Conformance Checking Using Object-Centric Behavioral Constraints

Wil van der Aalst and Marco Montali
www.processmining.org
Motivation

- Processes are not “flat”.
- It is difficult to model interactions between process instances, which are in fact typically considered in isolation.
- It is also difficult to model the data-perspective in a precise way.
- Cardinality constraints in the data model must influence behavior, but seldom do in existing models.
Idea

• Data and control-flow in the same diagram: coexistence of two aspects rather than a tight integration.
• Declarative style of modeling.
• Cardinality constraints in data models resemble constraints in Declare.
• Consider all instances at the same time or, better, abandon the classical “flat instance notion”.
• Constraints relate activities through the data model.
• Still abstract from concrete date values (can be added later).
Activities, object classes, relationship classes, cardinalities in data model, cardinalities between activities and object classes, behavioral constraints, reference events, target events, etc.
Consider a single instance for a moment

Before ≥ 0 and after ≥ 1
Before ≥ 0 and after = 1
Before = 1 and after ≥ 0
Before = 0 and after ≥ 0
Before + after ≥ 1
Before = 0 and after = 0

(response)
(unary-response)
(non-response)
(precedence)
(unary-precedence)
(non-precedence)
(co-existence)
(non-coexistence)
Like cardinalities in a data model

<table>
<thead>
<tr>
<th>constraint</th>
<th>formalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>response</td>
<td>{ (before, after) \in \mathbb{N} \times \mathbb{N} \mid after \geq 1 }</td>
</tr>
<tr>
<td>unary-response</td>
<td>{ (before, after) \in \mathbb{N} \times \mathbb{N} \mid after = 1 }</td>
</tr>
<tr>
<td>non-response</td>
<td>{ (before, after) \in \mathbb{N} \times \mathbb{N} \mid after = 0 }</td>
</tr>
<tr>
<td>precedence</td>
<td>{ (before, after) \in \mathbb{N} \times \mathbb{N} \mid before \geq 1 }</td>
</tr>
<tr>
<td>unary-precedence</td>
<td>{ (before, after) \in \mathbb{N} \times \mathbb{N} \mid before = 1 }</td>
</tr>
<tr>
<td>non-precedence</td>
<td>{ (before, after) \in \mathbb{N} \times \mathbb{N} \mid before = 0 }</td>
</tr>
<tr>
<td>co-existence</td>
<td>{ (before, after) \in \mathbb{N} \times \mathbb{N} \mid before + after \geq 1 }</td>
</tr>
<tr>
<td>non-co-existence</td>
<td>{ (before, after) \in \mathbb{N} \times \mathbb{N} \mid before + after = 0 }</td>
</tr>
</tbody>
</table>

allowed cardinalities

<table>
<thead>
<tr>
<th>notation</th>
<th>allowed cardinalities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1}</td>
</tr>
<tr>
<td>1..k</td>
<td>{1, 2, \ldots, k}</td>
</tr>
<tr>
<td>*</td>
<td>{0, 1, 2, \ldots}</td>
</tr>
<tr>
<td>1..*</td>
<td>{1, 2, \ldots}</td>
</tr>
</tbody>
</table>
Shorthands

\[
\begin{align*}
\text{register} & \rightarrow \text{play music} \\
\text{place order} & \rightarrow \text{pay}
\end{align*}
\]

equals

\[
\begin{align*}
\text{register} & \rightarrow \text{play music} \\
\text{place order} & \rightarrow \text{pay}
\end{align*}
\]
Definition 6 (Constraint Satisfaction). Let $BCM = (A, C, \pi_{ref}, \pi_{tar}, type)$ be a behavioral constraint model, and $E \subseteq U_E$ a set of events.

- Event set $E$ satisfies constraint $c$ if and only if

$$\left( |\partial_{\pi_{tar}}(c)(E \uparrow e_{ref})|, |\partial_{\pi_{tar}}(c)(E \downarrow e_{ref})| \right) \in type(c) \text{ for all } e_{ref} \in \partial_{\pi_{ref}}(c)(E)$$

- Event set $E$ satisfies $BCM$ if and only if $E$ satisfies each constraint $c \in C$.

Take reference event, determine target events, and count number of target events before and after the reference event. Then check cardinalities. Do this for all reference events (dots).
Class Model and Object Model

(a) class model

(b) object model
Definition 9 (Valid Object Model). Let $\text{ClaM} = (OC, RT, \pi_1, \pi_2, \text{card}_1, \text{card}_2)$ be a class model and $\text{OM} = (\text{Obj}, \text{Rel}, \text{class}) \in \mathcal{U}_{\text{OM}}$ be an object model. $\text{OM}$ is valid for $\text{ClaM}$ if and only if

- for any $(r, o_1, o_2) \in \text{Rel}$: $\text{class}(o_1) = \pi_1(r)$ and $\text{class}(o_2) = \pi_2(r)$,
- for any $r \in RT$ and $o_2 \in \partial_{\pi_2(r)}(\text{Obj})$, we have that

$$|\{o_1 \in \text{Obj} \mid (r, o_1, o_2) \in \text{Rel}\}| \in \text{card}_1(r), \quad \text{and}$$

- for any $r \in RT$ and $o_1 \in \partial_{\pi_1(r)}(\text{Obj})$, we have that

$$|\{o_2 \in \text{Obj} \mid (r, o_1, o_2) \in \text{Rel}\}| \in \text{card}_2(r)$$

Cardinality constraints are respected. Can be extended to any test between a class model and an object model (could also include attributes and multiple relations at the same time).
Putting it together …

• Event log: events refer to objects and the object model before and after.

• Object-Centric Behavioral Constraint Model composed of three parts:
  - Behavioral constraints
  - Class model
  - Relationships between both
    - Cardinality constraints between activities and objects
    - Correlation between reference and target events
Definition 10 (Event Log). An event log is a tuple $L = (E, act, attr, EO, pre, post, \leq)$, where

- $E \subseteq \mathcal{U}_E$ is a set of events,
- $act \in E \rightarrow \mathcal{U}_A$ maps events onto activities,
- $attr \in E \rightarrow (\mathcal{U}_{Attr} \rightarrow \mathcal{U}_{Val})$ maps events onto a partial function assigning values to some attributes,
- $EO \subseteq E \times \mathcal{U}_O$ relates events to sets of object references,
- $pre \subseteq E \rightarrow \mathcal{U}_{OM}$ maps each event to the object model before the event took place,
- $post \subseteq E \rightarrow \mathcal{U}_{OM}$ maps each event to the object model after the event took place, and
- $\leq \subseteq E \times E$ defines a total order on events.
Object-Centric Behavioral Constraint (OCBC) model

Definition 11 (Object-Centric Behavioral Constraint Model). An object-centric behavioral constraint model is a tuple $OCBCM = (BCM, ClaM, AOC, card_A, card_F^A, card_O^C, card_F^O^C, crel)$, where

- $BCM = (A, C, \pi_{ref}, \pi_{tar}, \text{type})$ is a behavioral constraint model (Definition 5),
- $ClaM = (OC, RT, \pi_1, \pi_2, card_1, card_2)$ is a class model (Definition 7),
- $A, C, OC$ and $RT$ are pairwise disjoint (no name clashes),
- $AOC \subseteq A \times OC$ is a set of relations between activities and object classes,
- $\text{card}_A \in AOC \rightarrow U_{\text{Card}}$ gives the cardinality of the source of a relation linking an activity and an object class (activity side) at any point in time,
- $\text{card}_F^A \in AOC \rightarrow U_{\text{Card}}$ gives the cardinality of the source of a relation linking an activity and an object class (activity side) that should hold from some point onwards,
- $\text{card}_O^C \in AOC \rightarrow U_{\text{Card}}$ gives the cardinality of the sink of a relation linking an activity and an object class (object-class side) at any point in time,
- $\text{card}_F^O^C \in AOC \rightarrow U_{\text{Card}}$ gives the cardinality of the sink of a relation linking an activity and an object class (object-class side) that should hold from some point onwards, and
- $crel \in C \rightarrow OC \cup RT \cup \tilde{RT}$ is the constraint relation\(^5\) satisfying the following conditions for each $c \in C$:
  - $\{ (\pi_{ref}(c), oc), (\pi_{tar}(c), oc) \} \subseteq AOC$ if $crel(c) = oc \in OC$.
  - $\{ (\pi_{ref}(c), \pi_1(r)), (\pi_{tar}(c), \pi_2(r)) \} \subseteq AOC$ if $crel(c) = r \in RT$.
  - $\{ (\pi_{ref}(c), \pi_2(r)), (\pi_{tar}(c), \pi_1(r)) \} \subseteq AOC$ if $crel(c) = \hat{r} \in \tilde{RT}$.
Object-Centric Behavioral Constraint (OCBC) model

- **activity**
- **constraint**
- **indicating the reference event**
- **constraint relation** (used to determine target events related to the reference event)

Relationship between activities and classes:
- Each a1 event refers to precisely one oc1 object.
- Each oc2 object refers to at least one a1 activity.
- Each oc3 object refers to at least one a3 activity.

Diagram details:
- a1 connects to c1, a2, and oc1.
- a2 connects to c2, a3, and oc2.
- c1 connects to c2.
- oc1 connects to r1, oc2.
- oc2 connects to r2, oc3.
- r1 connects to r2.
- constraint relation is indicated by arrows connecting the objects.
Navigation between reference event and target events (correlation through data model)

(a) the reference event and target events are related through common objects

(b) the reference event and target events are related through relations in the object model

(c) the reference event and target events are related through inverse relations in the object model
Example

create order
pick item
order line
order delivery
wrap item
deliver items

order
order line
delivery
product
customer

1 1
1
1 1..*
1..*
1
1 1
0..* 1..*
1 1
0..*
1
2
3
4
5
6
7
8
9

1
1
1
1
1
1
1
1
1
Traditional process models
Event log \( L \) satisfies model OCBCM

- All object models visited are valid.
- The object models are monotonic: objects do not disappear or change class.
Event log $L$ satisfies model OCBCM

- Event logs only refers to things that exist in process model or object model.
- Activities exist in model.
- Objects exist in object model after event.
- Activity-Object Class relations exist in model.
Event log \( L \) satisfies model OCBCM

- **Cardinality constraints between activities and object classes are respected.**
- **Two variants: stronger one (holds for all snapshots) weaker one (from some point in time onwards)**

There are no problems related to the object models observed by the event log:
- for any \( e \in E \): object models \( \text{pre}(e) \) and \( \text{post}(e) \) are valid for \( \text{ClaM} \) (Definition 9),
- for any \( e \in E \): \( \text{Obj}_{\text{pre}}^e \subseteq \text{Obj}_{\text{post}}^e \) and \( \text{class}_{\text{pre}}^e \subseteq \text{class}_{\text{post}}^e \) (objects do not disappear or change class through events),
- for any \( e_1, e_2 \in E \) such that \( e_1 < e_2 \): \( \text{Obj}_{\text{pre}}^{e_2} \subseteq \text{Obj}_{\text{pre}}^{e_1} \) and \( \text{class}_{\text{pre}}^{e_2} \subseteq \text{class}_{\text{pre}}^{e_1} \) (objects do not disappear or change class in-between events).

The event log refers to things that exist:
- \( \{ \text{act}(e) \mid e \in E \} \subseteq A \) (all activities referred to by events exist in the behavioral model),
- for all \((e, o) \in E O \): \( o \in \text{Obj}_{\text{pre}}^e \) (all objects referred to by events exist in the object model after the corresponding event),
- \( \{ (\text{act}(e), \text{class}_{\text{post}}^e(o)) \mid (e, o) \in E O \} \subseteq A O C \) (events do not refer to objects of unrelated classes).

The cardinality constraints between activities and object classes are respected:
- for any \((a, oc) \in A O C \) and \( e_{\text{now}} \in E \):
  - for any \( e \in \partial_a(E) \) with \( e \leq e_{\text{now}} \): \( \{ o \in \partial_{oc}(\text{Obj}_{\text{post}}^e) \mid (e, o) \in E O \} \in \text{card}_{oc}^a(a, oc) \) (each event refers to a permissible number of objects in \( oc \)),
  - for any \( o \in \partial_{oc}(\text{Obj}_{\text{post}}^e) \): \( \{ e \in \partial_a(E) \mid (e, o) \in E O \wedge e \leq e_{\text{now}} \} \in \text{card}_a^c(a, oc) \) (each object of \( oc \) refers to a permissible number of events),
  - there exists an \( e_{\text{later}} \in E \) such that for any \((a, oc) \in A O C \) and \( e_{\text{now}} \in E \) with \( e_{\text{later}} \leq e_{\text{now}} \):
    - for any \( e \in \partial_a(E) \) with \( e \leq e_{\text{now}} \): \( \{ o \in \partial_{oc}(\text{Obj}_{\text{post}}^e) \mid (e, o) \in E O \} \in \text{card}_{oc}^a(a, oc) \) (eventually each event refers to a permissible number of objects in \( oc \)),
    - for any \( o \in \partial_{oc}(\text{Obj}_{\text{post}}^e) \): \( \{ e \in \partial_a(E) \mid (e, o) \in E O \wedge e \leq e_{\text{now}} \} \in \text{card}_a^c(a, oc) \) (eventually each object of \( oc \) refers to a permissible number of events),

The behavioral constraints are respected:
- for each constraint \( c \in C \) and reference event \( e_{\text{ref}} \in \partial_{n_{\text{ref}}}(c)(E) \):
  \( (|\partial_{\text{ref}}|, |\partial_{\text{tar}}|, |\partial_{\text{ref}} \downarrow \partial_{\text{ref}}|) \in \text{type}(c) \) where
  - (\( \text{Obj}, \text{Rel}, \text{class} \) = \( \text{post}(e_{\text{ref}}) \) and
    \( \partial_{\text{ref}} = \{ e_{\text{ref}} \in \partial_{n_{\text{ref}}}(c)(E) \mid \exists o \in \partial_{oc}(\text{Obj}) \} \) \( (e_{\text{ref}}, o), (e_{\text{tar}}, o) \) \( \subseteq EO \) if \( \text{crel}(c) = oc \in OC \),
    \( \partial_{\text{tar}} = \{ e_{\text{tar}} \in \partial_{n_{\text{tar}}}(c)(E) \mid \exists o \in \partial_{oc}(\text{Obj}) \} \) \( \{ (e_{\text{ref}}, o), (e_{\text{tar}}, o) \} \subseteq EO \) if \( \text{crel}(c) = r \in RT \),
    \( \partial_{\text{ref}} = \{ e_{\text{ref}} \in \partial_{n_{\text{ref}}}(c)(E) \mid \exists o \in \partial_{oc}(\text{Obj}) \} \) \( \{ (e_{\text{ref}}, o) \} \subseteq EO \) if \( \text{crel}(c) = r \in RT \).
Event log \( L \) satisfies model OCBCM

- Behavioral constraints are respected.

There are no problems related to the object models observed by the event log:

- for any \( e \in E \): object models \( \text{pre}(e) \) and \( \text{post}(e) \) are valid for \( \text{CIA} \) (Definition 9),
- for any \( e \in E \): \( \text{Obj}^\text{pre}_e \subseteq \text{Obj}^\text{post}_e \) and \( \text{class}^\text{pre}_e \subseteq \text{class}^\text{post}_e \) (objects do not disappear or change class through events),
- for any \( e_1, e_2 \in E \) such that \( e_1 \prec e_2 \): \( \text{Obj}^\text{post}_{e_1} \subseteq \text{Obj}^\text{pre}_{e_2} \) and \( \text{class}^\text{post}_{e_1} \subseteq \text{class}^\text{pre}_{e_2} \) (objects do not disappear or change class in-between events).

The event log refers to things that exist:

- \( \{ \text{act}(e) \mid e \in E \} \subseteq A \) (all activities referred to by events exist in the behavioral model),
- for all \( (e,o) \in E_O \): \( o \in \text{Obj}^\text{post}_e \) (all objects referred to by events exist in the object model after the corresponding event),
- \( \{ (\text{act}(e), \text{class}^\text{post}(o)) \mid (e,o) \in E_O \} \subseteq \text{AOC} \) (events do not refer to objects of unrelated classes).

The cardinality constraints between activities and object classes are respected:

- for any \( (a,oc) \in \text{AOC} \) and \( e_{\text{now}} \in E \):
  - for any \( e \in \partial_a(E) \) with \( e \preceq e_{\text{now}} \): \( \{ o \in \partial_{oc}(\text{Obj}^\text{post}_e) \mid (e,o) \in E_O \} \subseteq \text{card}_{OC}(a,oc) \) (each event \( a \) refers to a permissible number of objects in \( oc \)),
  - for any \( o \in \partial_{oc}(\text{Obj}^\text{post}_e) \): \( \{ e \in \partial_a(E) \mid (e,o) \in E_O \wedge e \preceq e_{\text{now}} \} \subseteq \text{card}_{A}(a,oc) \) (each object class \( oc \) refers to a permissible number of events),
- there exists an \( e_{\text{later}} \in E \) such that for any \( (a,oc) \in \text{AOC} \) and \( e_{\text{now}} \in E \) with \( e_{\text{later}} \preceq e_{\text{now}} \):
  - for any \( e \in \partial_a(E) \) with \( e \preceq e_{\text{now}} \): \( \{ o \in \partial_{oc}(\text{Obj}^\text{post}_e) \mid (e,o) \in E_O \} \subseteq \text{card}_{OC}(a,oc) \) (eventually each event \( a \) refers to a permissible number of objects in \( oc \)),
  - for any \( o \in \partial_{oc}(\text{Obj}^\text{post}_e) \): \( \{ e \in \partial_a(E) \mid (e,o) \in E_O \wedge e \preceq e_{\text{now}} \} \subseteq \text{card}_{A}(a,oc) \) (eventually each object of class \( oc \) refers to a permissible number of events).

The behavioral constraints are respected:

- for each constraint \( c \in C \) and reference event \( e_{\text{ref}} \in \partial_{\text{act}(c)}(E) \):
  - \( |E_{\text{tar}}|, |E_{\text{tar}} \downarrow e_{\text{ref}}| \in \text{type}(c) \) where
- \( (\text{Obj}, \text{Rel}, \text{class}) = \text{post}(e_{\text{ref}}) \) and
- \( E_{\text{tar}} = \{ e_{\text{tar}} \in \partial_{\text{act}(c)}(E) \mid \exists o \in \partial_{oc}(\text{Obj}) \{ (e_{\text{ref}}, o), (e_{\text{tar}}, o) \} \subseteq E_O \} \) if \( \text{crel}(c) = oc \in \text{OC} \),
- \( E_{\text{tar}} = \{ e_{\text{tar}} \in \partial_{\text{act}(c)}(E) \mid \exists r, o_{1}, o_{2} \in \text{Obj} \{ (e_{\text{ref}}, o_{1}), (e_{\text{tar}}, o_{2}) \} \subseteq E_O \} \) if \( \text{crel}(c) = r \in \text{RT} \), or
- \( E_{\text{tar}} = \{ e_{\text{tar}} \in \partial_{\text{act}(c)}(E) \mid \exists r, o_{1}, o_{2} \in \text{Obj} \{ (e_{\text{ref}}, o_{1}), (e_{\text{tar}}, o_{2}) \} \subseteq E_O \} \) if \( \text{crel}(c) = \hat{r} \in \text{RT} \).
Additional Verification Questions

• Is there an event log that satisfies the model?
• Is there an event log that satisfies the model given an initial object model and/or progress requirements?
• Modulo assumptions on changes to the object model ...

Implementation?