# Introduction

In pursuit of the goal of making things from existing parts, we need to reuse existing specifications in the construction of domain-wide specifications such as the [Financial Industry Business Ontology](http://www.hypercube.co.uk/docs/20110623_fibo_background.pdf). To achieve maximum flexibility we need to manage reuse so that only the germane portions of the referenced specifications are incorporated. From our previous experience with making things from parts, the [Law of Demeter](http://en.wikipedia.org/wiki/Law_of_Demeter) provides guidance on controlling coupling between the parts. This document describes a technique for accomplishing these goals.

The model is composed of several components. The domain of the model is a small set of people: Arthur, Barbara, Charles, Debra, Edward and Felicia. There are two genders defined and each person has one of the other of them. These facts are all defined in people.ttl. There are two subsets of the model. One, female.ttl, contains the subset of facts about people that relate to female members of people.ttl. The other, male.ttl, contains the subset of facts about people that relate to female members of people.ttl. These two subsets are recombined in composite.ttl via owl:import.

|  |  |  |
| --- | --- | --- |
| **Prefix** | **Base URI** | **File** |
| people | <http://data.jpmchase.net/pedagogy/2011/11/04/people> |  |
| female | <http://data.jpmchase.net/pedagogy/2011/11/04/people/female> |  |
| male | <http://data.jpmchase.net/pedagogy/2011/11/04/people/male> |  |
| composite | <http://data.jpmchase.net/pedagogy/2011/11/04/people/male> |  |

# How the models were constructed

people.ttl was constructed by hand. SPARQL CONSTRUCT and INSERT INTO queries were used to create the other models. TopBraid Composer Free Edition was used as the editing tool and SPARQL execution environment. Some minor anomalies were observed. In comparison with people.ttl, female.ttl and male.ttl, composite.ttl indicates that there are 4 instances of Gender rather than 2, although the Instances pane only shows 2.



The results of the following SPARQL query revealed the cause.

|  |
| --- |
| SELECT ?graph ?gender WHERE { GRAPH ?graph { ?gender a <http://data.jpmchase.net/pedagogy/2011/11/04/people#Gender> . } . } ORDER BY ?graph ?gender |
| **graph** | **gender** |
| <http://data.jpmchase.net/pedagogy/2011/11/04/people> | people:Female |
| <http://data.jpmchase.net/pedagogy/2011/11/04/people> | people:Male |
| <http://data.jpmchase.net/pedagogy/2011/11/04/people/composite/flat> | people:Female |
| <http://data.jpmchase.net/pedagogy/2011/11/04/people/composite/flat> | people:Male |
| <http://data.jpmchase.net/pedagogy/2011/11/04/people/female> | people:Female |
| <http://data.jpmchase.net/pedagogy/2011/11/04/people/female> | people:Male |
| <http://data.jpmchase.net/pedagogy/2011/11/04/people/male> | people:Female |
| <http://data.jpmchase.net/pedagogy/2011/11/04/people/male> | people:Male |

As you can see, the same two definitions of people:Female and people:Male occur in four different graphs, two of which are imported into composite.ttl only one of which supplies instances for the instance pane.

# Manipulating the Model with Other Tools

In an attempt to determine which, if any, of the behaviors were specific to the TopBraid tool, I tried using the current production release of Protégé. It could not open composite.ttl. It didn’t fail, it just didn’t complete opening the file and loading the model. To overcome this problem I created flat.ttl from composite.ttl. Flat.ttl contains no imports. Everything is defined in a single file. Protégé did much better with this file, but it lacks SPARQL support. Fortunately, flat.ttl was well suited to processing flat.ttl from the command line. The results are show in the following figure.

|  |
| --- |
| bat\sparql --namedGraph=C:\TBCME-TinyWorkArea\EMDCOMG\flat.ttl "SELECT ?graph ?gender WHERE { GRAPH ?graph { ?gender a <http://data.jpmchase.net/pedagogy/2011/11/04/people#Gender> . } . } ORDER BY ?graph ?gender" |
| **graph** | **gender** |
| <C:%5CTBCME-TinyWorkArea%5CEMDCOMG%5Cflat.ttl> | <http://data.jpmchase.net/pedagogy/2011/11/04/people#Female> |
| <C:%5CTBCME-TinyWorkArea%5CEMDCOMG%5Cflat.ttl> | <http://data.jpmchase.net/pedagogy/2011/11/04/people#Male> |

These results are identical to the results of the same query being executed against the people.ttl graph.

# Conclusion

1. Through the judicious choice and use of named graphs we can decompose larger specifications into smaller more easily reused modules while addressing coupling between specifications
2. There are a number of minor problems with the existing toolset that need not limit the application of this technique at scale.