RDF and Inconsistency

Claudio Gutierrez, Carlos Hurtado
Center for Web Research
Computer Science Department
Universidad de Chile
www.ciw.cl

Contents

- The context
- RDF formal model
- RDFS and OWL
- Consistency issues

The context

- Semantic Web
  - Give semantics to the information on the Web
  - Make information on the Web machine-processable
- Database point of view:
  - Give structure to the information on the Web
  - Build tools to process such data
  - Deal with natural inconsistency of such data

A reflection

"[The Web] gives a completely new perspective to Leibniz's project of a universal framework for the management of knowledge. [...] Leibniz would be probably enthusiastic about this new arena of logic. [...] But in academic logic, these practical Leibnizian tasks do not attract much interest."

W. Thomas, in Dagstuhl Anniversary Conference, August 2000, LNCS 2000.

RDF: ask Google...

sex 184.000.000 hits

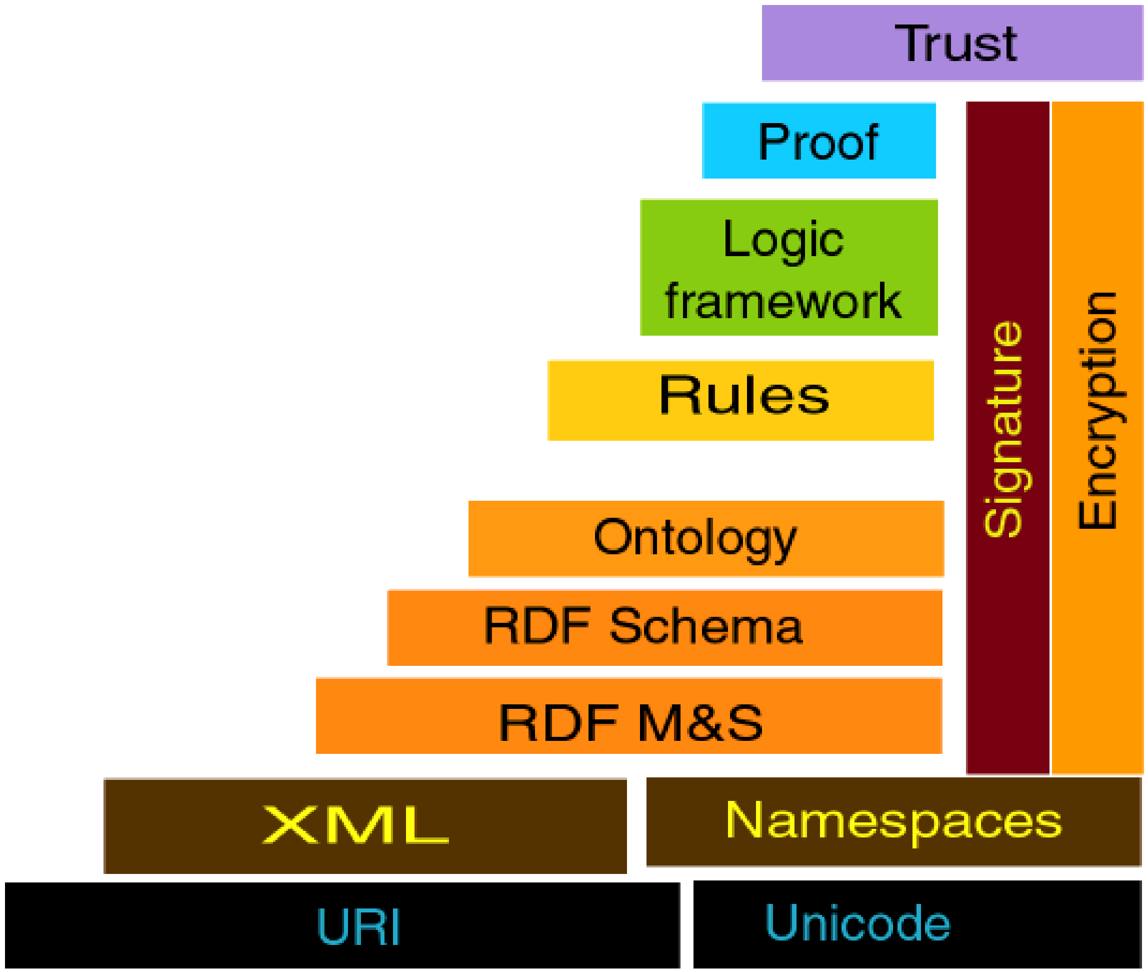
java 32.900.000 hits

xml 19.200.000 hits

rdf 2.250.000 hits

inconsistency 662.000 hits

inconsistency + rdf 2.460 hits


Trust
Proof
Logic framework
Rules
Ontology
RDF Schema
RDF M&S
Signature
Encryption
XML
Namespaces
URI
Unicode

Our focus

...
reasoning, proofs, etc.
OWL
RDF Schema
RDF
codification, transport, etc.
...

RDF, RDFS and OWL

RDF: basic data model for objects (resources) and relations between them. These data model can be represented in XML

RDFS: a vocabulary for describing properties and classes of RDF resources.

OWL: more vocabulary for describing properties and classes (e.g. relations, cardinality, equality, etc.)

RDF: a moving target...

- Resource Description Framework (RDF) Model and Syntax Specification, Lassila O., Swick R. Eds.
- RDF Vocabulary Description Language 1.0: RDF Schema, Brickley D., Guha R.V. Eds.
- RDF Semantics, P. Hayes, Ed.
- RDF/XML Syntax Specification (Revised), Beckett D. Ed.
- Resource Description Framework (RDF): Concepts and Abstract Syntax, Klyne G., Carroll J. Eds.
- OWL Web Ontology Language 1.0 Reference, Dean M., Connolly D., van Harmelen F., Hendler J., Horrocks I., McGuinness D.L., Patel-Schneider P.F., Stein L.A. Eds.

RDF Formal Model

U = infinite set of Uri references

$B = \{b_j : j \in N\}$ (Blank nodes)

L = infinite set of literals

Definition

- $(s, p, o) \in (U \cup B) \times U \times (U \cup B \cup L)$ is called an RDF triple.

- An RDF graph is a set of RDF triples.

RDF Formal Model (cont.)

- A mapping is a function μ : UBL $\rightarrow$ UBL preserving urirefs and literals

- $\mu(G)$ is the set of all $(\mu(s), \mu(p), \mu(o))$ such that $(s, p, o) \in G$ (called instance of G)

- $G_1 \cong G_2$ (isomorphic, "equal") iff G_2 is obtained from G_1 by renaming its blank nodes consistently.

- The merge of two graphs G_1, G_2 is defined as $G_1 \cup G'_2$, where $G'_2 \cong G_2$ and $blank(G'_2) \cap blank(G_1) = \emptyset$.

RDF Formal Model (cont.)

Theorem cf. RDF Semantics, Interpolation Lemma]
Let G_1, G_2 be RDF graphs. Then G_1 entails G_2 (denoted $G_1 \models G_2$) if and only if an instance of G_2 is a subgraph of G_1.

$G_1 \models G_2$ iff there is a mapping μ such that $\mu(G_2) \leq G_1$.

RDF: flexible model

- Descriptions:

  (John, `loves`, Maria)

- Structured properties:

  (Maria, `address`, Y)
  (Y, `street`, Goethe St.)
  (Y, `city`, Trier)

- Reification:

  (X, `type`, `statement`)
  (X, `subject`, John)
  (X, `predicate`, loves)
  (X, `object`, Maria)
  (X, `TruthValue`, false).

RDF: expressiveness/complexity

- Expresiveness: fragment $\exists, \wedge, \texttt{stat}(\texttt{X}, \texttt{Y}, \texttt{Z}), c_1, c_2, \ldots$ of first order
- Complexity: Deduction for RDF is NP-complete (Proof: codify subgraph isomorphism problem).

RDF: alt. formalization

Embedd RDF in F-logic (Yang, Kifer). Some differences with W3C view:

- Not same notion of deduction (although can simulate W3C notion): $G_1 \models G_2$ iff G_2 is isomorphic to a subgraph of G_1
- Reification: statements are given identifiers
(versus references to the components of a statement).

Vocabulary: RDF Schema

A. Classes:

rdfs:Resource

rdfs:Literal

rdf:XMLLiteral

rdfs:Class

rdf:Property

rdfs:Datatype

rdf:Statement

rdf:Bag

rdf:Seq

rdf:Alt

rdfs:Container

rdfs:ContainerMembershipProperty

rdf:List

Vocabulary: RDF Schema

B. Properties:

rdf:type The subject is an instance of a class.

rdfs:subClassOf The subject is a subclass of a class.

rdfs:subPropertyOf The subject is a subproperty of a property.

rdfs:domain A domain of the subject property.

rdfs:range A range of the subject property.

rdfs:label A human-readable name for the subject.

rdfs:comment A description of the subject resource.

rdfs:member A member of the subject container.

rdf:first The first item in the subject RDF list.

rdf:rest The rest of the subject RDF list after the first item.

Vocabulary: RDF Schema

Properties (cont.):

rdfs:seeAlso Further information about the subject resource.

rdfs:isDefinedBy The definition of the subject resource.

rdf:value Idiomatic property used for structured values

rdf:subject The subject of the subject RDF statement.

rdf:predicate The predicate of the subject RDF statement.

rdf:object The object of the subject RDF statement.

Ontology Web Language (OWL)

-(In)Equality: equivalentClass, equivalentProperty, sameIndividualAs, differentFrom, allDifferent

- Property Charact.: inverseOf, TransitiveProperty, SymmetricProperty, FunctionalProperty, InverseFunctionalProperty

- Prop. Type Restrictions: allValuesFrom, someValuesFrom

- Restricted Cardinality: minCardinality, maxCardinality, cardinality

- Header Information: imports, priorVersion, backwardCompatibleWith, incompatibleWith

-Class intersection: intersectionOf

- Datatypes

RDF and Inconsistency

- RDF gives a semantic layer to the web (base to reasoning)

- General design philosophy: as simple as possible (in particular, avoid axioms and predefined vocabulary)

- Challenge: tackle logical inconsistencies of information on the Web

RDF and Inconsistency

Logical inconsistencies in RDF specifications?

- Answer 1: No, there is no negation.
- Answer 2: No. But in any reasonable extension, yes.
- Answer 3: Yes and no (inconsistent...)

Issue I: conjunction

- Inconsistency needs negation + conjunction
- There are (at least) two kinds of conjunction on the Web:

  - two statements inside a page

  - two statements in different pages
- Example:

  - In Amazon, Neruda's "20 Love Poems" has two different ISBN: Inconsistency

  - Amazon and "Cheap Books" have different ISBN for the same book. Disagreement

Issue I (cont.)

- Two concepts:

  - Inconsistency (two contradictory statements made in the same source)

  - Disagreement (two statements that –forgetting its source– are contradictory.)

- Two related concepts in RDF

  - graph

  - merge of graphs

- Logical counterpart: two kinds of conjunctions: $\wedge$, $\bigwedge$
  (compare: "The SW needs two kinds of negation" G. Wagner)

Issue II: what is to be done?

- Classical:

  - Facts: inconsistency is an exception

  - Goal: Avoid inconsistency

  - Idea: Use adequate logic and/or makeup you KB

  - Procedure: implement the algorithms

- Web setting:

  - Facts: consistency is an exception

  - Goal: Work in the presence of inconsistencies

  - Idea: Build ontologies to deal with conceptual tools (logics, KB makeup, preferences, etc.)

  - Procedure: search for the “right” mechanism for particular case;

Issue II: example

- Leo Bertossi's Page is trustable

- Leo Bertossi's Page is not trustable

- Look in Yellow Pages of anti-inconsistency tools and choose your favorite method (ontology)

    - Run hubs and authorities alg. on certain pages

    - Use catalog of trustable people

    - Do not use this info in further reasoning

    - Use preferences

Issue III: Adding constraints

- General philosophy of RDF: avoid axioms that constraint the meaning of its vocabulary.

- Not true in OWL, for example:

  - FunctionalProperty and InverseFunctionalProperty

  - sameIndividualAs and differentFrom

- Use standard machinery to deal with inconsistency in OWL: Inconsistency and Description Logics

Issue IV: References

Reification, “published subjects”

- Sources of RDF data on the Web: Web pages, Data sources (dynamic Web pages)

- RDF statements must be somewhere located

- Is it reasonable to have a predicate “triple (s, p, o) belongs to uri u? or “uri u states triple (s, p, o)”?

- Several levels of statements, paradoxes

Issue IV: Example: a "paradox"

Reification of a triple (a, b, c) by reference:

$(X, \texttt{type}, \texttt{statement})$

$(X, \texttt{subject}, a)$

$(X, \texttt{predicate}, b)$

$(X, \texttt{object}, c)$

A "paradox":

$(X, \texttt{type}, \texttt{statement})$

$(X, \texttt{truthValue}, \texttt{false})$

$(X, \texttt{subject}, X)$

$(X, \texttt{predicate}, \texttt{truthValue})$

$(X, \texttt{object}, \texttt{false})$

Issue V: RDF graphs as databases

Idea: RDF Graph = database

- RDF graph = standard relational table
- Key difference: presence of blank nodes
- Database: set of RDF graphs (Warning!)

But, if we view RDF graphs a databases, it is natural to add minimal constraints at the data level.

Issue V: Example

Constraints for Reification. Need axioms like:

$\mathtt{stat}(a, b, c) \Leftrightarrow$

$\mathtt{stat}(X, \mathtt{subject}, a)$

$\mathtt{stat}(X, \mathtt{predicate}, b)$

$\mathtt{stat}(X, \mathtt{object}, c)$

plus functional dependencies like:

$X \rightarrow Y$ for $\mathtt{stat}(X, \mathtt{subject}, Y)$, etc.

What to do if we find two subjects for a statement?

Database issue; not (yet) deductive issue

Issue VI: Aggregation

- Aggregation as source of inconsistencies on the Web

- Not only issue of lost pages, non-accesible sites, but different forms of processing available RDF info

- Not easy to define aggregation (even at elementary levels) in RDF. “aggregate relations” instead of aggregate functions?

- Example: Joe belongs to the class of people having exactly 3 brothers, but no name is specified. There is no consistent way to count the number of Joe brothers.
(Ex. from Fikes, Hayes, Horrock, DQL)

The End

Ideas, comments, pointers, very welcomed

Thanks for your time!