at figure 2.6. We presented this figure as displaying a class/property hierarchy plus instances, but it is, of course, itself simply a labeled graph that can be encoded in RDF. Remember that RDF allows one to express any statement about any resource, and that anything with a URI can be a resource. So, if we wish to say that the class "apartment" is a subclass of "residential unit," we may.

1. Define the required resources for apartment, residential unit, and subClassOf;
2. define subClassOf to be a property;
3. write the triple (apartment subClassOf residential unit).

All these steps are within the capabilities of RDF. So, an RDFS document is just an RDF document, and we use one of the standard syntaxes for RDF.

Now we define the modeling primitives of RDF Schema.

2.5.1 Core Classes

The core classes are

rdfs:Resource, the class of all resources

rdfs:Class, the class of all classes

rdfs:Literal, the class of all literals (strings)

rdf:Property, the class of all properties

rdf:Statement, the class of all reified statements

2.5.2 Core Properties for Defining Relationships

The core properties for defining relationships are

rdf:type, which relates a resource to its class (see section 2.4.1). The resource is declared to be an instance of that class.