Concurrent Ontology Editing

These are a few notes from my trials with using version control systems (VCS) for concurrent ontology editing.

First, I should note the size of the ontology that I have been experimenting with. The FMA is about 223M when serialized in RDF/XML. This is part of the problem. VCS are designed for versioning code. Best practice, when writing code, is to avoid creating large monolithic files, but rather to break up an application into many small code blocks or modules. So, no code file should be this large. While large files may accompany code, they are typically binary assets and thus different rules of diff and merge apply. So, the first thing that I noticed was hanging in SVN.

While I was able to add ontologies to our SVN repository, and it was possible to commit something that I was the last to work on it, other situations presented problems. Most notably where diffs were required and possibly merges. Our SVN repo would just hang when doing an update or commit. There were a few reasons for this that might have been particular to our installation:

1. We've had previous memory issues on that server, so there could have been a resource problem
2. Our SVN repository is connected to a Trac issue tracking system. The Trac system performs operations that allow you to browse source, commits, comments, etc. through a web interface. It also emails the members of our group when commits have been made, and includes diffs in these emails. These are all implemented using SVN hooks (i.e. post-commit hook) that automatically perform tasks in response to SVN activity. The hanging could be connected to the hooks and not just the normal SVN operations.
3. Our SVN repository server is an old version. So it may not be the most optimal installation.

While investigating the problem(s), I needed to determine if it was our service or the size of the ontology that was at issue. In a previous project we used a Google Source hosted SVN server to manage concurrent ontology development with some success. We observed at the time that it the standard text diffs worked only because assertions were serialized in a predictable order each time (otherwise a purely text diff would report potentially huge "differences"). If that were ever true, it apparently isn't true any longer. I came across this blog post from Chris Mungall, that verifies what I have subsequently observed, that the current OWLAPI serialization is non-deterministic: https://douroucouli.wordpress.com/2014/03/30/the-perils-of-managing-owl-in-a-version-control-system/

So, now I had two problems, first could a VCS handle files of the size of a large ontology, in particular diffs and merges, and second, was there even a good diff utility for such an ontology?

I expanded my investigation to eliminate our hosted VCS server (SVN) as the possible issue. I tried using a VCS hosted elsewhere. For this experiment I used a GIT repo hosted at BitBucket. The results were basically the same, add, update, commit, and push seemed acceptable, so long as no one else had edited the ontology. As soon as diff and merge was required, I again had problems with the repository hanging.

But, it was still possible that the size of the diffs were to blame. The diffs were huge in the initial experiments (it turned out that one version of the file had full IRIs and the other had been given a shortened serialization by Protégé). For a better experiment I made the two versions of the ontology (two versions to be pushed, both slightly different from the master) almost identical. Version 1 had one comment added, version 2 modified a comment and added 1 more. The diffs should have been tiny. They still hung visual merge tools (in this case FileMerge from the XCode tools on a Mac), because 2 versions of the FMA are just a huge thing to load. The (text) diffs were still wrong due to serialization order.

So, next I experimented with two other diff utilities that I might be able to plug in to a VCS. The first was the OWLDiff tool used by Protégé, the second was an open source project called owl2vcs. The former tool has a visual component and hung indefinitely. The second simply produced erroneous differences. I tried multiple serialization formats. It made no difference with the OWLDiff tool. Owl2vcs couldn't read many of the OWL formats.

Some other things I tried:

1. XML diff utilities.
2. I at least looked for a reasonable RDF diff (I am only interested in diffing asserted facts).
3. Client-Server Protégé. This is apparently not currently in development and even after I was able to find and build server code, I could not get a client to connect.
4. Web Protégé. Installation of Web Protégé wasn't bad, though I had difficulty loading the FMA into it (64,000 entity limit in RDF/XML circumvented by first converting to Turtle format). The issue here was in the editing environment. The full OWL2 viewer plugin is fine, but the editing plugin was (in my experience) buggy and not at the level needed by ontology editors (e.g. assertions are hand typed in Manchester Syntax without sufficient hint/assistance/auto-complete).
5. Protégé OWL database backends. There appears to be two database backends available for Protégé 4 or 5 (the reason that I say appears is that it is easy to come across web pages and tutorials for Protégé that do not distinguish what version they are for, so some that I found may have been for the old DB backend from Protégé 3.x). One is from Protégé itself, and I could

not get it to work. The other is called OWLDB and similarly I do not know how to set up and configure/execute it. There is little documentation for either of these, what doc there is seems to be out-of-date, and neither seem to be in current development.

6. I also looked into the idea of non-conflict creating VCS, such as the old SourceSafe applications from MS, which required the current editor to hold a lock on any files he/she was editing. This, of course, would not allow concurrent editing, but might work in my situation. I determined that SVN could be configured to work this way, but, because the repository manager and editor are not linked (as SourceSafe was linked to VisualStudio), there was nothing in the editor application to prevent ontology editing.

So, I am still left without a sufficient solution for concurrent editing of large ontologies. Chris Mungall's post suggests that there may not presently be one. But I am still left to wonder what the rest of the community is doing?