How Much Event Data Is Enough?

A Statistical Framework for Process Model Discovery

Martin Bauer, Arik Senderovich, Avigdor Gal,
Lars Grunske, Matthias Weidlich

Processes and Events

Automated control
of business processes

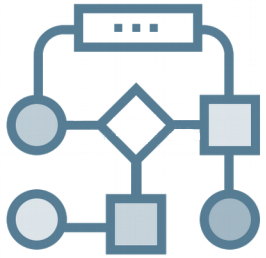
Recording of process
execution information

Event logs:
- Timestamps
- Case IDs
- Activity IDs
- …

The Question of Process Discovery

Event log

Discovery



Process model

Efficiency of process discovery becomes increasingly important

Pervasiveness of data sensing/logging

- => Large-scale event logs
- Tuning a large range of parameters of discovery algorithms
- => Repeated, exploratory analysis
- Software-as-a-Service solutions for process discovery
- => Online handling of event logs

A View on Related Work

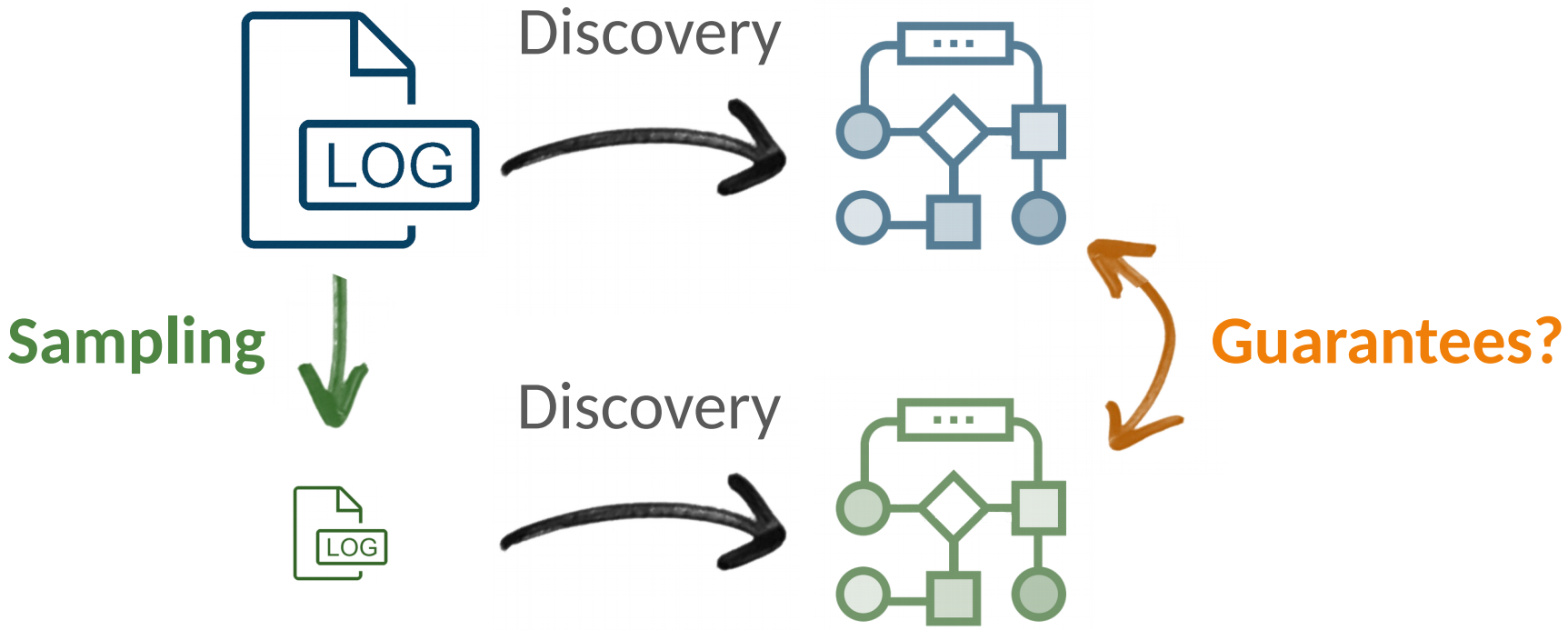
Plethora of discovery algorithms [Augusto et al. 2017]

Striving for scalability

- By divide-and-conquer: Decompose the discovery problem [van der Aalst & Verbeek 2015]
- By parallelization and distribution [Wang et al. 2015, Evermann 2016]

Recently: Idea of sampling event data [Busany & Maoz 2016]

Daring the Gap



How to determine how much of an event log to use to discover a process model?

Agenda

Background and Related Work on Process Discovery

A Statistical Framework for Process Discovery

- Log sampling
- Framework definition

Instantiating the Framework

- For control-flow discovery
- For performance discovery

Experimental Results

Log Sampling

- We want to stop when new information is unlikely
  - Log L, sampled Log $L' \subseteq L$
  - Trace abstraction function Ψ(ξ)
  - boolean predicates γ(L' ,ξ)
    - γ(L' ,ξ) ⇔ $\Psi(\xi) \notin \bigcup \Psi(\xi')$
    - γ(L' ,ξ) ⇔ d($\Psi(\xi), \bigcup \Psi(\xi')$) > ε

- $L' \subseteq L$ is (δ, ε, Ψ)-discovery sufficient if for newly sampled trace ξ:
  - P(γ(L',ξ) = 1) < δ

This is our Stopping criteria!

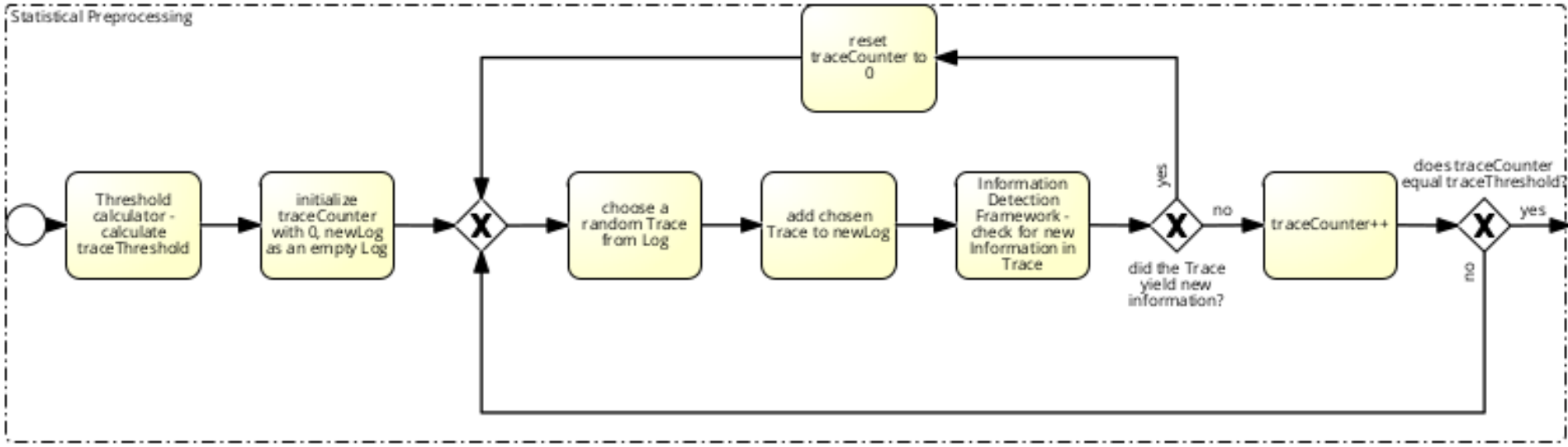
Minimum Sample Size

- „The longer we see no new information, the more likely it is that we have seen everything we need“
  - Each consecutive Trace Sampling until new Information is a binomial trial (parameters p, k, N)
    - Interval estimation can compute bounds for unknown p
    - rearranging interval for N, setting k=0, and p=δ we can compute smallest N for which p<δ holds
- BILD DER FORMEL
- For α=0.01 and δ=0.05 => N_min=128
  - After 128 consecutive trials without new information we stop

Sampling as multiple Binomial Trials

Trace #	Activities	New Info?	New binTrial?	Stop Sampling
1	A,B,C	Yes	Yes	No (1)
2	A,B,C	no	No	No (2)
3	A,B,D	Yes	Yes	No (1)
4-120	A,B,D	no	no	No (117)
121	A,B,E	Yes	Yes	No (1)
122-250	A,B,E	no	no	Yes (128)

Framework


Statistical Preprocessing
reset traceCounter to 0
Threshold calculator - calculate traceThreshold
initialize traceCounter with 0, newLog as an empty Log
choose a random Trace from Log
add chosen Trace to newLog
Information Detection Framework - check for new Information in Trace
yes
no
did the Trace yield new information?
traceCounter++
does traceCounter equal traceThreshold?
yes
no

Agenda

Background and Related Work on Process Discovery

A Statistical Framework for Process Discovery

- Log sampling
- Framework definition

Instantiating the Framework

- For control-flow discovery
- For performance discovery

Experimental Results

Control-Flow Perspective

A notion of “new control-flow information”

- New activity
- New directly-follows dependency
- New initial or final activity

Trace	New Information
A, B, C	Activity, Start, End, DF
A, B, D	Activity, End, DF
B, C, C	Start, DF
A, B, C, C	-

What about frequencies?

- Determine on sample (no guarantee on δ-similarity)
- Changes in relative frequencies are “new information”

Performance Perspective

Focus on cycle time of a process, a fine-grained numerical value

A notion of “new cycle-time information”

- Cycle time is more than ε-different
- Measuring granularity:
  - Per complete process
  - Per individual activities

A (30 ms), A (30 ms), A (30 ms)
A (30 ms, A (30 ms)

A (30 ms), A (10 ms)
A (5 ms) A (25 ms)

Agenda

Background and Related Work on Process Discovery

A Statistical Framework for Process Discovery

- Log sampling
- Framework definition

Instantiating the Framework

- For control-flow discovery
- For performance discovery

Experimental Results

Setup

Datasets

- BPI Challenge 2012
  - Loan/overdraft applications
  - >262k events of >13k traces
- BPI Challenge 2014
  - Incident management at Rabobank Group ICT,
  - >343k events of >46 traces

Discovery algorithm

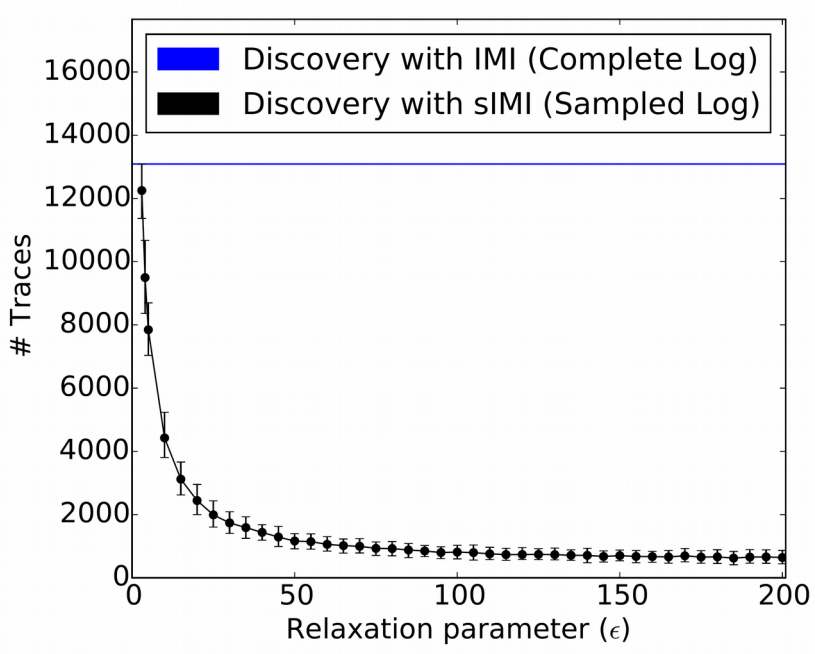
- Inductive Miner Infrequent [Leemans et al. 2013]
- Noise threshold set to 20%

Approach implemented as a ProM plugin (@Github)

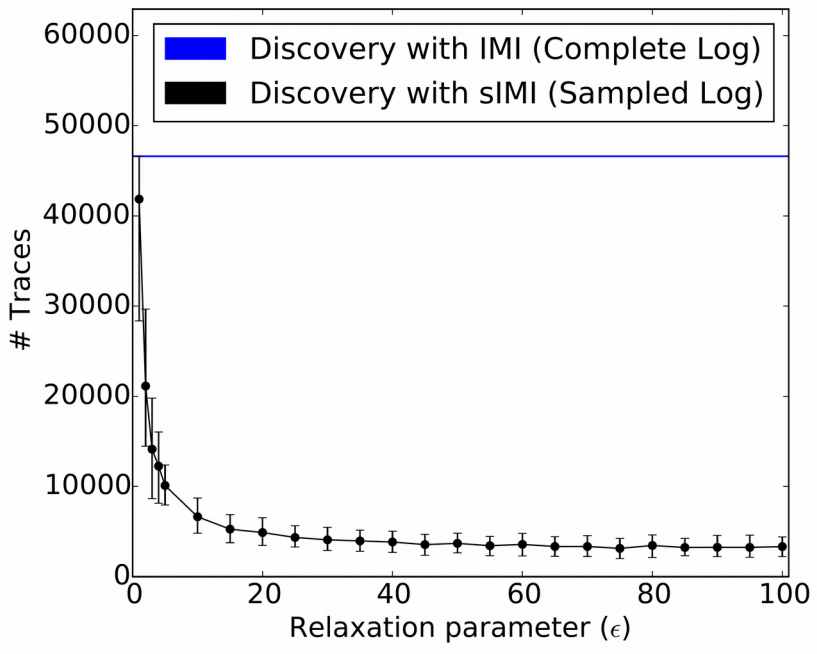
Measures

- Pre-processing effectiveness: #traces sampled
- Actual efficiency: runtime, memory footprint

Pre-Processing Effectiveness



BPI-2012

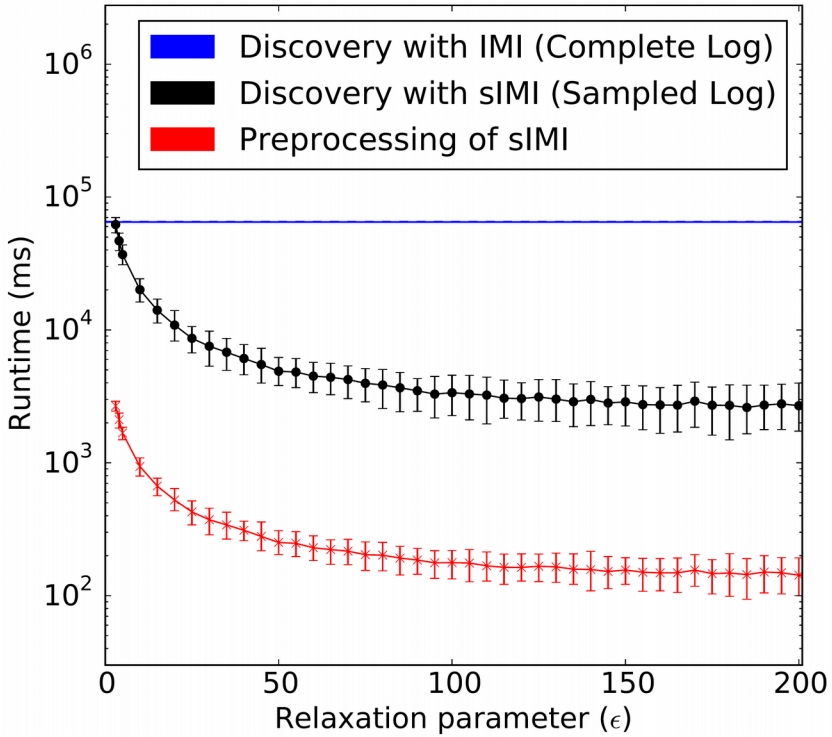


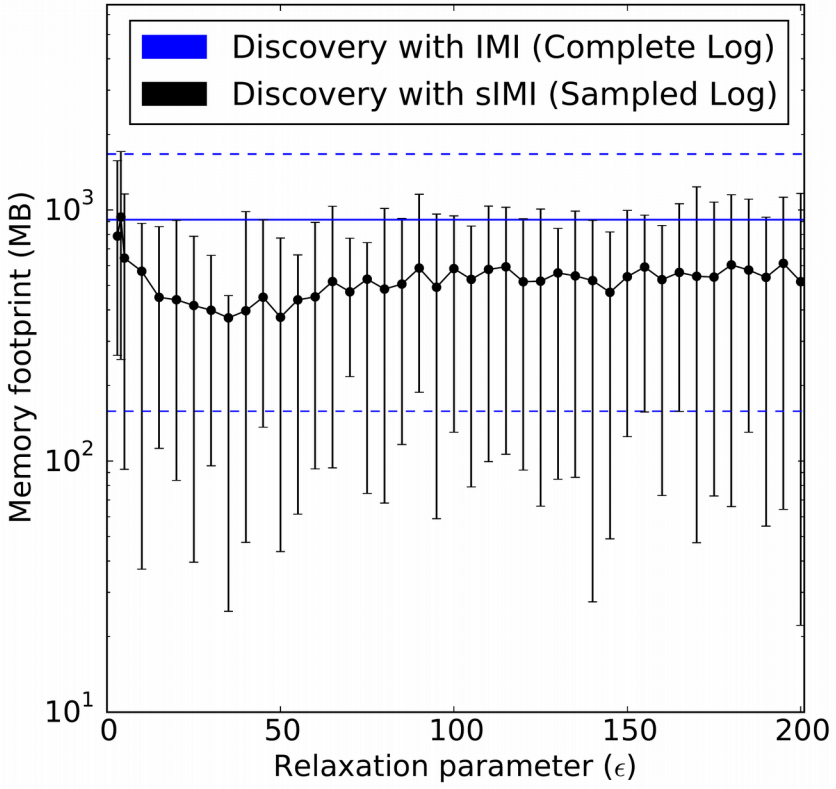
BPI-2014

Drastic reduction of number of traces considered for discovery

Trend is consistent for different datasets

Runtime and Memory Footprint

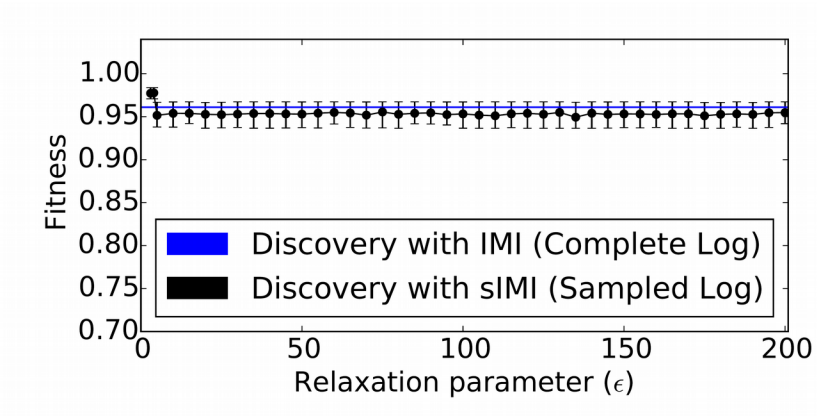


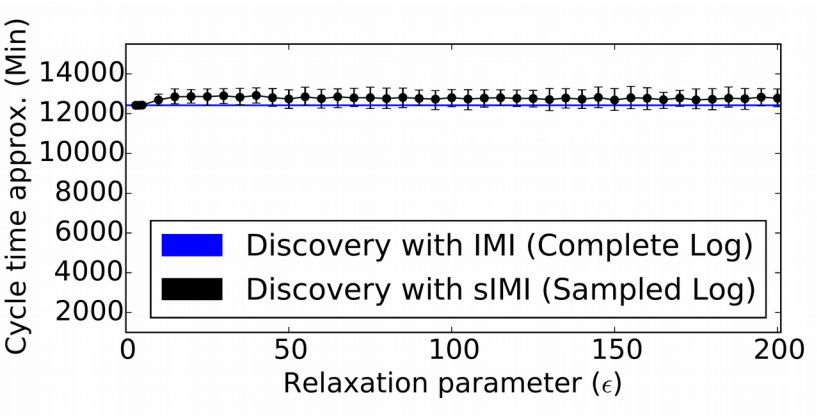


Pre-processing is efficient

Significant reduction of overall resource utilisation

Discovery Effectiveness





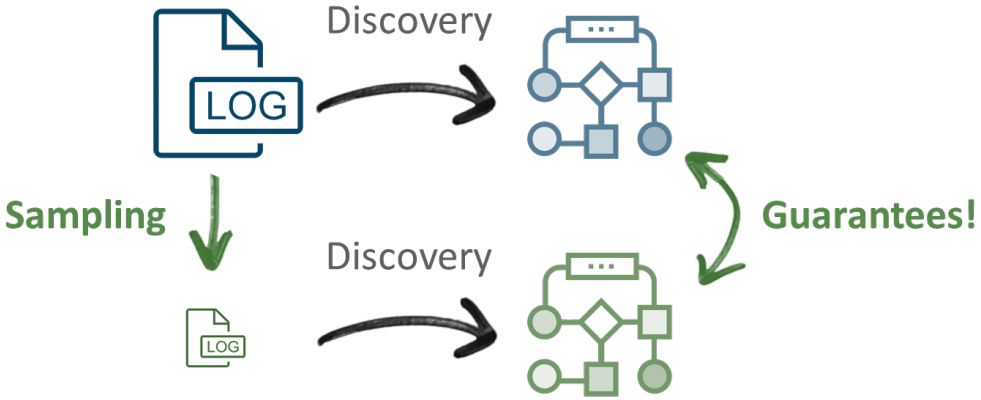
Negligible degradation of discovery quality

- For control-flow fitness
- For the cycle time approximation

Conclusions

Framework for statistical process discovery

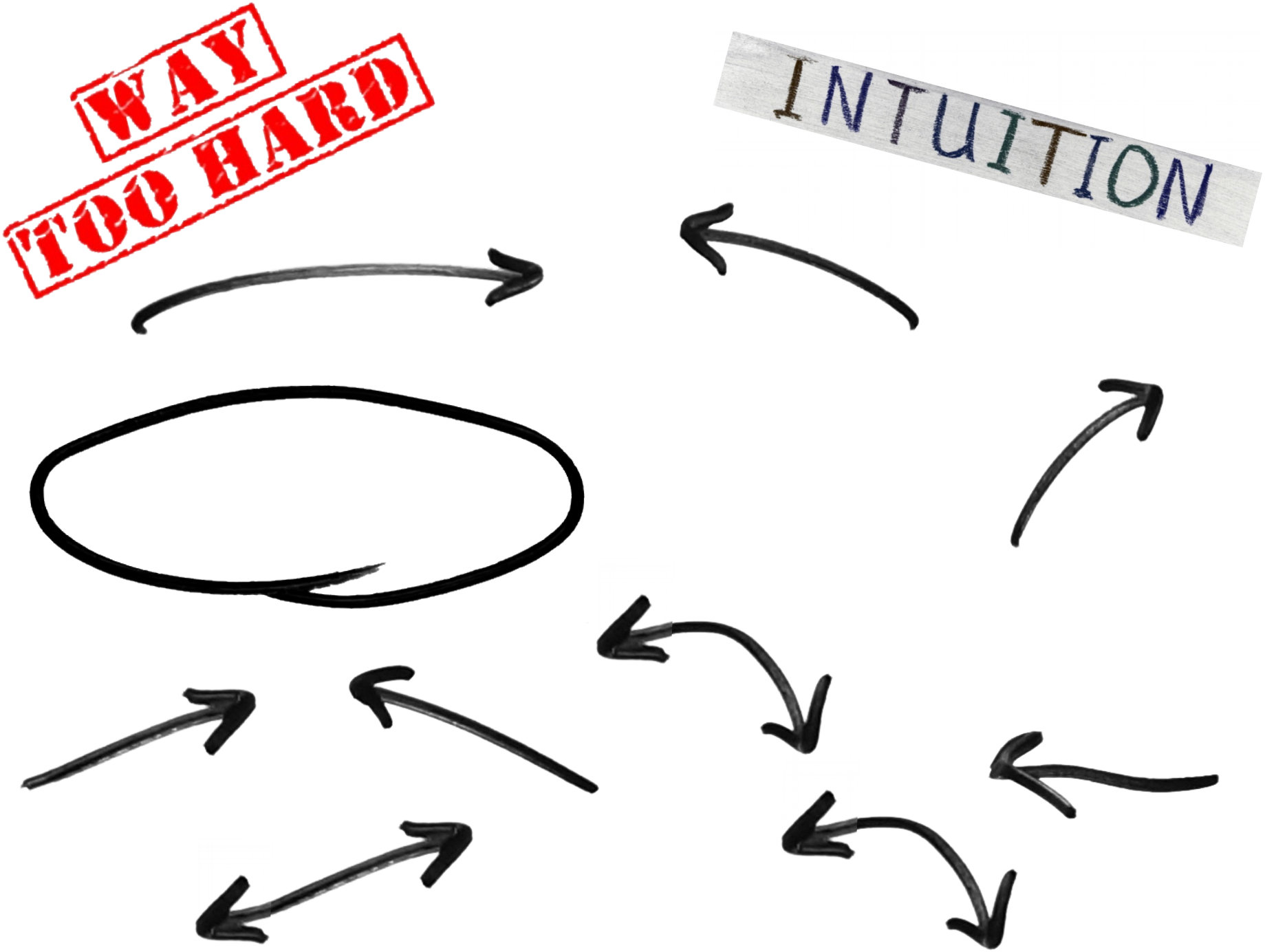
- Sample an event log
- Guarantees on the
- introduced error



Instantiation for control-flow and performance aspects

Next: Additional model perspectives

Thank you!


WAY
TOO HARD
INTUITION