Utilizing Views to Normalize `c_metadataxml` Values in Multi-Axial Metadata Hierarchies

Hubert Hickman
hhickman@essexmanagement.com

Introduction

With the increasing use of mutli-axial hierarchy metadata in i2b2 systems, the management of `c_metadataxml` values can become unwieldy. Since a given `c_conceptcd` can occur many times in the metadata, steps to normalize these values can be made without changing i2b2 source code.

Mutli-Axial Hierarchies

Many of the metadata sets used in the i2b2 ontology cell follow a simple parent/child pattern. That is, for a given node in a i2b2 metadata tree, a child node may have only one parent. Hence, the leaf node will only occur one time and no `c_metadataxml` needs to be defined more than one time.

However, in a multi-axial hierarchy there can be many paths to reach a particular leaf node. Hence, each time the leaf node is repeated, the corresponding `c_metadataxml` must be replicated in each row of metadata.

Conclusion

The creation of a separate table and a database view for each multi-axial metadata table allows the normalization of the `c_metadataxml` field. This simple solution eliminates the need for maintaining many copies of the `c_metadataxml` contents in complex i2b2 ontologies.

An additional normalization could be performed if there exists different data sources at a site with incompatible `c_metadataxml` contents for a given `c_conceptcd`. If an additional field for `c_sourcesystemcd` is added to the `metadataxml_map` table and the view is trivially extended, then the solution will work for many different datasources.

Acknowledgements

The author gratefully acknowledges the support of Essex Management for allowing me the time to work on i2b2 community oriented work. I would also like to thank the i2b2 group at the University of Nebraska Medical Center and NYU Langone Medical Center. Most especially I would like to thank Dr. James R. Campbell, M.D. for getting me involved in i2b2 and metadata development.

The LOINC example pathways are taken from LOINC metadata developed for the GPC CDRN.

Sample metadata paths for WBC in LOINC

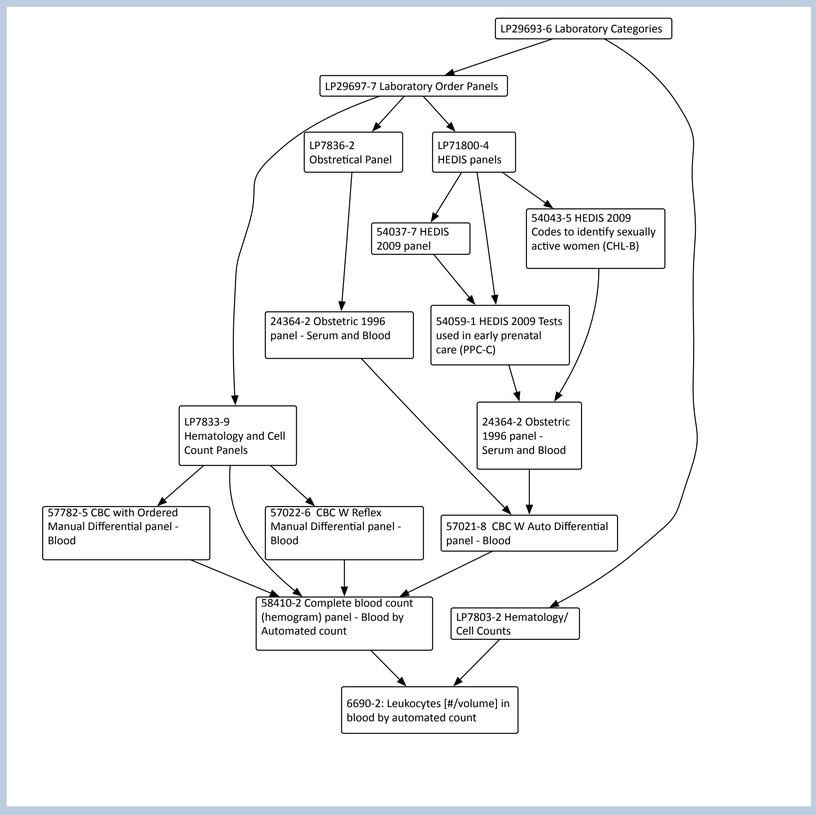

Figure 1: Example paths in LOINC

The graph shown above shows the different paths by which LOINC 6690-2 can be reached. There exist eight distinct paths from the top level node (Laboratory Categories) the the leaf node. The paths are enumerated in the table below.

c_fullname
\LP29693-6\LP29697-7\LP71800-4\54037-7\54059-1\24364-2\57021-8\58410-2\6690-2\
\LP29693-6 \LP29697-7 \LP71800-4 \54059-1\4364-2\57021-8\58410-2\6690-2 \
\LP29693-6\LP29697-7\LP71800-4\54043-5\24364-2\57021-8\58410-2\6690-2\
\LP29693-6\LP29697-7\LP7833-9\57022-6\58410-2\6690-2\
\LP29693-6\LP29697-7\LP7833-9 \57782-5\58410-2 \6690-2\
\LP29693-6\LP29697-7 \LP7833-9\58410-2\6690-2\
\LP29693-6 \LP29697-7\LP7836-2\24364-2\57021-8\58410-2\6690-2\
\LP29693-6\LP7803-2\6690-2 \

Table 1: Sample Metadata Paths for LOINC 6690-2

Adding Views to the Ontology Schema

The goal is to create a metadata view that encompasses the normalization of the `c_metadata` field. To accomplish this goal, we add a new table that contains the `c_metadataxml` field and the `c_basecode` field.

The view definition as shown above is very simple. One needs to use care when performing fact counting procedures that update the metadata tables and use the physical metadata table, and the not the view.

```
create table metadataxml_map (
  c_basecode  varchar2(50),
  c_metadataxml clob);

create view example_metadata_v as
(
  select em.c_hlevel,em.c_fullname, em.c_name,  em.c_synonym_cd,
         em.c_visualattributes,em.c_totalnum,em.c_basecode,
         mm.c_metadataxml,em.c_facttablecolumn  ,em.c_tablename  ,
         em.c_columnname,em.c_columndatatype,
         em.c_operator,em.c_dimcode,
         em.c_comment,em.c_tooltip  ,em.m_applied_path,
         em.update_date,em.download_date,em.import_date,
         em.sourcesystem_cd  ,em.valuetype_cd,
         em.m_exclusion_cd,em.c_path,em.c_symbol
         from example_metadata em
         left outer join metadataxml_map mm
         on em.c_basecode = mm.c_basecode

);
```

With the new table, each `c_basecode` value only occurs one time. In the `table_access` table, the view name is substituted for the the original table.