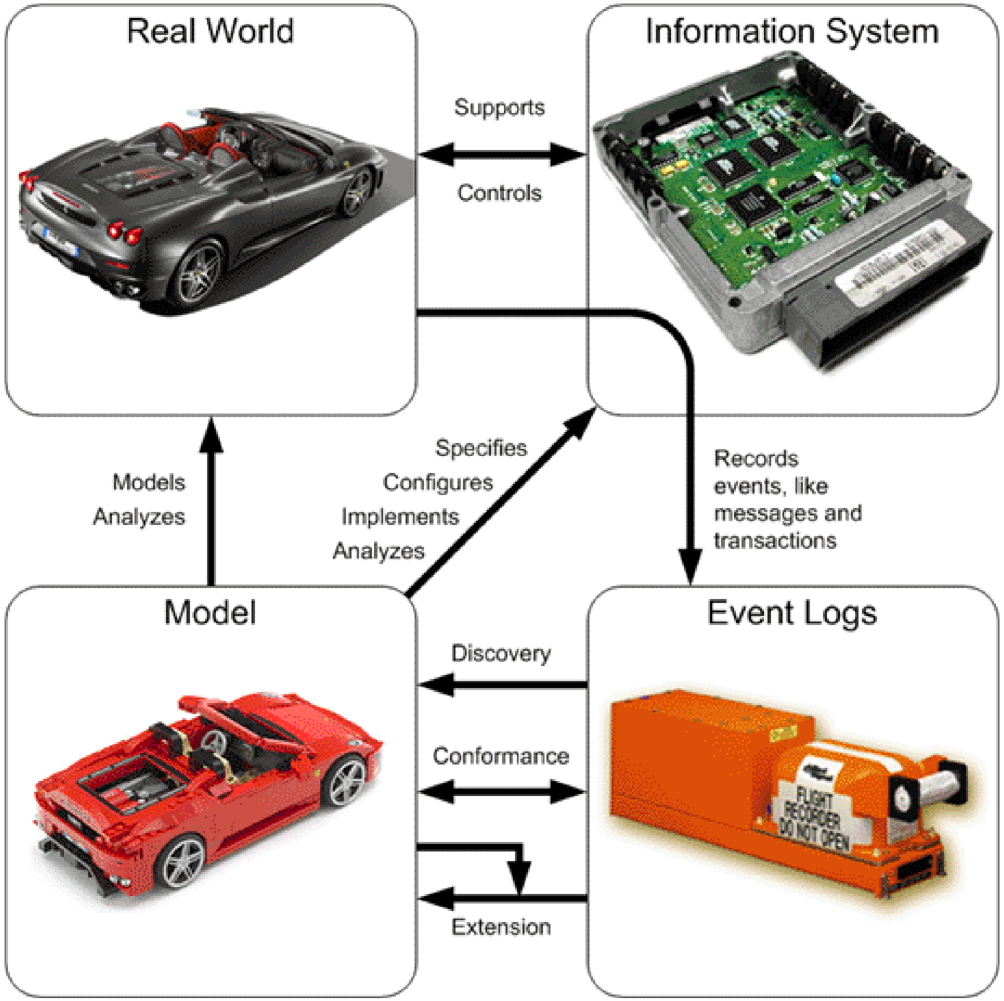
Enhancing Declare Maps Based on Event Correlations

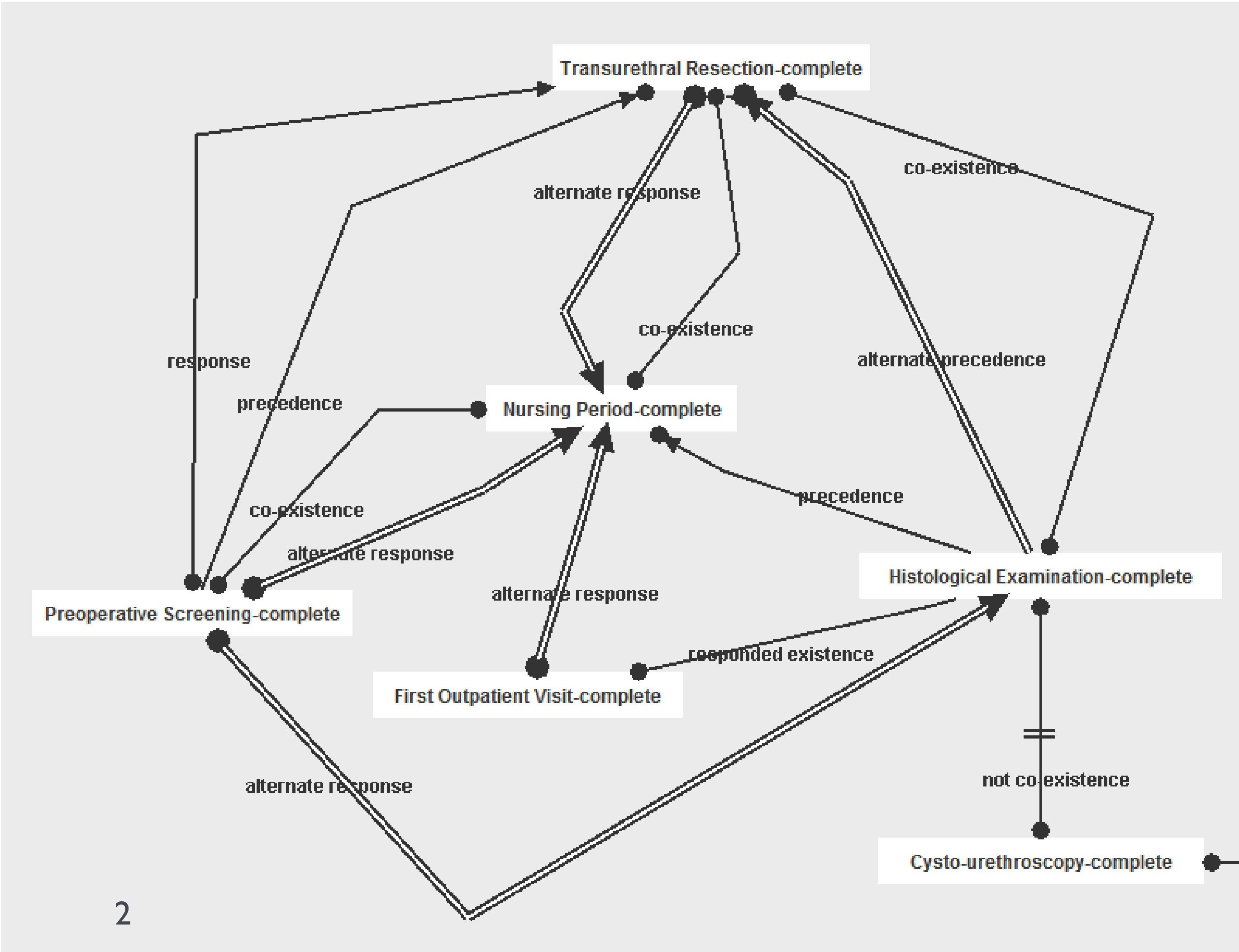
Jagadeesh Chandra Bose
Fabrizio Maria Maggi
Wil van der Aalst

Process mining

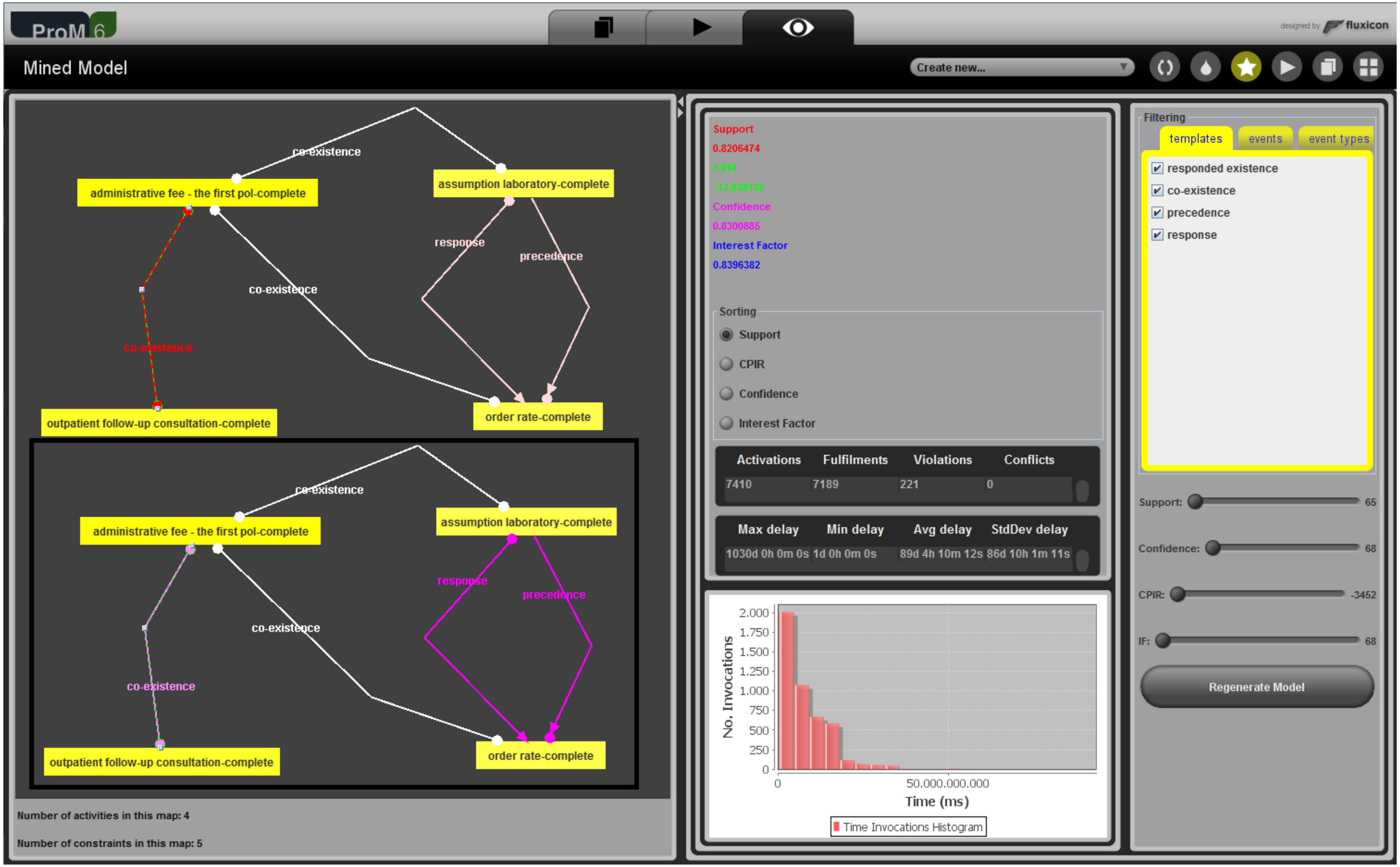


www.processmining.org

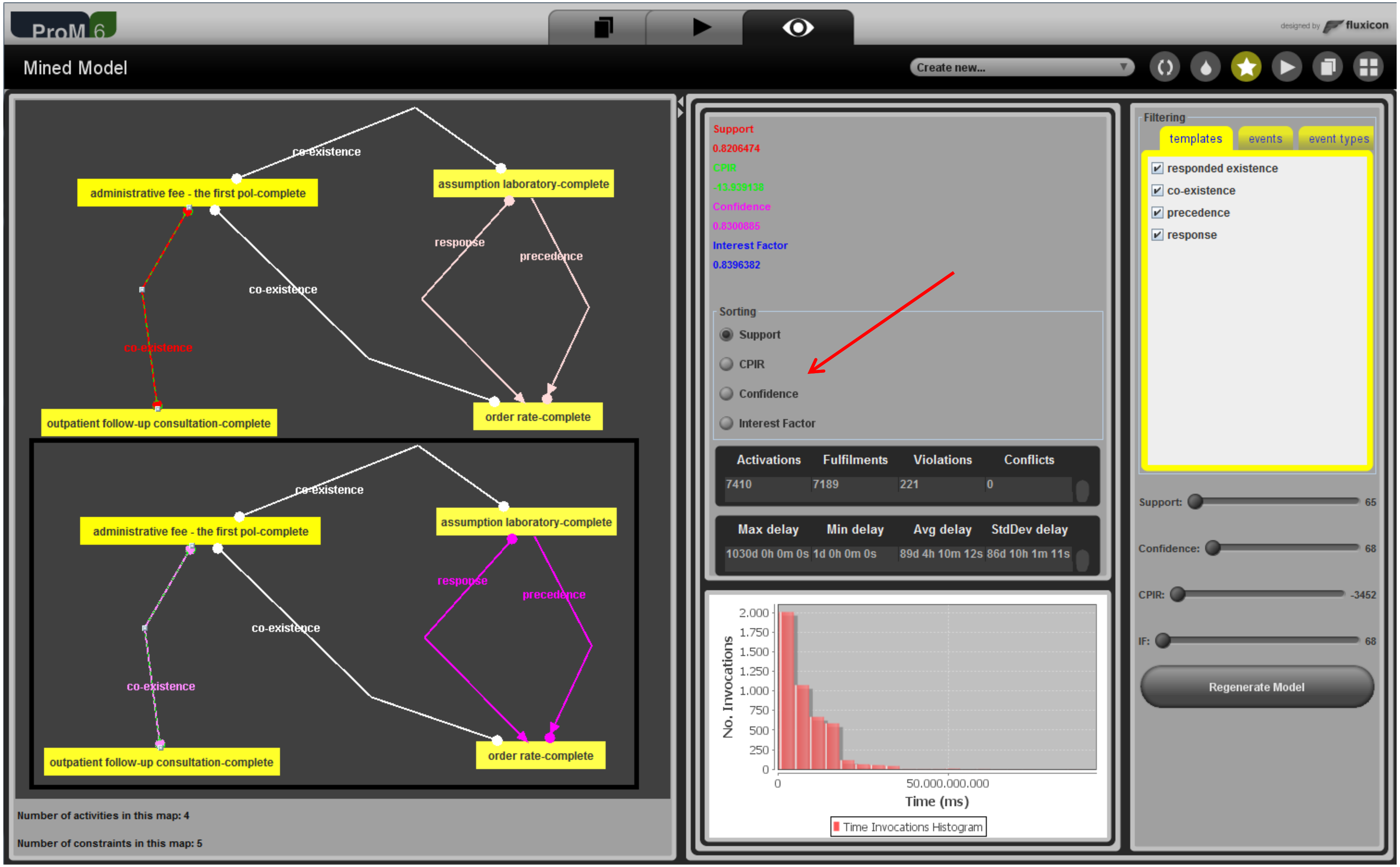
An example of Declare model



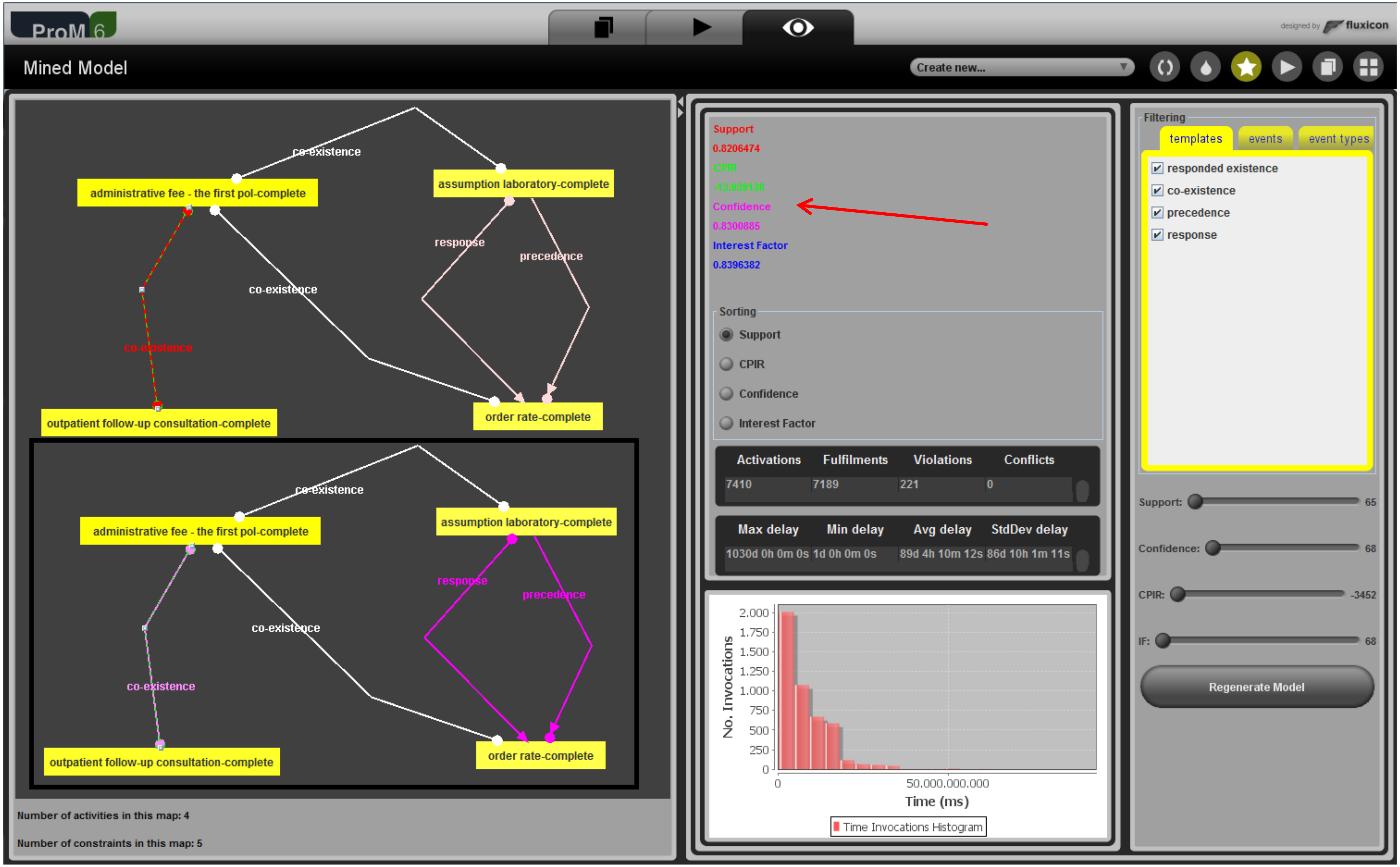
Declare Maps: a TomTom like visualization



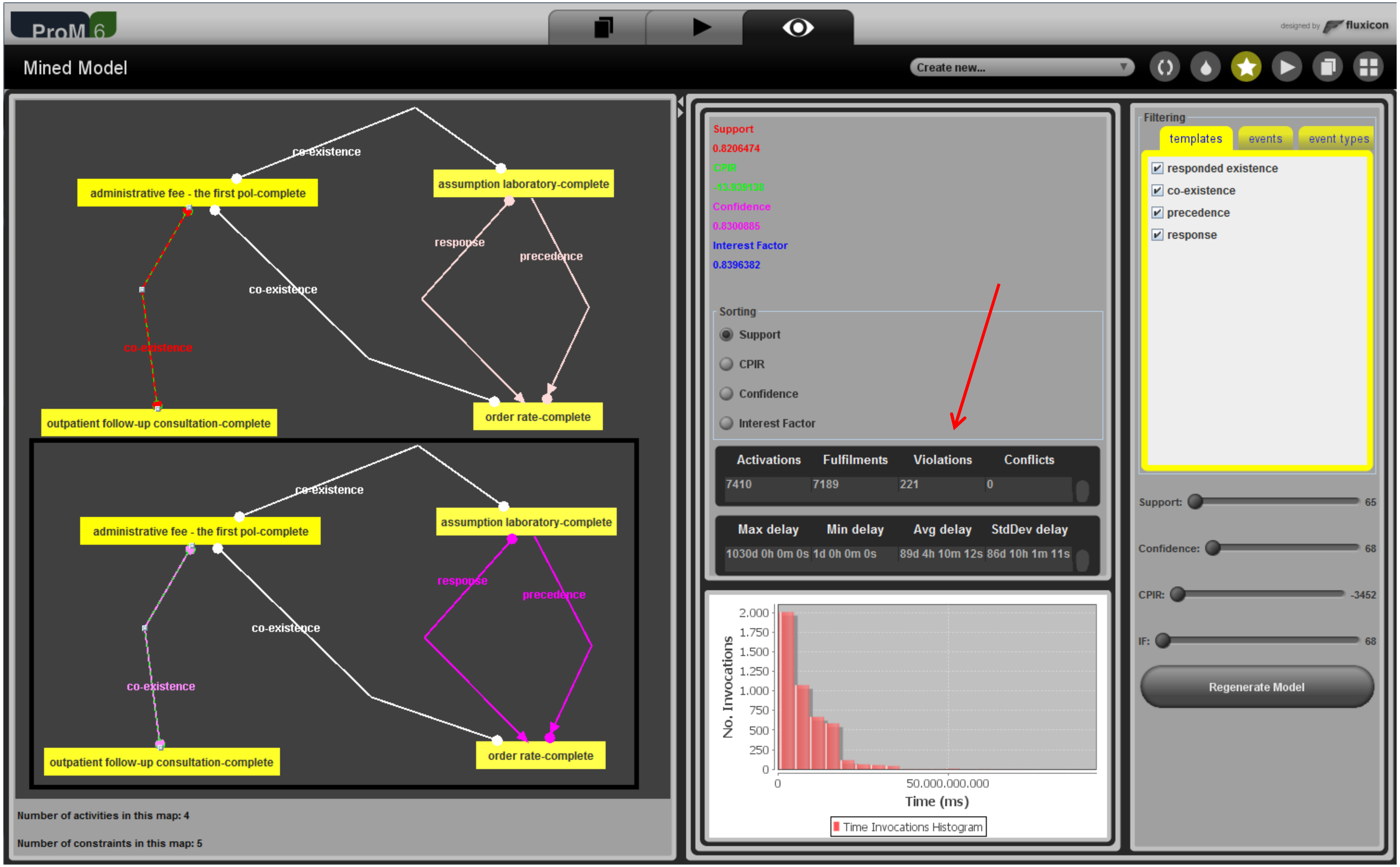
Declare Maps: a TomTom like visualization



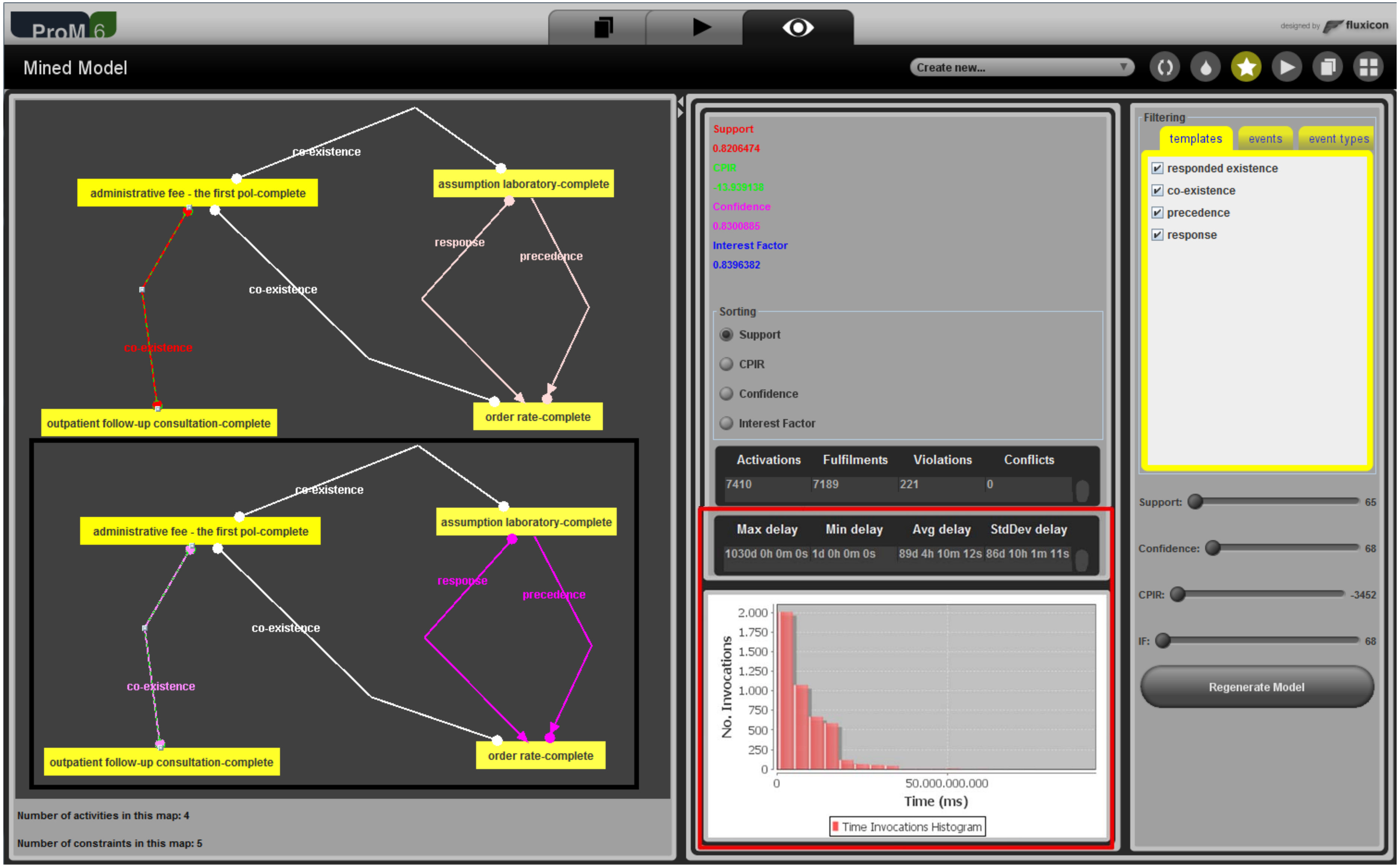
Declare Maps: a TomTom like visualization



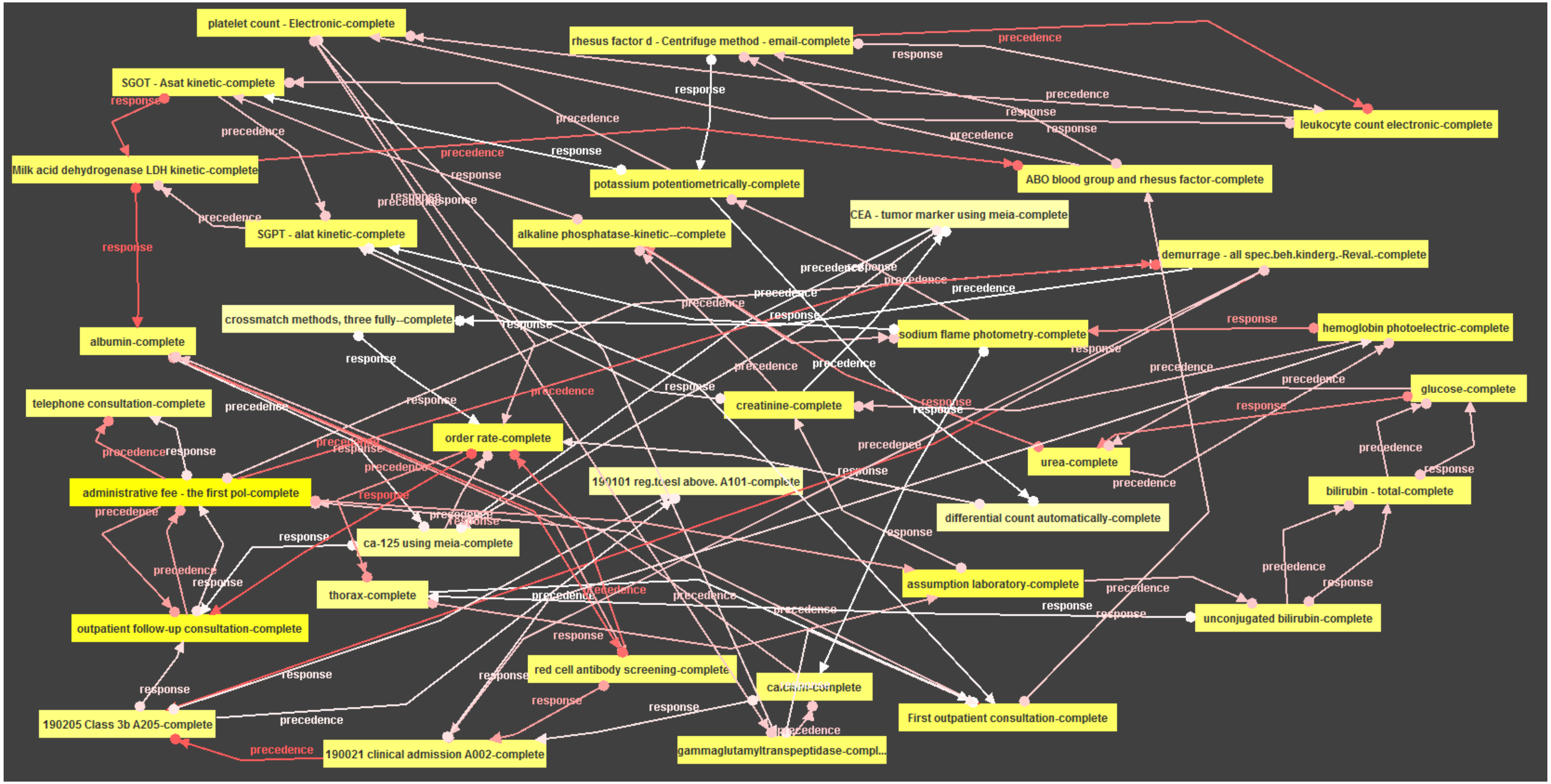
Declare Maps: a TomTom like visualization



Declare Maps: a TomTom like visualization



Spaghetti Declare



Addressing the problem of too many constraints

- Removing non-interesting constraints
  - Filter out non-interesting constraints based on activation support
  - Filter out redundant constraints "dominated" by other constraints
  - Filter out redundant constraints through transitive reduction
  - Filter out redundant constraints through reduction rules

- Guiding the discovery process with a-priori knowledge
  - Conceptual groupings: discovering inter- and intra-group constraints
  - Repair an existing Declare map with info retrieved from the log

F.M. Maggi, J.C. Bose, and W.M.P. van der Aalst
A Knowledge-Based Integrated Approach for Discovering and Repairing Declare Maps In CAiSE 2013

Addressing the problem of too many constraints (taking the data perspective into consideration)

- Considering interesting only those constraints between events that are highly correlated
  - constraints involving activities are interesting only if they share some common (data) elements of a process
    - **Example:** an *insurance claim process* could include a *claim application* and *filling in a questionnaire for feedback*. These two portions of the process sharing no or very few attributes (there are not significant correlations between them)
  - A constraint is considered satisfied only if some common attributes of the events involved in the constraint share the same value
    - **Example:** constraint *response(A,B)*;
      trace <A(x=1), B(x=1)> SATISFIED
      trace <A(x=1), B(x=2)> VIOLATED

Through correlations you can also…

- Disambiguate events
  - **Example:** trace <*A,B,C,B*>, *response(A,B)* (ambiguous) <*A(x=1),B(x=2),C,B(x=1)*> (can be disambiguated)
    - Disambiguation of events facilitates a correct association of events involved in a constraint and helps in *performance analysis* of a process
- Improve diagnostic capabilities
  - *Outliers*
    - **Example:** correlation *A.x* = *B.x* holds for 98% of the fulfillments of *response(A,B)*. The 2% of the activations where the correlation does not hold may potentially be outliers
  - *Discriminatory correlation patterns*
    - **Example:** if A and B are executed by the same department the response time between A and B is "Fast"
- Define conceptual groupings of activities
  - **Example:** activities involving all events that are executed in the same location as one conceptual group
    - Such conceptual groupings of activities can be used for guiding the discovery of Declare maps towards results that are more significant from an application domain point of view

F.M. Maggi, J.C. Bose, and W.M.P. van der Aalst
A Knowledge-Based Integrated Approach for Discovering and Repairing Declare Maps In CAiSE 2013

Categorization of correlations

Barros, G. Decker, M. Dumas, F. Weber ***Correlation Patterns in Service-Oriented Architectures*** In FASE 2007

- Property-based correlation
  - events are classified based on a function operating on their attributes
  - **Example:** all claim applications referring to an amount greater than 1000 euros are grouped together
- Reference-based correlation
  - two events are correlated if an attribute of the first event (identifier attribute) and an attribute of the second event (reference attribute) have the same value
- Moving time-window correlation
  - two events are correlated if they occur within a given duration of one another (e.g., one hour)

- We use an extended definition of reference-based correlation
  - events are correlated if there is a function connecting an attribute of the first event with an attribute of the second event ***(this function can include not only equality but also operators such as greater than, less than, and not equal to)***
  - **Example:** an event of producing a document is correlated to an event of checking it if the resource that produces the document is different from the resource that checks it

How to get correlations

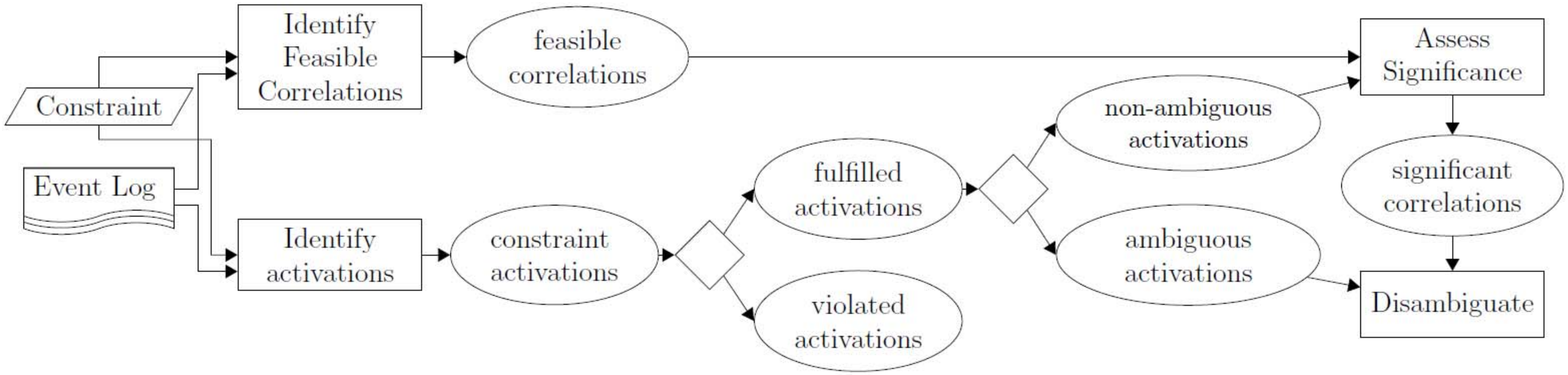
- Correlations can be
  - provided by a domain expert

  - automatically derived from event logs

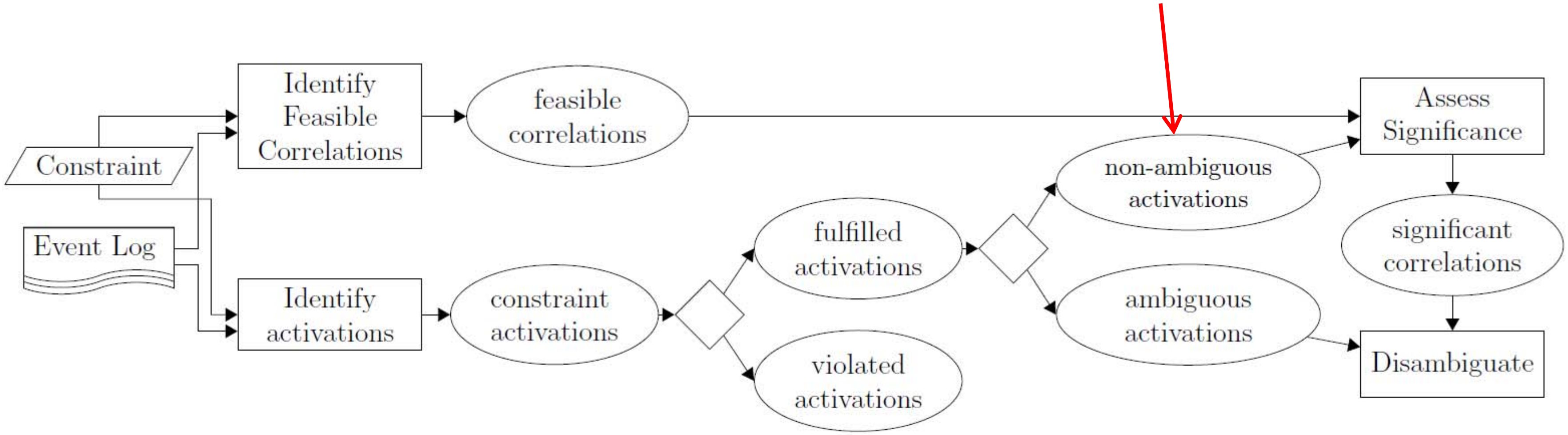
Discovering correlations from event logs (preamble)

- The XES standard for event logs allows for events having attributes
- Attributes can be comparable (feasible) or not
  - **Examples:** in an insurance claim process, attribute *amount claimed* is comparable to *amount issued*, but not to *location*.
  - *If a priori knowledge about the domain is available*, we can use that knowledge to identify/group attributes that are comparable.
  - *In the absence of prior domain knowledge*, we consider attributes having the same data type to be comparable.

Discovering correlations from event logs

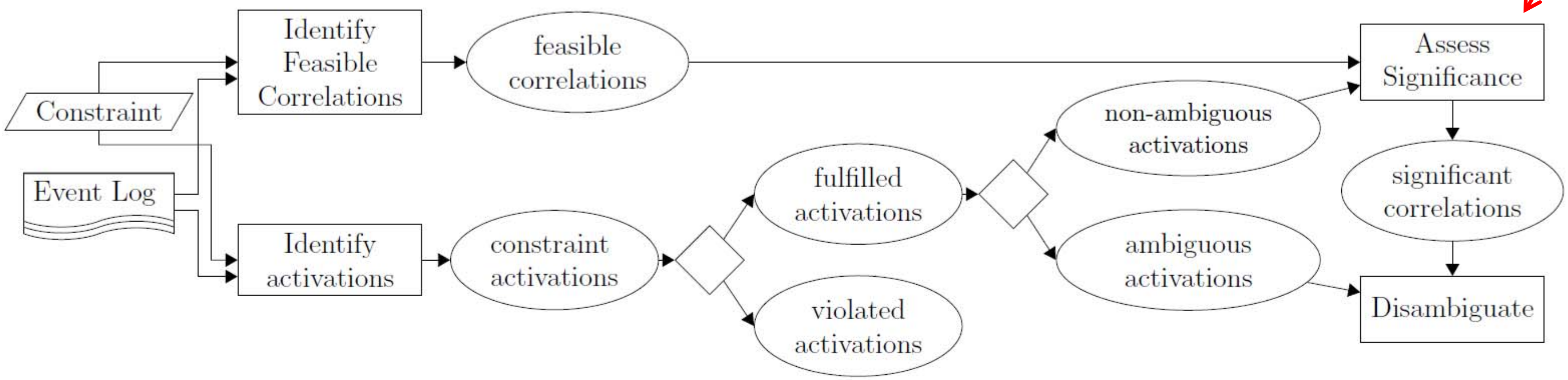


Discovering correlations from event logs



- Non-ambiguous activations
  - A fulfilled activation is non-ambiguous if there is only one possible target that can be associated to it
  - **Example:** for *response(A,B)*, the activations in traces <A,C,B> and <A,A,C,B> are <u>non-ambiguous</u> whereas the activations in traces <A,B,C,B> and <A,A,B,B> are <u>ambiguous</u>

Discovering correlations from event logs



- Correlation support
  - ratio between the number of non-ambiguous activations in which that correlation is true and the total number of non-ambiguous activations
  - We consider a feasible correlation as *significant* if its support is greater than a (user-specified) threshold

Discovering discriminant correlations

- *Step 1: Class Labeling*
  - **Example:** One can consider all the *fulfilled* activations of a constraint and classify them as *slow*, *medium* and *fast* based on their response time
- *Step 2: Feature Extraction and Selection*
  - The attributes of the events involved in a constraint are considered as primitive features for finding discriminatory patterns
- *Step 3: Discovering Discriminatory Patterns*
  - Discovery of the patterns over the features, which are strongly correlated to the class label (e.g., slow, medium and fast)
    - Standard data mining techniques, i.e., decision tree learning and association rule mining

Case study (BPI Challenge 2011)

- A = First outpatient consultation; B = administrative fee - the first pol; E = rhesus factor d - Centrifuge method; F = red cell antibody screening

constraint	support (constr.) (%)	#non-ambig. inst.	#ambig. inst.	**correlation**	support (correl.) (%)	degree of disambig-uation(%)
response (A,B)	57.39	517	559	A.org:group = B.org:group	94.00	57.96
				A.Producer code = B.Producer code	93.61	61.53
precedence (E,F)	54.85	603	932	E.time:timestamp = F.time:timestamp	100.00	96.45

Case study (BPI Challenge 2011)

constraint	correlation	#constraints			
		supp=30	supp=35	supp=40	supp=45
response(A,B)	⟨⟨ *no correlation* ⟩⟩	371	286	225	125
	A.org:group = B.org:group	229	180	163	114
	A.Producer code = B.Producer code	100	85	83	71
precedence(A,B)	⟨⟨ *no correlation* ⟩⟩	458	403	352	261
	A.org:group = B.org:group	274	249	240	237
	A.Producer code = B.Producer code	113	106	104	104

Conclusions and Future Work

- Motivation
  - Declarative process maps discovered from event logs without any consideration for event and case attributes tend to result in inaccurate and incomprehensible results.
- Contribution
  - We exploited the data present in event logs to discover process maps only showing relevant and accurate constraints.
  - We proposed a means of discovering significant correlations that exist between events and use these correlations to
    - *prune constraints*
    - *disambiguate event associations*
    - provide additional *diagnostic information*
  - We evaluated our approach using real-life logs
- Future work
  - extend the approach to also cover constraints involving multiple activities and negative relations
  - study the trade-off between completeness and efficiency of mining
  - evaluate our approach using more case studies