We used a tool called MappingMaster to map the contents of our X spreadsheet into OWL. MappingMaster is an open source Protege-OWL plugin for importing spreadsheet data into OWL ontologies. It provides a domain specific language (DSL) to define these mappings. The language is based on the Manchester syntax [X] for declaratively specifying OWL ontology constructs. It extends the Manchester syntax to support references to spreadsheet content in these construct declarations.

In the MappingMaster DSL, any clause in a declaration that refers to an OWL class, OWL property, OWL individual, or a data value can be substituted with a reference to one or more cells within a spreadsheet.

Any declarations containing such references are preprocessed and the relevant spreadsheet content specified by these references is imported.

These references may refer to a single cell or may be compound and refer to multiple cells. Each processed declaration has the appropriate spreadsheet content substituted for each reference and can then be processed by a standard Manchester parser. For example, the following declaration in the Manchester syntax defines the necessary and sufficient condition on class C that specifies that it has a property p with a value that is an individual of class B:

Class: C EquivalentTo: (p some B)

If, instead of a pre-existing class B, we wanted to use the MappingMaster DSL to specify that the class to come from column D, row 4 in a particular spreadsheet we could write:

Class: C EquivalentTo: (p some @D4)

The @ prefix identifies a reference and is typically followed by a standard Excel-style cell specification. If cell D4 contains, say, the value "blood", MappingMaster will by default create a class and use the cell value as its rdfs:label annotation. The generated class will then be substituted for the reference @D4 in the declaration and the expression will be processed as usual.

In most cases, we will wish to iterate over multiple cells in a spreadsheet.

To support this, MappingMaster provides the ability to iterate over multiple rows or columns. The wildcard character '\*' can then be used in references to refer to the current column or row in an iteration. For example, if we wished to extend the previous example to import classes from rows 1 to 10 in column D we can use MappingMaster to specify this range and then modify the expression with a wildcard:

Class: C EquivalentTo: (p some @D\*)

Here, a new necessary and sufficient condition will be created for each row with the appropriate value substituted for the reference @D\*.

This process effectively allows OWL classes, properties or individuals to be created from arbitrary spreadsheet data. The data can be used as the names of the created entities or can be used to annotate these entities.

Data literals can also be imported in the same way. The use of the Manchester syntax also allows created from spreadsheet content can be related to each other in complex ways. Since the Manchester syntax supports the full OWL specification, very complex interrelationships can be specified.

Optional details:

. Specifying superclasses of imported entities

. Missing values

. rdfs:label encoding vs. rdf:id encoding

. Reusing existing ontology entities if cell content refers to them

. Perhaps a detailed example of a real clause with screenshot of relevant spreadsheet