Knowledge representation: SPARQL, DBpedia, knowledge engineering and data model integration

Wojciech Jaworski

Institute of Informatics
University of Warsaw

Outline

Basic Graph Pattern

- SPARQL - query language for getting information from RDF graphs. It provides facilities to: extract information in the form of URIs, blank nodes, plain and typed literals.
  - extract RDF subgraphs.
  - construct new RDF graphs based on information in the queried graphs
  - matching graph patterns
- variables – global scope; indicated by ‘?‘ or ‘$‘
- terms delimited by ‘<>‘ are relative URI references
- Set of Triple Patterns
  - Triple Pattern – similar to an RDF Triple (subject, predicate, object), but any component can be a query variable; literal subjects are allowed

    ?book dc:title ?title
  - Matching a triple pattern to a graph: bindings between variables and RDF Terms
- Matching of Basic Graph Patterns
  - A Pattern Solution of Graph Pattern GP on graph G is any substitution S such that $S(GP)$ is a subgraph of G.

Basic Graph Pattern - Multiple Matches

- Data

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Johnny Lee Outlaw".
_:a foaf:mbox <mailto:jlow@example.com> .
_:b foaf:name "Peter Goodguy".
_:b foaf:mbox <mailto:peter@example.org> .
```

- Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE { ?x foaf:name ?name . ?x foaf:mbox ?mbox }
```

- Query Result

name	**mbox**
"Johnny Lee Outlaw"	<mailto:jlow@example.com>
"Peter Goodguy"	<mailto:peter@example.org>

Basic Graph Pattern - Blank Nodes

- Data

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Alice".
_:b foaf:name "Bob".
```

- Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?x ?name
WHERE { ?x foaf:name ?name }
```

- Query Result

x	**name**
_:c	“Alice“
_:d	“Bob”

Graph Patterns

- Basic Graph Pattern – set of Triple Patterns
- Group Pattern - a set of graph patterns must all match
- Value Constraints - restrict RDF terms in a solution
- Optional Graph Patterns .- additional patterns may extend the solution
- Alternative Graph Pattern – two or more possible patterns are tried
- Patterns on Named Graphs - patterns are matched against named graphs

Group Pattern

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE { ?x foaf:name ?name . ?x foaf:mbox ?mbox }
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE { {?x foaf:name ?name; foaf:mbox ?mbox } }
```

Value Constraints

- Data

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/book/> .
@prefix ns: <http://example.org/ns#> .
:book1 dc:title "SPARQL Tutorial".
:book1 ns:price 42 .
:book2 dc:title "The Semantic Web".
:book2 ns:price 23 .
```

- Query

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x ns:price ?price . FILTER ?price < 30 . ?x dc:title ?title . }
```

- Query Result

title	**price**
"The Semantic Web"	23

Optional graph patterns

- Data

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/book/> .
@prefix ns: <http://example.org/ns#> .
:book1 dc:title "SPARQL Tutorial".
:book1 ns:price 42 .
:book2 dc:title "The Semantic Web".
:book2 ns:price 23 .
```

- Query

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x dc:title ?title .
        OPTIONAL { ?x ns:price ?price . FILTER ?price < 30 }}
```

- Query Result

title	**price**
“SPARQL Tutorial“	
"The Semantic Web"	23

Multiple Optional Blocks

- Data

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
_:a foaf:name "Alice".
_:a foaf:homepage <http://work.example.org/alice/> .
_:b foaf:name "Bob".
_:b foaf:mbox <mailto:bob@work.example> .
```

- Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox ?hpage
WHERE { ?x foaf:name ?name .
        OPTIONAL { ?x foaf:mbox ?mbox }.
        OPTIONAL { ?x foaf:homepage ?hpage } }
```

- Query Result

name	**mbox**	**hpage**
“Alice“		<http://work.example.org/al
“Bob“	<mailto:bob@example.com>	

Alternative Graph Patterns

- Data

```
@prefix dc10: <http://purl.org/dc/elements/1.0/> .
@prefix dc11: <http://purl.org/dc/elements/1.1/> .
_:a dc10:title "SPARQL Query Language Tutorial".
_:b dc11:title "SPARQL Protocol Tutorial".
_:c dc10:title "SPARQL".
_:c dc11:title "SPARQL (updated)".
```

- Query

```
PREFIX dc10: <http://purl.org/dc/elements/1.0/>
PREFIX dc11: <http://purl.org/dc/elements/1.1/>
SELECT ?x ?y
WHERE { { ?book dc10:title ?x } UNION { ?book dc11:title ?y } }
```

- Query Result

x	**y**
	"SPARQL (updated)"
	"SPARQL Protocol Tutorial"
"SPARQL"	
"SPARQL Query Language Tutorial"	

Query forms

- SELECT: returns all, or a subset of the variables bound in a query pattern match
- CONSTRUCT: returns an RDF graph constructed by substituting variables in a set of triple templates
- DESCRIBE: returns an RDF graph that describes the resources found.
- ASK: returns whether a query pattern matches or not.

Support for SPARQL

- SPARQL and Jena
  - module called ARQ that implements SPARQL; also parses queries expressed in RDQL or its own internal language.
  - not yet part of the standard Jena distribution; availbale from either Jena‘s CVS repository or as a self-contained download
- Twinkle
  - simple Java interface that wraps the ARQ SPARQL Processor library (the add-on to Jena).
- Redland
  - set of free software packages that provide support for RDF, including querying with RDQL and SPARQL using the Rasqal RDF Query Library. .

Outline

Wikipedia

- Wikipedia — electronic encyclopedia.
- 4.699.911 articles in English, 1.087.696 articles in Polish (20 January 2015)
- Number of articles is dynamically increasing
  - 4.137.668 articles in English, 942.309 articles in Polish on 6 January 2013,
  - 3.840.850 articles in English, 863.021 articles in Polish on 7 January 2012.
- Greatest existing set of concepts.
- Corresponding articles in different languages are linked.
- Concepts do not form the structure of the ontology, however, they are grouped into categories and related by links.
- Descriptions of concepts are expressed in natural language, and not as a logical formula.
- Wikipedia is constantly updated.

DBpedia

- DBpedia is a project aiming to extract structured content from the information created as part of the Wikipedia project.
- The whole DBpedia data set describes 4,580,000 entities, out of which 4,220,000 are classified in a consistent ontology, including 1,445,000 persons, 735,000 places, 123,000 music albums, 87,000 films, 19,000 video games, 241,000 organizations, 251,000 species and 6,000 diseases.
- The DBpedia project uses the Resource Description Framework (RDF) to represent the extracted information and consists of 3 billion RDF triples, 580 million extracted from the English edition of Wikipedia and 2.46 billion from other language editions.
- DBpedia is generated automatically by extraction of structured data from Wikipedia content.

Querying DBpedia

- There is a public SPARQL endpoint over the DBpedia data set at http://dbpedia.org/sparql.
- Query

```
PREFIX dbprop: <http://dbpedia.org/property/>
PREFIX db: <http://dbpedia.org/resource/>
SELECT ?who, ?work, ?genre WHERE {
db:Tokyo_Mew_Mew dbprop:author ?who .
?work dbprop:author ?who .
OPTIONAL { ?work dbprop:genre ?genre } . }
```

- DBpedia ontology is available in OWL format at http://wiki.dbpedia.org/Downloads2014

Outline

Knowledge engineering

- According to a traditional methodology, shifting from a verbal description to a formal specification happens non-analytically: The system designer creates a model in his mind based on the description and expresses it in a formal way as a diagram of tuples, UML, or a hierarchy of classes.
- Knowledge engineering aims at systematic translation from a non-formal verbal specification to an executable program.

Knowledge engineering

- Knowledge engineering is the application of logic and ontology to construct computable models for given domains and a given aim.
- Specification is understood as natural language data that are to be translated to a form with overt structure and some fixed semantics.
- Computability, the application domain and the existence of an aim is what makes it different from pure mathematics, since the latter:
  - needs no domain for application.
  - the models need not be computable or even finite.
  - has no other aim but aesthetic satisfaction and the contemplation of elegant abstractions.

Non-formal specifications

There is a traffic light that automatically turns red or green, but it also has an option for manual control under special circumstances.

- The expert writes informally using a language unknown to the knowledge engineer.
- Each term that is trivial to the expert needs be analysed by the knowledge engineer and explained.
- The problem with automatic translation of informal specifications is the great amount of knowledge evoked by each word, not the language syntax itself.
- When translating specifications to a computational form one needs to make assumptions and add details.

Specification interpretation

The color of the traffic light X may be either red or green. X has an automatic control switch, which may be either on or off.
If the automatic control switch is on,
then X behaves according to the following two rules:
When the color of X becomes green,
it remains green for g seconds;
then it changes red.
When the color of X becomes red,
it remains red for r seconds;
then it changes green.

- The interpretation has additional assumptions, eg. the light remain green or red for a constant number of seconds.
- The assumptions are based on knowledge sources independent of the specification such as
  - general knowledge of the knowledge engineer
  - additional information sources
  - discussions with the expert

Rules for knowledge representation

- The objects incapable of storing directly in the computer memory are represented with surrogates. The computer program can manipulate them in order to simulate the outer system and to reason about it.
- When modeling a domain we make decisions about the concepts in the ontology and the relations between them. We determine what types of entities exist in the domain analysed.
- We create a partial theory to describe the domain.
- The representation must work efficiently on the available hardware.
- The representation should facilitate communication between the knowledge engineer and the expert.

Procedural implementation

- Each relevant entity such as the actual time or the color of the light has an associated variable, a symbolic representation of the entity.
- The values of those variables can be changed in order to simulate the behaviour of the system.
- In the procedural approach each event is an operation that changes the state of the model — a procedure.
- We create a control mechanism that executes procedures when their initial conditions are satisfied.
- We execute the program and observe its evolution in time.

Declarative implementation

- The declarative approach is based on constraints or axioms that define the initial conditions and the end terms as well as the changes that take place in the model with each action.
- Instead of executing programs we make use of theorem proving systems to infer consequences of performing actions.
- A theorem proving system is an interpreter that executes axioms.
- The effect of the procedural simulation is obtained with deduction.

Metalevel

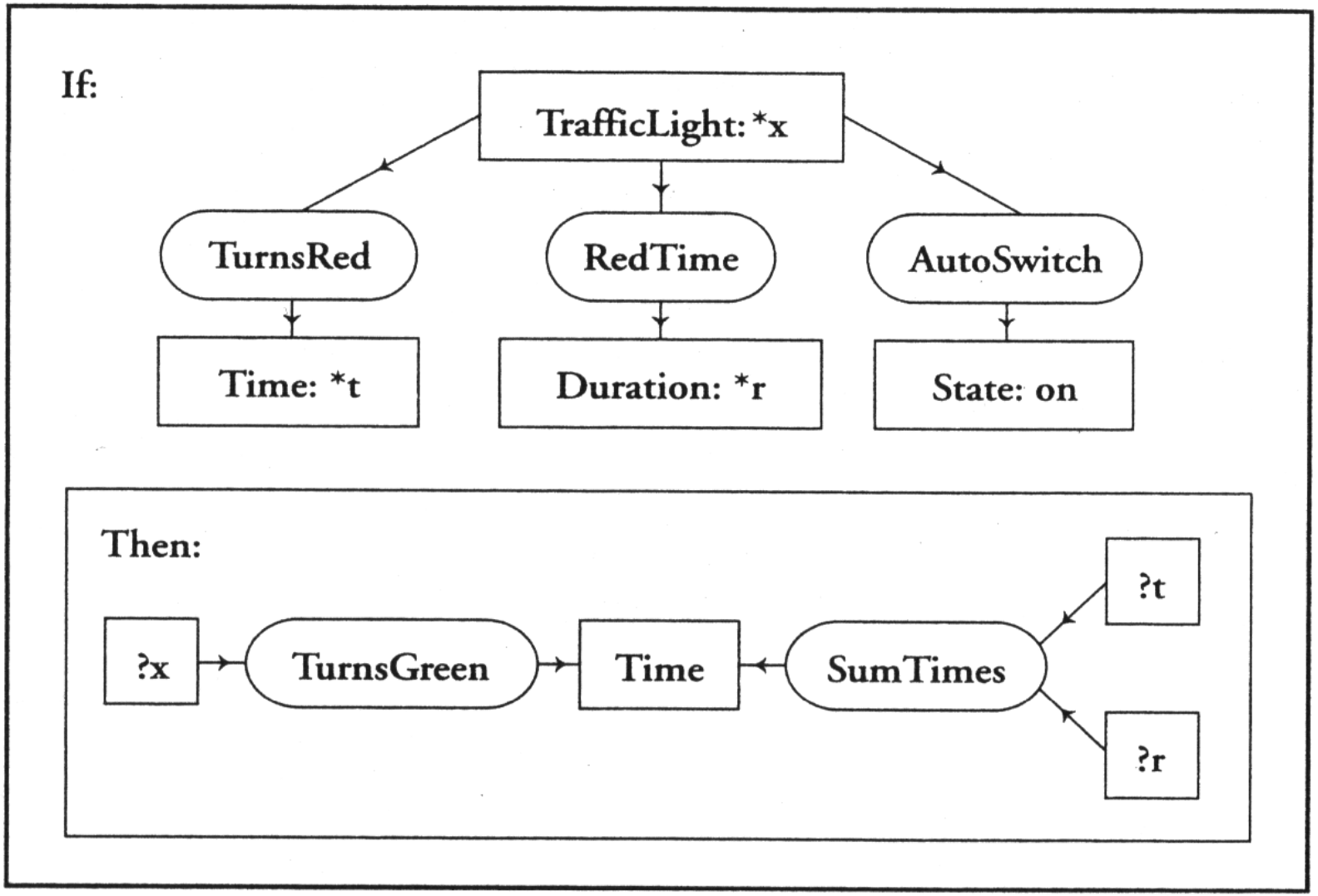
- Procedural approach does not make it possible to explain why the system works as it does.
- Does not allow versioning (no history of changes is being kept).
- It cannot say why the lights change colour.
- It does not answer questions like: what was the average time the lights had been operated manually.
- If we needed such information we would have to add instructions enabling us to make inferences about the lights from the metalevel.
- Description of those instructions might be more complicated than the initial program.
- In declarative approach streetlights are being simulated proving the claims about the condition they are in.
- System equipped with logical interpretation is capable of answering any question expressible in our ontology and logics provided that the answer is computable.

Communication with expert

- When the knowledge engineer interpret the specification differently than the author, errors appears.
- It is necessary to re-consult the specification with an expert after detailing and formalizing it. This requires a comprehensive formalism.
- This is especially important when laws, regulations, etc. are a part of the specification.
- Stylized English:
  If a traffic light x turns red at time t,
  has a red time of a duration r,
  and has autoswitch in the state on,
  then x turns green at a time, which is the sum of t and r.
- Experts often prefer diagrams used in their domains than verbal description.

Communication with expert: conceptual graphs

They are precise (formal semantics), readable, easy to explain to expert.



Outline

Data models integration (traditional approach)

- Set of source schemas (schemas of existing data bases).
- Global schema (base integration perspective).
- Query explaining mapping.
- Global as View (GAV):
  - Global schema is a collection of perspectives basing on source schemas.
  - Queries use those perspectives therefore do not require additional processing.
  - Adding new data source requires modifications in the code of existing perspectives.
- Local as View (LAV):
  - Source schemas are a collection of perspectives basing on global schema.
  - To execute query perspective needs to be inverted what requires searching the space of possible queries.
  - Adding new data source is independent from other sources.

Data models integration (using ontology)

- Data models collection.
- Ontology developed for integration tasks expressed in OWL.
- Relational database structure is mapped as follows:
  - Tables correspond to the concepts, attributes to columns (rdf:property)
  - Tables representing relations many-to-many are translated to attributes.
  - Row key in the table is an identifier of individual.
  - The value in column is the value of attribute.
- XML structure is mapped as follows:
  - XML attributes correspond to RDF attributes.
  - Elements having values belonging to simple types are translated to attributes.
  - Other elements correspond to concepts.
  - Individuals need to be given identifiers and duplicated data needs to be merged.
- Database queries are asked in SPARQL and translated to SQL (for relational databases) or XQuery (for XML bases) what needs coding the mapping procedure.

Data models integration (using knowledge base)

- Data models collection.
- Universal base of knowledge.
- Mapping concepts and relations from particular data models to concepts from knowledge base.
- Identical embedding data models in knowledge base.
- Defining constraints (in the form of axioms) between the concepts from data models and concepts from the knowledge base.
- Query translation is executed by inference system.
- Knowledge base serves as a universal interface that enables information search, data modification and making inferences using the knowledge representation language.
- Declarative mapping of model is considered better than procedural mapping because of system ability to determine its competencies (inferences about beliefs).
- What in Cyc is known under the name Semantic Knowledge Source Integration (SKSI), is generally the same as Semantic Integration Bus.

How to map concepts?

- UMBEL (Upper Mapping and Binding Exchange Layer)
- Ontology containing 28.000 concepts
- UMBEL Vocabulary serves to identify potential relations between concepts present in data models.
- By assumption such mapping is an approximation.
- The second objective of UMBEL is to provide fixed set of concepts that can be used to sort mappings established in previous step.
- This set of concepts is not aimed to model the world. It is supposed to provide fixed set of reference points that will allow to organize content.
- UMBEL is a subset of concepts selected from OpenCyc.

Outline

Semantic browsers

- Information Retrieval: finding documents by query.
- Information Extraction: finding facts.
- Query is a semantic pattern ie.
  - set of concepts,
  - logical formula, that is a set of concepts linked with relations.
- Semantic patterns usually need to be extracted from word sequence provided by the user.
- Sought documents do not need to contain expressions from the query. They may only relate to the given topic.
- Browser may try to guess user intentions and the context of the query.

Semantic browsers

- Semi-structured data:
  - text fields of different length,
  - field labels,
  - tags within the text.
- Meta-data: data about data.
- Other data(video, audio, numeric data) is transformed to structured text.
- Language models:
  - bag of words,
  - word sequence,
  - phrase(surface parsing),
  - sentence (deep parsing).

Bag of words

- Semantic capacity limited to concepts expressed by single words.
- Relations between concepts are lost.
- Linguistic analysis is limited to transforming words to the basic forms.
- List of first names, surnames, places etc. together with Wordnet are sufficient semantic resources.
- Bag of words represented as frequency vector allows to treat text as a point in linear space.
- Traditional model used in browsers.

Word sequences

- Covers all concepts represented by coherent word sequences (this is why it suits English better than Polish)
- Determining borders for representations of certain concepts and defining acceptable word permutations in concept description are problematic.
- Concept names and their synonyms can be extracted from Wikipedia.
- Wikipedia can also be used to resolve ambiguity.
- Relations between concepts can be inferred from their proximity in text.
- Initial division into sentences gives good results.
- Model is suitable for approximate fact extraction. Facts are here understood as sentences related to query (Textpresso).
- In texts from specified topic set of concepts uniquely determines relations connecting those concepts.

Surface parsing

- Includes relations between proximate concepts.
- Mapping syntactic and semantic relations is problematic.
- Such mapping is determined by knowledge base, or semantic grammar, which is a set of rules defining how facts related to ontological concepts are expressed in language.
- Regular expressions
- Surface parsing is a pattern-recognition from abundance of information.

Deep parsing

- Complete mapping of text with language formula for knowledge representation.
- Currently feasible only for text sets within limited topic.
- Field labels may indicate the topic of the specified field.

Information from field labels

- Field label may indicate concept, field value may indicate an individual ie. author, date.
- How to use information that a given field contains title, abstract or section title?
- How to use information that certain fields contain description of symptoms, diagnosis or treatment?
- (Assuming that user is not asking directly about abstract or diagnosis)
- Making use of information from field labels requires integration of structure set by labels with knowledge base.

Acknowledgements

Part of material has been taken from the lecture

SPARQL Query Language for RDF
Cristina Feier
http://www.wsmo.org/wsml/papers/presentations/sparql.ppt