- Required values. Often, classes are defined by virtue of a certain property's having particular values, and such required values can be specified in OWL, using owl:hasValue. Sometimes the requirements are less stringent: a property is required to have some values from a given class (and not necessarily a specific value, i.e., owl:someValuesFrom).

- Relational characteristics. The final family of facets concerns the relational characteristics of properties: symmetry, transitivity, inverse properties, and functional values.

After this step in the ontology construction process, it will be possible to check the ontology for internal inconsistencies. (This is not possible before this step, simply because RDF Schema is not rich enough to express inconsistencies.) Examples of often occurring inconsistencies are incompatible domain and range definitions for transitive, symmetric, or inverse properties. Similarly, cardinality properties are frequent sources of inconsistencies. Finally, requirements on property values can conflict with domain and range restrictions, giving yet another source of possible inconsistencies.

7.2.7 Define Instances

Of course, we rarely define ontologies for their own sake. Instead we use ontologies to organize sets of instances, and it is a separate step to fill the ontologies with such intances. Typically, the number of instances is many orders of magnitude larger than the number of classes from the ontology. Ontologies vary in size from a few hundred classes to tens of thousands of classes; the number of instances varies from hundreds to hundreds of thousands, or even larger.

Because of these large numbers, populating an ontology with instances is typically not done manually. Often, instances are retrieved from legacy data sources such as databases as discussed in Section 7.6. Another often used technique is the automated extraction of instances from a text corpus.