Temporality in Semantic Web

Ph.D student: Di Wu, Graduate Center, CUNY
Mentor: Abdullah Uz Tansel, Baruch College, CUNY
Committee: Sarah Zelikovitz, CIS, CUNY
Susan P. Imberman, CIS, CUNY
Abstract

• Semantic Web
• Semantic Web Technologies
• Time Ontology
• Temporal Database
• Temporal Relational Algebra
• Research Direction
Motivation

- Data
  - Centralized
  - Distributed
  - Decentralized

- More data
  - Semantic Web

- Better data
  - Accurate
  - Up to Date
  - Historical data
Semantic Web

• Definition is developed in 1960s.
  • an extension of the current WWW.
  • information is well-defined with meaning.
  • enable **computers** and people to understand.

• Data is saved over the WWW:
  • Row sharing: entity-based view
  • Column sharing: attribute-based view
  • Cell sharing: Semantic Web, RDF view
Semantic Web - Success

• Over 2.5 billion webpages have markup conforming to the schema.org format;

• Linked data is widely adopted by major libraries and museums as well as media sites such as BBC and New York Times;

• Web companies are developing knowledge graphs that link massive number of entities together such as Google, Bing, and Facebook;

• Commercial database systems are providing native support for Semantic Web languages such as Oracle;

• Recommender companies are taking advantage of semantics and tagging to improve their accuracy;

• The world health organization is developing the main international terminology for diseases as an ontology for semantic web users; etc.
Semantic Web - Challenges

• The development is still below the expectations of the founders of the Semantic Web.

• We still have few means to easily generate semantic web annotations.

• It is challenging that how to effectively query huge number of decentralized information repositories of varying scales.

• The research of Semantic Web has transitioned into larger, more applied systems.

• Although current Semantic Web implementations support some temporal elements, however, it is not enough to solve the problem “Finding accurate current data”.

Ontology

• An ontology is an explicit specification of a conceptualization - the terms in the domain and the relationship among them.

• It defines a commonly accepted vocabulary for a domain for information sharing - not only human beings but also machine agents.

• The main requirements for a good ontology include well-defined syntax, efficient support for reasoning, a clear and formal semantics, sufficient power of expression, and easy to express.

• A good ontology for Time is critical and fundamental.
Ontology Guide

• There are 7 steps to develop an ontology:
  • Step1: Domain and scope
  • Step2: Reuse existing ontologies
  • Step3: Defining important terms
  • Step4: Defining classes and class hierarchy
  • Step5: Defining properties of concepts
  • Step6: Defining facets of properties
  • Step7: Fill value to create an instance
Description Logic

• Description Logic is widely used in ontological modeling. Description Logic Ontologies consist of a set of statements, which are called axioms that must be true.

• Assertional axioms (ABox)
  • Represents attributes or facts of the object

• Terminological axioms (TBox)
  • Represent relationships between concepts

• Relational axioms (RBox)
  • Represent the property of roles
## DL Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Example</th>
<th>Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊤</td>
<td>⊤ is a special concept with every individual as an instance</td>
<td>⊤</td>
<td>top</td>
</tr>
<tr>
<td>⊥</td>
<td>empty concept</td>
<td>⊥</td>
<td>bottom</td>
</tr>
<tr>
<td>□</td>
<td>intersection or conjunction of concepts</td>
<td>C □ D</td>
<td>C and D</td>
</tr>
<tr>
<td>△</td>
<td>union or disjunction of concepts</td>
<td>C △ D</td>
<td>C or D</td>
</tr>
<tr>
<td>⊥</td>
<td>negation or complement of concepts</td>
<td>¬C</td>
<td>not C</td>
</tr>
<tr>
<td>∀</td>
<td>universal restriction</td>
<td>∀R. C</td>
<td>all R-successors are in C</td>
</tr>
<tr>
<td>∃</td>
<td>existential restriction</td>
<td>∃R. C</td>
<td>an R-successor exists in C</td>
</tr>
<tr>
<td>⊑</td>
<td>Concept inclusion</td>
<td>C ⊑ D</td>
<td>all C are D</td>
</tr>
<tr>
<td>⊑</td>
<td>Concept equivalence</td>
<td>C ⊑ D</td>
<td>C is equivalent to D</td>
</tr>
<tr>
<td>⊑</td>
<td>Concept definition</td>
<td>C ⊑ D</td>
<td>C is defined to be equal to D</td>
</tr>
<tr>
<td>:</td>
<td>Concept assertion</td>
<td>a : C</td>
<td>a is a C</td>
</tr>
<tr>
<td>:</td>
<td>Role assertion</td>
<td>(a, b) : R</td>
<td>a is R-related to b</td>
</tr>
</tbody>
</table>
Resource Description Framework

• Traditional Relational Database represents information either by Column (for specific topics) or by Row (for some entities).

• RDF makes information represented by Cell (a piece of information) possible.

• The RDF data model is based a set of statements, which are subject-predicate-object triples.
  • The subject in the triple represents a resource.
  • The predicate represents the relationship between the subject and the object.
  • The object represents attributes or properties of the resource.
RDF-Statement

William Weber \(\xrightarrow{\text{livesIn}}\) New York City

http://www.example.org/WilliamWeber
http://www.example.org/NewYorkCity

http://www.example.org/FullName

William Weber
Pseudo code:
<rdf:Description rdf:about="Subject">
    <Predicate rdf:resources="Object"></Predicate>
    <Predicate>Literal</Predicate>
</rdf:Description>

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:ex="http://example.org/">
    <rdf:Description rdf:about="http://www.example.org/WWeber">
        <ex:livesIn rdf:resource="http://www.example.org/NewYorkCity"></ex:livesIn>
        <ex:fullName>William Weber</ex:fullName>
    </rdf:Description>
</rdf:RDF>
RDFa

• RDFa (Resource Description Framework in Attributes) is a W3C Recommendation that adds a set of attribute level extensions to HTML and various XML-based document types for embedding metadata.

```html
<p author ="http://www.example.org/people" >William Weber
  <span property = "http://www.example.org/livesIn"> lives in</span>
  <span location = "http://www.example.org/location"> New York City</span>
</p>
```
Turtle

• A turtle document enables representing a RDF graph in a compact textual format.

• Beside the namespace definitions as we have seen in RDF, turtle is very compact because of the use of column to list predicates, and coma to list objects.
N-Triple

• N-Triples corresponds directly to the raw RDF triples and uses fully unabbreviated URIs. The subject, predicate and object of an RDF Triple are represented in a sequence and separated by white space. This sequence is terminated by a ’.’ and a new line.

<http://one.example/subject1> <http://one.example/predicate1> <http://one.example/object1> . # comments here
# or on a line by themselves
_:subject1 <http://an.example/predicate1> "object1" .
_:subject2 <http://an.example/predicate2> "object2" .
Notation 3

• Notation 3 (N3) is a more compact serialization of RDF. Namespaces are used to abbreviate the statements. Thus, N3 provides a compact representation. Similar to Turtle, ‘;’ is used to indicate another triple with the same subject, and ‘,’ is used to separate objects.
Reification

• Reification, is using a built-in vocabulary in RDF for describing statements so that we know more than the merely binary relationship, or, in other words, is making statements about other statements.

• In practice, Reification always uses four statements for describing statements, and thus is called “reification quad”.

• The standard reification has been criticized for its lack of efficiency:
  • Singleton Property: creates new property that embeds both the original and additional property
  • N-ary relations: creates an intermediate resource to denote the relationship.
Web Ontology Language

• OWL is a logic-based language, and machine agents are able to exploit knowledge expressed in OWL.

• The W3C Web Ontology Language is also a Semantic Web language. It is designed to express rich information.

• OWL 2 as the most current version was released in 2012 by W3C recommendation.

• The OWL 2 ontology provides structures as classes, properties, individuals, and data values, and are stored as Semantic Web documents.
Linked Data Initiative

• Linked Data Initiative is part of the Semantic Web in the way that resources over the web of hypertext can be linked so human and machine agents can reach out to more data if they have some.

• The four principles of linked data as Tim Berners-Lee proposed are:
  1) Use URIs as names for things;
  2) Use HTTP URIs so that people can look up those names;
  3) When someone looks up a URI, provide useful information, using the standards RDF, SPARQL);
  4) Include links to other URIs so that they can discover more things.
Dublin Core Metadata Initiative

• The Dublin Core Metadata Initiative is one of RDF vocabulary that has a metadata element and in the beginning it provides properties such as creator, publisher, and title, etc.

• This original 15 metadata elements are defined even before RDF so that they have a separate namespace.

• Nowadays DCMI is one of the most popular RDF vocabularies and it is inline with the Linked Data movement.

• Simple Dublin Core has 24 elements while Qualified Dublin Core has 3 additional elements.
FOAF

• As FOAF stands for Friend of a Friend, FOAF uses RDF technology to connect information on a person’s homepage with that of his/her friends, and the friends of his/her friends.

• FOAF enables machines to understand a page, and learn relations that connect people, places, and things on the web.

• FOAF can greatly expand the knowledge base of web as a whole.

• FOAF is very easy to use and there are various FOAF tools for generating FOAF files such as FOAF site, foaf-a-matic, etc
Google Knowledge Graph

• The Knowledge Graph is a knowledge base of Google to improve its search engine and enhance the result by adding a display of structured and detailed information besides the traditional list of links to other sites.

• The Knowledge Graph provides a convenient way for Google’s search engine users that they don’t need to go further for information if they are satisfied with the Knowledge Graph display.

• Google Now exploits the result of the Knowledge Graph and uses the short summary for the keywords/topic as feedback to its users.

• The Knowledge Graph takes a semantic search from various sources include Wikidata and Wikipedia etc.
Schema.org

• Schema.org was created in 2011 by major search engines Bing, Google, and Yahoo.

• It aims at providing a single schema that covers a wide range of topics. While webmasters only need to with the markup once, search engines can use the markup differently.

• The idea of Schema.org actually align with RDF that a piece of information is represented as a triple by data provides and how to retrieve the information is the work of query developers.
Linked Open Vocabularies

• Based on the view of vocabularies also data, the name Linked Open Vocabularies is derived from Linked Open Data.

• LOV provides a choice of many vocabularies based on a set of criteria such as URI stability and availability, proper version, etc.

• Most of the established vocabularies are collected in LOV, as well as their version history and relationships among each other.

• LOV provides a tool for webmasters to choose best vocabulary with less considerations.
Time ontology

• Time is one of the most important concept in most scientific fields. Many cultures regard time and space as the fundamental scale of the world.

• TemporalEntity
  • TemporalRule
  • TemporalConcepts
  • TemporalFeature
  • TemporalProperMeasure
Time ontology

• A temporal element is the primitive entity that is used to represent time. These elements for a theory of time are Time Points and Time Intervals.

• Time Points are the elements of a time line which can also be called instants. Time point (instant) t has no duration. The value of t differs in the specific programs. Some assign non negative values to an instant, some assign negative value to represent past and positive to represent future.

• Time Intervals are used to extend the point-based model of time to represent a period or a part of the time. Also, the relationship “during" is associated with a duration, which cannot be time points, but be an interval. Some theories argue that time interval is a compound structure of time points.
Temporal Properties

• Dates and Timestamps both are widely used in programs.
  • Dates are ordered triples of integers for year, month, and day.
  • Timestamps could be interpreted as a metric reference to a point in the time line. Timestamps can contain precise information and the preciseness is depended on the system.
Temporal Relations

- Temporal Relations are the binary properties among elements of a time model. Depends on the elements in the relationship, there are three categories: point to point, point to interval, and interval to interval.
Temporal database

• A Valid Time is a time period during which the statement / record is true with respect to the real world. The temporal attributes for valid time are independent of the transaction time in the database; they are reflecting the time in real world.

• A Transaction Time is a time period during which the statement / record in the database is considered to be true. It is associated with the time that the statement is added to the database and is assumed to be true till now.
Historical Database

• Historical database only supports valid time. Thus, it can be regarded as classical relational database with a new temporal attribute represents the valid time.

• Historical database represents the full life-span of an entity not just current state. We can have data already expired (past state) and will be valid in the future (future state)
Rollback Database

• Rollback database is the another type of database. It is different than the historical database and only supports transaction time.
• Rollback database keeps all records with their timestamps when they are entered.
• Rollback database is ideally for data recovery from a failure.
Bi-temporal Database

- Bi-temporal database is the most used temporal database in reality. This database supports both types of time.
Tuple Time-stamping

• There are two approaches for time-stamping in DBMS. One is tuple time stamping, the other is attribute time-stamping.

• Tuple time stamping approach keeps the 1NF relations, and adds timestamps to each tuple in a relation.
  • Each time we update an attribute associated with time, we have to insert a new tuple into the table, which will produce a lot of redundant information.
  • The advantage of tuple time stamping is minimum change needed for existing relation database.
Attribute Time-stamping

• Attribute time stamping uses a more complex way to store the temporal elements.
  • When an attribute has a temporal elements, it will be attached onto, no matter the elements are time points, intervals, etc.
  • Instead of a simple attribute, it becomes a relation that expresses the attribute value with associated temporal elements.
  • Thus, attribute time-stamping also called nested time stamping, because we can consider the temporal attribute as a sub-relation of the tuple relations.
Temporal Algebra

• Historical relational algebra (HRA)
  • Temporal elements represent valid time

• Nested Bi-temporal algebra (NBtA)
  • Temporal elements represent both valid and transaction time
HRA - Time

- Time is a continues variable;
- In databases, time is recorded as discrete points;
- If the origin of time is \( t_0 \) and current time is \( t_{\text{now}} \), time \( T \) can be represented as a set of \( \{ t_0, t_1, t_2, ..., t_{\text{now}} \} \), where \( t_0 < t_1 < ... < t_{\text{now}} \), \( t_0 + 1 = t_1 \), ..., \( t_{\text{now}-1} + 1 = t_{\text{now}} \).
HRA - Point and Interval

• Time points and intervals both represent time variable;
• They can be converted into each other.
• Time points can be regarded as the smallest interval;
• and time interval can be regarded as a set of time points.
• We use \([l, u]\) to represent the interval that starts at \(t_l\) (lower bound) inclusive and ends at \(t_u\) (upper bound) exclusive.
• Since interval representation can save more space, we just use interval representation from now on.
HRA - Attribute

• An attribute may have temporal element.

• If the attribute $A$ is time invariant, such as Social Security Number, the value itself forms an atom.

• If $A$ is time variant, that means the value $a$ of $A$ is valid for a certain time and not valid for other time, the attribute needs to include the temporal element as $\langle [l,u), a \rangle$.
There are four types of attributes in R.

- Atomic attributes contain atomic values which are subsets of U.
- Triplet-valued attributes contain triplets as the format of \( [l, u), a \).
- Set-valued attributes contain a set of atomic values.
- Set-triplet-valued attributes contain sets of triplets as values. Each set contains one or more triplets and represents the history of the attribute over \( T \).
HRA - Attribute Type
HRA - Basic operations

• Standard relational algebra operations can be applied to Historical Relations directly with minor modifications.
  • Projection ($\pi$) and Cartesian product ($\times$) operations are remain unchanged.
  • Set union ($\cup$) and Set difference ($-$) may have overlapping, adjacent, and contained intervals and then change the attribute type from triplet-valued to set-triplet-valued.
HRA - New operations

• Triplet-valued attributes can be packed to a set-triplet-valued attributes and unpack does the reverse.

• Atomic attributes can form triplet-valued attributes by triplet-formation to convert three attributes together while triplet-decomposition break triplet-valued attributes to its components.

• Slice operation restricts the time of an attribute based on the time of another attribute.

• Drop-time operation discards the time component and only keep the atom or atom set from triplet-valued attributes or set-triplet-valued attributes.
NBtA - Operations

• There are eight basic operations in : Set Operations, Projection, Selection, Cartesian Product, Unnest, Nest, Temporal Atom Decomposition, and Temporal Atom Formation.

• While Set Operation, Projection, Cartesian Product are exactly the same as relational algebra, others need modifications.

• The nested temporal model is a generalized model of the historical relational model where the latter one has order 1.

• Thus, the algebra operations remain the same except the nesting and unnesting operations.
NBtA - Atoms

• Bitemporal Atom A bitemporal atom has two temporal intervals to represent not only the valid time of the data value but also the transaction time of it.

• Thus, t in a bitemporal relation has a format as \( \langle TT, VT, V \rangle \).

• Also, since time points and time intervals can be converted from one to the other easily; and time interval is more general, we still use time interval to represent the transaction time and valid time in a bitemporal atom as \( \langle [TT_l, TT_u), [VT_l, VT_u), V \rangle \).
The unnesting operation makes a nested relation flatten.

- If attribute A of a bitemporal relation R is a set of atoms or bitemporal atoms, applying μ to A will create a new tuple for each atom or bitemporal atom in the set of A, and keep other attributes unchanged.
- If μ is applied to every bitemporal set-valued attribute in R recursively, then the final result will be a flat relation (1NF) with only atoms or bitemporal atoms, or nesting order of 0.
NBtA - Nesting

• Nesting operation packs a bitemporal schema to make it more compact.

• If certain conditions are satisfied, nesting operations can reverse a zero order flattened bitemporal schema get from a nested bitemporal schema using unnesting operation.
NBtA - Slice

• Slice operation modifies the $k^{th}$ attribute’s bitemporal time component based on the temporal component of the $p^{th}$ attribute by the operator $\theta \in \{\cup, \cap, -\}$. The result may not be empty.

• The $\cap$ represents the “where” predicate in queries.
NBtA - Decomposition

• Bitemporal atom decomposition operator splits the $k^{th}$ attributes $A$ of relation $R$ into five new attributes that were it’s components: $A.TT_l$, $A.TT_u$, $A.VT_l$, $A.VT_u$, $A.V$.
NBtA - Formation

• Bitemporal atom formation is the reverse of bitemporal atom decomposition (δ). It combines existing attributes together to build a bitemporal atom.
NBtA - Rollback

• Let R be a bitemporal relation schema, < r(1),...r(n) >, where r(k) is a bitemporal attribute.

• The result contains bitemporal atoms such that its $TT$ component includes the time $t$.

• If we want to cut the transaction time of these bitemporal atoms so that $y$ only includes time $t$, then we can replace in the above formula $z = x$ by $z = <[t,t+1)$, $x.TT$, $x.V$>. 
Connection and Future

• Temporal elements in current Semantic Web Technologies
• How to make semantic web more expressive by borrowing from Temporal Database?
• How to design queries and data structure so that searching result is more accurate and efficient?
Q & A
Thank you