Representing data as resources in RDF and OWL

Why and how?

Pierre-Antoine Champin
http://liris.cnrs.fr/~pchampin
Outline

- Background and motivation
  - RDF, OWL
  - wrapping OWL around relational DBs

- Rationale of our approach
  - statements about data (RDF)
  - inferences about data (OWL)

- Proposition and discussion
  - other datatypes
  - classification of values
  - efficiency
Resource Description Framework

W3C recommendation

aims at unifying the semantics of structured data

- RDF is to semantics what XML is to syntax

intended standard for the Semantic Web
Resources \approx\ any object

- **not** limited to web resources

Background & motivation
- RDF
- OWL
- wrapping RDBs

Rationale

Proposition & discussion

- this slideshow
- EROW 2007
- Champin
Properties (labelled arcs)

- relating resources to each other

- RDF
- OWL
- wrapping RDBs

Rationale

Proposition & discussion
Literals

- data with an XML-schema datatype
- also related to resources by properties
Background – OWL (1)

- Ontology Language for the Web
- W3C recommendation

- RDF vocabulary
  - resources and properties
  - with a pre-defined semantics
    - based on very expressive description logics (SHOQ-D)
    - with concrete values (literals)
    - no unique name assumption

Background & motivation
- RDF
- OWL
- wrapping RDBs

Rationale

Proposition & discussion
Background – OWL (2)

Backround & motivation
- RDF
- OWL
- wrapping RDBs

Rationale

Proposition & discussion

ScientificEvent ⊑ ∀ participant.Person

ScientificEvent ⊑ Person

ScientificEvent ⊑ Restriction

EROW 2007

Workshop

Scientific Event

Champin

participant

type

subclass

participant

onProperty

allValuesFrom

Type

Restriction
Background – OWL (3)

Background & motivation
- RDF
- OWL
- wrapping RDBs

Rationale

Proposition & discussion
Motivation - CROSS

OWL wrapper for relational databases
- with the possibility to provide additional knowledge to the original schema

Intended benefits
- bring legacy databases to the Semantic Web
- ease integration of heterogeneous data sources (relational, XML, RDF...)
- help reverse engineering by providing inferences

http://liris.cnrs.fr/~pchampin/dev/cross
The distinction between *objects* and *values* has a long history in computer science

- programming: C++, Java
- modelling: Entity-Relationship, UML

However, values can be considered as “primary citizen” objects

- programming: Smalltalk, Python...
Rationale – RDF

Literals are irregular w.r.t. resources

- this slideshow
  - presentedAt
  - EROW 2007

- EROW 2007
  - date
  - 2007-01-13 (date)

- 2007-01-13 (date)
  - dayOfWeek
  - Saturday
Inference with literals is somewhat weaker than inference with resources.

Problem of inverse-functional properties.

\[
\begin{align*}
&\text{Jane Doe} & \text{biologicalMotherOf} & \text{John} \\
&\text{Jane Smith} & \text{biologicalMotherOf} & \text{John} \\
\end{align*}
\]

\[
\begin{align*}
&\text{Jane Smith} & \text{sameAs} & \text{Jane Doe} \\
\end{align*}
\]
Properties to literals cannot be declared inverse-functional

Primary keys cannot be modelled in OWL!
Proposition

- A resource identifier (URI) for every
  - decimal number
  - unicode string

- Each value must have a unique identifier
  - not a theoretical requirements (no UNA)
  - but a practical one

- See the paper for more detail on the syntax of those identifiers
What about other datatypes? (1)

XML-Schema has a lot of other datatypes

- Other datatypes
- Classification
- Efficiency

Other datatypes can always be represented by numbers or strings
What about other datatypes? (2)

- We can relate their value resources to a **canonical representation**, with an *inverse-functional* property
  - e.g. ISO 8601 representation for dates
- Two resources will be unified as soon as they have the same canonical representation
- We do not need to provide standard identifiers for them
What about classification? (1)

XML-Schema provides a number of subtypes for *string* and *decimal*
What about classification? (2)

- Resources (unlike literals) can be related to other resources, including classes

- We can express those classifications in any RDF class system (e.g., OWL)

- We can relate any data resource to its (most specific) class

- We can even provide alternative classifications...
What about efficiency? (1)

- All this inference power comes at a price
  - expressiveness/complexity trade-off
- Inference engines delegate the reasoning about concrete domain values to a specialized *datatype oracle*
- Delegation is good for two reasons
  - reduces combinatorial complexity
  - allows for specialize delegates
Delegate reasoning about resources is an emerging topic in description logics

- Distributed Description Logics (DDL) Serafini and Tamilin (2004)
- $\varepsilon$-connections Grau, Parsia and Sirin (2004)

We could replace the datatype oracle by an inference engine

- more or less optimized / general
- according to our needs
Conclusion

- There are benefits in representing data as resources rather than literals in RDF and OWL.

- One of them being that it opens a number of interesting research opportunities!

- Thank you for your attention.

Any question?
Actually, the paper proposes a unique resource identifier for every

- decimal number
- unicode string
- tuple of arbitrary size, containing a mix of
  - decimal numbers
  - unicode strings
  - NULL values

Useful when representing relational data

- especially keys with multiple columns
Annex (2)

- Literals with different lexical values are always considered distinct
- Resources with different identifiers are **not** considered distinct (no UNA)
- Trick: use the best of both worlds, with a *functional* property

![Diagram](image_url)