Taxonomy Variants

Overview

Organizational areas often evolve their own particular taxonomies to communicate between the business and IT. These taxonomies are often *related-to* but **not** *identical-to* common enterprise taxonomies. They will often go into more detail in some areas and less detail in others as well as using slightly different terminology.

Within Waltz there is a desire to allow these areas to express themselves using their taxonomies but have a mechanism to describe how their taxonomy relates back to the enterprise taxonomy.

Using a mapping may allow for a fully or semi-automatic translation between taxonomies.

The purpose of this issue is to analyse the requirement, examine 'prior-art' and discuss potential approaches.

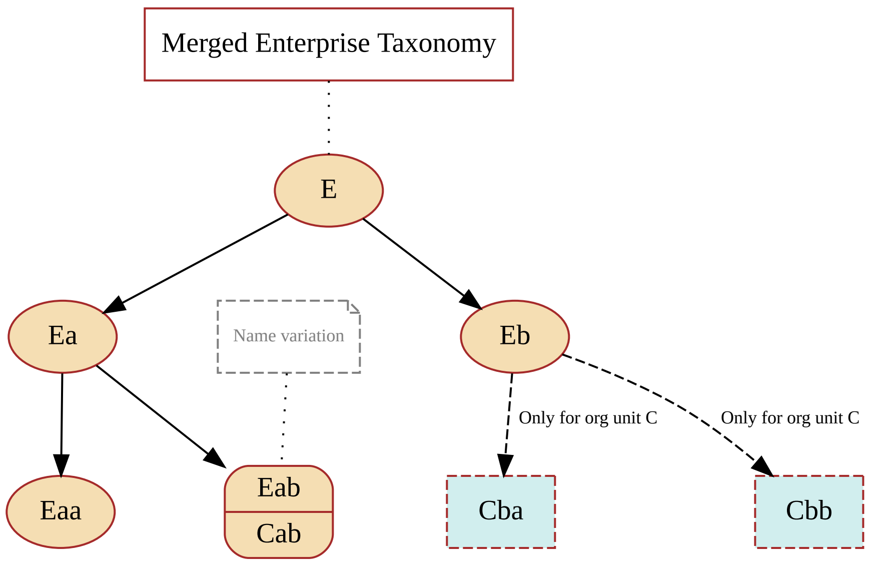
Approaches

We will consider two approaches to this task. The first is to model variations within the enterprise taxonomy and use filtering to allow sections to be enabled/disabled and also to allow for variations of the names of elements. We shall refer to this as the *Merged* approach

---

Merged Approach

The diagram below attempts to illustrate a merged taxonomy. The core enterprise taxonomy is shown using beige ovals prefixed with the letter `E`. The custom taxonomy is prefixed with the letter `C` and depicted with blue rectangles.



The custom taxonomy defines a variation of the name `Eab` in the enterprise taxonomy (the new name is `Cab`). The custom taxonomy also has specialisations of the `Eb` node (namely `Cba` and `Cbb`).

These two techniques (*filtering* & *name variation*)could be combined to allow for increased flexibility.

Pros

The merge approach has a several benefits:

- maximum leverage of the core enterprise taxonomy
- variations are localised to the relevant nodes

Cons

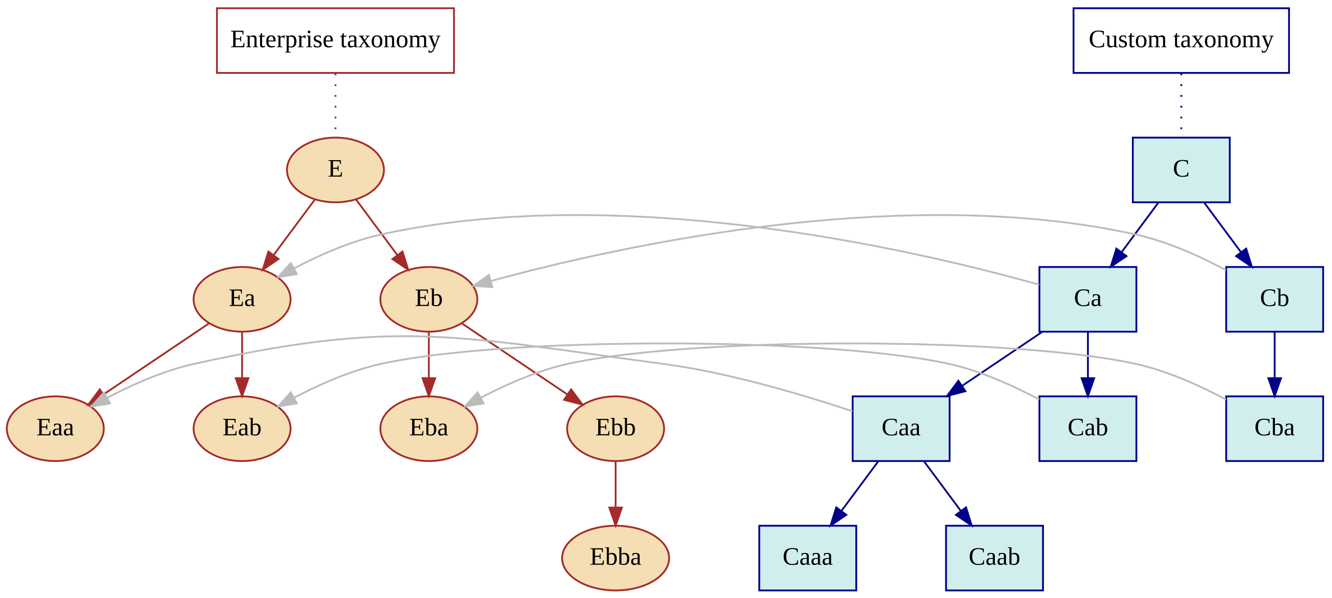
However, merging has several significant downsides:

- major structural differences will be very hard to express
- mapping must happen at runtime
- difficult to give ownership of taxonomy to organisational areas as the core structural 'shape' of the taxonomy must be managed from an enterprise perspective
- configuration screens may become complex
- will require a substantial Waltz taxonomy rework

Distinct Approach

Another approach is to use all for distinct taxonomies to be mapped back to an enterprise taxonomy.

This example shows both an enterprise taxonomy (beige ovals) and a custom taxonomy (blue rectangles) with the mapping shown between the two as grey arrows.



Notable aspect of this diagram includes the specialisation of the custom taxonomy under `Caa/Eaa`. Also worth noting is the *simplification* of the `Eb` subtree, indicating the custom taxonomy may not be interested in this area of the tree.

Pros

Taking the distinct approach has several advantages:

- custom taxonomies are much more flexible
- ownership of taxonomies is clear
- name variations not required, as the entire taxonomy is custom
- does not require runtime translation within a taxonomy tree
- this approach is a largely orthogonal addition to existing Waltz taxonomy capabilities giving more flexibility to implementation approach
    - i.e. we could do bulk translation from one taxonomy to another without the need to embed this in Waltz initially
- configuration screens will be simpler

Cons

- requires a large amount of mapping
- changes to enterprise taxonomy will need to be considered by all variant custom taxonomies

Conclusion

Currently the recommendation is to pursue the *Distinct* approach. This will give the most flexibility, has the cleanest underlying model and, importantly, seems to agree with the approach that taxonomy experts and methodologies recommend.

Notes

There is a Waltz Issue for tracking this proposal and to open the discussion to the wider Waltz community.

Some topics mentioned in this document are more fully explored in presentation written by Heather Hedden on Mapping Taxonomies, Thesauri, and Ontologies

Heather Hedden also has a blog post on Taxonomy Mapping which is informative.

There is an opensource taxonomy mapping application called Concoda which may be worth evaluating. However the project seems to have been archived, with no new contributions for 5 years.