Tutorial/Workshop Session 3: Frédéric Kaplan (Ecole Polytechnique de Lausanne)(@frederickaplan): Semantic Modelling for the Humanities: RDF and beyond

Goal: Feel the beauty and power of semantic encoding

Lot of (ugly) acronyms: RDF, URI, OWL, SPARQL, SWRL, CIDOC-CRM

No computer needed (pen & paper techniques)

Books: Seagaran et al: Programming the Semantic Web (O'Reilly)

Software: Protégé (OSS, Stanford) - http://protege.stanford.edu/

Examples from the Venice Project: Incanto System (1283-1453): Organization of commercial routes through an auction system (State owned). Source: Doris Stöckly, Le système de l'Incanto ... Brill 1995.

How to code that set of data ?

Simplest kind of dataset: Table (e.g. Excel), e.g.

Route	Departure	Destination	Departure_Date	Arrival_Date	Callling_at
R1	Venice	Alexandria	2/7/1422	5/8/1422	Corfu (2d), Candia (3d)
R2	Venice	Tana	23/6/1422	12/9/1422	Corfu (3d), Constantinople (4d)
R3	Venice	Bruges	12/3/1422	12/6/1422	Corfu (1d), Mallorca (3d) , Lisboa (3d)

With data in a spreadsheet, you can sort, display, print, edit

There are semantics in data table: Name of the columns

Limits : no (complex) querying, limited number of columns

-> Relational Database give you these options: multiple tables which you can join

```
Route
                                                              Place
ID
DepartureID ------------------------------------------->       ID
DestinationID                                                     GPS coord
Departure date                                          Stop
Arrival date
Stop1 ID ---------------------------------------------> ID
Stop2 ID                                                  Length of stay
Stop3 ID
```

Problem: As data grows and extends, Schema migration: painful process

Simple Excel like tools don't have capabilities for complex querying and also have limit on information that can be stored, so the way out is relational databases. These too become problematic when data grows and these have to be mmigrated along with schema. So using metadata can be a way out which is future proof as well. you can keep adding more data to exisiting one as and when there happens to be by improving the metadata.

Make schema generic (EAV: http://en.wikipedia.org/wiki/Entity%E2%80%93attribute%E2%80%93value_model)

Route

ID ->

Name

Logical encoding

Subject Predicate Object: RDF Statement

(R1 Departure Venice)

(R1 Departure-date 2/7/1422)

this can also be presented as a graph

R1 -> departure -> Venice <- departure - R2

|

departure-date

|

v

2/7/1422

Syntax of RDF: subject, predicate, object

Data presented in RDF can also be presented using graphs with subject and object forming the nodes (known as **resources**) and predicate as edge between the two.

In the RDF (Resource Description Framework) Model, nodes of graphs are called Resources

RDF was defined with the idea of making data sharable across teams, so each resource is given a: URI (Unique Resource Identfifiers) permits to avoid ambiguities

Best known form of URIs are URLs (Addresses of web pages). URIs extend this concept to any unique identifier of a resource (they don't need be related to a concrete Web page, e.g. isbn: 987123456 for a book)

In RDF: predicates are always resources. The URI for a resource represented in RDF is the URI Ref (e.g. http://www.w3.org/1999/02/22-rdf-syntax-ns#type) It is good practice that there is a corresponding site, if you type this into the Browser, but not a requirement

With namespaces, you can shorten long URIs as the one above to rdf:type

Serialized representations of RDF-data can become complicated. There are multiple serialization formats: e.g. RDF-XML, N3-Triples, RDFa (RDF in attributes e.g. of HTML-pages)

A set of URIRefs is known as a Vocabulary, e.g. for our maritine route examples

What SQL is as query-language for relational database, SPARQL is the language to query RDF-graphs:

- Departure (?x1, Venice) //find x1 which departures from Venice
- Captain (?x1, ?x2) && Gender(?x2, Women) //such that x2 is captain of x1 and x2 is of gender Women

(all the routes to Venice that have a Women Captain)

With RDF coding, we can also write rules to infer new triples

- hasParent (?x1, ?x2) ^ hasBrother (?x2, ?x3) => hasUncle (?x1, ?x3) // x1 has x2 as parent and x2 has x3 as brother that implies (means) x1 has x3 as uncle

This is also a way of detecting possible incoherence in the set of knowledge coded in the triple store -> SWRL (Sematic Web Rule Language) is one standard language to do this.

ONTOLOGY framing :

Once you think you know that something exists, the next step is to put it into a relation to other objects. We don't do this for every possible thing, just for the entities in our problem domain.

- (Venice is_a Place)
- (Corfu is_a Place)
- (Place has_lat latitude)
- (Place has_long longitude)
-

Ontology can be represented as RDF and stored as graph alongside RDF.
OWL (Web Ontology Language)

Terminology statements

- ex:Bridge rdf:type rdfs:class
- ex:Bridge rdfs:subclass ex:Place

Assertion statements

- ex:Rialto rdf:type ex:Bridge
- ex:RialtoCons ex:broughtIntoExistence ex:Rialto

It is expressed in the same language. Eg "Rialto" is a "Bridge".

Ontologies themselves are resources that can be associated with an URIref. It is relatively easy to create your own ontology using a software like **Protégé**. But some ontologies aim at being "Universal": **CIDOC-CRM** is an ontology for Cultural heritage (about 20 years of work, ISO 21127, 100+ schema, very stable)

https://en.wikipedia.org/wiki/ISO_21127

In CIDOC-CRM, the modelling is event-centric: The underlying idea is to model change, not state: Instead of coding the birthdate of an actor, it is better to code the event of its birth:

- not: (Person is_born 1900-1-1)
- but: (Person hasBirthEvent BirthEventP1)
- (BirthEventP1 hasDate 1900-1-1)

Actors are thought to related to physical items only through temporal events.

CIDOC-CRM has dozens of predefined Properties (had participants, carried out, ...)

CIDOC-CRM has also a very interesting model for places. What is hard about places? What is the coordinate of a city? Things can happen on mobile places (like a boat). The question "where is it" can either specify a position (coordinate) or a physical thing (the ship, at home, where you might not know where "home" is) -> E53 Place can only be defined relative to a larger object (Cities are moving over time)

Influence is very hard to model (Actor A1 has taken notice of Object O2).

Problem in the RDF-Model: Qualification of facts (who stated it, how certain is it).

Technical trick:

- (RialtoReconstruction hasTimeSpan 1588-1591)

-> Reifed RDF

- (s1 rdf:subject RialtoReconstruction)
- (s1 rdf:predicate hasTimeSpan)
- (s1 rdf:object 1588-1591)
- (s1 metardf:reliability 0.8)
- (s1 metardf:source FKaplan)