3 Current Version

Problems:

- Any class needing version control needs to inherit from VERSIONABLE, when it should be possible to be version controlled making no assumptions at all
- in this model, TRANSACTION has to inherit from VERSIONABLE, so as to get a VERSION_AUDIT
- however, logically, the audit trail is not part of what is being versioned, but something that the versioning mechanism records for each version when it is committed. It should not be part of the class VERSIONED.

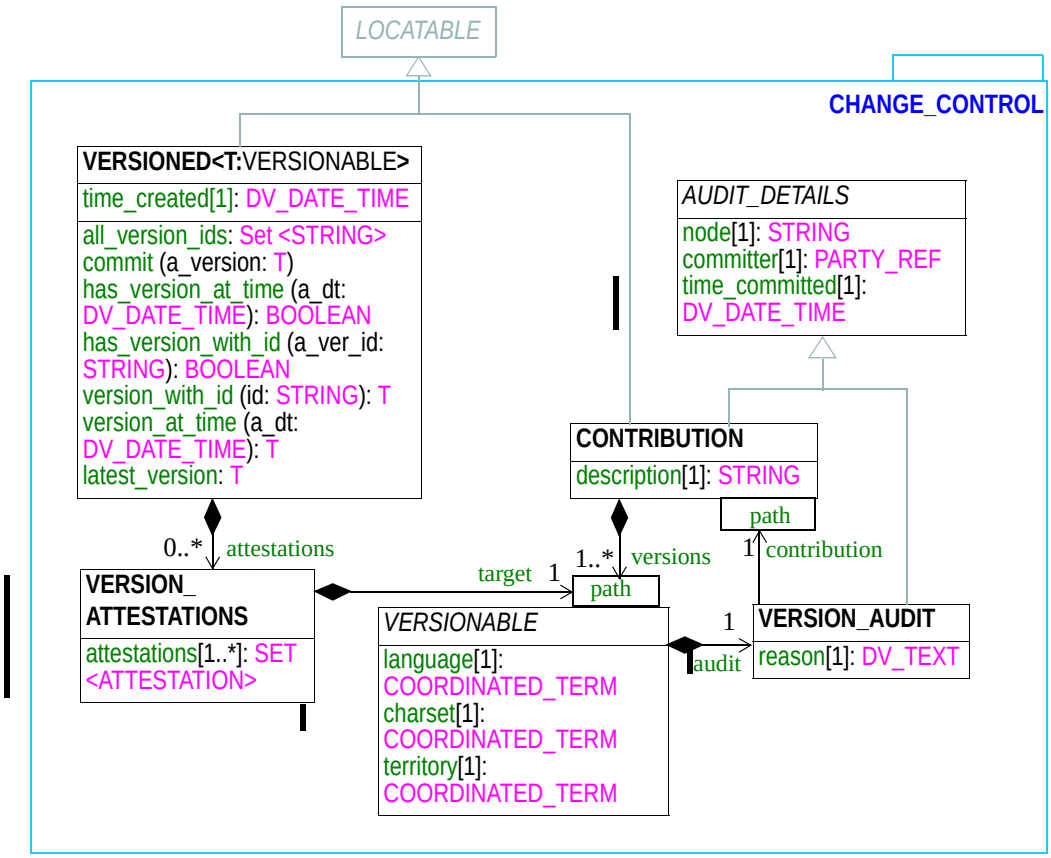

FIGURE 1 CHANGE_CONTROL Package

4 Proposed Version

Changes in this proposal:

- VERSIONED<T:VERSIONABLE> is renamed and redefined as VR<T> (i.e. “versioned repository”)
- VERSIONABLE is now the class VERSION<T>, and is no longer abstract. Because VERSION<T> is generic, it means that every version in a VR<T> is locked to a given type, e.g. VR<TRANSACTION>, VERSION<TRANSACTION>
- the relationship to what is versioned, e.g. TRANSACTION is no longer via inheritance but via a client/supplied relationship VERSION.data
- this mechanism can be used to version anything, including things that do not inherit from VERSIONABLE, as in the previous scheme.
- It also means that the definition of what is versioned is separate from the audit trail (VERSION_AUDIT)
- If we were to use this in the 13606 revision, then COMPOSITION should no longer have AUDIT_INFO directly connected to it.

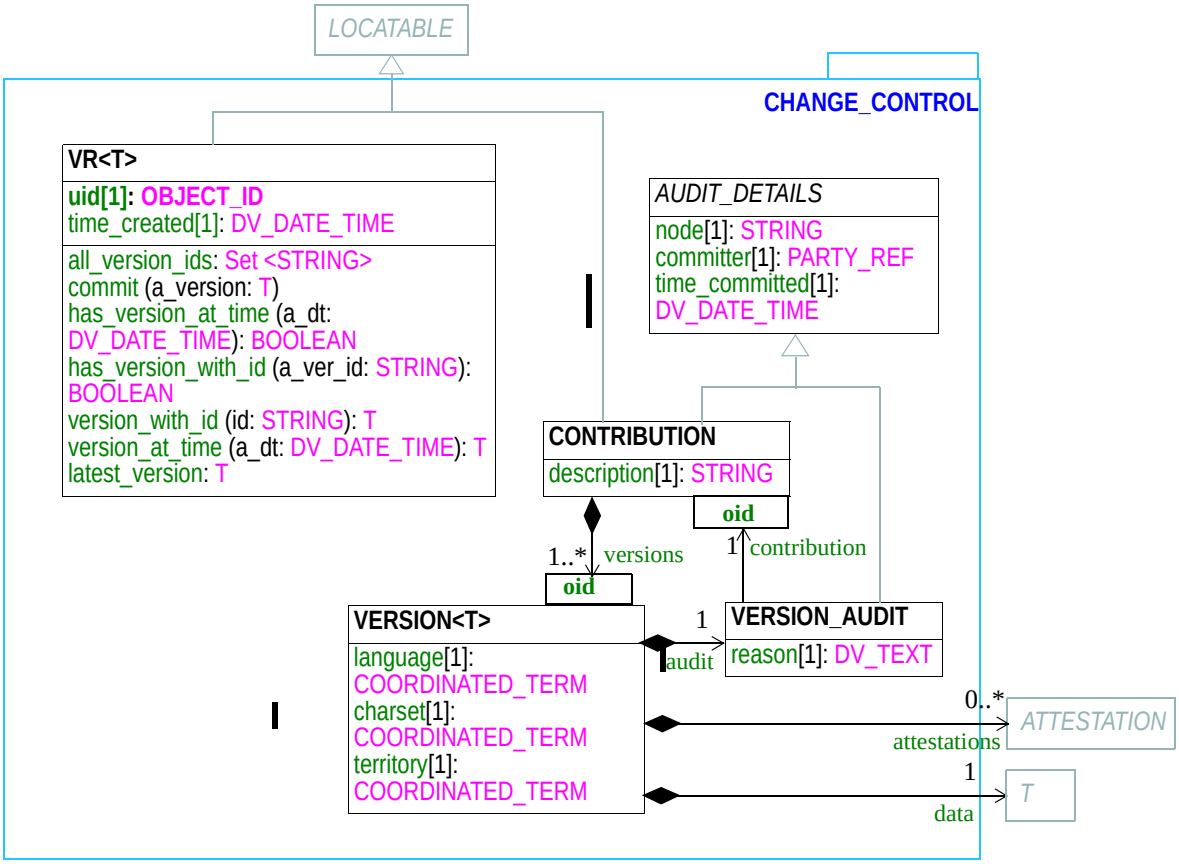

FIGURE 2 CHANGE_CONTROL Package

Other changes I believe would be advantageous:

- Contributions should be identified by an OID (this will make Dipak happy;-)

- VR<T>s are identified by an OID as well.
- THis means that a VERSION<T> is identified by the VR<T> OID + a local version identifier

The Extract cluster would also have to change to now include the Audit information.

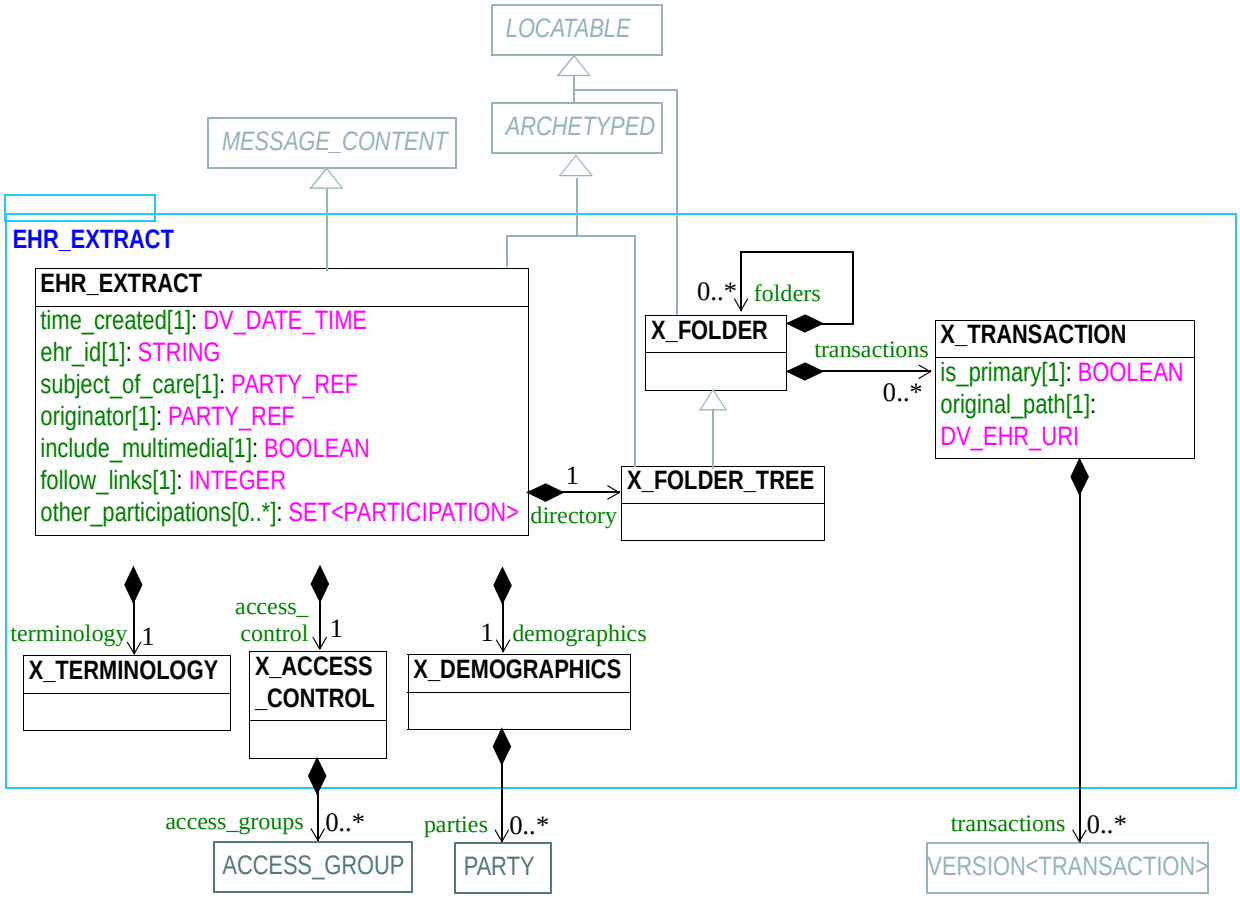

FIGURE 3 EHR_EXTRACT Package