Semantic Web

4. RDF

PD Dr. Matthias Thimm

thimm@uni-koblenz.de

Institute for Web Science and Technologies (WeST)
University of Koblenz-Landau

Some limitations of XML

- What is the relation between “Professor” and “Employee”
- Is “Professor” node in “Lecture” the same as “Professor” in “Seminar”, and does “Professor” has to be in the list of Professors?
- How can I refer to “Kant” on the Web?
- How do I link to external or internal data in/from the document?
- Is it possible to order information and what meaning it has?

XML and semantics 1/2

```
<bank>
  ...
</bank>
```

- a financial institution
- the shoreline of a lake
- a city in Iran

XML and semantics 2/2

- XML is a format for describing semi-structured information
- it has no requirements to
  - use a specific vocabulary
  - preserve the meaning of primitives
- needs standardization to be exchanged
- only feasible for closed collaboration
  - people in a small community
  - web pages on a small intranet

Outline

Outline

RDF

- RDF is a data model
  - it represents semantic relationships in the form of a directed labelled graph
  - the model is domain-neutral, application-neutral, and ready for internationalization
- The RDF data model is a conceptual layer that is, in general, independent of its representation using XML
  - XML is a representation language for RDF
  - RDF might be represented in other languages (Turtle, DB-specific syntax, . . .)

RDF model 1/2

The vocabulary for RDF comprises

- a set U of URIs,
- a set L of *literals*,
- a set B of *blank nodes*

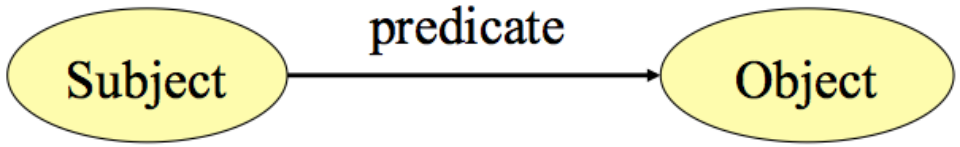
Definition (RDF statement, RDF graph)

An *RDF statement* is a triple $S \in (U \cup B) \times U \times (U \cup L \cup B)$.
An *RDF graph* $\mathcal{G}$ is a finite set of RDF statements.

For an RDF statement $S = (s, p, o)$ the element s is called *subject*, p is called *predicate*, and o is called *object*.

RDF model 2/2

An RDF statement is a triple connecting entities (subject, object) with a predicate (relationship)



- Abstracts from simple sentences like *Bob knows Mary*.
- Smallest unit of information

URIs, literals, blank nodes

The vocabulary of RDF comprises URIs, literals, and blank nodes. More specifically:

- A set U of URIs:
  - also called *resources*
  - A URI uniquely identifies an entity (class, individual, relationship)
  - Everything is a resource: living and non-living objects, attributes, abstract concepts, ...
- A set L of *literals*
  - strings that denote *fixed resources*
  - Used for names, labels, numbers, ...
  - can only be used as objects
- A set B of *blank nodes*
  - placeholder resources with anonymous label
  - used when the resource shall not be named
  - used for abstract concepts and for *reification* (later)
  - can be used for subjects and objects

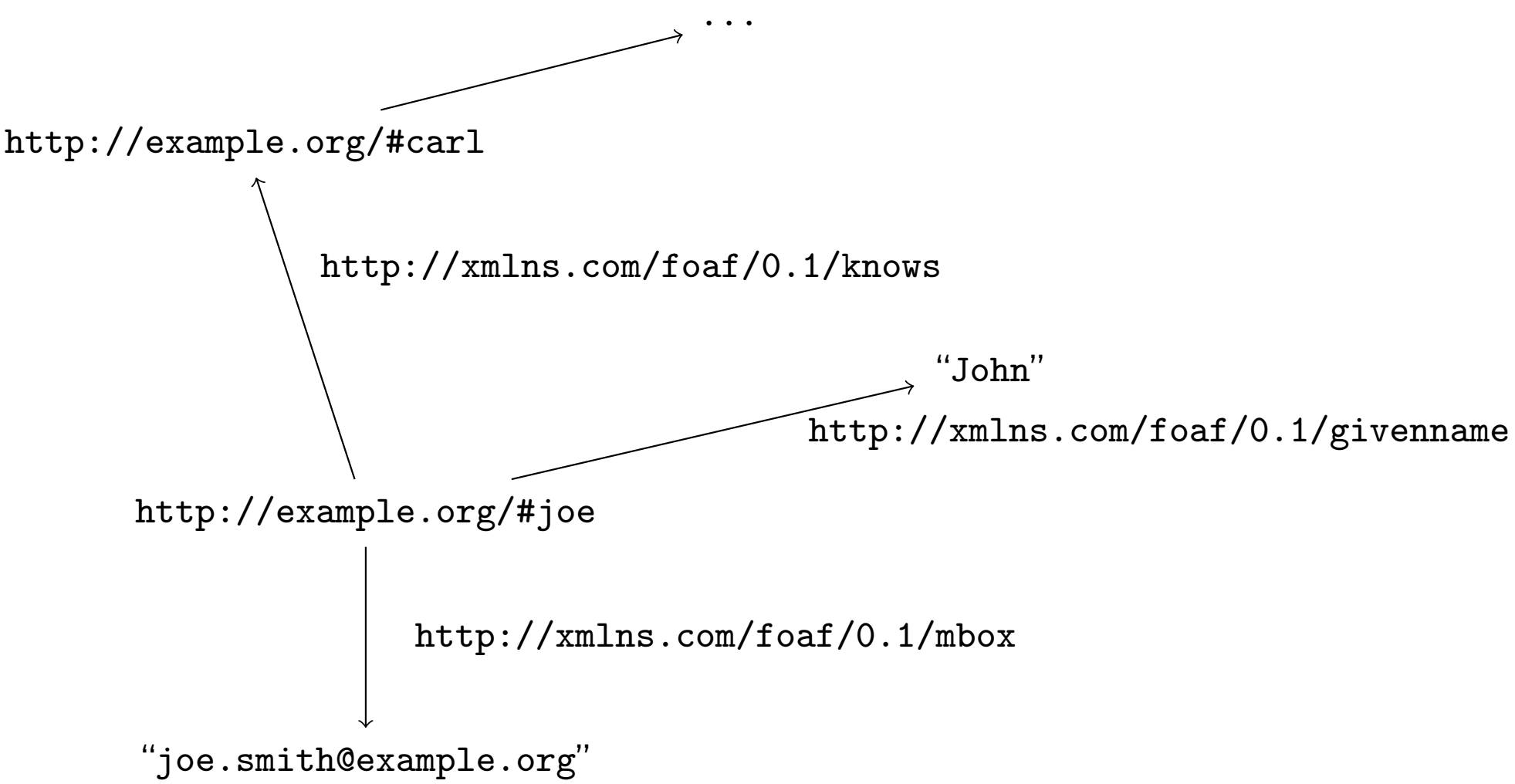
URIs: Best Practices

- (re)Use already known URIs:
  - Search engines: Swoogle, Okkam
- Use URLs as basis
  - Good: `http://west.uni-koblenz.de/#instituteWeST`
  - Bad: `http://ThisSiteDoesNotExist/#instituteWeST`
- Use known standards/conventions for specific types of URIs:
  - Phone number, ISSN, etc.
- Do not use URLs as URIs for people or organizations
  - Bad: `http://west.uni-koblenz.de` for WeST group
  - Better: `http://west.uni-koblenz.de/#groupWeST`
- Derive new URIs from the websites (addresses) you can control
  - Good: `http://west.uni-koblenz.de/#new` for me
  - Bad: `http://west.uni-koblenz.de/#new` for you

Example

- (http://example.org/#joe, http://xmlns.com/foaf/0.1/mbox, “joe.smith@example.org”)
- (http://example.org/#joe, http://xmlns.com/foaf/0.1/givenname, “John”)
- (http://example.org/#joe, http://xmlns.com/foaf/0.1/knows, http://example.org/#carl)

Example 2/2


...
http://example.org/#carl
http://xmlns.com/foaf/0.1/knows
"John"
http://xmlns.com/foaf/0.1/givenname
http://example.org/#joe
http://xmlns.com/foaf/0.1/mbox
"joe.smith@example.org"

Outline

RDF Vocabulary 1/2

- We use namespaces in the same way as for XML
- RDF namesspace:

  `xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"`
- Other standard namespaces:
  - FOAF: `xmlns:foaf="http://www.xmlns.com/foaf/0.1"`
  - For examples: `xmlns:ex="http://www.example.org"`
- Standard RDF vocabulary terms:
  - `rdf:XMLLiteral`: XML literal values
  - `rdf:Property`: class of properties
  - `rdf:Statement`: class of RDF statements
  - `rdf:Alt`, `rdf:Bag`, `rdf:Seq`: containers
  - `rdf:List`: class of RDF Lists
  - `rdf:nil`: the empty list

RDF Vocabulary 2/2

- Standard RDF vocabulary terms (cont'd):
  - `rdf:type`: type of an instance
  - `rdf:first`: first item in a list
  - `rdf:rest`: rest of a list
  - `rdf:value`: for structured values
  - `rdf:subject`: subject of a statement
  - `rdf:predicate` predicate of a statement
  - `rdf:object`: object of a statement

Typed Literals

- (`ex:thisLecture`,`ex:title`, `"Semantic Web"`)
  (untyped)
- (`ex:thisLecture`,`ex:title`, `"Semantic Web"@en`)
  (untyped, but assigned “English” (en) language)
- (`ex:thisLecture`,`ex:title`, `"Semantic Web"^^xsd:string`)
  (explicit type string)

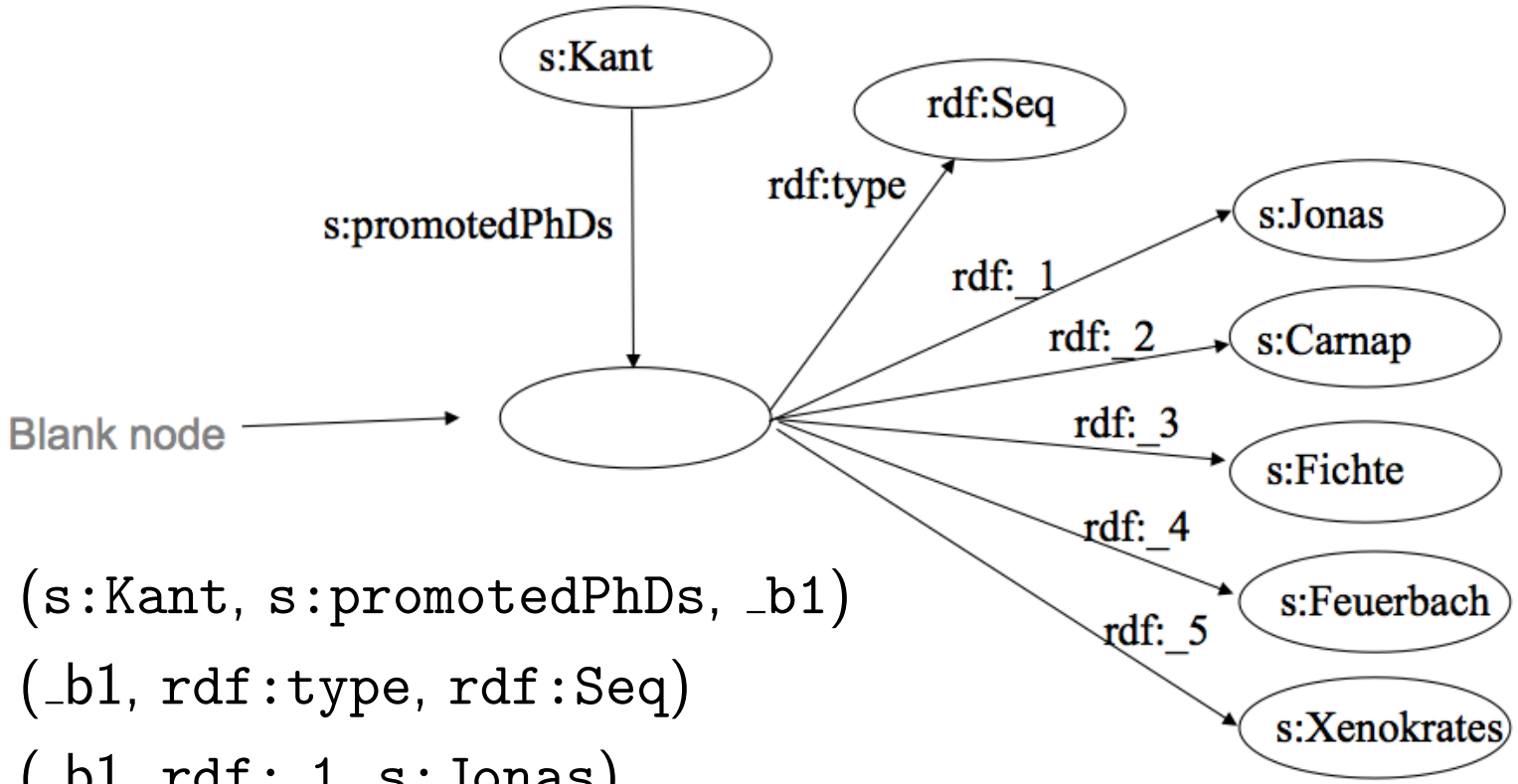
$\rightarrow$ three different literals

Other types:

- `xs:decimal`
- `xs:integer`
- `xs:float`
- `xs:boolean`
- `xs:date`
- `xs:time`

Containers 1/3

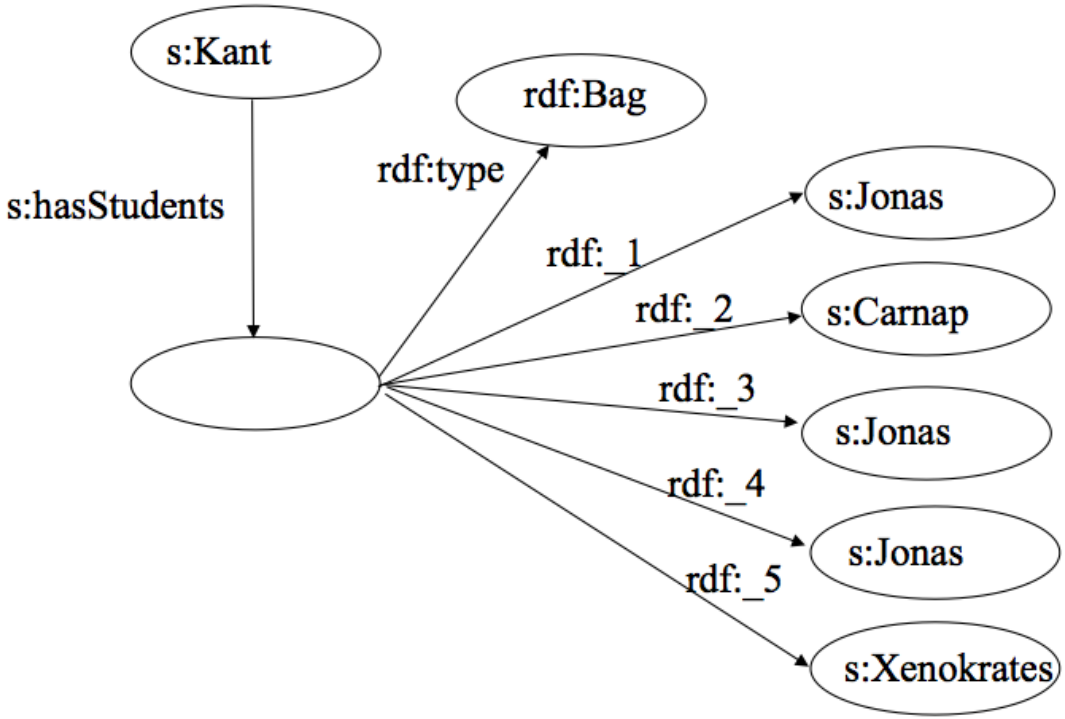
- `rdf:Seq` represent sequences using RDF statements
- order is important, elements may occur more than once
- uses blank nodes



- (`s:Kant`, `s:promotedPhDs`, `_b1`)
- (`_b1`, `rdf:type`, `rdf:Seq`)
- (`_b1`, `rdf:_1`, `s:Jonas`)
- ...

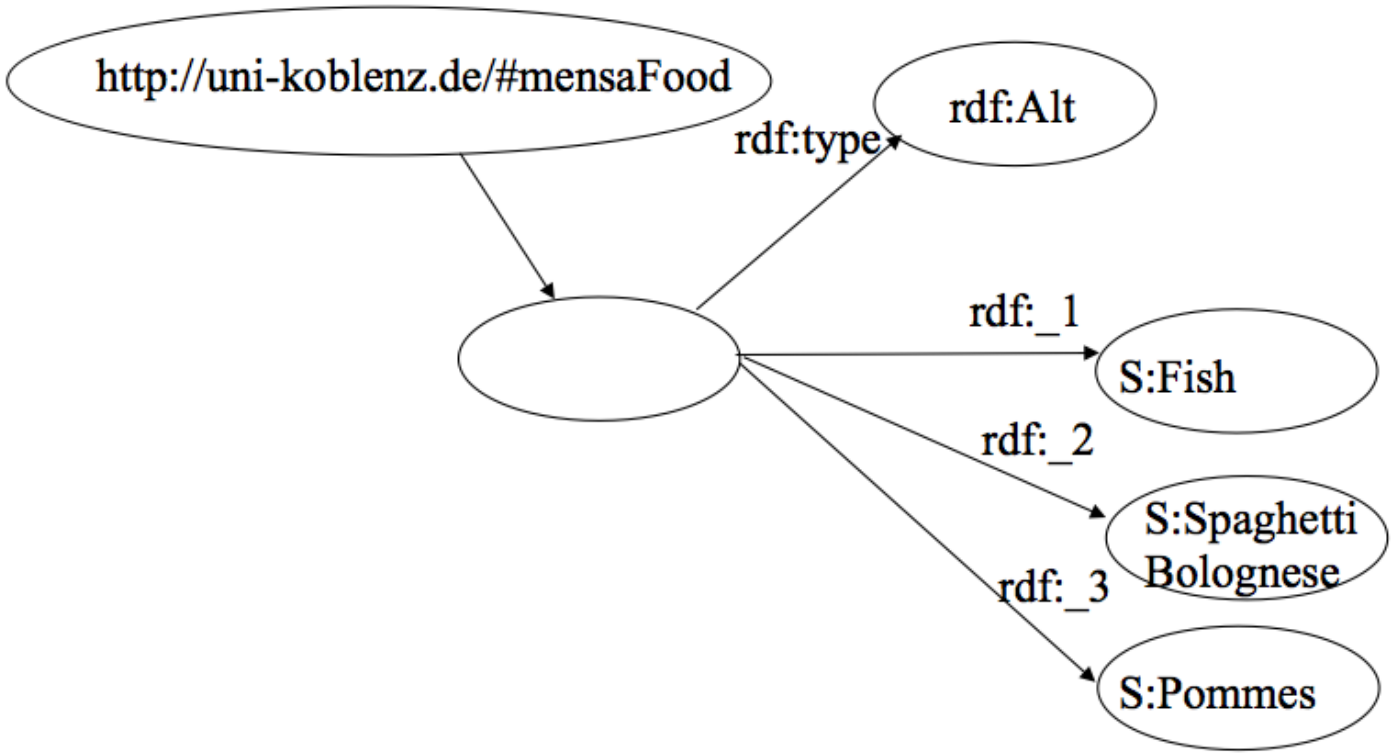
Containers 2/3

- `rdf:Bag` represent bags (multi-sets) using RDF statements
- order is not important, elements may occur more than once
- uses blank nodes

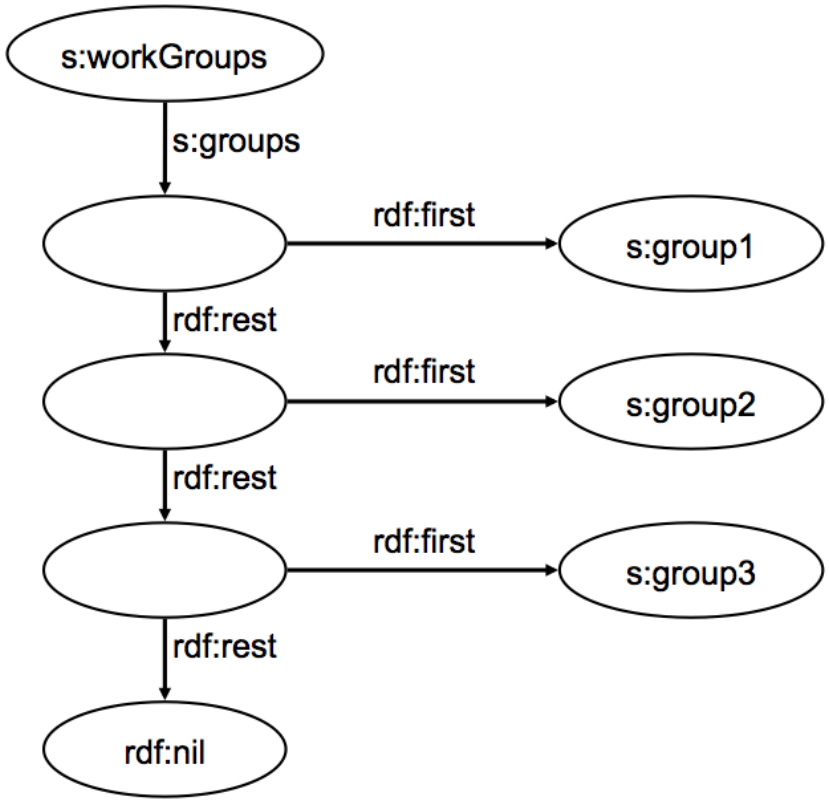


Containers 3/3

- `rdf:Alt` represents a selection
- order is not important
- uses blank nodes

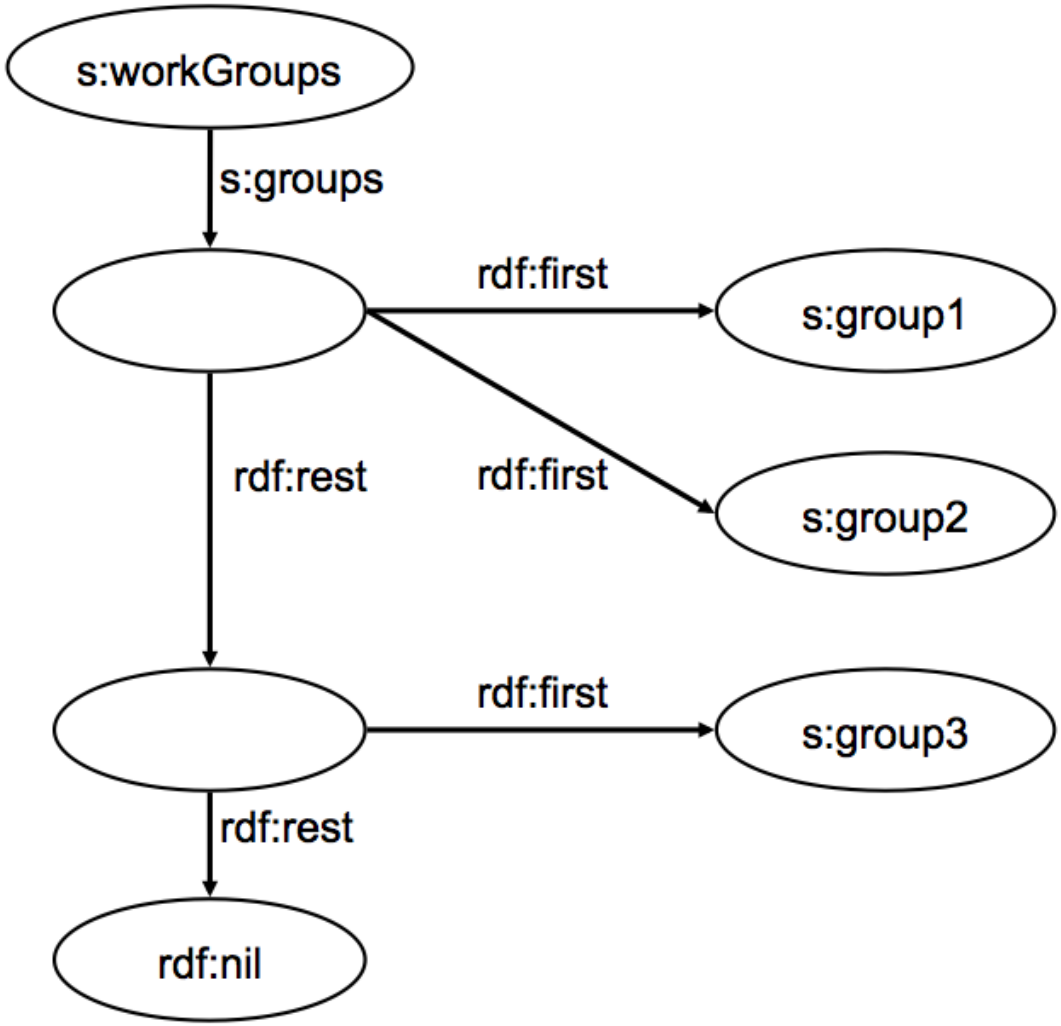


Linked List 1/2



- (`s:workGroups`, `s:groups`, `_b1`)
- (`_b1`, `rdf:first`, `s:group1`)
- (`_b1`, `rdf:rest`, `_b2`)
- (`_b2`, `rdf:first`, `s:group2`)
- ...

Linked List 2/2



- meaningless representation

Reification 1/4

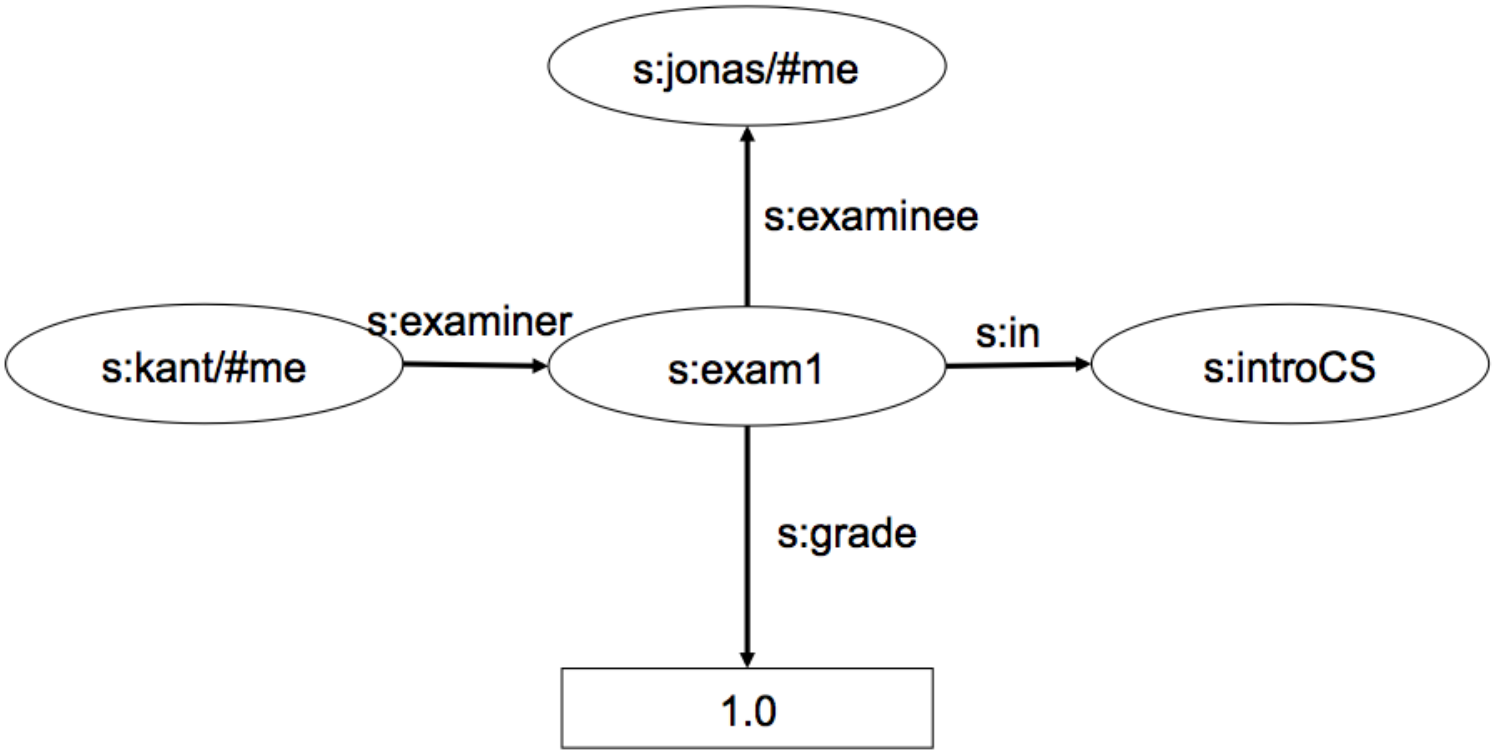
- Problem: How to represent the statement
  "Kant" examined "Jonas" in class "Introduction to CS" and gave him grade "1.0"
- needs more arguments than a triple has
- How to represent n-ary predicates?

Solution: Reification

- Refers to situation in natural language where statement is transformed so actions and events in it become quantifiable
- Here: "Jonas exam" becomes a described object
- Types of reification:
  - Ad-hoc Reification
  - RDF Reification
  - Named Graphs
  - Reification using other Design Patterns

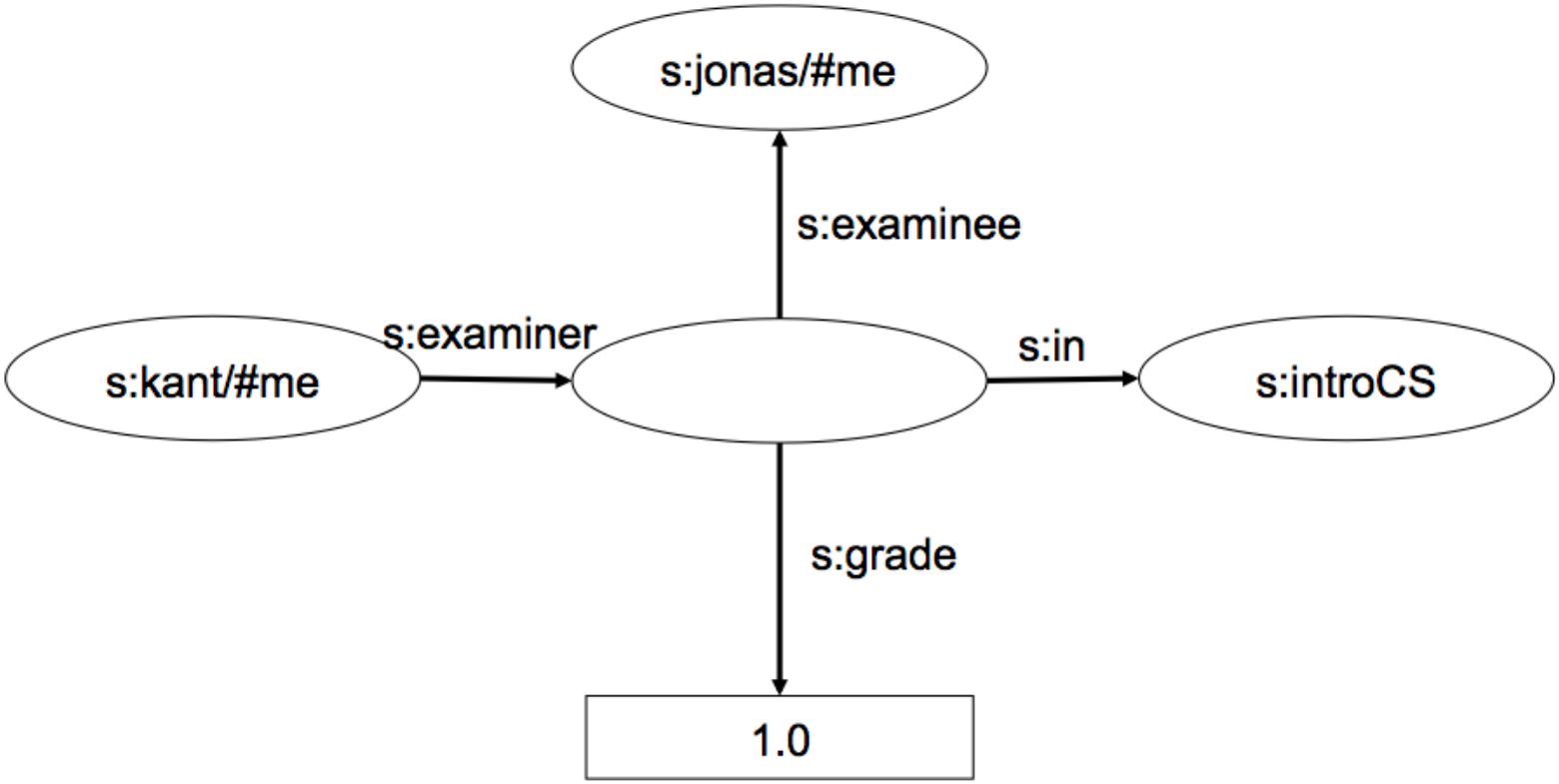
Reification 2/4: Ad-hoc reification (direct)

"Kant" examined "Jonas" in class "Introduction to CS" and gave him grade "1.0"



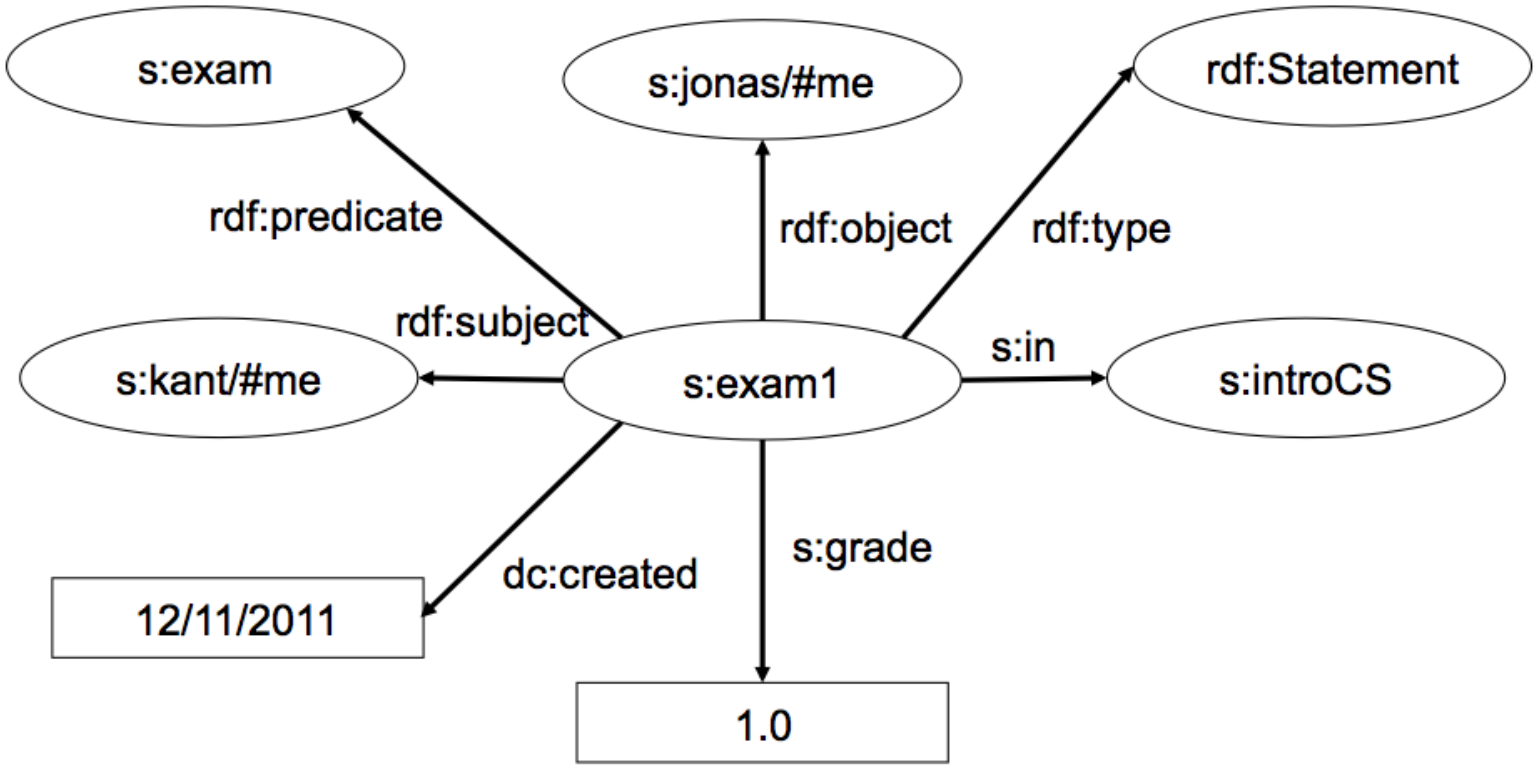
Reification 3/4: Ad-hoc reification (with blank node)

"Kant" examined "Jonas" in class "Introduction to CS" and gave him grade "1.0"



Reification 4/4: RDF reification

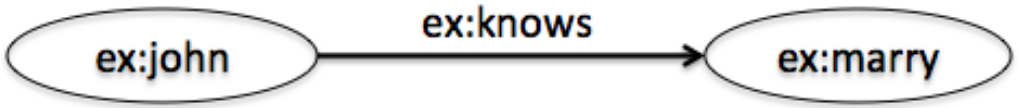
"Kant" examined "Jonas" in class "Introduction to CS" and gave him grade "1.0"



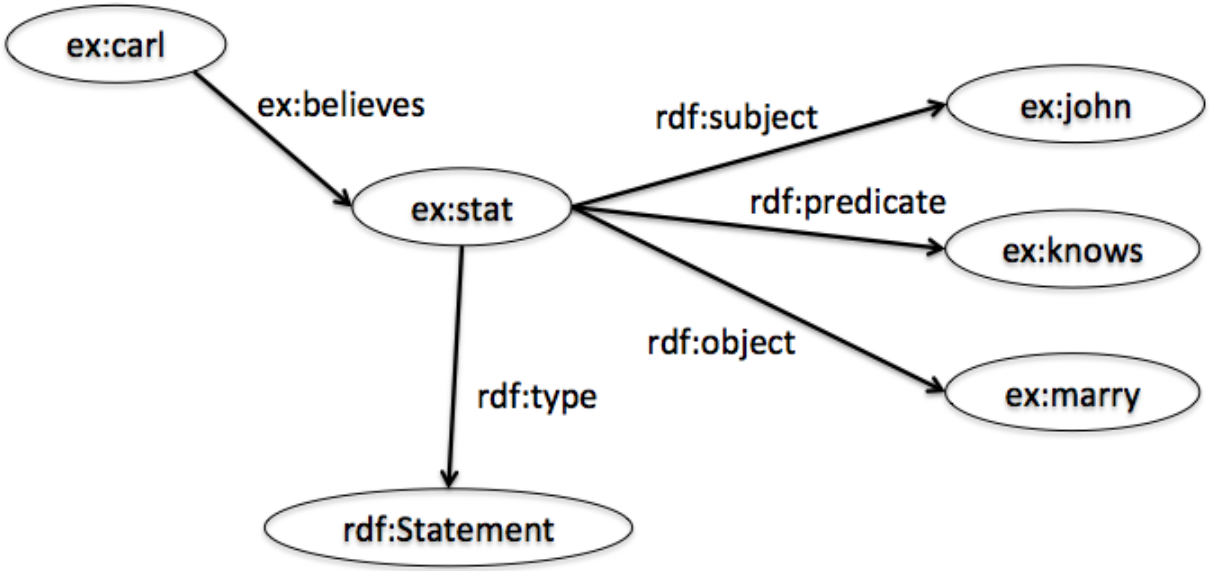
RDF Statements 1/3

Using the principle of reification and the property `rdf:Statement` one can also use RDF to describe RDF.

"John knows Marry":

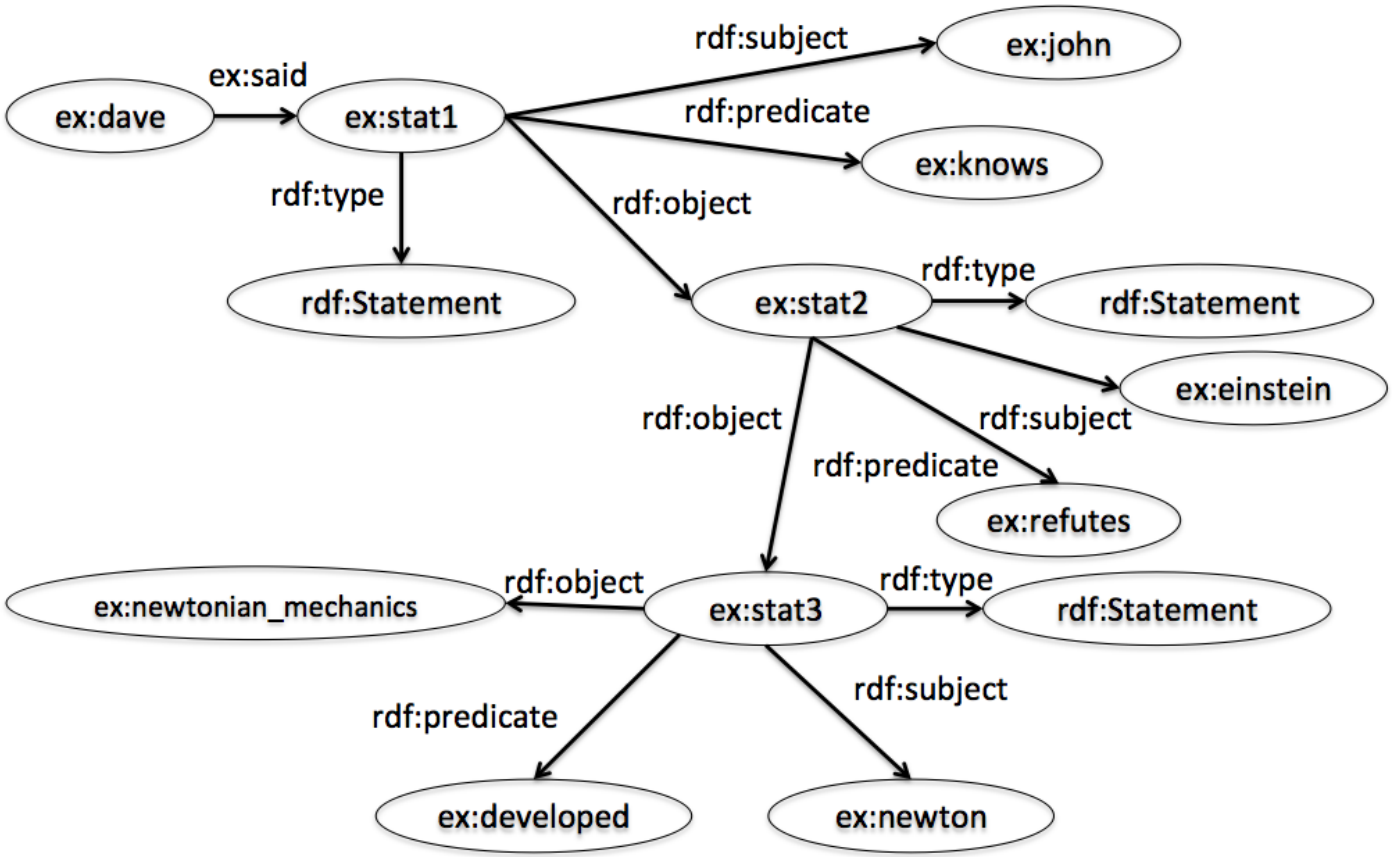


"Carl believes that John knows Marry":



RDF Statements 2/3

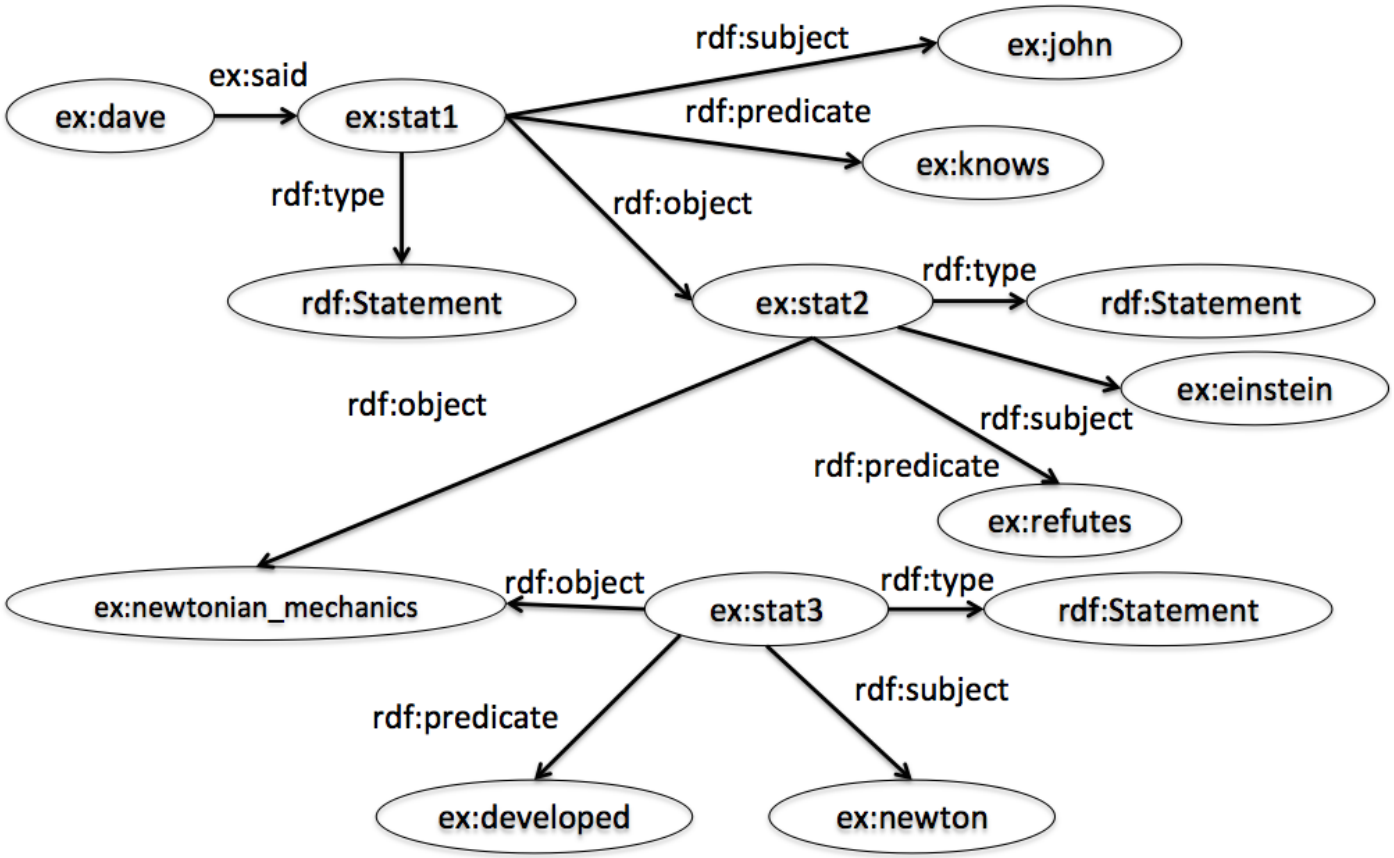
Arbitrary complex statements can be nested.



What does this statement say?

Is this what we want it to say?

RDF Statements 3/3



Outline

RDF Encodings

- The RDF model just describes a labeled directed graph (set of triples)
- It does not prescribe the format how a set of triples is stored
- Possibilities:
  - Graphical notation (informal)
  - triple-based notation (informal): `(subject, predicate, object)`
  - Turtle (formal)
  - RDF/XML (formal)
  - N3 notation (formal)

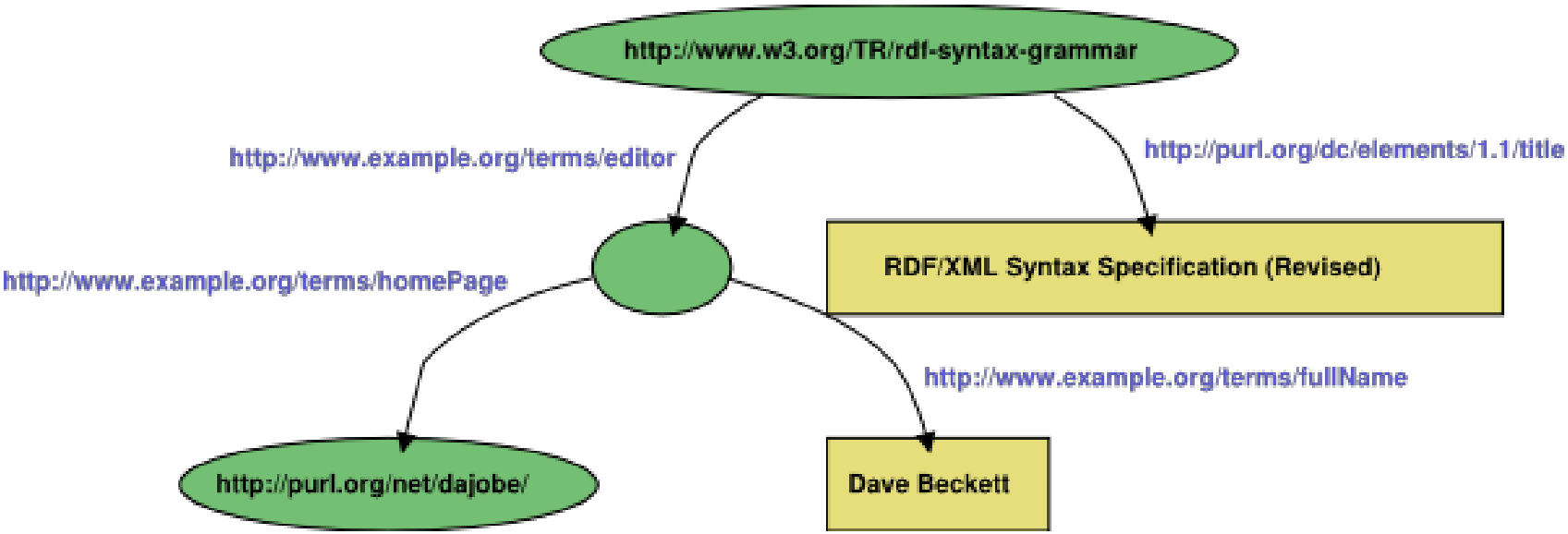
Turtle

Example of RDF in Turtle:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

:mary rdf:type <http://www.ex.org/Gardener>.
:mary :worksFor :ElJardinHaus.
:mary :name "Dalileh Jones"@en.
_:john :worksFor :ElJardinHas.
_:john :idNumber "54321"^^xsd:integer.
```

RDF/XML 1/2


http://www.w3.org/TR/rdf-syntax-grammar
http://www.example.org/terms/editor
http://purl.org/dc/elements/1.1/title
http://www.example.org/terms/homePage
RDF/XML Syntax Specification (Revised)
http://www.example.org/terms/fullName
http://purl.org/net/dajobe/
Dave Beckett

RDF/XML 2/2

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:ex="http://example.org/stuff/1.0/">
  <rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar"
   dc:title="RDF/XML Syntax Specification (Revised)">
    <ex:editor>
      <rdf:Description ex:fullName="Dave Beckett">
         <ex:homePage rdf:resource="http://purl.org/net/dajobe/" />
      </rdf:Description>
    </ex:editor>
  </rdf:Description>
</rdf:RDF>
```

Outline

Summary

- The RDF Model
  - URIs, literals, blank nodes
  - statements, graphs
- Complex statements
  - Typed literals
  - Containers: sequences, bags, alternatives
  - Linked lists
  - Reification
- RDF encodings: Turtle, RDF/XML

Pointers to further reading

- RDF/XML Syntax Specification:
  `http://www.w3.org/TR/REC-rdf-syntax/`
- Aidan Hogan. Exploiting RDFS and OWL for Integrating Heterogeneous, Large-Scale, Linked Data Corpora. PhD thesis, National University of Ireland, Galway, 2011.