721. Creating an optimal hierarchy

Universal vs project-focused

One of the first considerations when starting to design an i2b2 project ontology is the **primary target audience** that is expected to formulate the queries. Basically, there are two types of users:

- people that know the project and therefore know the structure of the data
- people that want to use the data but were not involved in the data collection process

The basic difference between those two groups is that the first one is able to handle labels like "day 14" since they know what to expect under that node. They know which events and visits have been scheduled. They know what data is recorded in "initial visit". Here, one can stick to the original naming. On the contrary, such users would be annoyed by changes that maybe couldn't track, e.g. Items not at the same position where they were on the Case Report Form. This is a case for **project-focused navigation ontologies**.

Things are different for external researchers or when data from different sources are combined in a research database. Here, an **universal ontology** has to be created. That means "translating" project-specific label into generic ones and moving data. Events can be classified into *screening*, *baseline*, *intervent ion*, *follow-u*p. Data should be classified into unambiguous groups like demography, diagnosis, laboratory, medications. Wherever possible, mapping data to medical standards or terminologies should be considered.

Depth of the navigation

In most cases, it is advisable to just stick to the original depth of navigation. But depending on the source format the data came from, this can lead to **very deep hierarchies**. Exemplary for an ODM import, one will find seven levels of hierarchy for a clinical trial:

1. Study
    a. MetaDataVersions
        i. Events
            1. Forms
                a. ItemGroups
                    i. Items
                        1. CodeLists

Some levels can be simply deleted, e.g. if there is only one MetaDataVersion or one event, they can be spared.

> The first level "Ontology" can also be renamed to something more expressive.

Another idea is to abandon hierarchy levels that where useful for data collection, but not for presentation. More than 250 concepts, on the other hand, should not be listed under a single i2b2 folder for reasons of usability.

Splitting large value sets without natural hierarchy

A concept might have a large number of possible values without a normative hierarchy. Examples are code systems like zip codes, genetic information, or costs for billing. In this case, it is not feasible to represent every possible code: 1 Euro, 2 Euro, 3 Euro, ...

The basic idea is to **find an artificial, but yet reasonable substitute ontology**. A possible solution is shown in the Boston Demodata: not every plausible patient age is coded right below *age*, instead, there is an intermediate level for every decade (0-9 years, 10-19 years, ...). So, one can select a bulk of ages with one click.

> Splitting ages into decades might be suboptimal for some use cases in clinical research. For instance, when recruiting participants for clinical trials, inclusion criteria hardly match decades. In most scenarios, it would be better to have categories like 0-17 years (harder regulations for trials with minors), 18-80 years and 81-130 years (special screening for elderly). Categories should have subcategories where appropriate: 18-80 years might be further splitted into 18-49, 50-65, 66-80 years.

Costs could have categories at ten thousands, thousands, hundreds and so on. Alphanumeric codes could have categories defined by the first letter (0-9, A-Z). Postal codes could have states and counties as classifying attributes.