Description of Information Resources: RDF/RDFS (an Introduction)

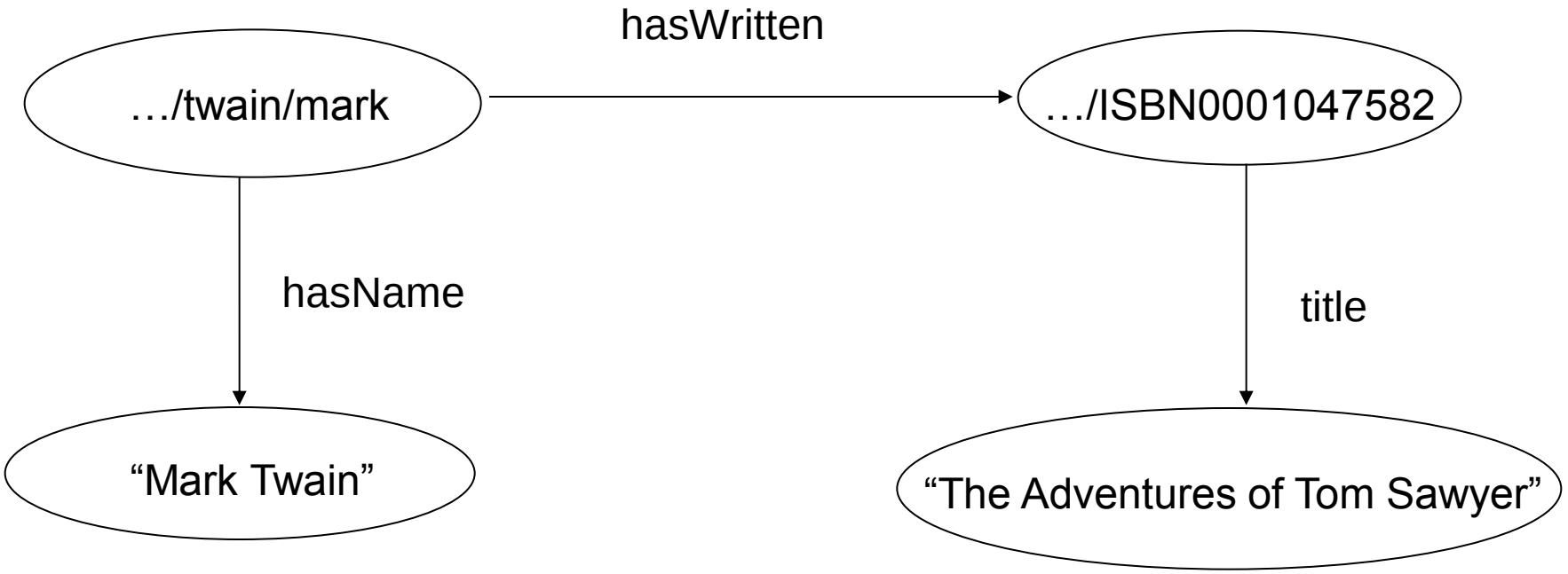
RDF

RDF (Resource Description Framework) is a W3C recommendation originally designed to standardize the definition and use of metadata descriptions of Web-based resources. However, RDF is equally well suited to representing arbitrary data, whether they are metadata or not.

The basic building block in RDF is an object-attribute-value triple, commonly written as A(O;V). That is, an object O has an attribute A with value V.

Another way to think of this relationship is a labeled edge between two nodes: *[O]_A![V].* This notation is useful, because RDF allows objects and values to be interchanged. Thus, any object from one triple can play the role of a value in another triple, which amounts to changing two labeled edges in a graphic representation.

Example RDF data graph, capturing three statements:



The graph in the figure expresses the following relationships:

hasName(‘http://www.famouswriters.org/twain/mark/’, “Mark Twain”)

hasWritten(‘http:// www.famouswriters.org/twain/mark/’, ‘http://www.books.org/ISBN0001047582’)

title(‘http://www.books.org/ISBN0001047582’, “The Adventures of Tom Sawyer”)

RDF also allows a form of reification in which any RDF statement itself can be the object or value of a triple. This means graphs can be nested as well as chained. On the Web this allows us, for example, to express doubt of support for statements created by other people.

Finally, it is possible to indicate that a given object is of a certain type, such as stating that “ISBN0001047582” is of the type Book, by creating a type edge referring to the Book definition in an RDF schema:

```
type(‘http://www.books.org/ISBN0001047582’,
‘http://www.description.org/schema#Book’)
```

The RDF Model and Syntax Specification also proposes an XML syntax for RDF data models. One possible serialization of the above relations in this syntax would look like this:

```
  <rdf:Description
rdf:about="http://www.famouswriters.org/twain/mark">
  <s:hasName>Mark Twain</s:hasName>
  <s:hasWritten
rdf:resource="http://www.books.org/ISBN0001047582"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.books.org/ISBN0001047582">
  <s:title>The Adventures of Tom Sawyer</s:title>
  <rdf:type
rdf:resource="http://www.description.org/schema#Book"/>
  </rdf:Description>
```

Since the proposed XML syntax allows many alternative ways of recording information (and indeed still other syntaxes may be introduced), the above XML syntax is just one of many possibilities for writing an RDF model in XML. It is important to note that RDF is designed to provide a basic object-attribute-value model for Web data.

Other than this intentional semantics – described only informally in the standard – RDF makes no commitments with respect to data modeling. In particular, no reserved terms are defined for further data modeling. As with XML, the RDF data model provides no mechanisms for declaring property names that are to be used.

RDF Schema

RDF Schema is a mechanism that lets developers define a particular vocabulary for RDF data (such as hasWritten) and specify the kinds of objects to which these attributes can be applied (such as Writer).

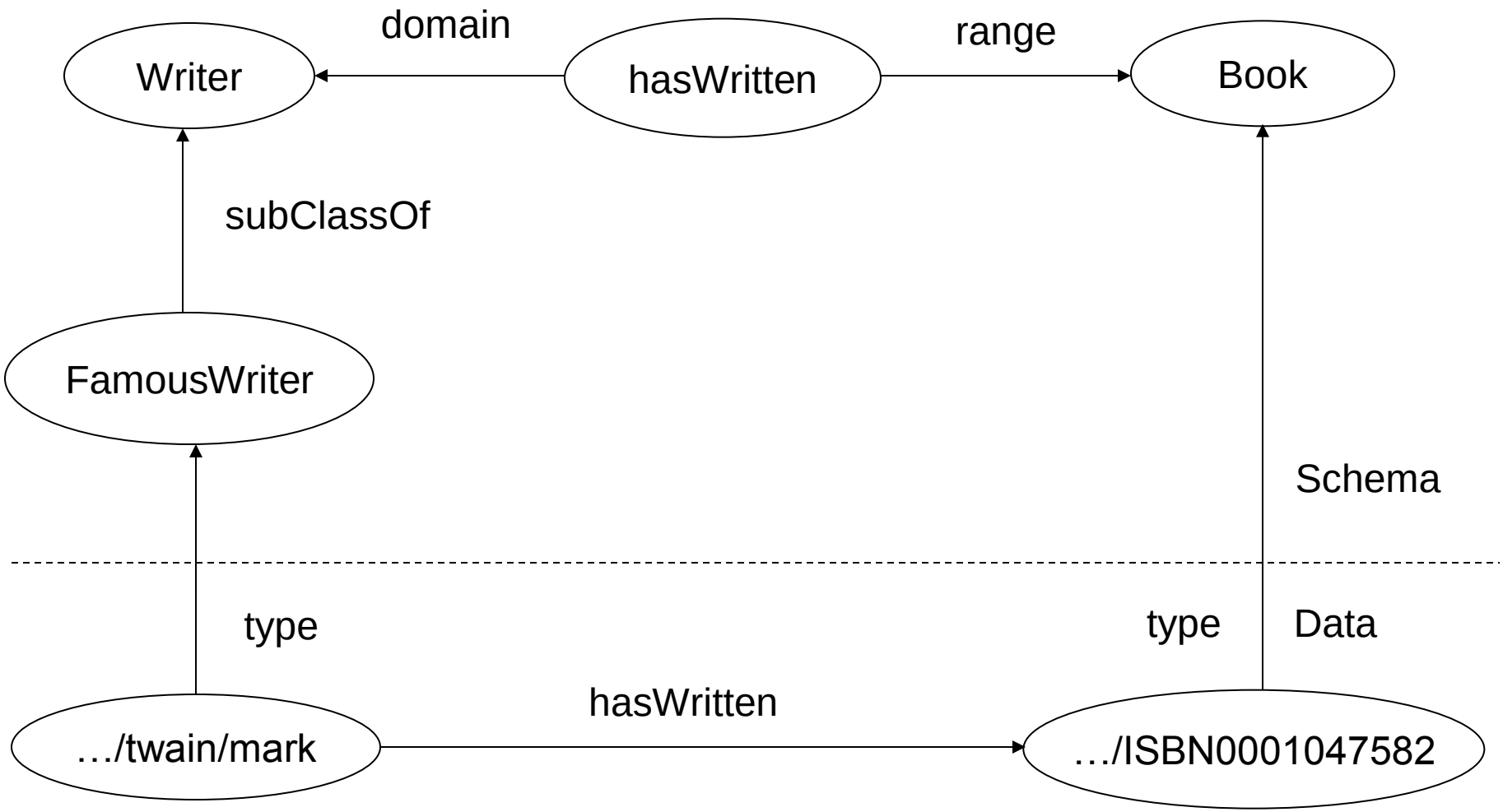
RDF Schema does this by prespecifying some terminology, such as Class, subClassOf, and Property, which can then be used in application-specific schemata.

RDF Schema expressions are also valid RDF expressions – in fact, the only difference between RDF Schema expressions and “normal” RDF expressions is that in RDF Schema an agreement is made on the *semantics* of certain terms and thus on the *interpretation* of certain statements.

For example, the subClassOf property allows the developer to specify the hierarchical organization of classes. Objects can be declared to be instances of these classes using the type property. Constraints on the use of properties can be specified using domain and range constructs.

Above the dotted line in the next figure, we see an example RDF schema that defines vocabulary for the RDF example we saw earlier: Book, Writer, and FamousWriter are introduced as classes, and hasWritten is introduced as a property. A specific instance is described in terms of this vocabulary below the dotted line.

Example RDF schema, defining vocabulary and a class hierarchy:



Summary

RDF documents and RDF schemata can be considered at three different levels of abstraction:

1. At the *syntactic* level they are XML documents.
2. At the *structure* level they consist of a set of triples.
3. At the *semantic* level they constitute one or more graphs with partially predefined semantics.