Introduction to Conformance Checking

Fabrizio Maria Maggi
f.m.maggi@ut.ee

UNIVERSITY OF TARTU
Process Analysis in the BPM Lifecycle

- **Process discovery**
- **Process architecture**
- **As-is process model**
- **Insights on weaknesses and their impact**
- **To-be process model**
- **Executable process model**

**Steps in the BPM Lifecycle**

1. **Define Vision**
2. **Develop Strategy**
3. **Implement Strategy**
4. **Manage Personnel**
5. **Manage Assets**
6. **Core Processes**
7. **Support Processes**
8. **Management Processes**
9. **Manage Risk**
10. **Manage Information**
11. **Procure Materials**
12. **Procure Products**
13. **Market Products**
14. **Deliver Products**
15. **Manage Customer Service**
Process Mining
Process Mining
Replay: the Origin of Conformance Checking

Replay: the Origin of Conformance Checking

A B C E
Replay: the Origin of Conformance Checking

B C E
Replay: the Origin of Conformance Checking

B C E
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking

E
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking

A C E
Replay: the Origin of Conformance Checking

C E
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking

missing token
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking

remaining token
Quality Criteria for Conformance Checking

Dealing with Noise and Incompleteness

The **ideal process model** allows for the behavior coinciding with the frequent behavior seen when the process would be infinitely observed.

Mature process mining algorithms allow to **abstract** from infrequent behavior.
Four competing quality criteria

In general, the **quality** of a process mining result refers to **four quality dimensions**:

1. **Fitness**: the discovered model should allow for the behavior seen in the event log.
   - A model has a **perfect fitness** if all traces in the log can be replayed from the beginning to the end.
Four competing quality criteria

In general, the quality of a process mining result refers to four quality dimensions:

1. **Fitness**
2. **Precision** (avoid underfitting): the discovered model should not allow for behavior completely unrelated to what was seen in the event log.
Four competing quality criteria

In general, the quality of a process mining result refers to four quality dimensions:

1. **Fitness**
2. **Precision** (avoid underfitting)
3. **Generalization** (avoid overfitting): the discovered model should generalize the example behavior seen in the event log.
Four competing quality criteria

In general, the quality of a process mining result refers to four quality dimensions:

1. **Fitness**: the discovered model should allow for the behavior seen in the event log.
   - A model has a *perfect fitness* if all traces in the log can be replayed from the beginning to the end.

2. **Precision** (avoid underfitting): the discovered model should not allow for behavior completely unrelated to what was seen in the event log.

3. **Generalization** (avoid overfitting): the discovered model should generalize the example behavior seen in the event log.

4. **Simplicity**: the discovered model should be as simple as possible.
   - Occam’s Razor: The simplest model that can explain the behavior seen in the log is the best model.
Flower model (underfitting)

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^5, <a,f,g,i,j,k,l>^{360}\} \]
Enumerating model (overfitting)

\[ L = \{ <a, b, i, j, k, l>^{10}, <a, b, g, j, k, i, l>^{140}, <a, f, g, j, i, k>^{5}, <a, f, g, i, j, k, l>^{360}\} \]
Something in the middle...

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360}\} \]
Computing fitness: basic approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \} \]
Computing fitness: basic approach

\[ L = \{ <a, b, i, j, k, l>^{10}, <a, b, g, j, k, i, l>^{140}, <a, f, g, j, i, k>^5, <a, f, g, i, j, k, l>^360 \} \]
Computing fitness: basic approach
Computing fitness: basic approach
Computing fitness: basic approach

B I J K L
Computing fitness: basic approach

B I J K L
Computing fitness: basic approach
Computing fitness: basic approach

I J K L
Computing fitness: basic approach
Computing fitness: basic approach

L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \}\n
A “basic approach” to compute fitness is to count the fraction of cases that can be “parsed completely” (i.e., the proportion of cases corresponding to firing sequences leading from [start] to [end]).
Computing fitness: basic approach

L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \}\n
A “basic approach” to compute fitness is to count the fraction of cases that can be “parsed completely” (i.e., the proportion of cases corresponding to firing sequences leading from [start] to [end]).

Fitness = 0.97
Computing fitness: Event-based approach

- In the simple fitness computation, we stopped replaying a trace once we encounter a problem and mark it as non-fitting.
- An event-based approach to calculate fitness consists of just continue replaying the trace on the model and:
  - record all situations where a transition is forced to fire without being enabled, i.e., we count all missing tokens.
  - record the tokens that remain at the end.
- Use of four counters:
  - \( p \) = produced tokens
  - \( c \) = consumed tokens
  - \( m \) = missing tokens
  - \( r \) = remaining tokens
Computing fitness: Event-based approach

L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \}
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360}\} \]
Computing fitness: Event-based approach

A B I J K L
Computing fitness: Event-based approach

A B I J K L

\[ p = 1 \]
\[ c = 0 \]
\[ m = 0 \]
\[ r = 0 \]
Computing fitness: Event-based approach

B I J K L
Computing fitness: Event-based approach
Computing fitness: Event-based approach
Computing fitness: Event-based approach

\[
\begin{align*}
\text{I} & \quad \text{J} \quad \text{K} \quad \text{L} \\
\end{align*}
\]
Computing fitness: Event-based approach
Computing fitness: Event-based approach

I J K L
Computing fitness: Event-based approach

J K L
Computing fitness: Event-based approach

\[ K = L \]
Computing fitness: Event-based approach
Computing fitness: Event-based approach
Computing fitness: Event-based approach
Computing fitness: Event-based approach

\[ p = 1 \quad c = 1 \quad m = 0 \quad r = 0 \]

\[ p = 1 \quad c = 1 \quad m = 0 \quad r = 0 \]

\[ p = 1 \quad c = 1 \quad m = 0 \quad r = 0 \]

\[ p = 1 \quad c = 1 \quad m = 0 \quad r = 0 \]

\[ p = 12 \quad c = 12 \quad m = 1 \quad r = 1 \]
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^5, <a,f,g,i,j,k,l>^{360}\} \]

\[
\text{fitness}(\sigma, N) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right)
\]
Computing fitness: Event-based approach

\[ L = \{ <a, b, i, j, k, l>^{10}, <a, b, g, j, k, i, l>^{140}, <a, f, g, j, i, k>^5, <a, f, g, i, j, k, l>^{360} \} \]

\[
\text{fitness}(\sigma, N) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right)
\]

\[
p = 12, \quad c = 12, \quad m = 1, \quad r = 1
\]
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \} \]

\[ p = 12 \]
\[ c = 12 \]
\[ m = 1 \]
\[ r = 1 \]

fitness(\(\sigma, N\)) = \(\frac{1}{2} \left(1 - \frac{m}{c}\right) + \frac{1}{2} \left(1 - \frac{r}{p}\right)\)

Fitness = 0.9166
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^10, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^5, <a,f,g,i,j,k,l>^{360} \} \]

\[ \text{fitness}(\sigma, N) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right) \]
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360}\} \]

\[ p = 13 \]
\[ c = 13 \]
\[ m = 0 \]
\[ r = 0 \]

\[ \text{fitness}(\sigma, N) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right) \]
Computing fitness: Event-based approach

\[ L = \{ <a, b, i, j, k, l>^{10}, <a, b, g, j, k, i, l>^{140}, <a, f, g, j, i, k>^5, <a, f, g, i, j, k, l>^{360}\} \]

\[ p = 13 \]
\[ c = 13 \]
\[ m = 0 \]
\[ r = 0 \]

fitness(\(\sigma, N\)) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right)

Fitness = 1
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \} \]

\[
\text{fitness}(\sigma, N) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right)
\]
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \} \]
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, [a,f,g,j,i,k>^5, <a,f,g,i,j,k,l>^{360}\} \]

\[ p = 1 \]
\[ c = 1 \]
\[ m = 0 \]
\[ r = 0 \]
Computing fitness: Event-based approach

\[ L = \{ \langle a, b, i, j, k, l \rangle^{10}, \langle a, b, g, j, k, i, l \rangle^{140}, \langle a, f, g, j, i, k \rangle^{5}, \langle a, f, g, i, j, k, l \rangle^{360} \} \]

\[ p = 12 \]
\[ c = 11 \]
\[ m = 0 \]
\[ r = 1 \]

\[ \text{fitness}(\sigma, \hat{N}) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right) \]

Fitness = 0.9583
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \} \]

\[
\text{fitness}(\sigma, \hat{N}) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right)
\]
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360}\} \]

\[ p = 13 \]
\[ c = 13 \]
\[ m = 0 \]
\[ r = 0 \]

\[
\text{fitness}(\sigma, N) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right)
\]
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360}\} \]

\[
p = 13 \\
c = 13 \\
m = 0 \\
r = 0
\]

\[
fitness(\sigma, \hat{N}) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right)
\]

Fitness = 1
Computing fitness at log level

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \} \]

\[
\text{fitness}(L, N) = \frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times c_{N,\sigma}} \right) + \frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times p_{N,\sigma}} \right)
\]
Computing fitness at log level

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360}\} \]

Number of occurrences of a specific trace in the log (e.g., if a trace \( \sigma \) appears 200 times in the log, \( L(\sigma) \) will be equal to 200.)

\[
\text{fitness}(L, N) = \frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times c_{N,\sigma}} \right) + \frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times p_{N,\sigma}} \right)
\]
Computing fitness at log level

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360}\} \]

\[
\begin{align*}
\text{p} &= 12 & \text{c} &= 12 & \text{m} &= 1 & \text{r} &= 1 \\
\text{p} &= 13 & \text{c} &= 13 & \text{m} &= 0 & \text{r} &= 0 \\
\text{p} &= 12 & \text{c} &= 11 & \text{m} &= 0 & \text{r} &= 1 \\
\text{p} &= 13 & \text{c} &= 13 & \text{m} &= 0 & \text{r} &= 0
\end{align*}
\]

\[
\text{fitness}(L, N) = \frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times c_{N,\sigma}} \right) + \frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times p_{N,\sigma}} \right)
\]
Computing fitness at log level

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^5, <a,f,g,i,j,k,l>^{360} \} \]

\[
\begin{align*}
    p &= 12 & p &= 13 & p &= 12 & p &= 13 \\
    c &= 12 & c &= 13 & c &= 11 & c &= 13 \\
    m &= 1 & m &= 0 & m &= 0 & m &= 0 \\
    r &= 1 & r &= 0 & r &= 1 & r &= 0 \\
\end{align*}
\]

\[
\text{fitness}(L,N) = \frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times c_{N,\sigma}} \right) + \\
\frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times p_{N,\sigma}} \right)
\]

Fitness = 0.998
Computing Precision

Computing Precision

\[ L = \{ <a, b, i, j, k, l>^{10}, <a, b, g, j, k, i, l>^{140}, <a, f, g, j, i, k>^5, <a, f, g, i, j, k, l>^{360} \} \]
Computing Precision

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \} \]

# distinct traces in the log conformant with the model (2) / # possible traces conformant with the model (6) = 0.33
A common approach to compute generalization in order to check the quality of a discovered model.
A common approach to compute generalization in to check the quality of a discovered model
A common approach to compute generalization in to check the quality of a discovered model
A common approach to compute generalization in to check the quality of a discovered model
A common approach to compute generalization in order to check the quality of a discovered model
A common approach to compute generalization in

to check the quality of a discovered model
A common approach to compute generalization in
to check the quality of a discovered model
A common approach to compute generalization in
to check the quality of a discovered model
A common approach to compute generalization in
to check the quality of a discovered model

Diagrams:

- Nodes: N1, N2, N3, N4, N5
- Links: L1, L2, L3, L4, L5
- Average Fitness (Avg Fitness)