

# A Fresh Approach to Analyze Process Outcomes

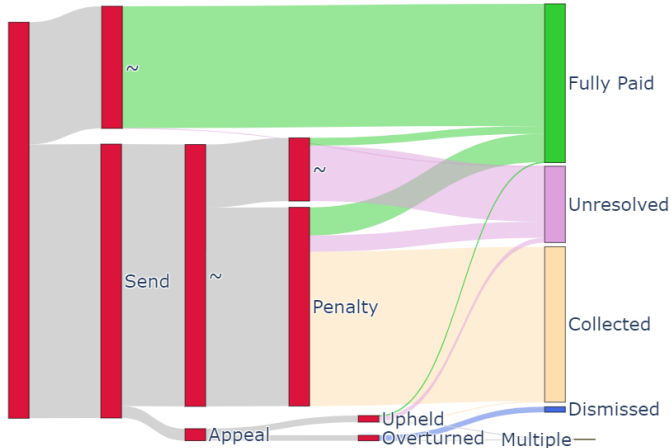
Hagen Völzer, Francesca Zerbato, Timothy Sulzer and Barbara Weber  
 University of St. Gallen, St. Gallen, Switzerland  
 Email: hagen.voelzer@unisg.ch

**Abstract**—We propose a set of techniques to analyze final or intermediate process outcomes. The main novelties are (i) outcome flow diagrams - a visualization of a process as a sequence

process outcomes as well as intermediate process milestones, exceptional cases such as normative deviations, etc.

We demonstrate the feasibility and effectiveness of the

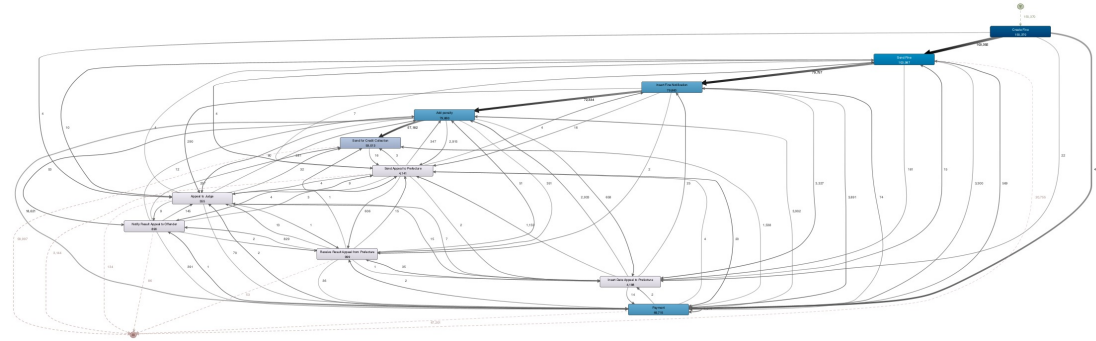
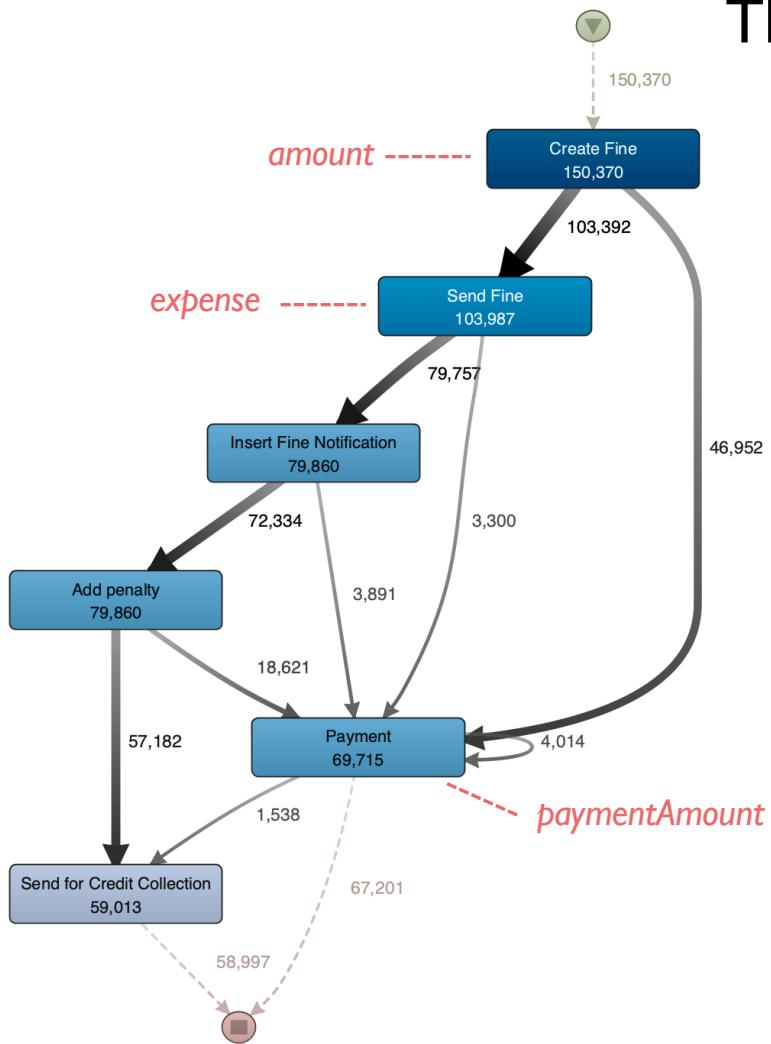
## (1) Outcome Flow Diagrams



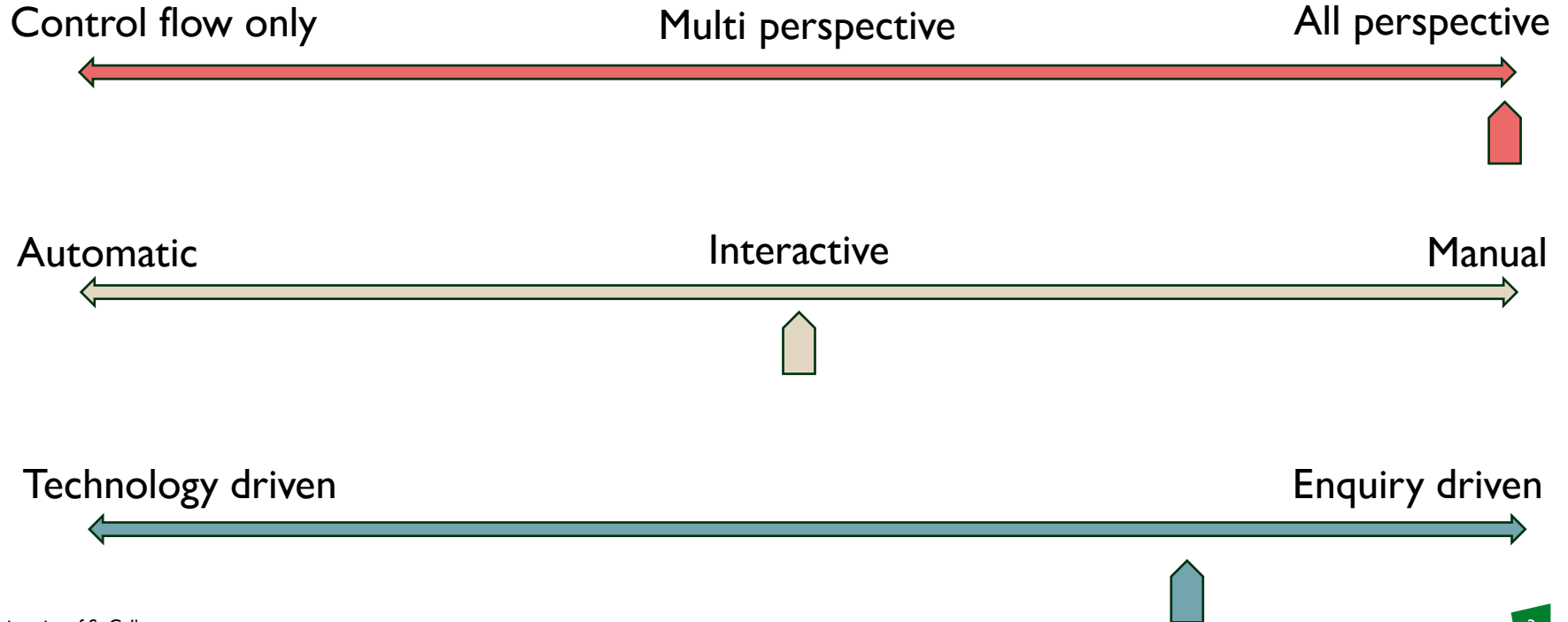
## (2) Method to Explain Outcomes a.k.a. Root Cause Analysis, Deviance Explanation

$r_{\#}$	Rule	Precision	Recalled Cases
$r_1$	<b>if</b> $\neg$ InsertFine <b>then</b> $\neg$ Collect	1.0	21089
$r_2$	<b>if</b> $\text{delay}(\text{Penalty}, \text{Payment}) \leq 3 \wedge \text{FINAL.outstandingBalanceWithoutPenalty} \leq 0.01$ <b>then</b> $\neg$ Collect	0.999	5749
$r_{2.1}$	<b>if</b> (if-part of $r_2$ ) $\wedge$ Payment.count = 1 <b>then</b> $\neg$ Collect	0.998	2214
$r_{2.2}$	<b>if</b> (if-part of $r_2$ ) $\wedge$ Payment.count $\geq 2$ <b>then</b> $\neg$ Collect	1.0	3535
$r_{3.1}$	<b>if</b> caseStart $\geq 4481$ <b>then</b> $\neg$ Collect	1.0	1694
$r_{3.2}$	<b>if</b> caseStart $\geq 4401 \wedge$ Payment <b>then</b> $\neg$ Collect	0.996	228
$r_4$	<b>if</b> InsertAppeal $\wedge$ $\neg$ Notify <b>then</b> $\neg$ Collect	1.0	1284
$r_5$	<b>if</b> FINAL.outstandingBalance $\leq 10$ <b>then</b> $\neg$ Collect	1.0	1064
$r_6$	<b>if</b> appeal $\wedge$ FINAL.outstandingBalanceWithoutPenalty $\leq 0.01$ <b>then</b> $\neg$ Collect	0.994	482
$r_7$	<b>if</b> Create.dismissal $\in \{2, 3, 5, A, B, E, F, I, J, K, M, N, Q, R, T, U, V\}$ <b>then</b> $\neg$ InsertFine	1.0	444
$r_8$	<b>if</b> Create $\wedge$ Send $\wedge$ eventCount = 2 <b>then</b> $\neg$ InsertFine	1.0	20385

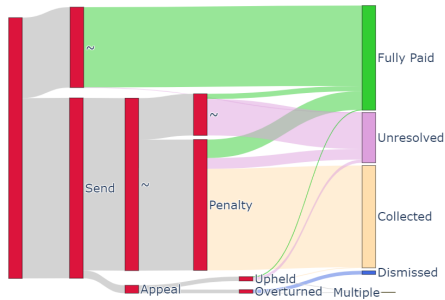
# The Road Traffic Fines Management Process



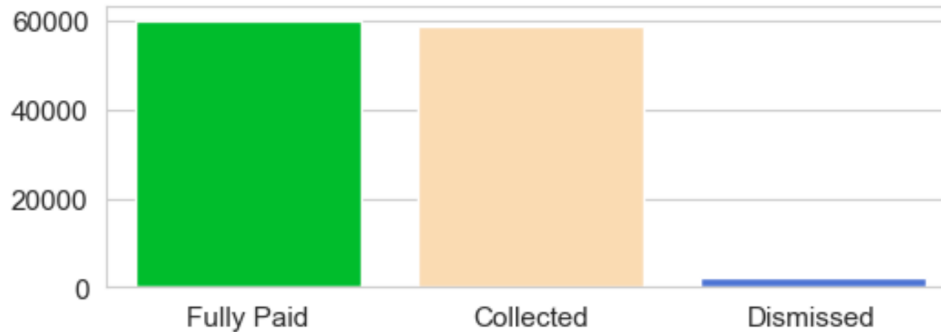
# Model Building Characteristics (for what follows)



# Part 1: Outcome Flow Diagrams



# Normative Outcomes - Specification



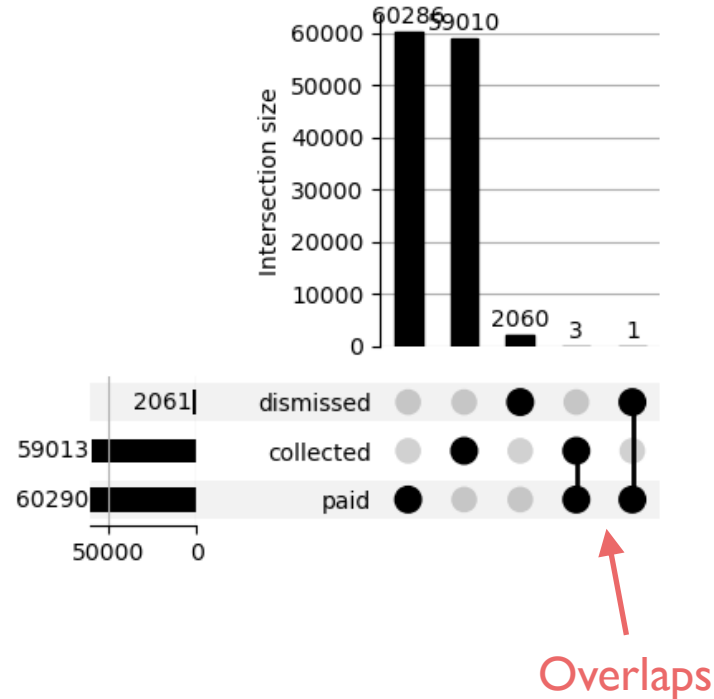
*Multi-perspective case predicate*

- **FullyPaid** =  $\text{Final.outstandingBalance} \leq 0$ , where
  - $\text{outstandingBalance} = \text{amount}::\text{last} + \text{expense}::\text{sum} - \text{paymentAmount}::\text{sum}$
- **Collected** =  $\text{Collect.count} > 0$
- **Dismissed** =  $\text{Final.dismissal}::\text{last} \in \{\#,G\}$

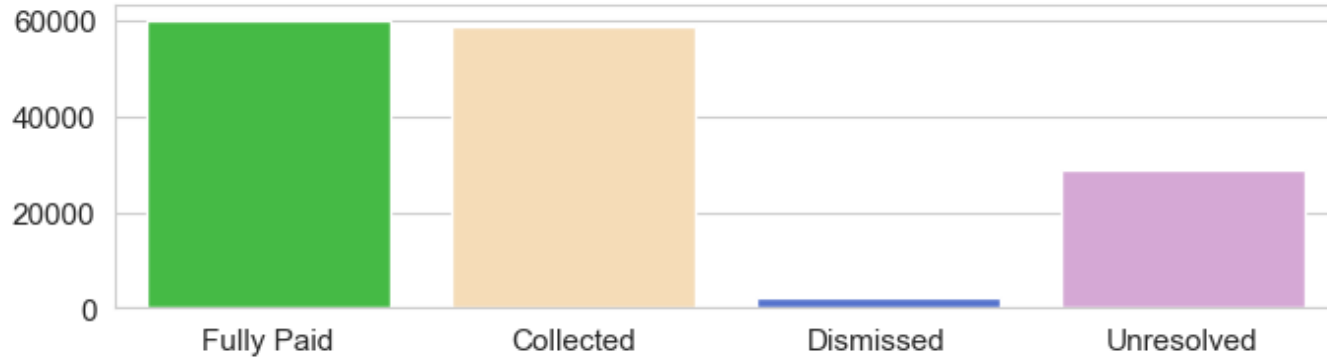
*Data perspective*

*A little language for case predicates  
cf. paper.*

# Normative Outcomes – Conformance Check (1/2): Disjointness



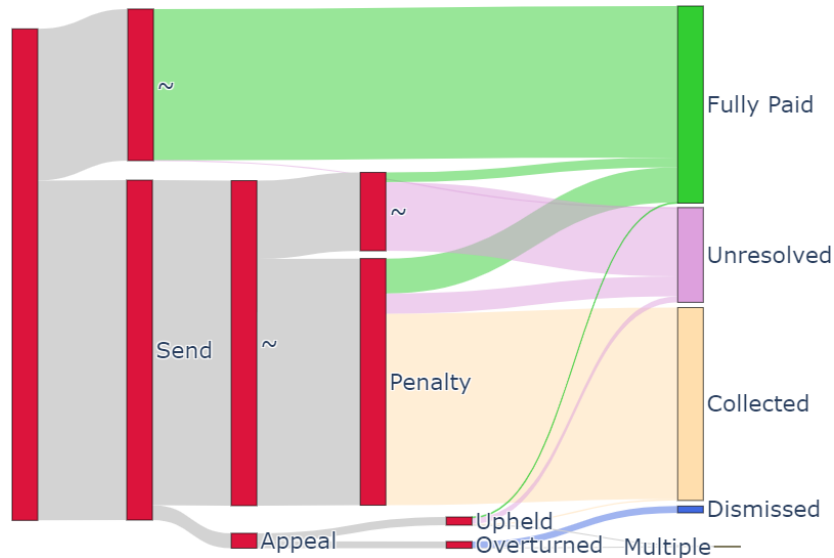
# Normative Outcomes – Conformance Check (2/2): Completeness



Uncovered cases

# Introducing ... Outcome Flow Diagrams

How does the *process* influence the outcome?



4 business-level decisions

Design principles

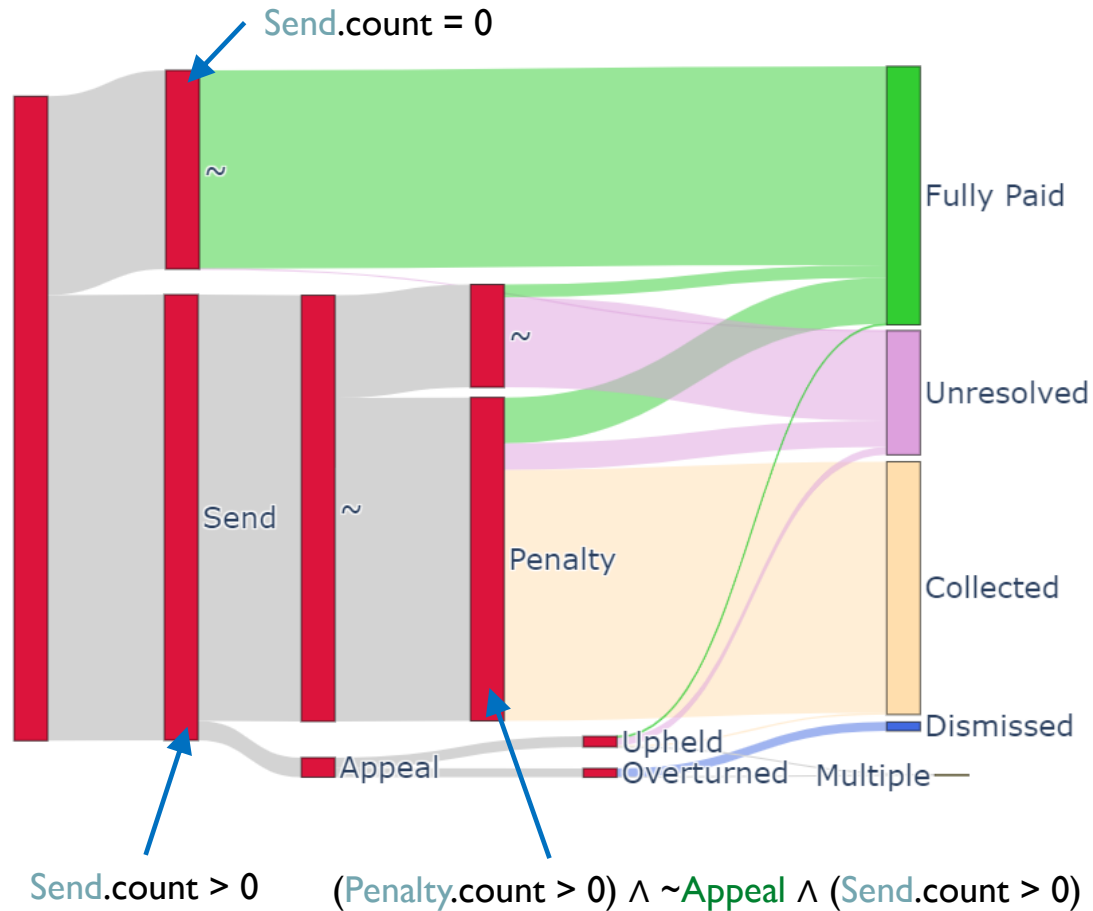
- Control complexity
- Do not leak or add cases

Idea

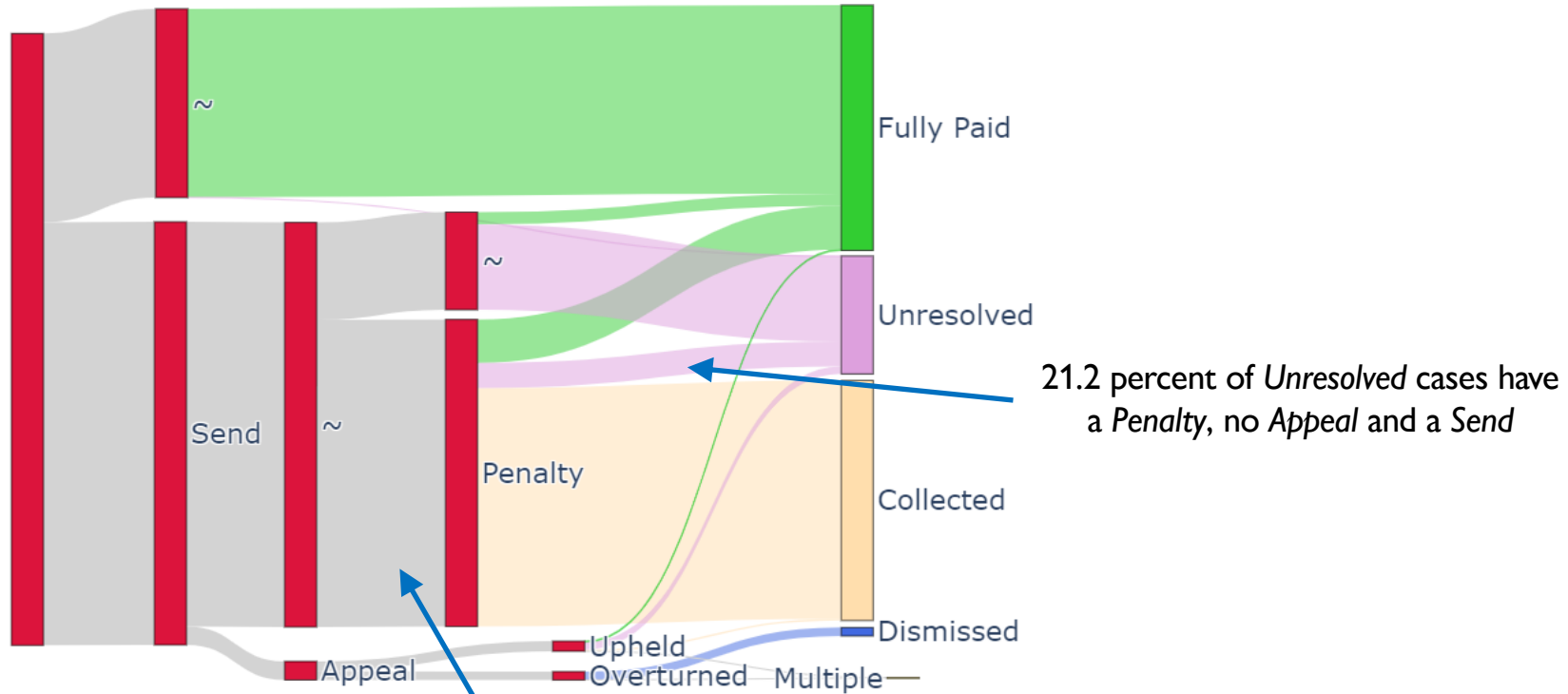
- Process = sequence of business level decisions



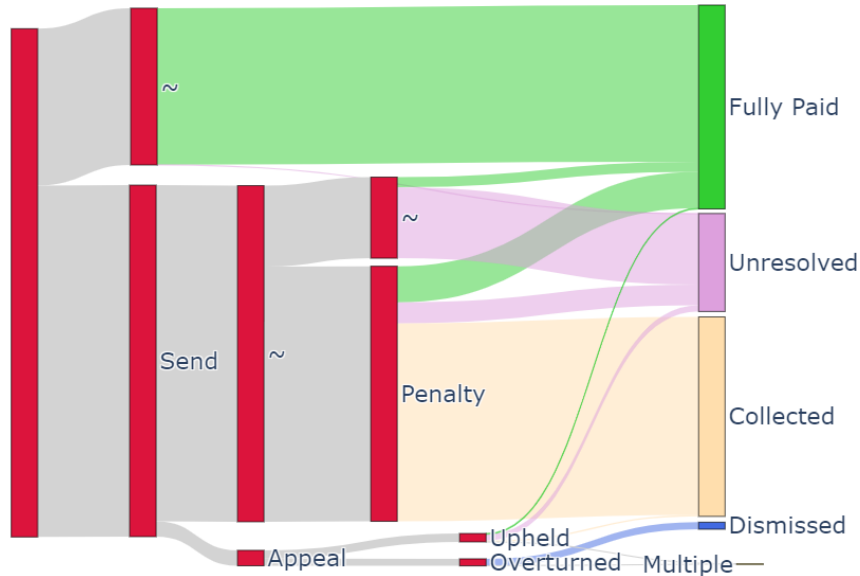
# Outcome Flow Diagram – Some Details



# Outcome Flow Diagram – Benefits



# Outcome Flow Diagram – Summary



- Quantify contribution of business level decisions to outcomes
  - Also possible as control-flow only, e.g. *knock-out processes*, e.g., BPIC 2012
- Control *complexity vs. information* by
  - Using only *selected* case predicates (relevant business level decisions)
  - **Order decisions by control-flow** whenever possible
  - **Omit decisions / orderings** deliberately
  - Details in the paper

# Part 2:

# Explanations

<i>r</i> #	Rule	Precision	Recalled Cases
<i>r</i> <sub>1</sub>	if ¬InsertFine then ¬Collect	1.0	21089
<i>r</i> <sub>2</sub>	if delay(Penalty, Payment) ≤ 3 ∧ FINAL.outstandingBalanceWithoutPenalty ≤ 0.01 then ¬Collect	0.999	5749
<i>r</i> <sub>2.1</sub>	if (if-part of <i>r</i> <sub>2</sub> ) ∧ Payment.count = 1 then ¬Collect	0.998	2214
<i>r</i> <sub>2.2</sub>	if (if-part of <i>r</i> <sub>2</sub> ) ∧ Payment.count ≥ 2 then ¬Collect	1.0	3535
<i>r</i> <sub>3.1</sub>	if caseStart ≥ 4481 then ¬Collect	1.0	1694
<i>r</i> <sub>3.2</sub>	if caseStart ≥ 4401 ∧ Payment then ¬Collect	0.996	228
<i>r</i> <sub>4</sub>	if InsertAppeal ∧ ¬Notify then ¬Collect	1.0	1284
<i>r</i> <sub>5</sub>	if FINAL.outstandingBalance ≤ 10 then ¬Collect	1.0	1064
<i>r</i> <sub>6</sub>	if appeal ∧ FINAL.outstandingBalanceWithoutPenalty ≤ 0.01 then ¬Collect	0.994	482
<i>r</i> <sub>7</sub>	if Create.dismissal ∈ {2, 3, 5, A, B, E, F, I, J, K, M, N, Q, R, T, U, V} then ¬InsertFine	1.0	444
<i>r</i> <sub>8</sub>	if Create ∧ Send ∧ eventCount = 2 then ¬InsertFine	1.0	20385

# Explanation

**Why are unresolved cases not credit collected?**

Example hypothesis:

**if** delay(Create, Send) > 90 **then**  $\neg$ Collect

Precision = 0.37



# Explanation

**Why are unresolved cases not credit collected?**

Example hypothesis:

**if caseStart  $\geq$  4481 then  $\neg$ Collect**

Meaningful to analyst  
(viz. e.g. incomplete cases)

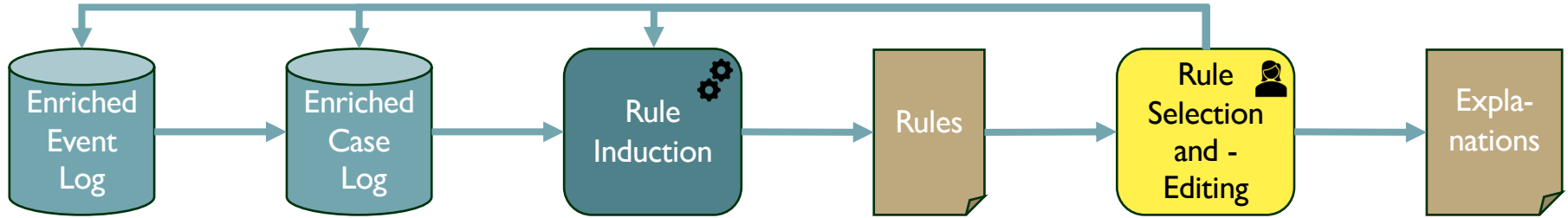
Substantial recall

Precision = 1.0  
1694 recalled cases

High precision



# Outcome Explanation Method Overview



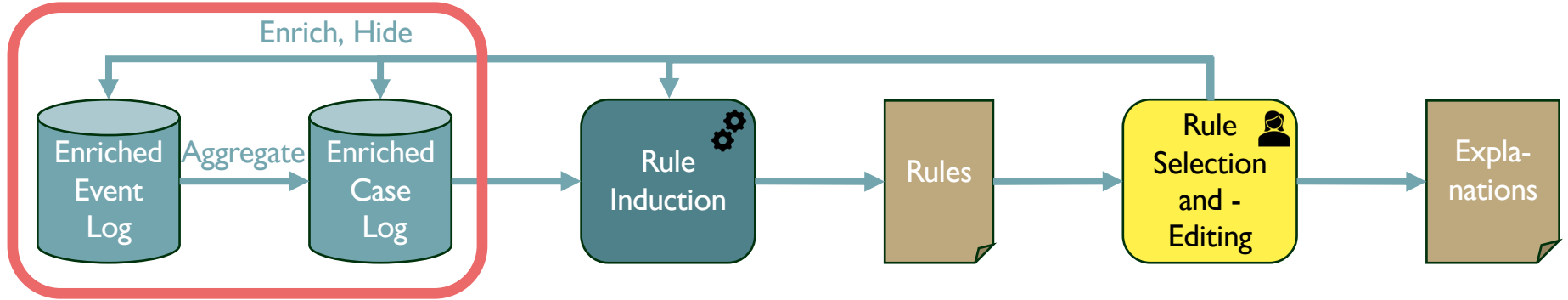
Generating *rich* set of *multiperspective* case attributes

E.g. RIPPER, not decision trees

Driven by how *meaningful* a rule is to the analyst

Address *multi-collinearity*

# Enriched (all-perspective) Case Log



case_id	event_count	caseStart	duration	Send.count	Payment.count	Payment:2.start	delay(Send, Payment)
C10852	16	89	518	1	10	189	165

Payment:2.paymentAmount	Payment:2.amount::last	Payment:2.paymentAmount::sum	Payment:2.outstanding_balance
62.59	625.95	125.18	500.77

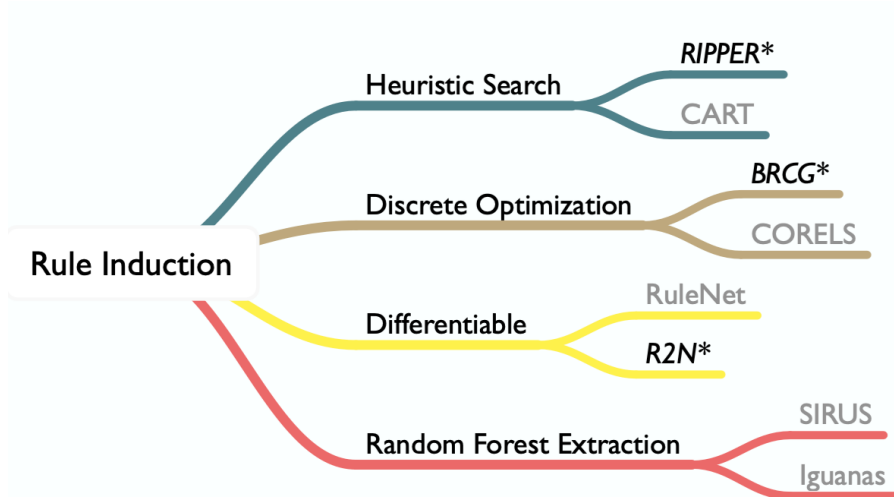
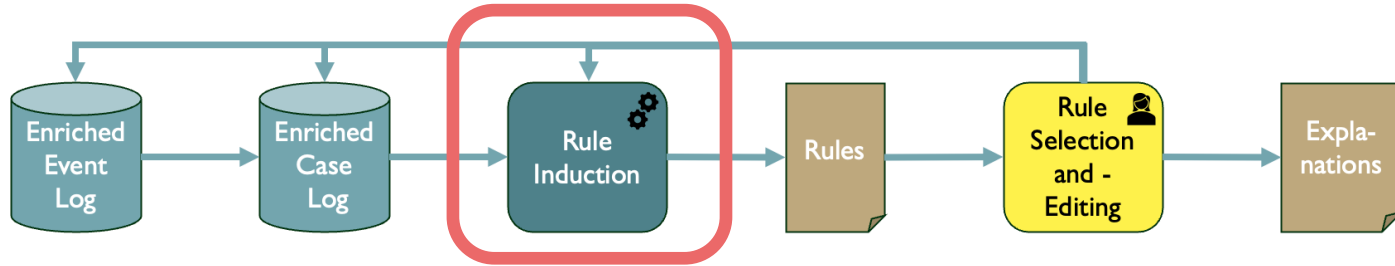
Final.paymentAmount::sum	Final.outstanding_balance	Final.dismissal::last
625.9	0.05	NIL

...

383 columns, i.e., case attributes for *Road Fines*



# Rule Induction

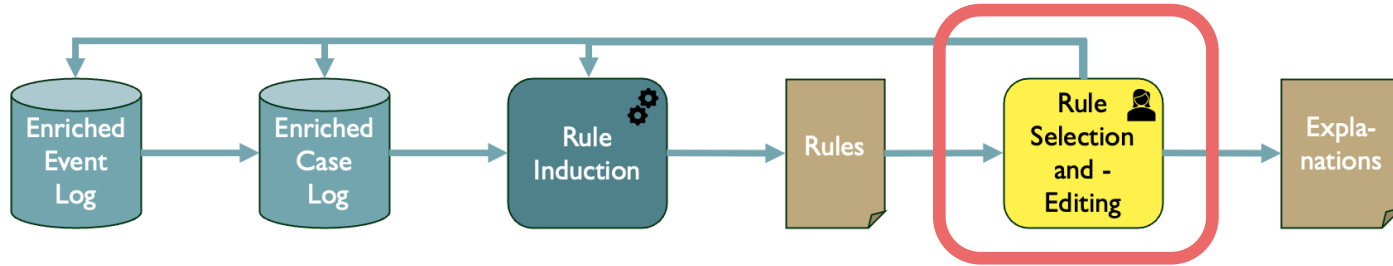


\* included in rule mining package *rulelearn*

- which we released in collaboration with IBM Research
- Also contains
  - GLRM (rule mining for regression)
  - Rule set interchange, export, and some tools

<https://github.com/IBM/rulelearn>

# Rule Selection and -Editing (1/3)



*Why are unresolved cases not credit collected?*

Meaningful rule:

**if** Final.outstandingBalance  $\leq$  10 **then**  $\neg$ Collect

Precision = 1.0  
1064 recalled cases

# Rule Selection and -Editing (2/3)

*Why are unresolved cases not credit collected?*

Partially meaningful rule:

**if** caseStart  $\geq$  4401  $\wedge$  delay(Send, Payment)  $\geq$  54 **then**  $\neg$ Collect



Simplification, Threshold Optimization

**if** caseStart  $\geq$  4481 **then**  $\neg$ Collect

Precision = 1.0  
1694 recalled cases

# Rule Selection and -Editing (3/3)

Why are unresolved cases not credit collected?

Multi-collinearity



Less meaningful rule:

**if** delay(Penalty, Payment)  $\leq 3 \wedge$  Payment.count  $\geq 2 \wedge$  Final.outstandingBalance  $\leq 37.55 \wedge$   
Final.paymentAmount::sum  $\geq 25.49$  **then**  $\neg$ Collect

Precision 1.0, 3080 recalled cases

Inspecting true positives:

time	concept:name	amount	paymentAmount	expense	outstandingBalance
02.01.00	Create Fine	31.3			31.3
15.02.00	Send Fine			11.41	42.71
27.03.00	Insert Fine Notification				42.71
15.04.00	Payment		37.75		4.96
26.05.00	Add penalty	62.59			36.25
28.08.00	Payment		4.96		31.29

# Case Inspection

time	concept:name	amount	paymentAmount	expense	outstandingBalance
02.01.00	Create Fine	31.3			31.3
15.02.00	Send Fine			11.41	42.71
27.03.00	Insert Fine Notification				42.71
15.04.00	Payment		37.75		4.96
26.05.00	Add penalty	62.59			36.25
28.08.00	Payment		4.96		31.29

case S49055



Formulate new hypothesis and test it



Enrich log and re-run rule induction

time	concept:name	amount	paymentAmount	expense	outstandingBalance	outstanding_balance_without_penalty
02.01.00	Create Fine	31.3			31.3	31.3
15.02.00	Send Fine			11.41	42.71	42.71
27.03.00	Insert Fine Notification				42.71	42.71
15.04.00	Payment		37.75		4.96	4.96
26.05.00	Add penalty	62.59			36.25	4.96
28.08.00	Payment		4.96		31.29	0

# Results for Road Fines

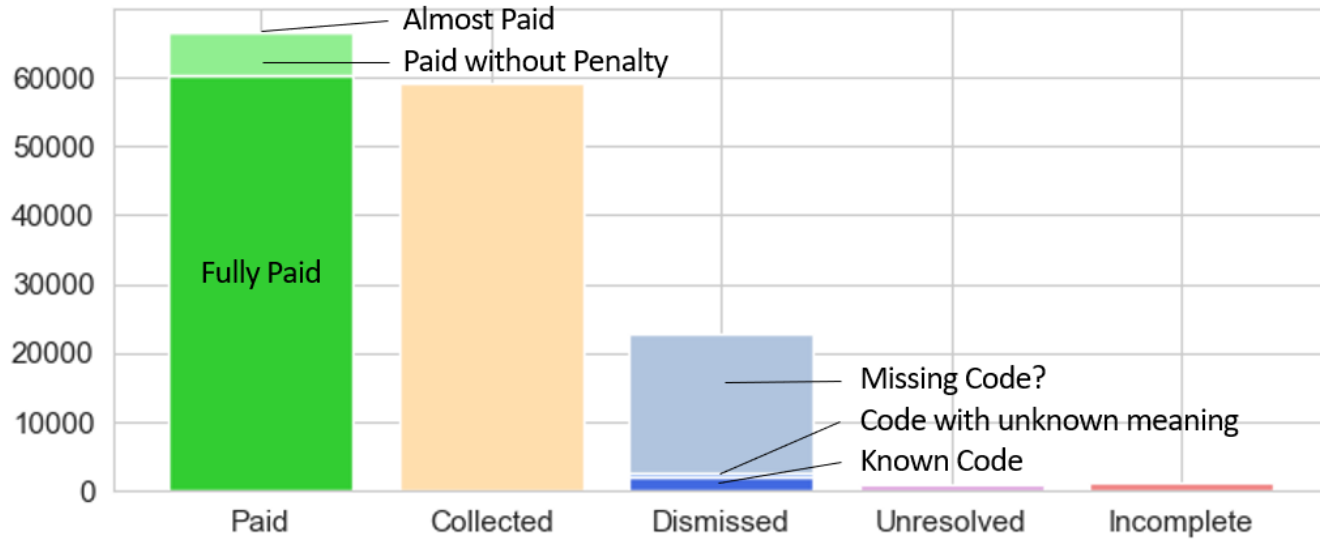
<i>r</i> #	Rule	Precision	Recalled Cases
<i>r</i> <sub>1</sub>	<b>if</b> $\neg$ InsertFine <b>then</b> $\neg$ Collect	1.0	21089
<i>r</i> <sub>2</sub>	<b>if</b> delay(Penalty, Payment) $\leq$ 3 $\wedge$ FINAL.outstandingBalanceWithoutPenalty $\leq$ 0.01 <b>then</b> $\neg$ Collect	0.999	5749
<i>r</i> <sub>2.1</sub>	<b>if</b> (if-part of <i>r</i> <sub>2</sub> ) $\wedge$ Payment.count = 1 <b>then</b> $\neg$ Collect	0.998	2214
<i>r</i> <sub>2.2</sub>	<b>if</b> (if-part of <i>r</i> <sub>2</sub> ) $\wedge$ Payment.count $\geq$ 2 <b>then</b> $\neg$ Collect	1.0	3535
<i>r</i> <sub>3.1</sub>	<b>if</b> caseStart $\geq$ 4481 <b>then</b> $\neg$ Collect	1.0	1694
<i>r</i> <sub>3.2</sub>	<b>if</b> caseStart $\geq$ 4401 $\wedge$ Payment <b>then</b> $\neg$ Collect	0.996	228
<i>r</i> <sub>4</sub>	<b>if</b> InsertAppeal $\wedge$ $\neg$ Notify <b>then</b> $\neg$ Collect	1.0	1284
<i>r</i> <sub>5</sub>	<b>if</b> FINAL.outstandingBalance $\leq$ 10 <b>then</b> $\neg$ Collect	1.0	1064
<i>r</i> <sub>6</sub>	<b>if</b> appeal $\wedge$ FINAL.outstandingBalanceWithoutPenalty $\leq$ 0.01 <b>then</b> $\neg$ Collect	0.994	482
<i>r</i> <sub>7</sub>	<b>if</b> Create.dismissal $\in$ {2, 3, 5, A, B, E, F, I, J, K, M, N, Q, R, T, U, V} <b>then</b> $\neg$ InsertFine	1.0	444
<i>r</i> <sub>8</sub>	<b>if</b> Create $\wedge$ Send $\wedge$ eventCount = 2 <b>then</b> $\neg$ InsertFine	1.0	20385

Recursion (points to Rule column)  
 Actual deadline for penalty (points to delay(Penalty, Payment)  $\leq$  3 in *r*<sub>2</sub>)

Incomplete cases (points to *r*<sub>3.1</sub>)  
 Lenience (points to *r*<sub>5</sub>)  
 No penalty when appealed (points to *r*<sub>6</sub>)

"Special" dismissal codes (points to {2, 3, 5, A, B, E, F, I, J, K, M, N, Q, R, T, U, V} in *r*<sub>7</sub>)  
 Missing dismissal code? (points to eventCount = 2 in *r*<sub>8</sub>)

# Revised Outcome Distribution



Not so much non-conformance anymore

# Meta Insights

Plain event ordering is sometimes the wrong abstraction

Events are not always what they seem

$r\#$	Rule	Precision	Recalled Cases
$r_1$	<b>if</b> $\neg$ InsertFine <b>then</b> $\neg$ Collect	1.0	21089
$r_2$	<b>if</b> delay(Penalty, Payment) $\leq 3 \wedge$ FINAL.outstandingBalanceWithoutPenalty $\leq 0.01$ <b>then</b> $\neg$ Collect	0.999	5749
$r_{2.1}$	<b>if</b> (if-part of $r_2$ ) $\wedge$ Payment.count = 1 <b>then</b> $\neg$ Collect	0.998	2214
$r_{2.2}$	<b>if</b> (if-part of $r_2$ ) $\wedge$ Payment.count $\geq 2$ <b>then</b> $\neg$ Collect	1.0	3535
$r_{3.1}$	<b>if</b> caseStart $\geq 4481$ <b>then</b> $\neg$ Collect	1.0	1694
$r_{3.2}$	<b>if</b> caseStart $\geq 4401 \wedge$ Payment <b>then</b> $\neg$ Collect	0.996	228
$r_4$	<b>if</b> InsertAppeal $\wedge \neg$ Notify <b>then</b> $\neg$ Collect	1.0	1284
$r_5$	<b>if</b> FINAL.outstandingBalance $\leq 10$ <b>then</b> $\neg$ Collect	1.0	1064
$r_6$	<b>if</b> appeal $\wedge$ FINAL.outstandingBalanceWithoutPenalty $\leq 0.01$ <b>then</b> $\neg$ Collect	0.994	482
$r_7$	<b>if</b> Create.dismissal $\in \{2, 3, 5, A, B, E, F, I, J, K, M, N, Q, R, T, U, V\}$ <b>then</b> $\neg$ InsertFine	1.0	444
$r_8$	<b>if</b> Create $\wedge$ Send $\wedge$ eventCount = 2 <b>then</b> $\neg$ InsertFine	1.0	20385

Incompleteness can be detected (approx.)

Conformance thresholds can be shifted or fuzzy

Beware of missing data

Also: Some rules are applied manually



# Conclusion

- New results on *Road Fines* log shows first evidence on the benefits of the methods
  - Run time is sufficient for an interactive setting (< 1 min)
  - More case studies are needed
- Future: Tool building for ease of use and supporting user studies

**Thanks to all log providers!**