On Temporal Cardinality in the Context of the tOWL Language
Outline

- The tOWL Language – Overview
- A Discussion on Temporal Cardinality
- Conclusions
The tOWL Language

- For the current purpose, a clear definition of time is required.

- We distinguish between:
  - Temporal ‘infrastructure’ (timepoints & intervals);
  - Change.

- Providing support for the representation of these aspects of time in a Semantic Web context is the general goal of the tOWL language.
The tOWL Language

Temporal infrastructure

- Describes the quantitative aspect of time
- Provides a basic texture for complex temporal representations
- Common example: intervals + Allen’s relations
- Very concrete

Requirements:
- Rely on standards (we are extending a standard!)
- Represent timepoints and intervals
- Represent temporal constraints
- Level of granularity
The tOWL Language (Change)

Change

- Most entities change some of their traits in time

- Think of:
  - Changing height of a person, from child- to adulthood
  - Changes in the price of a company’s share
  - Changes in variables (fundamental & technical indicators, etc.)

- Representing change = enabling context-awareness

- Context-awareness \(\rightarrow\) better decision-making (though not invariably)

- Think of reasoning over several versions of an OWL-DL ontology (snapshots). In the same time!
The tOWL Language

Change as complex process

- Many phenomena can be described as processes

- Think of:
  - Obtaining a driver’s license
  - Drug trials
  - Leveraged Buy Outs

- A process is described by its states (phases)

- Each process has certain ‘transition rules’ (axioms)

- A proper representation of processes and their associated axioms enables automated reasoning
The tOWL Language

Until now:

- Time is a relevant dimension of knowledge on the Semantic Web
- Two state-of-the-art Semantic Web languages have currently been standardized: RDF & OWL
- Although a (somewhat limited) temporal extension exists for RDF, none has been yet devised for OWL
- We seek to:
  - Extend OWL-DL into a temporal dimension;
  - Enable the representation of quantitative time, as well as change.
The tOWL Language

The tOWL Layer Cake

- Layered approach for the design of the tOWL language;
- The extensions are built on top of the OWL-DL layer;
- Concrete domains enable a meaningful time representation (intervals & Allen’s interval relations);
- The timeslices & fluents approach employs the time representation for the semantics of change.
The tOWL Language

The OWL-DL Layer

- Based on Description Logics (DL)
- OWL-DL offers the means to:
  - formalize a domain by defining classes and properties of those classes,
  - define individuals and assert properties about them, and
  - reason about these classes and individuals to the degree permitted by the formal semantics of the OWL language.
- Tools & Reasoners: Protégé, Pellet, Racer, FaCT++

The OWL-DL Layer Cake

- TimeSlices & Fluents
- Time Representation
- Concrete Domains
- The OWL-DL Layer
The tOWL Language

The Concrete Domains Layer

- OWL-DL has only limited support for concrete domains
- We seek to:
  - Enable feature chains
  - Enable complex temporal restrictions based on the concrete domain (binary predicates)
- Temporal concrete domain = constraint system
  - Intervals and Allen’s 13 interval relations

StockGoodDay $\equiv$ (priceBegin, priceEnd).<
The tOWL Language

The Time Representation Layer

- Constraint system based on intervals and Allen’s 13 interval relations
- We define intervals in terms of their endpoints (start & end)
  \[ \text{Interval} = (\text{start}, \text{end}).< \]
- The endpoints are defined by relying on XML Schema dateTime

Example: In an LBO process, the early stage (may) be followed by the abort stage; in case this happens, the two stages follow each other immediately.

\[ \exists \text{earlyStage} \circ \text{time}, \text{abort} \circ \text{time}).\text{meets} \]
The tOWL Language

The TimeSlices & Fluents Layer

- Represent temporal aspects of entities other than timespan
- This layer regards change and state transitions
- TimeSlice = temporal part of an individual
- Fluent = indicates the changing attribute value
- Two types of fluents:
  - fluentObjectProperty
  - fluentDatatypeProperty

The tOWL Layer Cake
Two timeslices are equal (identical) if the following holds:

\[(TS_1, TS_2).eq_{TS} \equiv (TS_1.time, TS_2.time).equal \land \land (TS_1.timeSliceOf, TS_2.timeSliceOf).sameAs\]

Timeslice representation:
How does a temporal setting influence the OWL-DL constructs?
Cardinality

- OWL-DL implements three constructs for cardinality:
  - minCardinality
  - maxCardinality
  - cardinality

- If stated to have the value $a$ on a property $P$, with respect to a class $C$, then any instance of $C$ will be related through $P$ to at least/at most/exactly $a$ individuals (of which the type may further be restricted by the range of $P$).
Temporal Cardinality

- An extension of the static concept of cardinality may be envisioned in the sense that, at any point in time, only a restricted number of timeslices may describe a concept.

- In other words, temporal cardinality is meant to restrict the number of timeslices that may overlap, at any point in time for the same individual.

- These restrictions should be stated on fluents, with respect to static individuals whose timeslices are described by those fluents.
Temporal Cardinality in tOWL

- Example: represent the fact that, at any point in time, a company must have exactly 1 Chief Executive Officer (CEO)
- Two types of cardinality:
  - fluent cardinality: the (static) cardinality of the hasCEO fluent should be equal to 1
  - overlapping timeslices: the (temporal) cardinality of the hasCEO fluent should be equal to 1
Temporal Cardinality in tOWL

- We define the following temporal equivalents for the static OWL-DL cardinality constructs:
  - temporalMinCardinality
  - temporalMaxCardinality
  - temporalCardinality

**temporalMinCardinality (definition)**

Given a fluent property $f$, a class $C$, an individual $i$ of type $C$ and a value $a$ such that $a$ in $\mathbb{N}$, we represent by $\text{temporalMinCardinality}(f; a)$ the restriction on $f$ with respect to timeslices of $i$ for which $f$ is defined that, at any point in time, any timeslice of $i$ is described by at least $a$ timeslices through $f$. 
Temporal Cardinality in tOWL

- Define a function $g$ that, given a fluent $f$, a static individual $i$ and a point in time $t$, returns the number of timeslices of different individuals $j$ holding at $t$, for which $f$ is explicitly defined and linked from a timeslice of $i$ that also holds at $t$.

$$g_{(f,i,t)} = |\{j \in C^I \mid \exists x, y, s, e \text{ s.t. } x, y \in TS^I \land (x, i) \in \text{timeSliceOf}^I \land (y, j) \in \text{timeSliceOf}^I \land (x, y) \in f^I \land s = \text{start}(\text{time}(y)) \land e = \text{end}(\text{time}(y)) \land s \leq t \leq e\}|$$
The semantics of the three constructs relating to temporal cardinality can be represented as follows, where $a$, $f$ and $t$ preserve their meaning as previously, and $C$ denotes a concept.

\[
(\geq_T a f)_I = \{x \in TS^I \mid \forall i \forall t, i \in C^I \land (x,i) \in \text{timeSliceOf}^I \land g_{(f,i,t)} \geq a\}
\]

\[
(\leq_T a f)_I = \{x \in TS^I \mid \forall i \forall t, i \in C^I \land (x,i) \in \text{timeSliceOf}^I \land g_{(f,i,t)} \leq a\}
\]

\[
(=^T a f)_I = (\geq_T a f)_I \cap (\leq_T a f)_I
\]
Conclusions

- The tOWL language is a temporal ontology language built on top of OWL-DL

- tOWL enables the representation of different aspects of change in the language, based on a clearly defined temporal infrastructure

- Temporal cardinality in tOWL is closely related to the concept of timeslices

- In a temporal setting, we seek to represent restrictions on the number of overlapping timeslices
Questions