Manual Ontology Alignment

Manual ontology alignment is the process by which a user can see one ontology for related terms even though there are multiple underlying ontologies (VCARD and FOAF in this case). The user is thus presented a unified view of the underlying ontologies. There are various steps in the process of manual ontology alignment:

1. Construction of meta-TBox: The first step is the construction of a meta-TBox (ALIGNMENT.owl) that allows us to relate 1...N ontologies. For the process of manual ontology alignment we construct the meta-TBox with properties such as equivalent, subset, superset etc.

2. Relating VCARD and FOAF: The next step is to use the properties from the meta-TBox to construct an ABox that relates terms from the VCARD and FOAF ontologies. An example relationship in this ABox would be, VCARD:Name ALIGNMENT:equivalent FOAF:name. All related classes and properties between VCARD and FOAF are mapped in this same way. Each such assertion is reified with the user who made the assertion as well as the particular alignment that this assertion belongs to.

3. Implementation of JUnit test: This step tests the process of manual alignment between FOAF and VCARD with a JUnit test in Blackbook. The process has two sub-processes:
   - De-homogenization: This is the process of taking a user request for one or a set of ontologies and mapping it to instances of the meta-TBox to select the appropriate ontology to represent the request. This translated request is then federated across data stores using the appropriate ontology. In Blackbook, this process takes the input request and based on the mapping changes the parameters of the request before it is federated across datasources.
   - Homogenization: This is the process of representing the data returned from the federated query using the appropriate ontology as specified by the user. In Blackbook, this process takes place when a URI is materialized.

These two algorithms are implemented in Blackbook with Aspect Oriented Programming (AOP) by using an interceptor. The JUnit test is written as an extension of the JUnit tests provided in Blackbook to test the AlgorithmManager.

We have finished the implementation of the De-homogenization algorithm in Blackbook and will finish the implementation of the Homogenization algorithm by the end of the month.