Process Ontologies in Action

Michael Grüninger

2nd Workshop on Ontology and Engineering in Tandem

September 20, 2017
Motivating Scenarios

- Construction project management
- Procedures
- Manufacturing process routing
Construction Project Management

- **Issue:**
  - Coordinate knowledge existing across organizations and disciplines

- **Problem:**
  - Modify project schedules based on updated information from contractors and subcontractors.

- **Solution:**
  - Contractors use smartphones to send updates to the project server; determine if these updates are inconsistent with the project schedule, and then communicate the changes to other contractors.
The Hamilton Beach 4200 Series Water Dispenser was identified as a concrete example with which it is possible to extract process specifications similar to those in manufacturing maintenance processes.

The specific focus was the Sanitize process extracted from the use and care manual for this device.
Maintenance Process Descriptions

1. Before cleaning, set both power switches (on the back of the dispenser) to the OFF position and unplug the water dispenser. For first time installation, skip next three steps.

2. Empty the water bottle, then remove it from the dispenser.

3. Drain the water tanks

4. Reinstall the silicon stoppers and drain clips.

5. **DO NOT ADD BLEACH FIRST OR DIRECTLY TO THE DISPENSER.** Concentrated bleach may damage plastic. Add 1/2 teaspoon (2.5 ml) of ordinary 6.0 maximum household bleach to a 2 quart (2 liter) pitcher filled with tap water.

6. Pour the solution in the opening at the top of the dispenser.

7. Repeat last two steps until the dispenser is full.

8. Allow to sit for 15-20 minutes.
Manufacturing Process Routing

- We are interested in dynamic self-routing of objects through the various process plans within the set of manufacturing processes.
- Each product is associated with a set of process plans, which are partially ordered sequences of manufacturing processes.
- In more general scenarios, such process plans may also be nondeterministic (that is, involve different choices of sequences of manufacturing processes).
- Objects “flow” through the sequence of processes. At any point in a process plan, there are multiple activities that can possibly occur next.
- Furthermore, different process plans may have manufacturing processes in common, so that an object may participate in an activity that is part of multiple process plans.
Process Routing

Manufacturing Process Scenario

1. Production order released
2. Translated to PSL & stored on RFID tag
3. Which activities must occur?
4. What's next?
5. What happened?
6. Feeding back the activities that occurred

Confirmation

Enterprise

Plant

Grüninger (Semantic Technologies Lab)
We can use these motivating scenarios and their associated informal competency questions to identify the key ontological commitments that we will need in a process ontology.

These scenarios will also guide us in the specification of the axiomatization of the process ontology.
Activities and Occurrences

Commitment 1
There are four kinds of entities required for reasoning about processes – activities, activity occurrences, timepoints, and objects.

Commitment 2
Activities may have multiple occurrences, or there may exist activities which do not occur at all.
Activity Occurrences and Time

Commitment 3
Activity occurrences and objects are associated with unique timepoints that mark the begin and end of the occurrence or object.

Commitment 4
Timepoints are linearly ordered, forwards into the future, and backwards into the past.
Participation

Commitment 5

Objects may participate in activity occurrences at timepoints during the occurrence.
Composition of Activities

Commitment 6

Activities can be composed together to construct complex activities.
Composition of Activity Occurrences

Commitment 7
Occurrences of complex activities correspond to sets of occurrences of their subactivities.

Commitment 8
Different occurrences of complex activities may contain occurrences of different subactivities or different orderings on the same subactivity occurrences.
Commitment 9

There are different ordering relations on activity occurrences

- an ordering on possible activity occurrences
- linear ordering of subactivity occurrences of a complex activity occurrence
- partial ordering of subactivity occurrences for a set of complex activity occurrences
State

Commitment 10
There are constraints on which activities can possibly occur in some domain (preconditions).

Commitment 11
State is changed by the occurrence of activities, and state can only be changed by the occurrence of activities.

Commitment 12
State does not change during the occurrence of a primitive activity
Intermezzo

- We have identified twelve key ontological commitments related to activities and their occurrences.
- We now turn to the axiomatization of a process ontology that captures these commitments.
PSL Ontology

PSL Core
- http://colore.oor.net/psl_core/

Subactivities
- http://colore.oor.net/psl_subactivity

Occurrence Trees
- http://colore.oor.net/psl_occtree/

Complex Activities
- http://colore.oor.net/psl_complex/

Complex Activity Occurrences
- http://colore.oor.net/psl_actocc/

Subactivity Occurrence Orderings
- http://colore.oor.net/psl_soo/
A process ontology provides the underlying semantics for the process terminology but people use the ontology primarily to specify descriptions of processes as repeatable patterns of behaviour.

A process description for an activity in some class imposes constraints on activity occurrences corresponding to the definition of the class.

Each class of activities in the PSL Ontology is associated with a specific class of sentences that are the correct process descriptions for that class.

The ontology is used to support reasoning with the process descriptions.
State-based Preconditions

Mixing is not performed unless the moulding machine is clean.

\[(\forall o, x) \ occurrence\_of(o, \text{mixing}(x)) \land \text{legal}(o) \supset \text{prior}(\text{clean}(x), o) \quad (1)\]
Time-based Preconditions

The pre-heating operation can only be performed on Tuesday or Thursday.

\[(\forall o, x) \text{ occurrence of } (o, \text{preheat}(x)) \land \text{legal}(o) \supset (\text{beginof}(o) = \text{Tuesday}) \lor (\text{beginof}(o) = \text{Thursday})\] (2)
Occurrence Constraints

If we do not fold the metal after fabrication, we need to reheat it

\[(\forall o_1, x) \ occurrence\_of(o_1, reheat(x)) \land legal(o_1) \supset
\neg(\exists o_2) \ occurrence\_of(o_2, fold(x)) \land earlier(o_2, o_1) \land legal(o_2) \] (3)
Time-based Occurrence Constraints

Drill bits are replaced every 10 minutes.

\[(\forall o_1, x_1) \ occurrence\_of(o_1, replace(x_1)) \land legal(o_1) \supset \]
\[(\exists o_2, x_2) \ occurrence\_of(o_2, replace(x_2)) \land earlier(o_2, o_1) \]
\[\land legal(o_2) \land (duration(beginof(o_2), beginof(o_1)) = 10) \]  \hspace{1cm} (4)
Effects

*If the object is fragile, then it will break when dropped; if the object is elastic, then it will bounce when dropped.*

\[
(\forall o, x) \ occurrence\_of(o, \ drop(x)) \land prior(\ fragile(x), o) \\
\implies holds(broken(x), o)
\]  

(5)
Occurrence-based Effects

If we remove the coffee pot before the brewing activity completes, then the burner will be wet.

is axiomatized by

\[(\forall o_1, o_2, x, y) \text{occurrence}\_of\( (o_1, \text{brew}(x, y)) \wedge \text{occurrence}\_of\( (o_2, \text{remove}(x, y)) \wedge \text{before}(\text{beginof}\( (o_2) \wedge \text{beginof}\( (o_1) \supset \text{holds}(\text{wet}(y), o_1) \quad (6)\)

Grüninger (Semantic Technologies Lab)
Deterministic Process Descriptions

Making the frame consists of cutting, punching, and pressing.

\[(\forall o, x) \ occurrence\_of(o, \ make\_frame(x))\]

\[\supset (\exists o_1, o_2, o_3) \ occurrence\_of(o_1, \ cut(x))\]

\[\land occurrence\_of(o_2, \ punch(x)) \land occurrence\_of(o_3, \ press(x))\] (7)
Fabrication consists of cutting the metal together with either pressing or punching.

\[
(\forall o, x) \ occurrence\_of(o, \ fabricate(x))
\]

\[
\supset (\exists o_1, o_2) \ subactivity\_occurrence(o_1, o) \land \ subactivity\_occurrence(o_2, o)
\]

\[
\land occurrence\_of(o_1, \ cut(x))
\]

\[
\land (occurrence\_of(o_2, \ press(x)) \lor occurrence\_of(o_2, \ punch(x)))
\] (8)
Ordering Constraints

Making the car chassis involves making the body and making the frame in parallel, followed by final assembly.

\[
\left( \forall o, o_1, o_2, o_3, x, y \right) \ occurrence\_of\left( o, make\_chassis(x, y) \right) \ \\
\wedge occurrence\_of\left( o_1, make\_body(y) \right) \wedge occurrence\_of\left( o_2, make\_frame(x) \right) \ \\
\wedge occurrence\_of\left( o_3, final\_assembly(x, y) \right) \ \\
\succeq min\_precedes\left( o_1, o_3, make\_chassis(x, y) \right) \ \\
\wedge min\_precedes\left( o_2, o_3, make\_chassis(x, y) \right) \quad (9)
\]
Conditional Activities

Within the painting activity, if the surface of the product is rough, then sand the product:

\[(\forall s, o_1, x)\text{occurrence}_o(o_1, \text{paint}(x)) \land \text{root}_o(o_1) = s \land (\text{prior}(\text{rough}(x), s))\]

\[\supset (\exists o_2) \text{occurrence}_o(o_2, \text{sand}(x)) \land \text{subactivity}_o(o_2, o_1) \land (\text{root}_o(o_2) = s)\]  \hspace{1cm} (10)
Conditional Orderings

*If the machine is not ready, then perform the painting before final assembly*

\[
(\forall o, o_1, o_2, x, y) \ occurrence\_of(o, \ assembly(x, y)) \\
\land occurrence\_of(o_1, \ paint(x)) \land occurrence\_of(o_2, \ final(x)) \\
\land \neg prior(ready(y), \ root\_occ(o)) \\
\supset min\_precedes(root\_occ(o_1), \ root\_occ(o_2), \ assembly(x)) \quad (11)
\]
Triggered Activities

*Deliver the product when we have received three orders.*
State determines when an activity must occur, so that the process description is written as

\[
(\forall s, x) \ prior(order\_quantity(x, 3), s) \supset
\]

\[
(\exists o) \ occurrence\_of(o, deliver(x)) \land s = root\_occ(o) \tag{12}
\]
Maintenance Procedures

\[(\forall o) \text{occurrence}_o(o, P_1) \supset \]
\[[(\exists o_1, o_2, o_3, o_4) \text{occurrence}_o(o_1, \text{empty}) \]
\[\land \text{occurrence}_o(o_2, \text{drain}) \land \text{occurrence}_o(o_3, \text{reinstall}) \]
\[\land \text{occurrence}_o(o_4, \text{reattach}) \]
\[\land \text{min preceding}_o(o_1, o_2, P_1) \land \text{min preceding}_o(o_2, o_3, P_1) \]
\[\land (\text{min preceding}_o(o_4, o_2, P_1) \equiv \neg \text{min preceding}_o(o_3, o_4, P_1)))] \quad (13)\]
A *no_split* activity is a class of poset activity in which there is no forward branching.

\[(\forall a) \text{ no_split}(a) \equiv \]
\[\left( (\forall s_1, s_2, s_3) \text{ soo_precedes}(s_1, s_2, a) \land \text{ soo_precedes}(s_1, s_3, a) \right.\]
\[\lor \left. \left( \text{ soo_precedes}(s_2, s_3, a) \lor \text{ soo_precedes}(s_3, s_2, a) \lor (s_2 = s_3) \right) \right) \quad (14)\]

A maximum activity is one that contains all process plans as subactivities.

\[(\forall a) \text{ maximum}(a) \equiv (\forall a_1) \text{ activity}(a_1) \supset \text{ subactivity}(a_1, a)) \quad (15)\]
Process Routing Competency Questions

Which subactivities can possibly occur next after an occurrence of $a_1$?

$$(\exists a) \ maximum(a)$$

$$\land (\forall s_1) \ occurrence\_of(s_1, a_1) \supset (\exists a_2, s_2) \ subactivity(a_2, a)$$

$$\land occurrence\_of(s_2, a_2) \land min\_precedes(s_1, s_2, a) \land next\_subocc(s_1, s_2, a))$$

(16)

Which subactivities must occur later after an occurrence of $a_1$?

$$(\exists a) \ maximum(a)$$

$$\land (\forall o, s_1) \ occurrence\_of(s_1, a_1) \land occurrence\_of(o, a)$$

$$\land subactivity\_occurrence(s_1, o)$$

$$\supset (\exists a_2, s_2) \ occurrence\_of(s_2, a_2) \land min\_precedes(s_1, s_2, a)$$

$$\land subactivity\_occurrence(s_2, o))$$

(17)
The Question of Semantics

- How do we know whether or not these process descriptions are correct?
- What are the models of these process descriptions?
Models of PSL-Core

Figure: Example of an occurrence structure (incidence structure for activities, activity occurrences, and timepoints).
Subactivities

Figure: Composition of subactivities.
The semantics of activities must consider what can possibly happen when activities occur.

Furthermore, we need to specify different ordering relations on activity occurrences.

Different branches of a tree correspond to different possible sequences of activity occurrences.
Occurrence Trees

Figure: Example of an occurrence tree.
Complex Activities

- Complex activities are associated with activity trees, which are subtrees of legal occurrence trees.
- All elements of an activity tree are legal occurrences of atomic activities that consists of all possible sequences of atomic subactivity occurrences beginning from a root subactivity occurrence (denoted by the relation $\text{root}(s, a)$).
- Each branch of an activity tree corresponds to a possible sequence of occurrences of subactivities of the complex activity, ordered by the $\text{min\_precedes}$ relation.
- Occurrences of complex activities are associated with branches of the activity trees.
Activity Trees

Figure: Example of a nondeterministic permuted activity tree.
Figure: Example of a nondeterministic permuted activity tree.
Summary

- Find the processes around you
- Go forth and axiomatize!