Ontologies 101

by i2b2 tranSMART Foundation Ontology Working Group

7/16/2020

Matvey Palchuk <matvey@yahoo.com>

Jay Pedersen <jay.pedersen@unmc.edu>

- What is an i2b2 ontology?
  - See our definitions of various terms used in this tutorial
  - Building blocks of an i2b2 query
    - This is what “drives” the upper-left corner of the i2b2 webclient - it represents the “building blocks” of your i2b2 query. In order to make facts or observations you wish to load into your i2b2 queryable, you must have corresponding (accounting for codes used to represent these facts or observations) entries in your ontology.
  - Typically based on standards
    - Some commonly used standards to represent basic structured clinical patient data collected in EHRs

Data domains	Typical Standards
Demographics	HL7 Administrative
Diagnoses	ICD
Procedures	ICD, CPT, HCPCS
Medications	RxNorm + VA Classes hierarchy
Labs	LOINC
Vital Signs	LOINC

  - Rendered as a collection of nested folders
    - Assumed to represent child-parent (or is-a) relationships
  - Inspired by hierarchical data representations of old (and not so old - think MUMPS and Caché), hence the importance of concepts paths
    - A notable aside (perhaps too technical but worthwhile mentioning): querying on paths using SQL affords the use of LIKE operator which optimizes the selection of entire sub-trees
  - Resides in a SQL table in metadata schema of i2b2
  - Do not be intimidated by i2b2 ontology!

- Prequel (JP)
  - Medical terminologies are commonly involved
    - Medical terminologies vary widely in their purpose and release format
    - The ONC prescribes standard medical terminologies for interoperable use
    - Medical terminology releases are not directly usable by i2b2
    - Standardized I2b2 metadata must be created for each desired medical terminology to make it usable within i2b2.
- Coded entities, including code systems, and i2b2’s notion of prefixes
  - Drawing on and continuing the discussion of vocabularies, terminologies, and ontologies above, we need to point out that it is a common practice to assign codes to concepts. As an aside, “Desiderata for Controlled Medical Vocabularies in the Twenty-First Century” by Jim Cimino is a must-read article if you’d like an in-depth introduction to the world of controlled medical terminologies.
  - A code is a way to uniquely identify a concept. Codes are organized into coding systems. A good example is a coding system for diagnoses - ICD-10-CM. An example of an ICD-10-CM code is “E11” and it identifies the diagnosis of “Type 2 Diabetes Mellitus.”
  - Note that E11 by itself might be ambiguous, but if you are told that it’s an ICD-10-CM diagnosis, it becomes a unique identifier.
  - It is a convention in i2b2 to combine code systems and codes into a single string separated by “:” (again, a convention), with the left side of the resulting expression typically referred to as the “prefix.”
- How to quickly add a few concepts to i2b2 ontology
  - This content was presented at 2020 Harvard Symposium and is available in this video recording: https://www.youtube.com/watch?v=A5bIvNDIb3c (starting at about 5 minute mark and lasting approximately 20 minutes)
  - Not everyone is adventurous to look “under the hood” so to say, but it is rather straight-forward and not scary. The beauty of i2b2 is its flexibility - you can fairly easily add new concepts to the ontology and you are immediately ready to load corresponding data and work with it in your webclient (or other compatible interfaces).
  - There are many ways of adding rows to database tables, but I will discuss using a tool with a graphical user interface, such as Oracle SQL Developer for Oracle database (there are similar tools for every RDBMS, there’s a plethora of choices). I will assume that you know how to connect to your i2b2 database using such a tool.
    - Export the content of your ontology table to .csv file and work with it in Excel. Subsequently, you will need to drop the table and replace it with the content of the .csv file you modified by importing it into your database, or you can do this work directly in your database table.
  - By default, the ontology lives in a table called “i2b2” although you can change that name and have it live in multiple tables.
  - The basic structure of the ontology table(s) in i2b2
    - The table might look a little intimidating at first glance, but once you understand the idea behind it, it is straightforward to work with. Here are the main columns in the ontology (also frequently referred to as the metadata) table:

- C_HLEVEL
  - This column carries a whole number which indicates the depth of the hierarchical level for a given ontology concept. Typically 0 means the very top - root level - of the hierarchy, 1 is the first “child,” 2 is the child the first, and so on.
- C_FULLNAME
  - This is the most important column. It is my practice whenever I work with i2b2 ontologies in a spreadsheet or a database to sort the table by c_fullname - you will understand why in a moment
  - Please note that I am simplifying here and in order for this simplification to work and not cause problems, you must ensure that your c_fullname is kept in sync (values be identical) with c_dimcode
  - C_fullname is a path. Backslash acts as a separator, and individual elements of the path indicate the relationship of a given concept to the top of the hierarchy, referred to as the root. In other words, it’s the way you can get from the root to the current concept - a path!
  - The element in the path that has no “children” is a terminal node or a “leaf.” Its ancestors are “folders.” In i2b2 webclient, folders get rendered with a folder icon and leaves - with a page icon. See c_visualattributes below for the continuation of the leaf and folder discussion.
  - By convention, the root is “\i2b2\” - doesn’t have to be, but frequently is.
- C_NAME
  - Concept description or a name
  - This is shown in the webclient, so think about readability
  - Webclient will alphabetize “sibling” elements in your ontology, so you might need to get clever about C_NAME if you want to enforce a certain order of concepts
    - Tip: try “zz “ (that’s z-z-space) in front of your C_NAME and see what happens in webclient
- C_SYNONYM_CD
  - Default is “N” and we’ll concentrate on those. If you see “Y” it means that the row you are looking at contains a synonym - out of scope for this tutorial.
- C_VISUALATTRIBUTES
  - Three-letter values populate this column, but we will focus on the first 2 only:
    - 1st letter determines if it’s a folder (F) or a leaf - a terminal node (L). There are other possible values, but it’s out of scope. Just remember to ignore rows where the 1st letter is R - those are modifiers, and that C behaves almost but not quite like F.
    - 2nd letter - we’ll always keep it “A” (stands for “active”) for the purposes of this tutorial
- C_TOTALNUM
  - Really important column but not for this discussion
- C_BASECODE
  - This one is optional but convenient - fill in the code for your concepts here and it’ll be easier for you to visually understand and manage your ontology
- C_METADATAXML
  - This controls popups in i2b2 webclient for concepts like lab results. This will be explained elsewhere.
- C_FACTTABLECOLUMN
  - Default is “concept_cd”
- C_TABLENAME
  - Default is “concept_dimension”
- C_COLUMNNAME
  - Default is “concept_path”
- C_COLUMNDATATYPE
  - Default is “T”
- C_OPERATOR
  - Default is “LIKE”
- C_DIMCODE
  - “Dim” code stands for *dimension* code
  - Technically concept_path in concept_dimension table equates to c_dimcode.
  - It is important to remember to keep c_dimcode the same as c_fullname, but beware that this is a simplification
  - For more advanced behavior, you may need to allow your c_fullname and c_dimcode to diverge, but we will not cover it in this presentation
  - Might want to ask Lori Phillips <LCPHILLIPS@PARTNERS.ORG> to chime in with more details here
- C_COMMENT
  - I do not remember whether and how it gets rendered in the webclient
  - Ok to leave blank
- C_TOOLTIP
  - Self-explanatory - appears on hover-over in the webclient
  - Ok to leave blank
- M_APPLIED_PATH
  - Default is “@”
- UPDATE_DATE
  - Date in the format of “dd-mmm-yy”
- DOWNLOAD_DATE
  - Date in the format of “dd-mmm-yy”
- IMPORT_DATE
  - Date in the format of “dd-mmm-yy”
- SOURCESYSTEM_CD
  - You can put in the name of your organization
- VALUETYPE_CD
  - Ok to leave blank
- M_EXCLUSION_CD
  - Ok to leave blank
- C_PATH
  - Ok to leave blank
- C_SYMBOL
  - Ok to leave blank

- As an example, let's go through the process of adding a U07.1 "COVID-19" diagnosis to your existing i2b2 ontology
    - Let's assume that you are starting with an existing diagnoses ontology and c_fullname for your Diagnosis folder looks something like "\i2b2\Diagnosis\"
    - We will need to add a folder for the U chapter to accommodate our U07.1 code because up until April 1, 2020, ICD-10-CM did not have any codes beginning with U.
    - C_hlevel will be 2 because the root ("\i2b2\) is 0 and your "Diagnosis" folder is 1
        - C_fullname will be "\i2b2\Diagnosis\(U00-U85)\"
        - C_name is "Codes for special purposes"
        - C_synonym is kept at "N"
        - C_visualattributes is "FA " for Active Folder, and it's a folder because it will have "children" beneath it
        - Lastly, you can add the optional c_basecode as "ICD10CM:U00-U85"
        - All other values should be set in accordance with descriptions above, being consistent with other rows of your ontology file and being mindful of keeping c_dimcode the same as c_fullname
    - If at this point you can add this row to your i2b2 metadata table, commit the change and look at your webclient - you will see a brand new "Codes for special purposes" folder in your Diagnosis folder. It's that easy!
    - Let's add remaining rows. The pertinent info is:

C_HLEVEL	C_FULLNAME	C_NAME	C_VISUALATTRIBUTES	C_BASECODE
2	\i2b2\Diagnosis\U00-U85\	Codes for special purposes	FA	ICD10CM:U00-U85
3	\i2b2\Diagnosis\U00-U85\U00-U49\	Provisional assignment of new diseases of uncertain etiology or emergency use	FA	ICD10CM:U00-U49
4	\i2b2\Diagnosis\U00-U85\U00-U49\U07\	Conditions of Uncertain Etiology	FA	ICD10CM:U07
5	\i2b2\Diagnosis\U00-U85\U00-U49\U07\U07.1	2019-nCoV acute respiratory disease (COVID-19)	LA	ICD10CM:U07.1

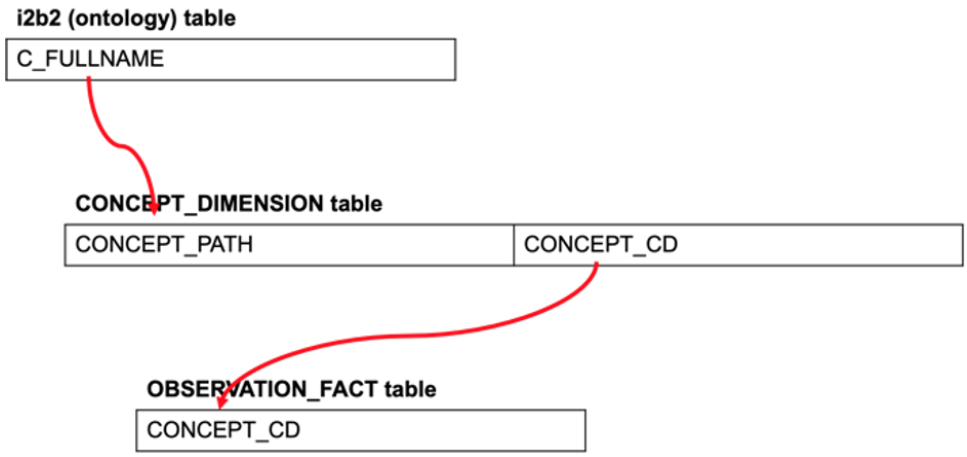
    - In this table, notice the last row with c_visualattribute of "LA " - that's the last element in this sub-tree (or hierarchy) and so it's the leaf.
    - There is one more crucial step in this process. If you recall, we said that c_basecode was optional. But you will recognize the content of that column as being the exact observation codes you will be storing in your observation_fact table. There must be a way to link ontology concepts with codes in observation_fact table. I2b2 accomplishes that via concept_dimesion table. So in addition to adding new ontology concepts, you also must make necessary changes to your concept_dimension table.
    - Making changes to concept_dimension table is covered in one of the subsequent sections - please see below.
    - Let's assume that you added the desired concepts to your ontology table and made the necessary changes to your concept_dimension table. Now, when you load patient observations into your observation_fact table and populate the concept_cd with "ICD10CM:U07.1" you will be able to query for patients with COVID-19 diagnosis.
- Address c_fullname vs c_dimcode
- How to deploy, incl. TABLE_ACCESS; in multiple tables
- How to make changes

- I2b2 user interface and its relationship to metadata
    - Describe how metadata provides the bulk of the i2b2 user interface
        - Browsing through folders in the i2b2 user interface literally browses through the rows of installed metadata tables of that i2b2 instance
        - The browsing behavior differs slightly for rows with metadataxml
        - An i2b2 query is created by making selections in the folder browsing interface, each selection is tied to a specific metadata row
    - Describe the process i2b2 uses to resolve queries, and how the attributes in metadata are used in that process.
- METADATA XML – see the METADATA XML subpage for information on setting the C_METADATAXML column for fine-tuning the associated query to look at data values in the associated Facts
- Relationship between i2b2 ontology and concept_dimension table
    - C_fullname in i2b2 ontology is the unique identifier of a concept. Individual patient facts in observation_fact table are identified by concept_cd - typically a code_system:code construct, for example, "ICD10CM:E11" for "Type 2 Diabetes Mellitus." To enable queries, the concept path in the ontology must be linked with concept code. Concept dimension accomplishes that and provides the link between concept path in the ontology with concept code
    - There is an important nuance which you must be aware of - the naming convention for columns in i2b2's ontology table(s) differs from that in observation_fact and its dimension tables, including concept_dimension. Column names in the table below illustrate this - each row shows different column names even though they carry the same data. In other words, C_FULLNAME in ontology carries the same information as CONCEPT_PATH in concept_dimension table.

DATA	METADATA (ontology)
CONCEPT_PATH	C_FULLNAME
CONCEPT_CD	C_BASECODE
NAME_CHAR	C_NAME

    - Let's review pertinent columns:
        - CONCEPT_PATH - same as c_fullname in the ontology

        - CONCEPT_CD - same as c_basecode in the ontology, only it's no longer optional here! This is the link between ontology (metadata) and patient data - from ontology's c_fullname to concept_dimension's concept_path (which is the same thing as c_fullname) and from concept_cd to observation_fact's concept_cd. See the illustration:



        - NAME_CHAR - concept name or description. It is good practice to populate this field.
    - As you can see, concept_dimension is a crucial table and it is very important to populate it properly, else the data in your i2b2 will not queriable.
    - Out of scope for this tutorial, but concept_dimension is fundamental for significant additional flexibility in i2b2 ontology and concept mapping.
- Link to i2b2 doc describing relevant topics
    - Synonyms
    - Modifiers
    - ?
- Do not forget to plan for regular updates of your ontology
- What to do with deprecated codes?
- COVID-19
    - It is important to note that many COVID-19 related facts or observations, such as the diagnosis of COVID-19 (ICD-10-CM:U07.1), procedure codes for SARS-CoV-2 testing, laboratory codes for actual tests, etc., are brand-new and will not be in your i2b2 ontology unless you add them. It is crucial to do so in order to ensure that COVID-19 related observations flowing into your i2b2 are queriable by its users.
    - New codes: https://docs.google.com/spreadsheets/d/1rbZfsZxsx16dsmleb7TFdM4lGQ-rypLG7IFcApwdAL0/edit?usp=sharing
    - Innovations introduced in ACT's COVID-19 ontology