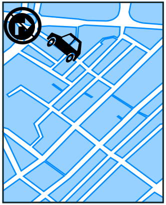
Representing Change Over Time in OWL

Megan Katsumi
Mark S. Fox

Enterprise Integration Lab
University of Toronto Transportation Research Institute

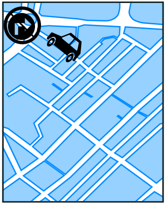
October 21, 2017

A Motivating Scenario: Transportation Network



- Transportation networks contain nodes and arcs.
- Nodes and arcs may be subject to various signals and controls that impact the flow of traffic, e.g. HOV (High Occupancy Vehicle) lanes, turning restrictions).
- The network can be traversed by various things, such as vehicles.
- A vehicle has a make, a vin, a colour, a number of occupants, an owner, a location, and many other properties.

Questions About Transportation Systems



We want to know things like:

- What arcs is this node connected to?
- How many vehicles travelled through these arcs during morning rush hour?
- Can this vehicle turn left at some node right now? (HOV lane, turning restrictions...)
- Where did this vehicle go today?
- ...

Change over time plays a big role!

Existing Work

The traditional representation of fluents is not possible, but there are approaches for capturing change in OWL.

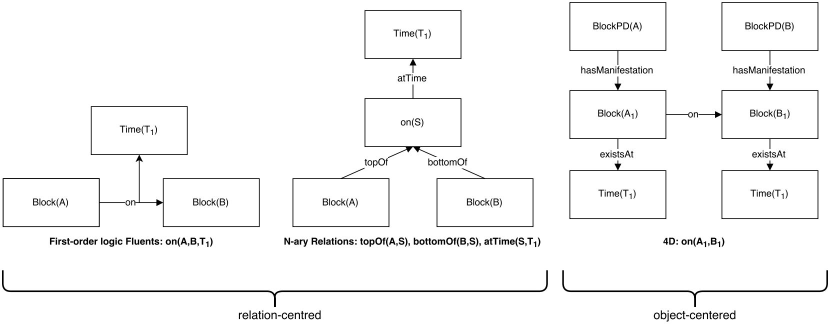


Figure: Approaches to capturing change over time.

- N-ary approach overloads the objects with relations, imposes limitations on what we can express about the fluent relationship.
- We prefer the 4D approach.

A Revised 4D Approach

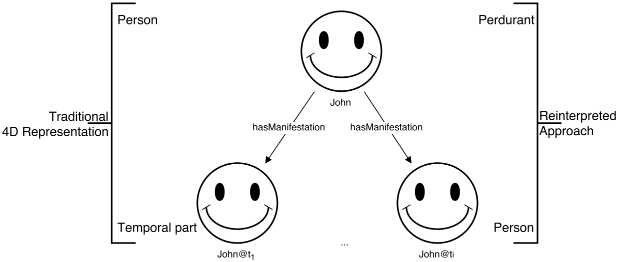


Figure: We adopt the reinterpretation of the 4D approach from[1] due to[2].

[1] Chris Welty, Richard Fikes, and Selene Makarios. "A reusable ontology for fluents in OWL". In: *Formal Ontology in Information Systems (FOIS)*. vol. 150. 2006, pp. 226–236.

[2] Hans-Ulrich Krieger. "Where temporal description logics fail: Representing temporally-changing relationships". In: *Annual Conference on Artificial Intelligence*. Springer. 2008, pp. 249–257.

A Design Pattern to Guide the Solution

- We identified a clear, compact design pattern to guide the capture of change over time in the design of domain ontologies.
- Given some concept that is subject to change (from an existing ontology, or when designing from scratch), the design pattern guides:
  - The import and extension of a simple ontology of change
  - The specification of additional rules

A Simple Ontology for the 4D Approach to Change

A small set of axioms introduces the foundation required to adopt this approach:

- TimeVaryingEntity: the *perdurant* class that exists and changes through time.
- Manifestation: the class of "time-slices" of a perdurant.
- hasManifestation: a TimeVaryingEntity has some manifestation(s).
- existsAt: TimeVaryingEntity and Manifestation objects exist at some interval or instant in time.

Implementing the Pattern

1. Import the Change Ontology
2. Identify time-variant classes
3. Distinguish between the variant and invariant properties for these classes
4. Fill-in the blanks...

The Pattern I

Example

Consider a simple, atemporal representation of a Vehicle:

$$\text{Vehicle} \sqsubseteq =1\text{hasVin.Vin}$$
$$\text{Vehicle} \sqsubseteq \forall\text{hasColour.Colour}$$
$$\text{Vehicle} \sqsubseteq =1\text{hasMake.Manufacturer}$$

The Pattern II

1. Define the concept C as a subclass of Manifestation.

Axiom Type

$\underline{C} \sqsubseteq Manifestation$

$$\text{Vehicle} \sqsubseteq {=}1\text{hasVin.Vin}$$

$$\text{Vehicle} \sqsubseteq \forall\text{hasColour.Colour}$$

$$\text{Vehicle} \sqsubseteq {=}1\text{hasMake.Manufacturer}$$

$$\text{Vehicle} \sqsubseteq \text{Manifestation}$$

The Pattern III

2. Define the perdurant (TimeVaryingEntity) counterpart class for the concept.

Axiom Type

$\underline{PD_C} \sqsubseteq \texttt{TimeVaryingEntity}$

$$\begin{aligned}
\text{Vehicle} &\sqsubseteq {=}1\text{hasVin.Vin} \\
\text{Vehicle} &\sqsubseteq \forall\text{hasColour.Colour} \\
\text{Vehicle} &\sqsubseteq {=}1\text{hasMake.Manufacturer} \\
\text{Vehicle} &\sqsubseteq \text{Manifestation} \\
\text{VehiclePD} &\sqsubseteq \text{TimeVaryingEntity}
\end{aligned}$$

The Pattern IV

3. The Manifestation subclass will have both variant and invariant properties. Include any invariant properties from the Manifestation subclass in the axioms for the TimeVaryingEntity subclass. Where invariantProperty is the invariant property and CE is the class expression:

Axiom Type

$\underline{PD_C} \sqsubseteq \underline{invariantProperty}.\underline{CE}$

(invariant) $\text{Vehicle} \sqsubseteq {=}1\text{hasVin.Vin}$

$\text{Vehicle} \sqsubseteq \forall\text{hasColour.Colour}$

(invariant) $\text{Vehicle} \sqsubseteq {=}1\text{hasMake.Manufacturer}$

$\text{Vehicle} \sqsubseteq \text{Manifestation}$

$\text{VehiclePD} \sqsubseteq \text{TimeVaryingEntity}$

$\text{VehiclePD} \sqsubseteq {=}\ 1\text{hasVin.Vin}$

$\text{VehiclePD} \sqsubseteq {=}\ 1\text{hasMake.ManufacturerPD}$

The Pattern V

4. Restrict the hasManifestation relationship for this new pair of TimeVaryingEntity and Manifestation subclasses.

Axiom Type

$\underline{PD_C} \equiv \exists\ hasManifestation.\underline{C}$
$\sqcap \forall\ hasManifestation.\underline{C}$

Axiom Type

$\underline{C} \equiv \exists\ manifestationOf.\underline{PD_C}$
$\sqcap \forall\ manifestationOf.\underline{PD_C}$

The Pattern VI

$$
\begin{aligned}
\text{Vehicle} &\sqsubseteq {=}1\text{hasVin.Vin} \\
\text{Vehicle} &\sqsubseteq \forall\text{hasColour.Colour} \\
\text{Vehicle} &\sqsubseteq {=}1\text{hasMake.Manufacturer} \\
\text{Vehicle} &\sqsubseteq \text{Manifestation} \\
\text{VehiclePD} &\sqsubseteq \text{TimeVaryingEntity} \\
\text{VehiclePD} &\sqsubseteq = 1\text{hasVin.Vin} \\
\text{VehiclePD} &\sqsubseteq = 1\text{hasMake.ManufacturerPD} \\
\text{VehiclePD} &\equiv \exists\text{hasManifestation.Vehicle} \sqcap \forall\text{hasManifestation.Vehicle} \\
\text{Vehicle} &\equiv \exists\text{manifestationOf.VehiclePD} \sqcap \forall\text{manifestationOf.VehiclePD}
\end{aligned}
$$

Summary

- **Our goal:** simplify the process of incorporating change over time and increase its accessibility for a wider audience of Semantic Web practitioners.
- The design pattern we've proposed:
  - elicits temporal information in an effective, efficient manner.
  - is particularly useful for reusing atemporal ontologies.
  - also specifies a possible extension beyond OWL in order to support reasoning about the relationship between the properties of a TimeVaryingEntity and those of its Manifestations
- See the paper! Come visit the poster!
- We gratefully acknowledge support provided by the Ontario Ministry of Research and Innovation through the ORF-RE program.