LTAT.05.025

Business Process Mining

Lecture 9: Declarative Process Mining

Anti Alman

Junior Research Fellow @ University of Tartu
The Declarative Approach
The Declarative Approach in a Nutshell

Describes a business process in terms of
… what rules are followed by the process

Allows flexibility in deciding
… how these rules are fulfilled

For example: X and Y can only occur after A has occurred
Note that the exact order and possibility of repetitions of X and Y is left open
The Declarative Approach in a Nutshell

For example: X and Y can only occur after A has occurred

*Note that the exact order and possibility of repetitions of X and Y is left open*

Positive examples:

- ✓ A → X → Y → ...
- ✓ A → X → ...
- ✓ A → ...
- ✓ B → ...

Negative examples:

- ✗ X → Y → A → ...
- ✗ X → Y → ...
- ✗ X → ...
- ✗ Y → ...
Motivation for the Declarative Approach

Business processes are different
• Some are simple and can be modeled both fully and also in an understandable way
Motivation for the Declarative Approach

Business processes are different
• Some are simple and can be modeled both fully and also in an understandable way
• Some are too big to model both fully and in an understandable way
Motivation for the Declarative Approach

Business processes are different
• Some are simple and can be modeled both fully and also in an understandable way
• Some are too big to model both fully and in an understandable way
• And some are inherently too flexible to be modeled fully
  • For example, treatment process of some disease
  • … combined with every potential set of allergies each individual patient may have
  • … and multiplied by different severity levels
Business Process Models – Conceptually
Paradigms

Procedural (imperative) paradigm
• Suitable for predictable processes
  • limited number of exceptions and deviations
• Focus on **how** things must be done

Declarative paradigm
• Suitable for unpredictable processes
  • high number of exceptions and deviations
• Focus on **what** must be accomplished
Procedural Paradigm

Let's take a blank model
• “Model” is in the box below

Nothing can happen
• As restrictive as possible
• Also, the model is invalid – we are even missing a place to start in
Procedural Paradigm

Let’s add something

• ER Registration must occur – still invalid
• ER Registration must occur first – still invalid
• ER Triage occurs immediately after ER Registration – still invalid
**Procedural Paradigm**

We need at least one full path from start to end

Supporting more process behaviour means adding more paths

- How do we cope with an ever-increasing model?
- Can we avoid overspecifying the process?
- Do we also add a path for something that occurs once in a million executions?
Theory vs. Practice

In theory, a procedural model will include all allowed execution paths … while also excluding all prohibited execution paths

In practice, we very rarely follow the theory … we might represent only the main flows … we might exclude rare execution paths … we might make simplifications even in the main flow

omitting intricacies of each decision, upper and lower bounds on repetition, etc.
Declarative Paradigm

Let’s start again with a blank model

- “Model” is in the box below

The crucial difference – everything can happen

- As flexible as possible
- And technically it is a valid model, not very useful, but valid nevertheless
Declarative Paradigm

Let’s add something

- ER Registration can occur – valid model and a bit more useful
- ER Registration must occur first – valid model, even more useful
- ER Triage occurs immediately after ER Registration – valid again
Declarative Paradigm

But we are not required to go in the execution order
• If ER Sepsis Triage occurs, then it is eventually followed by Leucocytes
• If Admission NC occurs, then CRP also occurs and vice versa
Declarative Paradigm

In fact, we can go in order of importance

• Describing the most important rules first, one by one, moving closer to the level of detail we wish to achieve

• And whatever is left undescribed, is up to the process worker to decide
Closed World vs Open World

Closed world – Imperative
• If it is not in the model, then it is not allowed
• Always explicit control flow

Open world – Declarative
• If it does not contradict the model, then it is allowed
• Mostly implicit control flow
• Sometimes tricky to understand
“You have seen it before”
An Analogy for the Declarative Approach

Most procedural process mining tools (including AproMore) have a “declarative component”

Filters

• A filter is basically a declarative rule

• Filter types are basically pre-prepared templates for these rules
  • Retain all cases that contain Activity X – where X is defined while creating the filter

• A trace can match a filter/rule, regardless of what occurs in unrelated parts of the trace
  • Retain all cases that contain Activity 'ER Registration' – leaves all other activities open
Filters and Declarative Rules

ER Registration must occur

Note: Using ‘Existence’ here for simplicity

Filter: "Retain all cases that contain Activity 'ER Registration'"
Filters and Declarative Rules

ER Triage occurs immediately after ER Registration

Filter: Retain all cases that contain the directly-follows relation 'ER Registration' -> 'ER Triage' between Activity nodes
Filters and Declarative Rules

If ER Sepsis Triage occurs, then it is eventually followed by Leucocytes

Filter: "Retain all cases that contain the eventually-follows relation 'ER Triage' -> 'ER Registration'"
Filters as Declarative Rules

However

• If admission NC occurs, then CRP also occurs and vice versa

• You can do “both exist” or “neither exist”, but can't really combine these nicely in Apromore

• And this relation can be tricky to insert into procedural models
  • What if the activities are on different parallel branches in BPMN model?
  • What if the one of the activities is at the very beginning of the process and the other at the very end?
Set of Rules as a Process Model

One could argue that a set of filters/rules also defines a business process, at least partially.

But remember, we used this mainly as an analogy.

- There is significant overlap and many similarities, but filtering is not intended for describing a business process, it is intended for filtering.
A Note on Conformance Checking

We have seen how filters can be used for conformance checking

*For example: ER Triage must occur exactly once*

If your process model is a set of declarative rules (similar to filters) … then conformance checking will come more naturally

Each declarative rule can be checked individually and automatically … and violations can be pinpointed precisely to specific events
Declarative Process Models using Declare
Declare Process Modeling Language

One of the earlier and more popular declarative languages
Maja Pesic, Helen Schonenberg, Wil M. P. van der Aalst:

A Declare model consists of constraints
• Each constraint defines some important aspect of the process

A constraint consists of...
• Template – The semantic meaning of the constraint
• Activity reference(s) – Activity or activities to which this meaning applies

An example of a simple constraint
• Template – The activity must exist
• Activity reference – 'ER Registration'
**Visual Notation of Declare**

Graph-based like BPMN, but not a flowchart

- Already used in previous parts of the lecture

Activity nodes are not duplicated

- For example, if we also had: 'ER Registration' is eventually followed by 'ER Sepsis Triage'

![Diagram showing activity nodes and constraints]

- Constraint on a single activity (unary constraint)
- Constraint between two activities (binary constraint)
Visual Notation of Declare

Furthermore
• Model can be disconnected
• Activities can appear without constraints
  • Mostly for informative purposes, but can be otherwise useful also

Constraint on a single activity (unary constraint)
Constraint between two activities (binary constraint)
Unary Declare Templates

Used to define the cardinality or position of an activity

- **Existence**
  - A occurs at least once

- **Exactly2**
  - A occurs exactly twice

- **Absence3**
  - A occurs at most twice

- **Init**
  - A occurs first

- **End**
  - A occurs last
Binary Declare Templates

Used to define a relation between two activities

Response(A, B)

- If A occurs, then B occurs after A

Alternate Response(A, B)

- Each time A occurs, then B occurs afterwards before A recurs

Chain Response(A, B)

- Each time A occurs, then B occurs immediately afterwards
**Binary Declare Templates**

Additional examples (non-exhaustive)

Succession (A, B)

A occurs if and only if it is followed by B
(Alternate and Chain Succession also possible)

Not Succession (A, B)

A can never occur before B
(All binary templates can be negated)

Choice (A, B)

A or B must occur at least once
Alternative Representations of Declare

A Declare model can be represented in three equivalent ways

Constraints
1) ER Registration occurs at least once
2) Each time ER Registration occurs, then ER Triage occurs immediately afterwards

Textual description

Finite-state automaton
(LTL-based, allows for multiple formal techniques)
MP-Declare

Data-Aware extension of Declare
Supports
• Conditions on event data
• Temporal conditions

If A occurs with ‘booleanValue = True’, then within 3 to 5 hours
B occurs with ‘integerValue > 5’ after A
Declarative Process Mining using Declare
The main application for Declare – RuM

- Available at: https://rulemining.org/

RuM supports

- Automated process discovery
- Conformance Checking
  - Fulfilments, Violations
  - Log alignment
- Log Generation (Simulation)
- Process Model Editor
- Monitoring
A Scenario for Demonstration

Synthetic event log mimicking behavior found in a real event log

- An example to cover main functionalities in a logical manner
- 2 event logs of sepsis cases
- Completed cases only (50/50 split)

Main steps

1. Discover a model
2. Validate the model
3. Improve the model
4. Create an Event Log
Demo Time!
Some Other Declarative Languages

• DCR Graphs
  Thomas T. Hildebrandt, Raghava Rao Mukkamala:
  Declarative Event-Based Workflow as Distributed Dynamic Condition Response Graphs. PLACES 2010: 59-73

• GSM Models

• OCBC Models
  Alessandro Artale, Alisa Kovtunova, Marco Montali, Wil M. P. van der Aalst:
  Modeling and Reasoning over Declarative Data-Aware Processes with Object-Centric Behavioral Constraints. BPM 2019: 139-156.

• Hybrid (mixed-paradigm) models
  Amine Abbad Andaloussi, Andrea Burattin, Tijs Slaats, Ekkart Kindler, Barbara Weber:
  On the declarative paradigm in hybrid business process representations: A conceptual framework and a systematic literature study. Inf. Syst. 91: 101505 (2020)
The Declarative Approach in a Nutshell

**Describes** a business process in terms of
... *what* rules are followed by the process

**Allows flexibility** in deciding
... *how* these rules are fulfilled

For example: X and Y can only occur after A has occurred

*Note that the exact order and possibility of repetitions of X and Y is left open*