

726. Miscellaneous

Representing a concept as a folder and a leaf

Sometimes it is necessary to **represent a concept as a folder and a leaf**, because the folder used for subsumption is also a valid code itself. One example is ICD-10 for coding diseases and diagnoses:

- XVIII - Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
 - R50-R69 - General symptoms and signs
 - R65 - Systemic Inflammatory Response Syndrome (SIRS)
 - R65.0 - Systemic Inflammatory Response Syndrome of infectious origin without organ failure
 - R65.1 - Systemic Inflammatory Response Syndrome of infectious origin with organ failure
 - ...

Most diagnoses will be at leaf level (R65.0, R65.1), but some source data were coded just with R65. You won't find them when using i2b2 folders as parameters because a folder is only a substitute for its containing leafs (R65.0 - R65.9).

To be able to query such fact in i2b2, you need an extra leaf even if this is a defeat with regards to usability.

Multi-axial hierarchies

i2b2 has no support for multi-axial hierarchies. This causes problems for a number of post-coordinated knowledge organization systems.

In this case, you need to **define redundant sub-hierarchies**, one for each parent of the current node.

Rights

i2b2 ontologies are not made for user and rights management. It is not possible to hide or disable parts of the navigation based on the rights of a user account. Every folder or leaf is visible and selectable for querying to everyone.

If you want to limit visibility, it's more practical to **create different projects**, assign user accounts correspondingly and load only the desired data.