

## Guide to the Set of Slides on Remaining RDF Topics

These slides cover four rather unrelated topics (which we can identify as sections 13-16) and complete the coverage of RDF. (The next topic will be RDF Schema, which allows us to define vocabularies.)

The first topic is containers, which allow us to refer to groups. A container is a resource containing members, which are themselves resources or literals. There are three kinds of containers.

- A Bag may contain duplicate members, and there is no significance to the order of the members.
- A Sequence (or Seq) may also have duplicate members, but the order of its members is significant.
- The members of an Alternative (or Alt) are alternatives (usually for a single value of a property)

There is no way to indicate that the members specified for a container are all the members that the container in fact has. In contrast, collections let us describe groups that contain only the specified members. A collection is similar to a list in some programming languages (but a collection is a description, not a construct).

Reification lets us describe other RDF statements (e.g., when the statement was made and by whom) but is not (quite) the same as quotation. The reification vocabulary introduces the type `rdf:Statement` and (to refer to the parts of a statement/triple) the properties `rdf:subject`, `rdf:predicate`, and `rdf:object`. All the examples here are given in RDF/XML since the community that pushes the N3 notation considers reification to be largely out of date.

Although RDF directly supports only binary relations, we earlier saw how `bnodes` allow us to represent relations of higher arity. A special case is where a quantity is qualified with a unit of measure (as in “5 kilograms”). Typically, this involves a typed literal (e.g., using `rdf:datatype="&xsd:decimal"`). We need a separate resource to represent the structured value as a whole and to serve as the object in a triple. This resource needs a property for the typed literal (the main value) and a property for the unit. RDF provides the predefined `rdf:value` property to describe the main value. How to represent the unit is up to the particular application. Proper use of `rdf:value` allows us to represent quantities using any units of measure as well as values taken from different classification schemes or rating systems

Finally, sometimes the value of a property needs to be a fragment of XML. To facilitate writing XML literals, RDF/XML provides a third value of the `rdf:parseType` attribute: given an element, the attribute `rdf:parseType="Literal"` indicates that the contents of the element are to be interpreted as an XML fragment. Using `rdf:parseType="Literal"`, an RDF user avoids dealing directly the various transformations required for representing XML fragments in the accepted (“canonical”) way.