r, \ldots z^{\wedge}_{\zeta^*}$

Abstraction (2/4)

Discovering Taxonomical Process Models

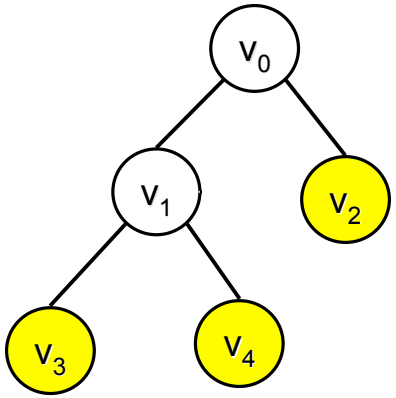
- Output: A tree of workflow schemas

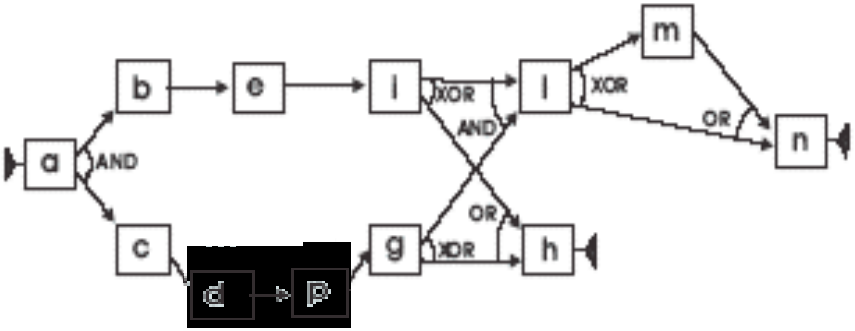


  - The tree describes the process behavior at different level of details
  - At the highest level of detail (leaves of the tree), the schemas could be used to support the design of concrete workflow models
  - At lower levels, the schemas are abstract views over heterogeneous behaviors, which could support analysis and monitoring tasks

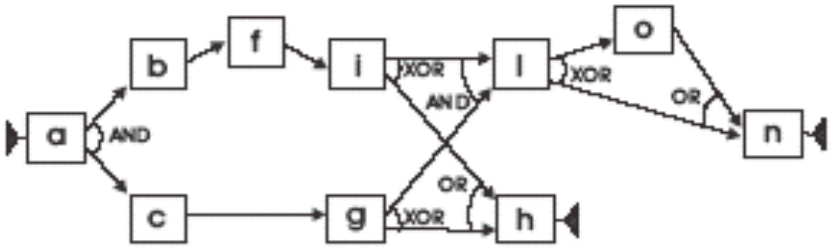
- Method: two-phase approach, combining mining and abstraction methods
  - First, we mine a tree of workflow schemas, by using a hierarchical, top-down, clustering algorithm
  - Then, the mined model is restructured at several levels of abstraction, in a bottom-up way (i.e., from the leaves to the root)

Leafs as Concrete Schemas

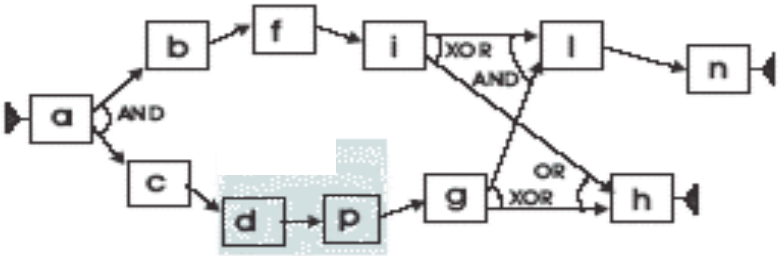




Workflow schema W_2 for node v_2



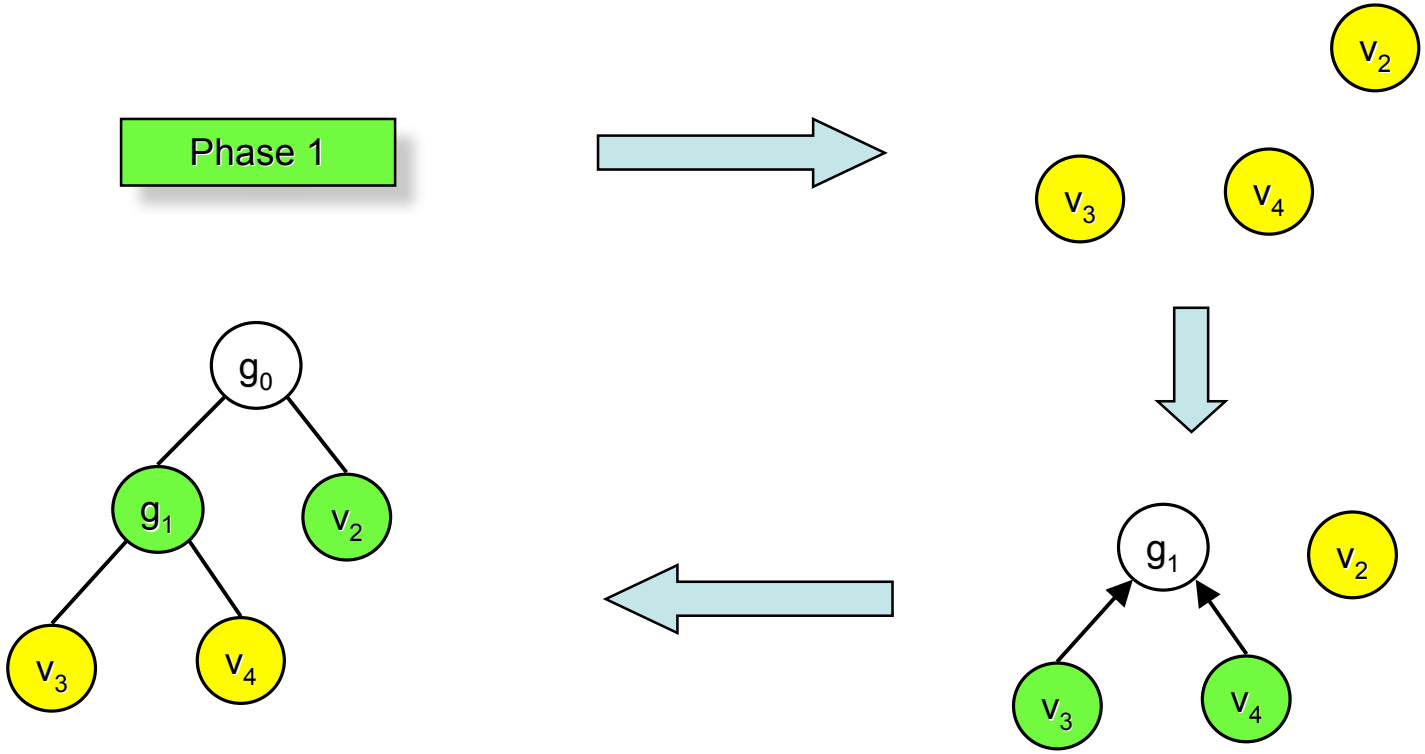
Workflow schema W_3 for node v_3



Workflow schema W_4 for node v_4

Abstraction Process (1/3)

- Every non-leaf schema in the hierarchy is replaced with an abstract schema that generalizes those of its children
  - The process is applied in a bottom-up way, i.e., form the leaves to the root of the hierarchy



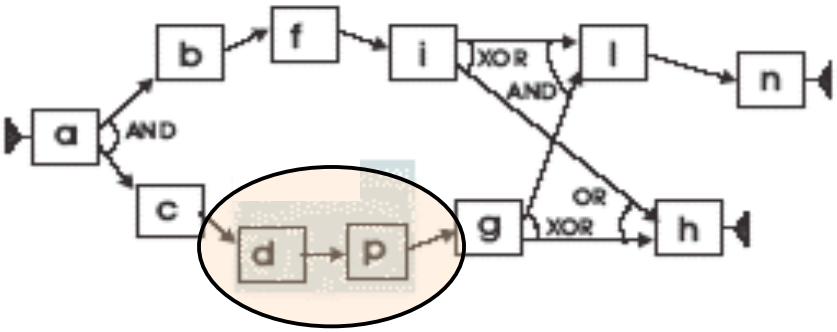
g_0 is computed that abstracts v_2 and g_1

g_1 is computed that abstracts v_3 and v_4

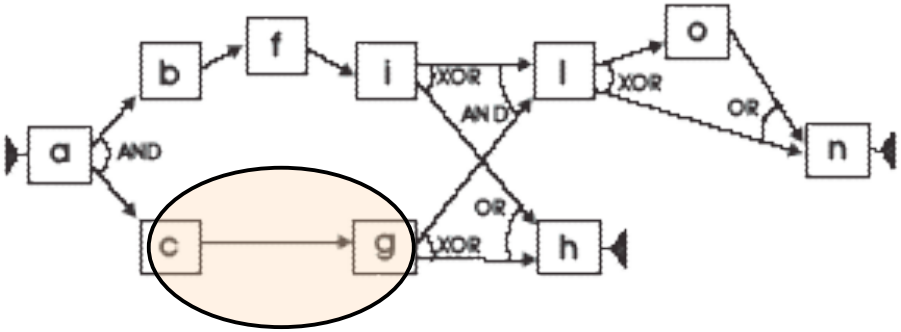
How two schemas are generalized?

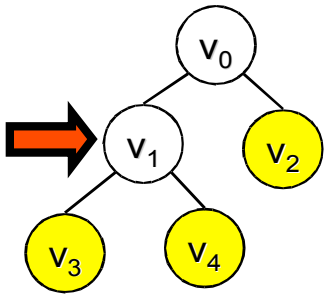
- Computation of the generalized schema for a non-leaf node
  - For each child schema *abstract* "specific" activities (activities that do not occurring in all children)
  - Merge all the children schemas into a single one
    - compute the union of the graphs, and adjust all constraints
  - *Abstract* "specific" activities appearing in the merged schema

Schema of v_3



Schema of v_4





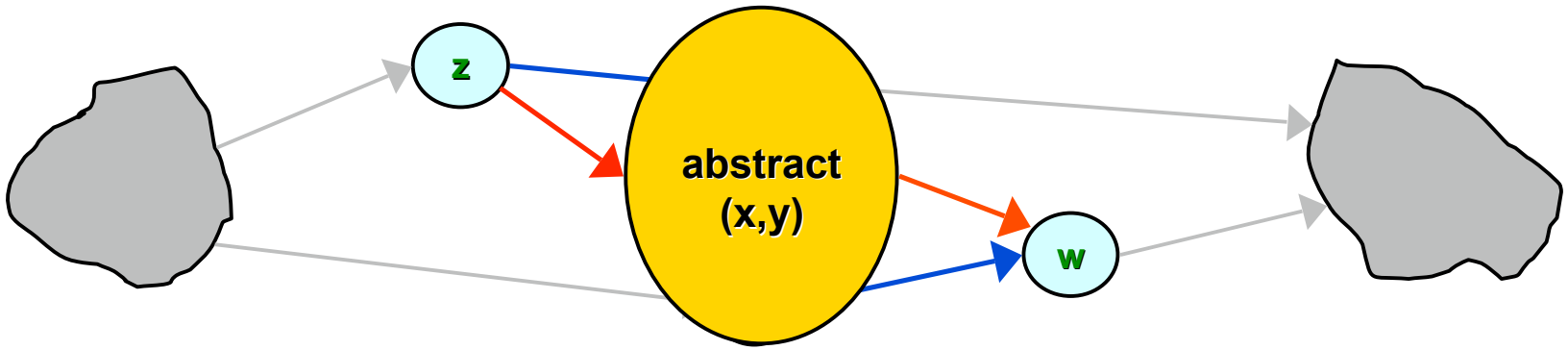
- We need a strategy to recognize groups of "specific" activities that can be abstracted by the same higher-level activity ….

Merging Activities to be Abstracted

- Pair-wise approach
  - A pair of “specific” activities is greedily chosen for being abstracted together into a single higher-level activity
- A notion of safety w.r.t. merge for pairs of activities
  - for preventing the creation of “spurious” dependencies among not abstracted activities, in the generalized schema
- A series of affinity measures assessing how much two any “specific” activities are suitable to be merged
  - A “topological” affinity measureTopological $sim^E(x,y)$
    - how similar the neighborhoods of *x* and *y* are w.r.t. the flow graph
  - Two “semantical” affinity measures, $sim^D{}_P(x,y)$ and $sim^D{}_G(x,y)$
    - how similar *x* and *y* are w.r.t. the generalization/aggregation relationships stored in an abstraction dictionary *D*

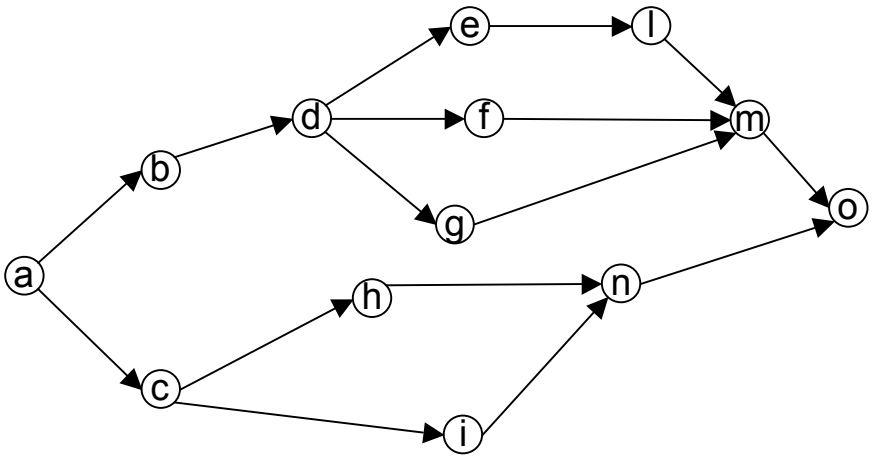
Merge-safe Activities

- A pair of activities (*x, y*) is *merge-safe* w.r.t. a given an edge set *E*, if one of the following conditions holds:
  - *x* and *y* are directly linked by some edges in *E* and after removing these edges no other path exists between them
  - there is no path in *E* connecting *x* and *y*



Outlier Detection (3/4)

Outlier Detection Challenges

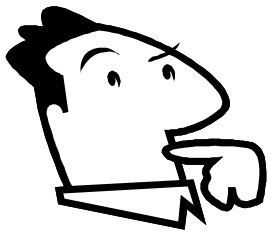


Process Log: a set of execution traces

- abcdef….
- acbdef….
- abcdfe….

Application of traditional sequential outlier techniques may be misleading

- a lot of traces that only differ in the ordering between parallel tasks may be interpreted as anomalous (***false positive***)

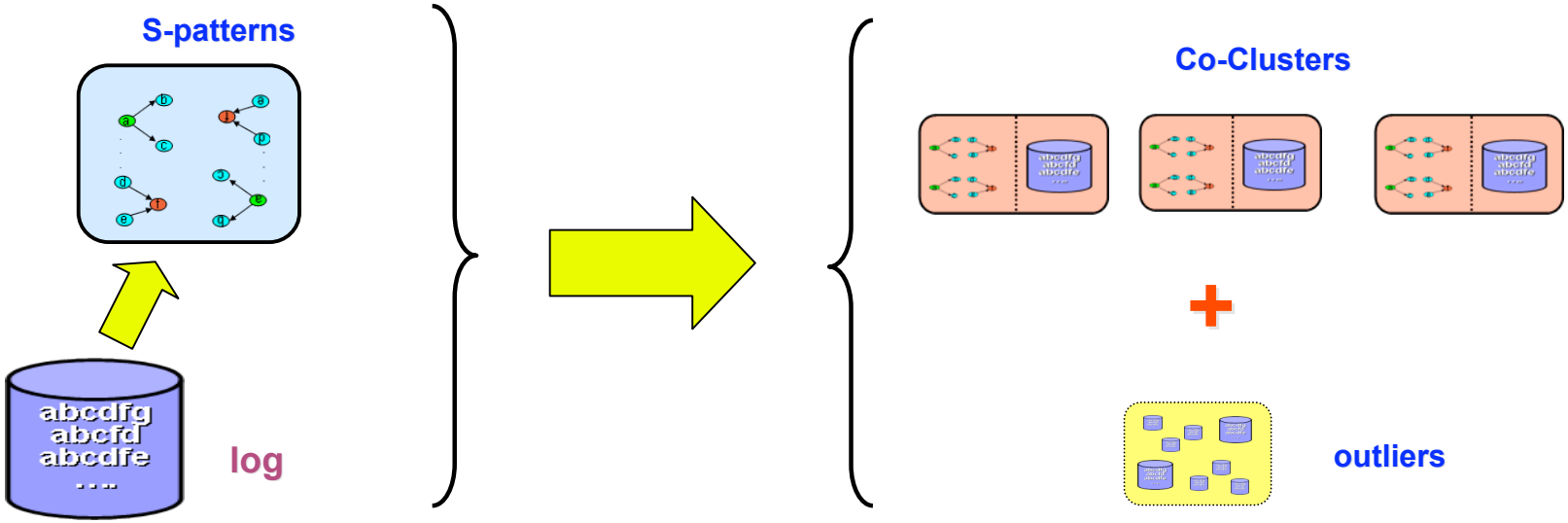


Considering the compliance with an ideal schema may fails too

- some trace might well be supported by a model, yet representing anomalous behavoiurs (***false negative)***

Outlier Detection

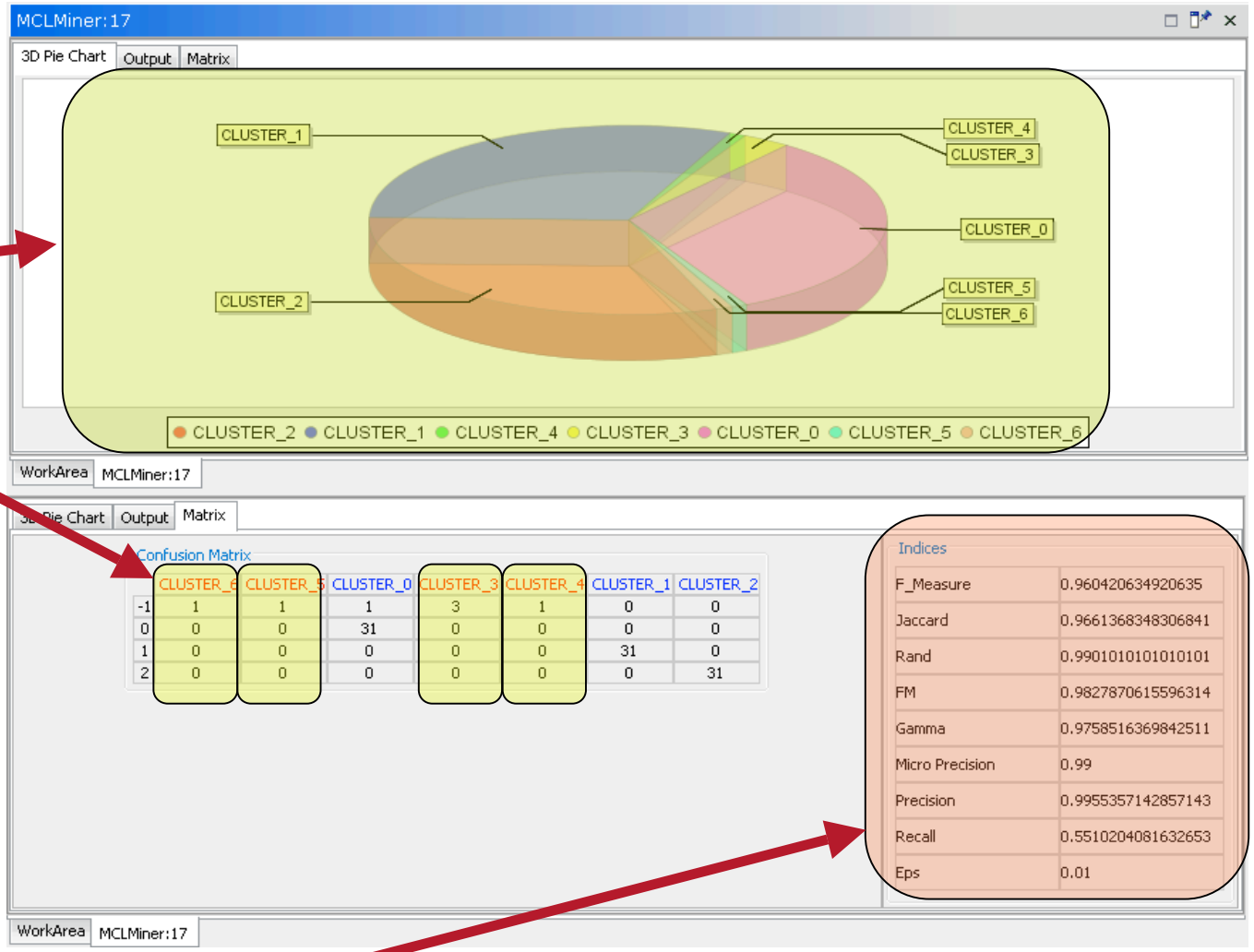
- Basic Idea:
  - Structural patterns are co-clustered with the traces, based on what an extent these latter support them
  - Mark as outlier each trace *t* such that either
    - *t* has not been assigned to any cluster
    - *t* belongs to a cluster whose cardinality is “appreciably smaller” than the average cluster size



Outlier Detection …in Action

Statistics on found clusters

Red clusters correspond to outliers



Accuracy measures
-- computed only on a test dataset, where real classes/outliers are known

Classification (4/4)

Non-Structural Aspects

They focus on "process structure" (tasks' execution)

- Other "non-structural" data are disregarded

Structural Data:

- Tasks, Timestamps

Non-structural Data:

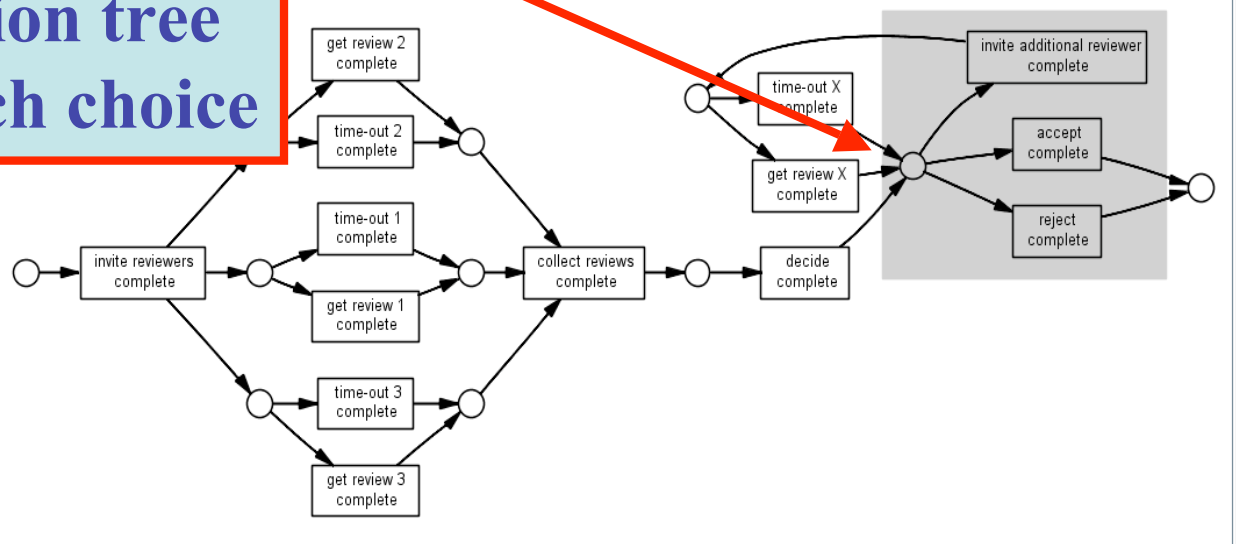
- Humans, Parameters, Performances, …

Two directions:

- Extend workflow models by capturing the dependence of the execution flow on non-structural features
- <u>Execution data co-clustering:</u> the clustering of log traces is performed in a synergic way w.r.t. both structural and non-structural aspect

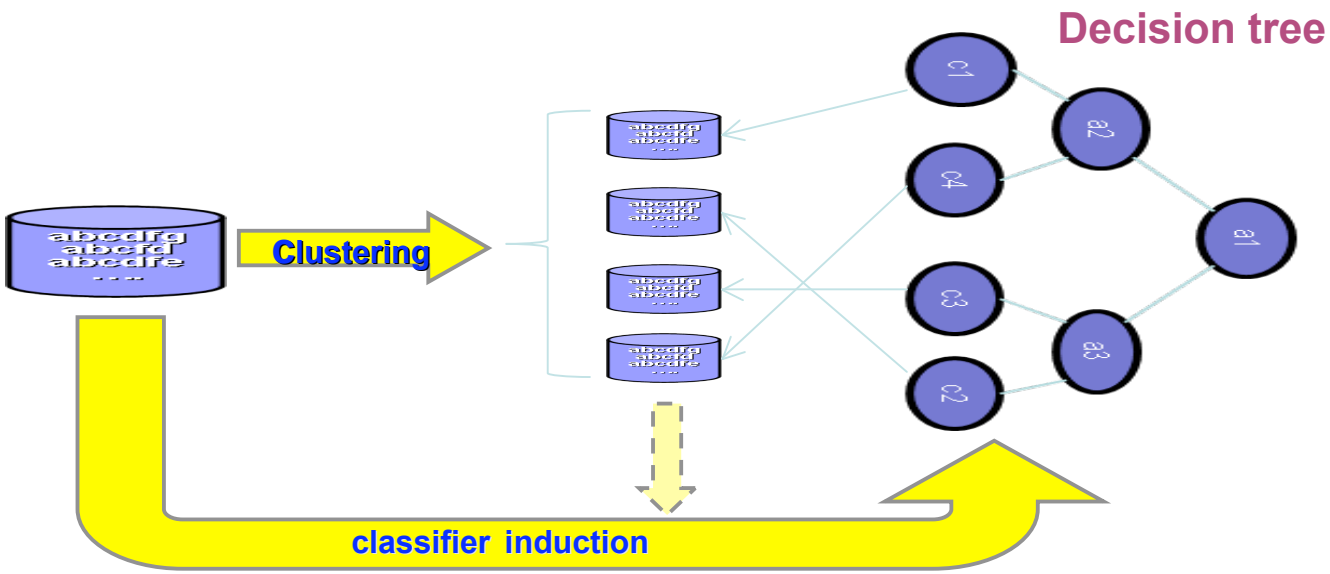
Decision Point Analysis

builds a decision tree for each choice



Decision Trees

- Basic idea:
  - find a comprehensive representation for the process, describing both structural and non-structural aspects
  - A rule-based classifier is induced to discriminate among given structural clusters, based on process/task data
  - help interpreting/predicting the different ways of executing the process, based on properties of process/task instances



Thank you!

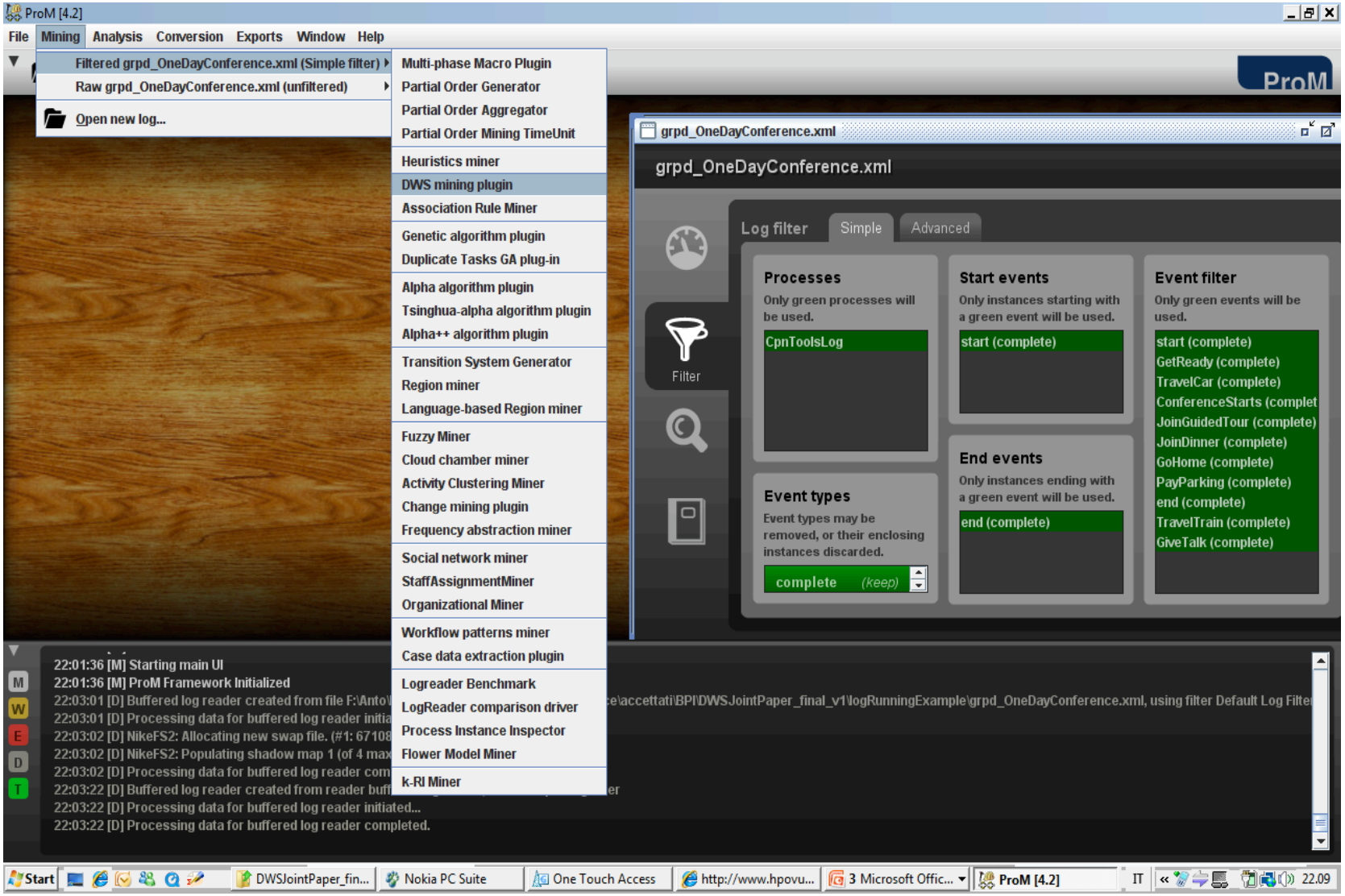
References(1/2)

- J. Cook, A. Wolf, Event-based detection of concurrency, in: Proc. 6th Int. Symposium on the Foundations of Software Engineering (FSE'98), 1998, pp. 35-45.
- W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. Data and Knowledge Engineering, 47(2):237-267, 2003.
- A.J.M.M. Weijters and W.M.P. van der Aalst. Rediscovering Workflow Models from Event-Based Data using Little Thumb. Integrated Computer-Aided Engineering, 10(2):151-162, 2003.
- G. Schimm, Mining most specific workflow models from event-based data, Business Process Management (2003) 25-40.
- M. Golani, S. S. Pinter, Generating a process model from a process audit log, in: Intl. Conf. on G. Greco, A. Guzzo, L. Pontieri, D. Saccà: “Mining Expressive Process Models by Clustering Workflow Traces”. In Proc. of 8th Pacific-Asia Conference (PAKDD 2004), Sydney, Australia, May 26-28, 2004: pp. 52-62.
- W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering…..
- J. Herbst, D. Karagiannis, Integrating machine learning and workflow management to support acquisition and adaptation of workflow models, Journal of Intelligent Systems in Accounting, Finance and Management 9 (2000) 67-92.
- R. Agrawal, D. Gunopulos, F. Leymann, Mining process models from workflow logs, in Proc. 6th Int. Conf. on Extending Database Technology (EDBT'98), 1998, pp. 469-483
- W.M.P. van der Aalst and B.F. van Dongen. Discovering Workflow Performance Models from Timed Logs. EDCIS 2002, volume 2480 of Lecture Notes in Computer Science, pages 45-63. Springer-Verlag, Berlin, 2002.

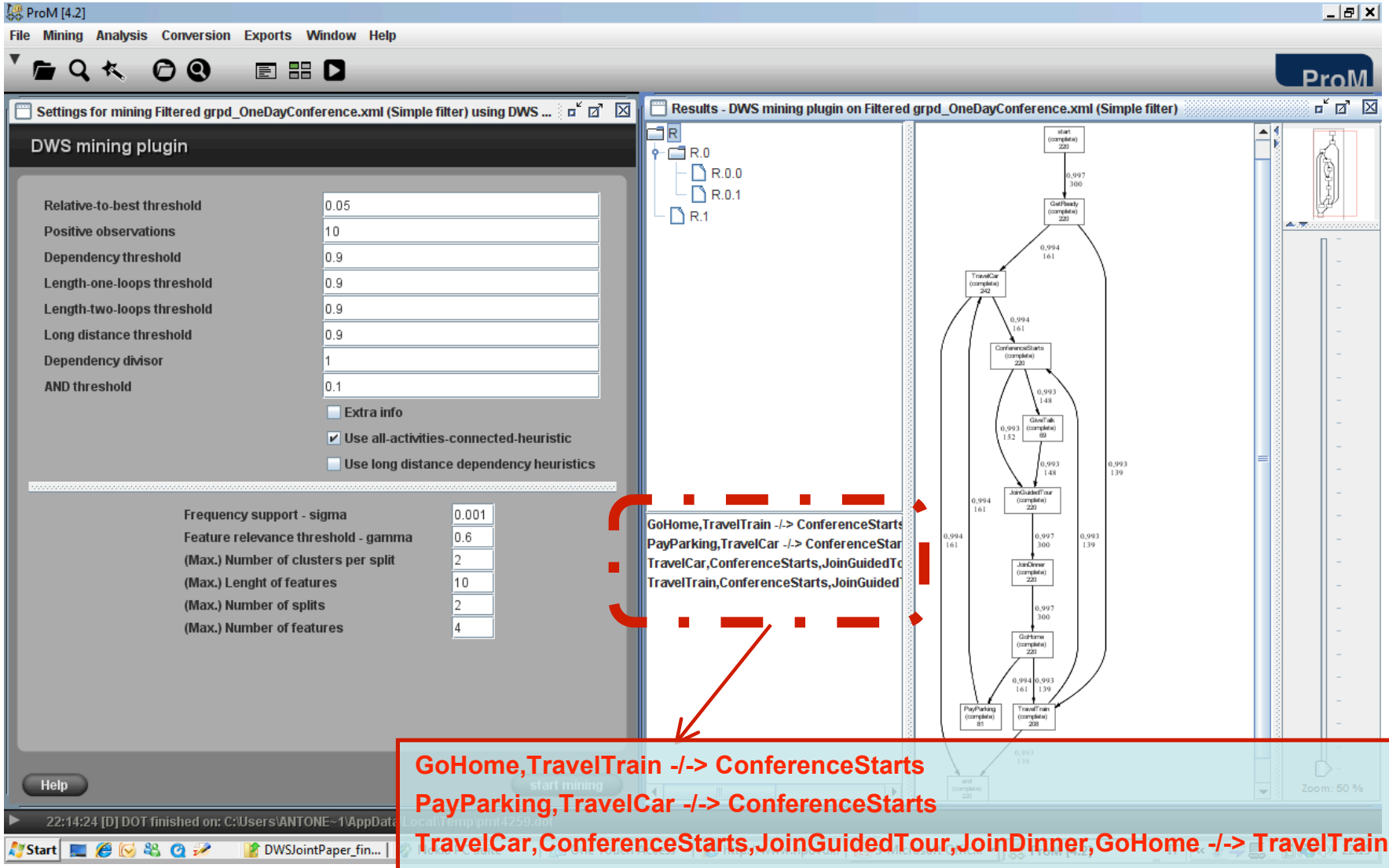
References(2/2)

- G. Greco, A. Guzzo, L. Pontieri, D. Saccà: Discovering Expressive Process Models by Clustering Log Traces. In IEEE Transactions on Knowledge and Data Engineering, 18 (8), pp. 1010-1027, IEEE Computer Society (2006).
- G. Greco, A. Guzzo, L. Pontieri: Mining Taxonomies of Process Models. Data & Knowledge Engineering, 67 (1), Elsevier Science: pp. 74-102 (2008).
- F. Folino, G. Greco, A. Guzzo, L. Pontieri. Discovering Multi-perspective Process Models: The Case of Loosely-Structured Processes. ICEIS 2008, LNBIP 19: pp. 130–143. Springer-Verlag Berlin Heidelberg (2009).
- A. D. Chiaravalloti, G. Greco, A. Guzzo, L. Pontieri: An Information-Theoretic Framework for Process Structure and Data Mining. In Proc. of 8th Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK 2006), Krakow, Poland, September 4-8, 2006, Lecture Notes in Computer Science 4081, Springer 2006: pp. 248-259.
- G. Greco, A. Guzzo, L. Pontieri: An Information-Theoretic Framework for Process Structure and Data Mining. International Journal of Data Warehousing and Data Mining, 3 (4), pp. 99-119, IGI Publishing (2007).
- L. Ghionna, G. Greco, A. Guzzo, L. Pontieri: Outlier Detection Techniques for Process Mining Applications. In Proc. of the 17th Intl Symposium on Foundations of Intelligent Systems (ISMIS 2008, Toronto, Canada, May 20-23, 2008): pp. 150-159
- L. Maruster, A.J.M.M.Weijters,W. M. P. van der Aalst, and A. van den Bosch. A rule-based approach for process discovery: Dealing with noise and imbalance in process logs. Data Mining and Knowledge Discovery, 13(1):67–87, 2006.

ProM in Action



Discriminant Rules …in Action

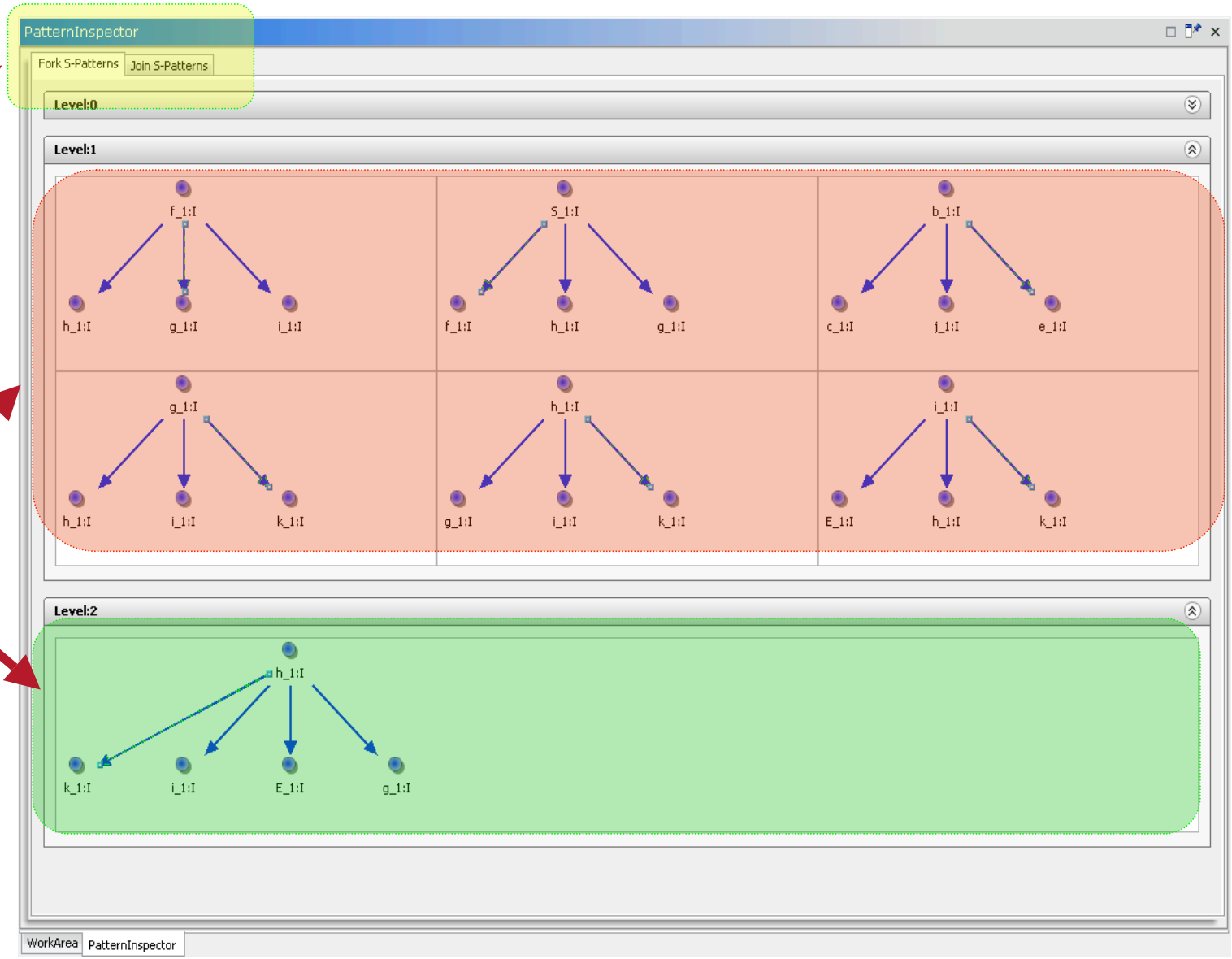


GoHome,TravelTrain -/-> ConferenceStarts
PayParking,TravelCar -/-> ConferenceStarts
TravelCar,ConferenceStarts,JoinGuidedTour,JoinDinner,GoHome -/-> TravelTrain
TravelTrain,ConferenceStarts,JoinGuidedTour,JoinDinner,GoHome -/-> PayParking

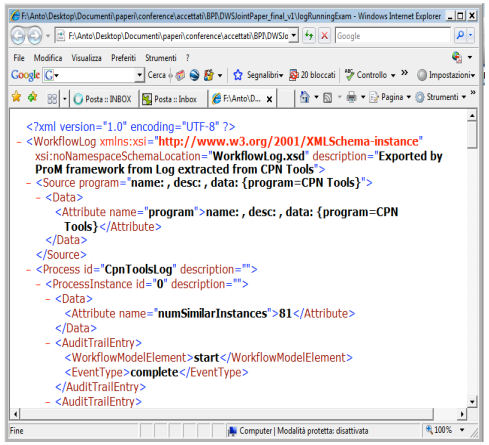
Frequent Pattern Mining …in Action

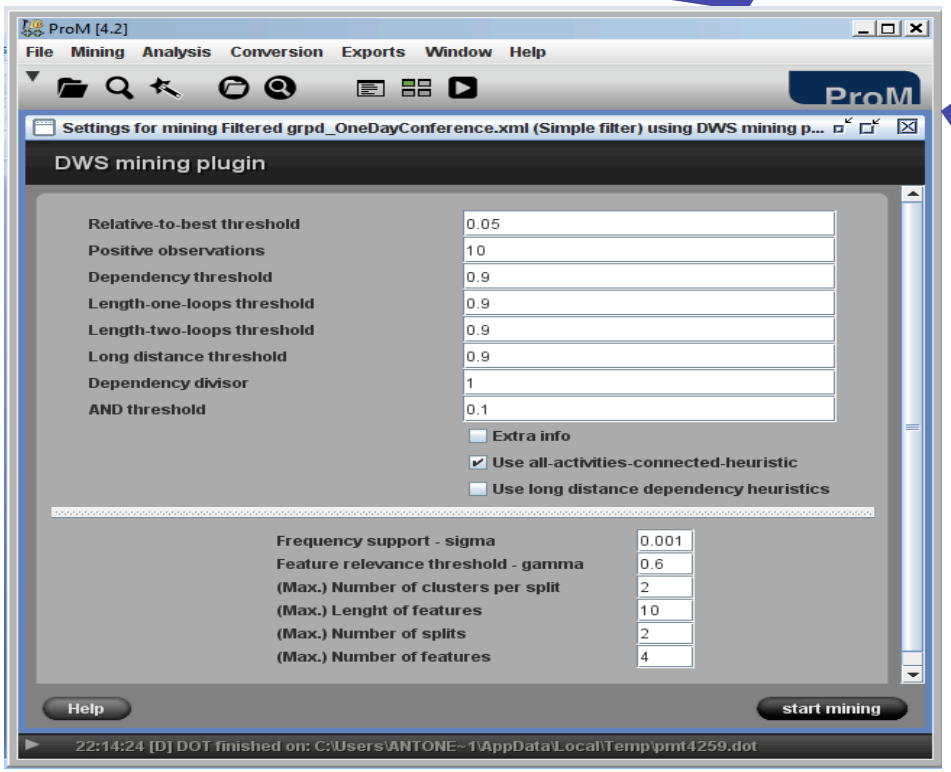
Kind of patterns (fork/join)

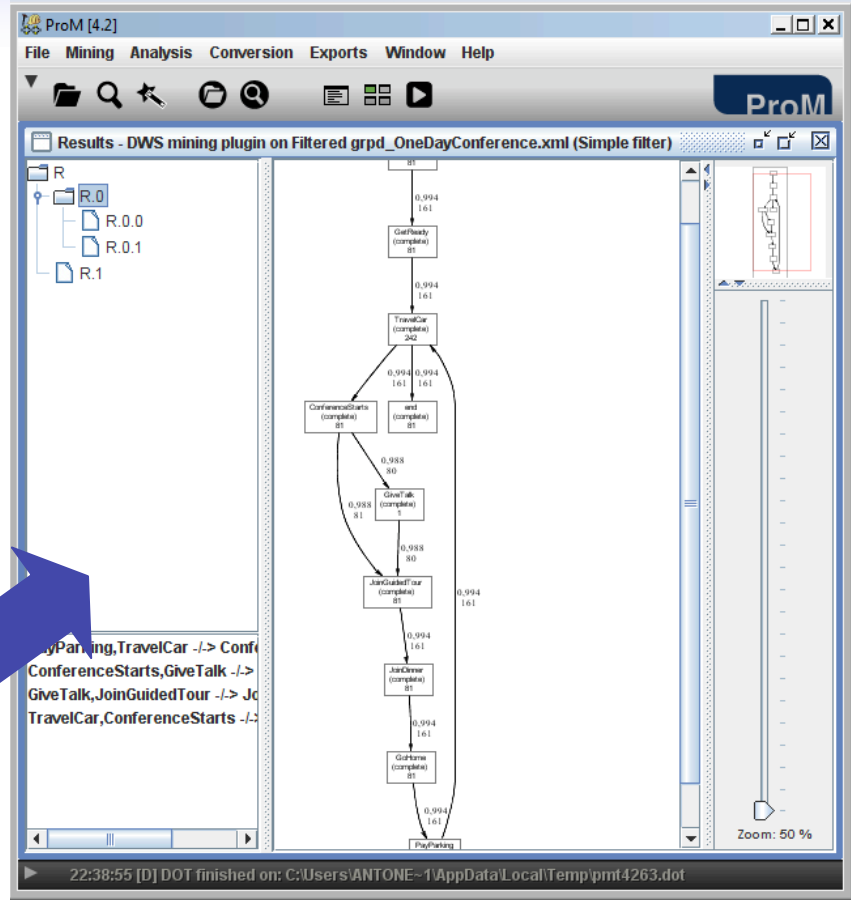
Patterns are grouped by size (#edges)



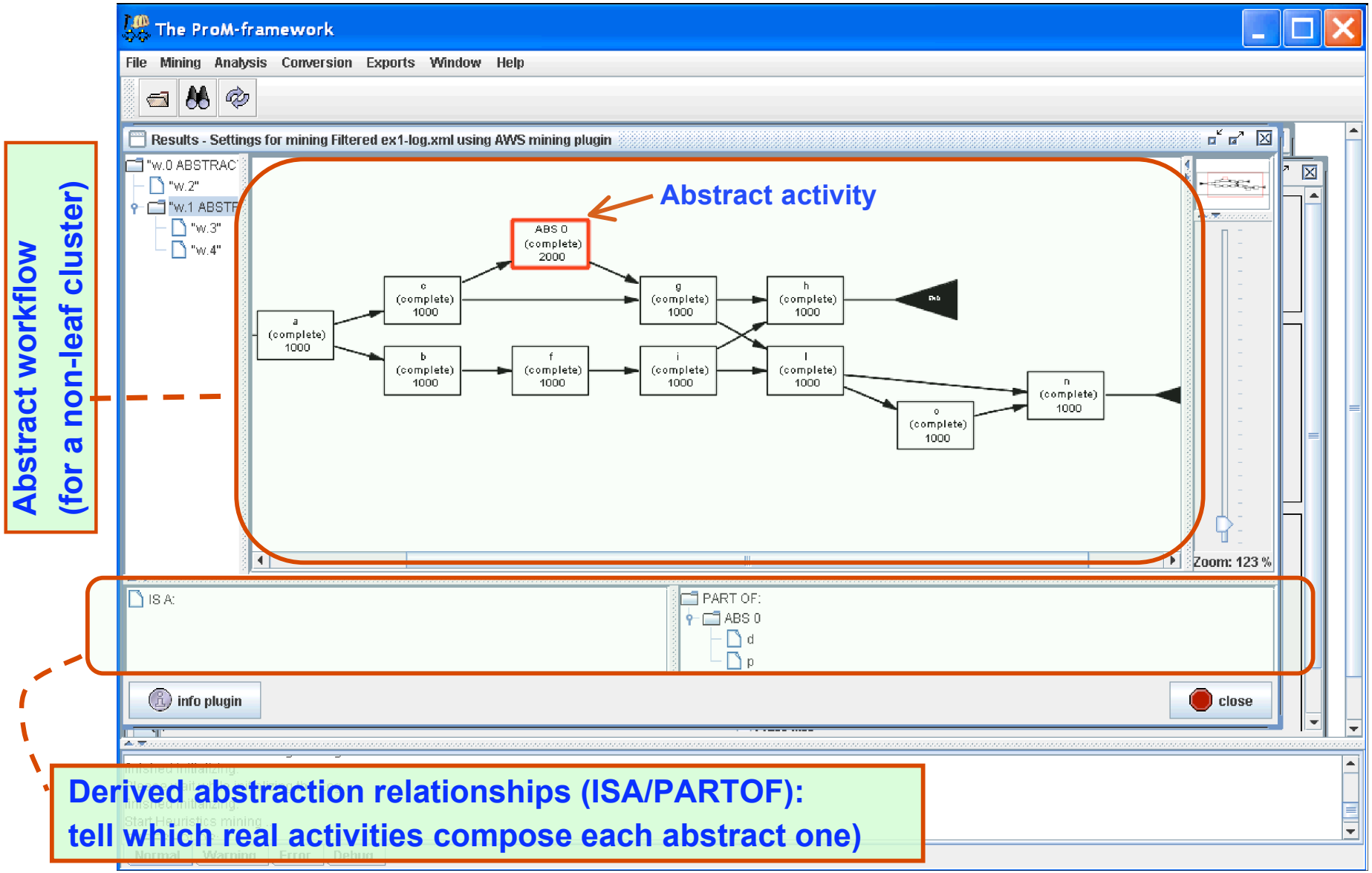
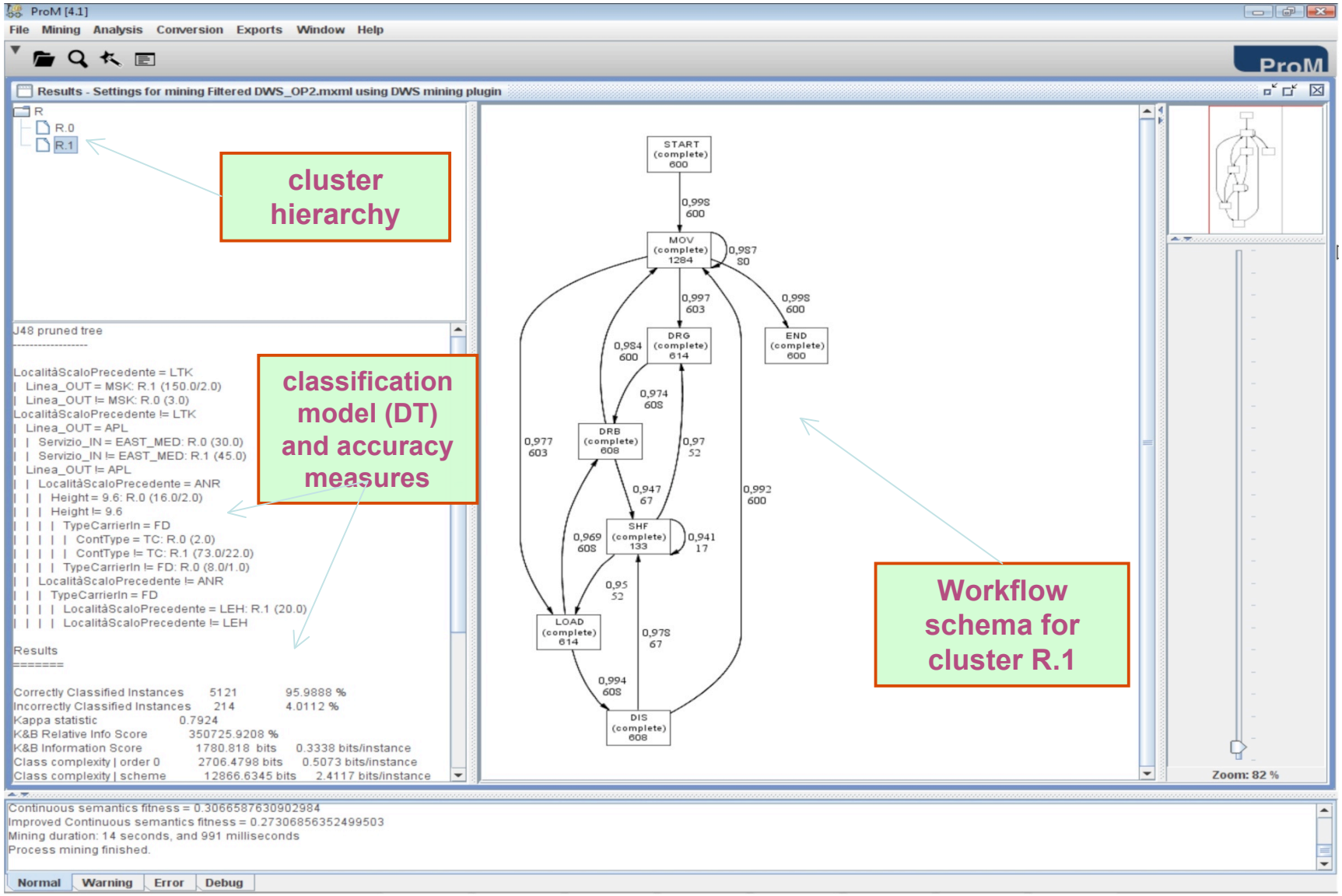
Clustering …in Action







Abstraction …in Action



Classification …in Action