The folder Test (https://svn.kwarc.info/repos/MMT/src/OWLMMT/Test) contains small ontologies in subdirectories per test case.

For example, a subdirectory called ObjectPropertyAxiom contains the following six files:

objectPropertyAxiom.owl : the input file of the translation from OWL to OMDoc

objectPropertyAxiom.omdoc : the output file of the translation from OWL to OMDoc and also the input file of the translation from OMDoc to OWL

objectPropertyAxiomToOWL.owl : the output file of the translation from OMDoc to OWL

objectPropertyAxiomChanged.doc: the second version of the ontology in the file objectPropertyAxiom.omdoc

objectPropertyAxiomDiff.xml: the output file of the MMTDiff which compares objectPropertyAxiom.omdoc and objectPropertyAxiomChanged.doc

Translation from OWL to OMDoc

Note for Florian:
We need to write <?xml version="1.0"?> in OMDoc files automatically when we translate OWL files to OMDoc.

Comparing versions of ontologies

We applied MMTDiff to the test ontologies to find the diff between versions of the ontologies.

MMT Diff: (add, delete, rename, update, identical)

*In baseDiff:
individual James is renamed as John, and individual Bob is added to the ontology.
MMTDiff outputs that James is renamed as John, and
James is also renamed as Bob instead of Bob is added.

*In objectPropertyAxiomDiff:
objectProperty hasParent and objectProperty hasWife are renamed as hasparent and haswife, respectively.
MMTDiff outputs that hasparent and haswife are added, and hasParent and hasWife are deleted.

*If we rename at least two constants which have the same type, MMTDiff outputs that they are added instead of they are renamed.
(when we rename constants which have different types, MMTDiff outputs that they are renamed.)

Change Impact Analysis

(identifying consequences of changes and modifications to accomplish changes.)

*In objectPropertyAxiomDiff:
the type of the class Man is changed to an objectProperty, so the axiom which includes the constant Man should complain about the type.

If we change the type of a constant, then the axiom which includes it should complain about the type.

* Florian wants to work on the change impact analysis later, because Maria (undergraduate student)will also join us. So, we skipped this part and we will work on it later.

Computing the changes between ontologies
(A paper written by Timothy Redmond and Natasha Noy)
It describes a highly optimized pluggable difference engine that searches for alignments between entities in different ontology versions and applies those alignments to display the differences in the ontologies.

*aligns entities when the names of entities have changed
*understands structures in OWL2 ontology
  calculates structural differences
  handles refactor operations
*first compares entities, then uses the result to compare axioms
*alignment: it maps entities and axioms
*two phases: alignment phase and explanation phase
    Alignment phase:
    *determines the diff between signatures of two ontologies
    *calculates refactors
    *identifies axioms added or removed
    *as it aligns entities, it keeps track of axiom mappings
    *heuristic algorithms for changes in names of entities
    *priority for algorithms

    Explanation Phase:
    *reorganizes and represents axiom changes into a more readable format

it does not support:
*detecting when an axiom has changed
*merging and detecting conflicts
*conflict resolution
*entailment

future work:
*porting more of the algorithms used by PROMPTDIFF to the difference engine

Visualizing differences of ontologies
*Prompt provides two views:
1)Tree view presents the result of comparing two versions of an ontology in a single tree.
A Class is deleted - crossed out in red
      added – underlined in blue
      renamed or changed – in bold
a warning icon shows a subtree which has some changes
a tooltip shows old name of a renamed class
2) Table view shows all frames in two compared ontologies and lists similarities and differences.