Evolution of Ontology in Time

Andrzej Bassara
Department of Management Information Systems
The Poznań University of Economics
A.Bassara@kie.ae.poznan.pl

Abstract

This paper presents an overview of a solution for maintaining the knowledge on evolution of ontology in time. Firstly, an overview of requirements for the solution is summarized. Then, theoretical background for temporal reasoning is provided. Finally, based on identified requirements and theoretical information, methods for incorporating temporal information into ontology are discussed.

1. Introduction

Ontologies, as technology, are used to encode knowledge about world, mainly about a specific domain, and to provide common understanding of these domains. In other words they are "explicit specification of a conceptualization" [Gruber 1993]. The word conceptualization refers to the way a certain agent or a group of agents percepts certain idea.

From the above definition at least three important sources of ontologies dynamism may be identified. Change in each of them may lead to significant changes in ontology:

- domain (external),
- conceptalization (internal),
- requirements (technical).

Domains, that are modeled are often dynamic, therefore are subject to permanent changes: new object appears, existing objects case to exist, properties of existing object are changing. [Jensen 2000] highlights following classes of applications where temporal extension is necessary:

- financial applications (portfolio management, accounting, banking)
- record-keeping applications (HR, medical-record, inventory)
- scheduling applications (airline, train, hotel reservations)
- project management
- scientific applications

The need for change may also arise not from the nature of domain but from the nature of human. In this case change in ontology is determined by the change in conceptualization (change in the way that given domain is percept). Let us consider a concept "person in productive age". The extension of this concept depends on the definition which is changing over time and is different among various countries.

The least important need for change is related to changing requirements. In this case ontologies are changed to improve system performance and reliability, or just to cover permanently changing users needs.

One may be tempted by the idea of disallowing changes. The change, however, is not to be discouraged while it is inevitable. Many software engineering (RUP [Kruchten 2003], Agile Development [Boehm & Turner 2005]) or infrastructure management (ITIL [ITIL 2000]) communities have already realized that the only solution to this problem of change is to provide formalized processes for change capture, change propagation as well as processes for releasing updated versions of a product.

Since it is unavoidable to have changes during the ontology lifecycle, the question that also must be raised is, whether it is possible to have only one conceptualization of a given domain at a given point of time. The answer to this question comes not easily and is depended on many factors, mainly the domain and ontology usage patterns being considered.

For instance, the ontology of bibliographic information used to structure books in a library may be created once for quite a long time and may be easily used by most members of the community, while the librarianship has a long lasting history and different taxonomies has been created for ages.

Contrary the ontology of content in P2P networks may be quite dynamic. This dynamism arise from the variety of content being shared. It may be even impossible to come to an agreement on one common ontology, while the community consist of users from different cultures and with different backgrounds. Or the ontology of legislature used to compare legal system of different countries over time. In this case, changes to ontology will be applied according to geographic region or time range being considered.

Summarizing, it may be stated that some information enclosed in ontology are valid only within certain scope (context). This context may be characterized by many dimensions. Above example of "*person in productive age*" is dependent on time and location but many more dimensions of context may be introduced. The rest of this paper will utilize time as an example dimension, which influences the overall ontology shape.

2. Requirements

The presented solution is part of the Black Ocean project, which aims at discovering relations between business entities. In summary, a knowledge base (KB) is constructed. This KB is permanently supplied with information on business entities. Then the filtering mechanism is used for discovering new documents that may contribute to the KB. If such a document is found, new information is extracted and added to KB. Much attention is paid to the determination of a time scope of newly added information. For instance, when it is discovered that person A works for company B, it should be stated when such a statement is true.

This approach implies that data (a.k.a. ABox) is temporal and is valid only in specific periods. The KB also learns how to structure data in new ways. If, for instance, the law has changed there may be a need for restructuring the ontology. Therefore, also the schema (a.k.a. TBox, a.k.a. Terminology) is subject to change and is valid in its own periods. The concept *"a person in productive age"* is still a good example.

For the purpose of the project, OWL [Bechhofer et al 2004] has been selected for the implementation. Therefore, a clarification is needed, while distinction between data and the schema is somehow sparse. We follow convention that data includes statements about individuals and relations that hold between them, everything else is enclosed in schema (the project follows OWL-DL as strictly as it is only possible, therefore sets of: classes, individuals and relations must be mutually disjoint).

From a general analysis of the problem a number of requirements for ontology temporality have been identified. Although, they are project specific, they may be applied to the vide range of other problems that involve ontology evolution in time. These requirements are related to:

- Statements temporality
- Time references
- Ontology snapshots

2.1. Statements temporality

For the Black Ocean it is infeasible to have an ontology which consist of statements (facts) that are unconditionally true, which are valid at any point in time. It is necessary to have a mechanism which would allow to store in a single OWL document different perspectives on the considered domain – in this case time constitutes the perspective. Since, it is a rear case that the whole ontology is changed at once at particular point in time but the process of change is gradual, the temporality should be implemented at lower level of granularity then the whole document.

We consider a single statement (i.e. *ex:subject ex: predicate ex: object*, which may be translated into English as *subject is in relation predicate with object*) as an element which may be consider to exist in time. Basically, RDF [Hayes 2004] statements are considered to be unconditionally true, therefore there is a need to incorporate mechanism, which would allow to state that particular statement is valid only during a certain time period.

As noted before it should be considered that both data as well as the schema are temporal. In case of data the situation is rather simple. The Data Warehousing (DW) paradigm may be followed. In this case, data is never removed from DW. Each entry to the DW is described with a set of DW's dimensions, which describes precisely the situation (context) in which the data is true - for instance, to which period this data applies. The same approach may be used to describe data within OWL. Data temporality refers to the properties of individuals and existence of individuals, both of which are valid only in time ranges.

The more challenging aspect of temporality refers to the temporality of the schema. The schema (terminology) provides means for structuring data, therefore any change in schema may possibly influence ABox, and this influence is discussed in further sections.

2.2. Time references

Much of information available today on the Internet, which is a main source of information for the KB, is uncertain and ambiguous. Very often events (see below) that are described in news feeds do not have precisely defined time boundaries – exact start and end dates are unknown. However there usually exist many constraints that relate described events to one another, and which are sufficient for meaningful description.

Example: *"Parts of the nation's midsection are cleaning up today after a strong storm system downed power lines…" cnn.com*

There are two events described in the above sentence: *event A – cleaning of the nation's midsection* and *event B – storm, which downed dower lines*. No precise definition of temporal boundaries of *event B* are provided, but there exists constraint that states that *event B* took place before *event A*. And we also know that *event A* took place today with respect to information issuing date.

The designed mechanism should allow for the usage of both precisely defined time boundaries for events (absolute references) as well as relatively defined boundaries by relating events to each other (relative references).

2.3. Ontology snapshots

The Black Ocean KB contains all data and terminology that has been put in it since the creation. That implies that the ontology consists of various very often contradictory facts, which are not suitable for direct processing. It is however assumed that at any given time there exist, so called, ontology snapshot, which is an ontology for this particular point in time and that this ontology is valid OWL-DL ontology (snapshot shall be free of any temporal extensions).

One should be also able to track changes that appear in the ontology over time. To accomplish this task presented approach should provide infrastructure for computing and presenting in formalized manner the difference between two different snapshots. This difference should consist of information that would allow for transformation from older to newer snapshot.

3. Temporal theories

One of the most fundamental questions arising from the requirement that statements are valid in time is - how to encode temporal information? The project itself is not about the development of new temporal theory, therefore an existing theory, which has proved to be sufficient, shall be utilized.

There exist two theories that may easily be applied to the presented problem. Theory of intervals [Allen 1983] and theory of semi-intervals [Freksa 1992]. Both theories use the notion of interval as a central point. Intervals are percept as some extend of time within some event occurs. Each interval has its beginning and ending (i.e. boundaries). Both theories allow for relating intervals to one another based on relations between their boundaries.

Assuming that a boundary of an interval is an instant, which is a point in time and that the time is linear then there are only three possible relations between any two boundaries: they may be equal (happening on the same time); one may happen before another or vice versa.

With this assumption one may define relations between intervals as a function of relations between their boundaries. For instance, if *interval A* starts and ends between *interval B* starts and ends then, according to Allen theory of intervals, *interval A* is before *interval B*.

The major difference between both theories is that theory of semi-intervals does not require the full knowledge abut their boundaries in order to relate one to another. The above example requires knowledge on beginnings and endings of both intervals. But actually, according to human perception, it is sufficient to know that ending of first event is before beginning of other event, to state that one event is before the other one. This is a central assumption that stands behind the theory of semi-intervals.

In fact, both theories have similar capabilities (expressiveness) and computational complexity. Therefore, although theory of semi-intervals may seem to be more appropriate for uncertain information other criteria has been used to select theory.

Temporal information, with respect to presented requirements, must be incorporated into RDF documents in a way that is easily processable by contemporary tools for RDF processing. Therefore the theory of choice needs to be translated into OWL ontology. Fortunately, there exist an ontology of time called OWL-Time [Hobbs & Pan 2004], which is an implementation of Allen's temporal theories in OWL. As both theories are claimed to be comparable, it has been decided that the existing ontology will be reused for the project purposes and that the Allen's theory of intervals will serve as a backbone for the temporal framework.

3.1. Theory of intervals

As described before, the central point of Allen's theory are intervals and relations between them. Intervals are required to have beginning and ending (boundaries) defined by instants (i.e. points in time). Boundaries may be defined in one of following ways:

- Interval may have precisely defined boundary, instant which points into given date
- Interval may have undefined boundary, instant which does not point into any specific date. This approach is highly usable while often it is known that interval has ending/beginning but the precise date is not known. However it may be stated that the ending/beginning of this interval is related to other instant/interval. For instance two different intervals share ending.
- Interval may have no boundary; an interval may have no ending, or no beginning or even both.

Such defined intervals may be now related to each other. The relation between two intervals is based on relations between their boundaries. Following relations may be defined:

- Before (inverse of After)
- Equal
- Meets (inverse of MetBy)
- Overlaps (inverse of OverlappedBy)
- During (inverse of Contains)
- Starts (inverse of StartedBy)
- Finishes (inverse of FinishedBy)

For the precise definition of these relations please refer to [Allen 1983] [Hobbs & Pan 2004].

```
<te:IntervalEvent rdf:ID="meeting1">
  <te:begins rdf:resource="#meeting1Start" />
</te:IntervalEvent>

<te:IntervalEvent rdf:ID="meeting2">
  <te:begins rdf:resource="#meeting2Start" />
    <te:durationDescriptionDataType rdf:datatype="&xsd;duration">
      PT45M
    </te:durationDescriptionDataType>
</te:IntervalEvent>

<te:Instant rdf:ID="meeting1Start">
  <te:inCalendarClockDataType rdf:datatype="&xsd;dateTime">
    2003-11-05T18:00:00-5:0
  </te:inCalendarClockDataType>
</te:Instant>

<te:Instant rdf:ID="meeting2Start">
    <te:inCalendarClockDataType rdf:datatype="&xsd;dateTime">
      2003-11-05T14:00:00-8:00
    </te:inCalendarClockDataType>
</te:Instant>
```

Above example is a test case from OWL-Time which shows how one may describe events. In this case there exist two events *meetint1* and *meeting2*. *Meeting1* starts at 2003-11-05T18:00:00-5:0, and *meeting2* starts at 2003-11-

05T14:00:00-8:00 and it is also said that *meeting2* last for 45 minutes. The question from this task case is whether both events collide. To answer this question a proper temporal reasoner is required.

3.2. Temporal reasoning

With such a rich notation one may define intervals in different ways: either by providing precise boundaries for interval which describes the interval or by relating this interval to other intervals or instants or just by providing duration of interval. However regardless of how intervals are described the infrastructure should be ale to answer the question of how each of these intervals relates to any date, any interval or any instant.

Unfortunately, there is no available free for academic use temporal reasoner that can deal with OWL-Time ontology. Therefore one had to be created. As the functionality of reasoner may be, for the sake of simplicity, defined as: derive all relations between any two intervals/instants that may be derived based on available input information, the reasoner may be then implemented as a set of "if-then" rules.

Rules that have been implemented may be classified in following way:

- Duration rules – new instants are estimated based on supplied duration information. Example: if one of the boundaries and the duration is know, then it is possible to define second boundary.
- Instant rules – based on exact dates and relations between instants, instants are related to one another. Example: if dates of two instants are known, then it is possible to define all relations that hold between these instants.
- Interval rules – based on information on instants and relations between intervals, intervals are related to one another. Example: if all boundaries of two intervals are known, then it is possible to define all relations that hold between these intervals.
- Boundaries propagation rules – based on relation between intervals, instants are related to one another. Example: If the relation between two intervals is known, then it is possible to define relations between their boundaries.

As the Black Ocean project uses Jena API[1] a Generic Rules Reasoner from Jena framework has been utilized to implement these rules. Complete temporal reasoner consists of around 70 rules each of following form:

```
[intMeets:
  (?T1 tm:intMeets ?T2) <- (?T1 rdf:type tm:ProperInterval),
  (?T2 rdf:type tm:ProperInterval), (?T1 tm:ends ?t1end),
  (?T2 tm:begins ?t2begin), (?t1end tm:equalInstants ?t2begin)]
```

[1] http://jena.sourceforge.net/documentation.html

The above rule states that if there exist two intervals *T1* and *T2* and *T1* ends with *instant t1end* and *T2* begins with instant *t2begin* and both instants are equal then interval *T1* meets interval *T1*.

Applying the set of rules the temporal reasoner should:

- Define missing boundary for *meeting2*
- Relate existing instants to one another
- Based on available information conclude that *meeting1* takes place after *meeting* 2 (they are not overlapping)

In order to optimize the reasoner in terms of required memory, one may use instead forward chaining rules, backward changing rules which are allowed in Jena's Generic Rules Reasoner. However this approach does not introduce any significant gain in terms of processing speed.

4. Temporal extension to ontology

With such a tool as time ontology and temporal reasoner it is possible to define that certain entities exist in time or certain events happens in time. This mechanism is also sufficient to define statement existence in time. It is said that statement is valid in time if it happens to be true during this time interval. For instance we may say that statement "*person A works at company B"* is valid only from year 2000 to year 2005 what happens to be the time of the employment.

Having in mind one of requirements that ontology must be processable by existing tools for RDF processing, the project should not introduce any specific extensions to RDF model. In this case the most natural way, which allows making statements about statements (which allows saying that statement is valid within time interval) is to use RDF reification.

In general, reification it is a mechanism which allows treating an abstraction like a concrete thing. In case of RDF, reification allows treating RDF statements as any other resource. Going back to example of *person A*, who works at *company B*. This statement may be encoded as *ex:A ex:worksAt ex:B* (using N3 notation). It is not possible to make statements directly about this statement. Therefore, this statement must be reified, which can be formalized as:

```
ex:r1 rdf:type rdf:Statement;
        rdf:subject ex:A;
        rdf:predicate ex:worksAt;
        rdf:object ex:B.
```

This way, it is now possible to make statements about this statement just referring to it by *ex:r1* as to any other resource in RDF. In the considered framework an object property *validWithin* has been introduced which allows to say that a given statement is valid within a given interval.

At least two different semantics may be applied to this property, which have quite different implication during snapshot construction process. One definition of this property is based on open world assumption. In this case *validWithin* defines that the statement must be valid within this period but nothing is said about the rest of the time. References to and from other statements may imply that this statement is also true in other intervals. This problem is analyzed in more detail in next section.

Second interpretation uses closed word assumption. In order to highlight the difference the property may be named *validOnlyWithin.* The semantic of this property enforces that the property is valid only within defined period (not in any other). The validity period of this statement may be further constrained by references from other statements. Let us consider statement *"person A works at company B"* (*ex:A ex:worksAt ex:B*), which is valid since 2000 to 2005. But we also have statement that "*the person A exist*" only since ever to 2004 (*ex:A rdf:type ex:Person*). According to this semantics the statement "*person A works at company B"* is valid only since 2000 to 2004, while this is constrained by the existence of individual A.

As the second approach is against the open word assumption, which is one of fundaments behind OWL-DL, next paragraph focuses only on open world approach.

5. Ontology snapshots

An ontology constructed with presented structures is inadequate for direct usage. It contains very often contradictory facts, traditional tools for RDF processing are not aware of semantics defined by *validWithin* property, moreover traditional tools does not process temporal data. Therefore, presented infrastructure introduces the notion of ontology snapshots.

Ontology snapshot is an ontology extracted from temporal ontology which contains only statements that are valid for certain time (time that is requested). It is crucial that this ontology is free from temporal extensions and statements that have been reified are now presented directly in RDF graph.

The snapshot is just an ontology which consists of all statements that are valid for given point in time (or interval). Let us consider an ontology which consists of following statements:

```
ex:A ex:worksAt ex:B. [2004,2005]
ex:A rdf:type ex:Person. [1980,2003]
ex:Person rdf:type owl:Class. [ever]
ex:worksAt rdf:type owl:ObjectProperty. [ever]
ex:worksAt rdfs:domain ex:Person. [ever]
```

In order to take snapshot, one should iterate through all statements and select only these, whose boundaries cover the date of consideration. For instance for 2004, the list of relevant statements is as follows:

```
ex:A ex:worksAt ex:B.
ex:Person rdf:type owl:Class.
ex:worksAt rdf:type owl:ObjectProperty.
ex:worksAt rdfs:domain ex:Person.
```

Using the open word assumption that boundaries does not define the existence of concept exclusively and following the OWL-DL semantics it may be inferred from the above set of statements that if *A* works at *B* and domain of *worksAt* is *Person* then *A* is of type *Person*. So, the statement that has been previously removed is again added to the ontology, which finally has the form of:

```
ex:A ex:worksAt ex:B.
ex:A rdf:type ex:Person.
ex:Person rdf:type owl:Class.
ex:worksAt rdf:type owl:ObjectProperty.
ex:worksAt rdfs:domain ex:Person.
```

6. Conclusions

This approach presents means for incorporating temporal information into ontology. It may be now stated that individuals, properties of individuals as well as terminology used to describe individuals exist and can change in time. In presented examples no distinction has been made on what kind of statements are being described. This approach is general enough to provide temporality of data as well as temporality of schema.

Although the solution does not deal with aspects of undecidability of some of temporal information (it is possible to relate intervals for which boundaries are unknown), the solution is complete and provide simple means for maintaining different version of single ontology in one document.

7. References

[Allen 1983] J. E. Allen, "Maintaining knowledge about temporal intervals", *Commun. ACM 26, 11* November 1983, pp 832-843.

[Bechhofer et al 2004] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, L. A. Stein, "OWL Web Ontology Language Reference", *W3C Recommendation,* 10 February 2004

[Boehm & Turner 2005] Boehm B., Turner R., "Management Challenges to Implementing Agile Processes in Traditional Development Organizations", *IEEE Software,* September/October 2005

[Freksa 1992] C. Freksa, "Temporal Reasoning Based on Semi-Intervals", *Artificial Intelligence, vol. 4, issue 1,* 1992, pp 1992-227

[Gruber 1993] T. Gruber, "A translation approach to portable ontology specifications", *Knowledge Acquisition, vol. 5, issue 2,* June 1993, pp 199-220

[ITIL 2000] Office of Government Commerce, "ITIL Service Support", *Stationery Office*, Jun 2000

[Jensen 2000] C. Jensen, "Temporal Database Management", *dr.techn. thesis,* April 2000

[Kruchten 2003] P. Kruchten, "The Rational Unified Process: an Introduction", *Addison-Wesley Longman Publishing Co., Inc*.. 2003

[Hayes 2004] P. Hayes, "RDF Semantics", *W3C Recommendation*, 10 February 2004

[Hobbs & Pan 2004] J. R. Hobbs, F. Pan. 2004, „An Ontology of Time for the Semantic Web", *ACM Transactions on Asian Language Processing (TALIP): Special issue on Temporal Information Processing, Vol. 3, No. 1,* March 2004, pp. 66-85.