

Use Case 3 - Generate Vocabulary Terms

Use case for generating new vocabulary terms. This happens when there are no recommendations actions for leftover data terms or the user does not want to use any or transform the data manually and just wants the end result generated. The application then presents vocabulary generation options the these leftover terms which is mainly described in this use case.

The use case uses food product data (used in use case 2) for which there are few vocabularies represent a product along with nutrition information and ingredients - FoodOn, The FoodOntology and some ontologies for categorizing food products (but without properties, just thesaurus) - AGROVOC.

The user does not want to fully use the aforementioned vocabularies for some reason (not enough properties, different focus - e.g. to agricultural, too complex to understand, weird, ...) they create their own. This tool helps with that and is capable of creating new vocabularies to represent unmapped data per users bidding. The tool also provides a way to add custom triples to the new vocabulary and add it to the catalog. If the tool matches only a term (concept) from thesaurus, for example nutrients, but without any property and term to represent that e.g. “food” with property “nutrients”, we want to the final vocabulary to e.g. include the term in range of some generated “ex:nutrients” property.

Figma Use Case

The starting point of this use case is the middle results of use case 2. The uses matched property values of property “countries” to countries scheme general purpose matching methods.

The user selects **Nutriments** recommendation which found matching “nutrients” term in AGROVOC thesaurus. The user can also preview what properties it has. The user inspects it but based on the definition does not found the terms matching. However, rather than “nutrients” he finds a related term while browsing the data which fits the match - “food composition” and manually adds the entity to this class (next frame).

The “nutriments” entity then contains a lot of properties such as calcium or energy values for given product. The calcium composition has at least three properties so it is further refactored manually to new entity representing calcium component in the food composition. The same thing is done for all such component properties such as energy, carbohydrates, etc.

On the next frame, the tool detected a recommendation to convert a unit for food composition values from literal to URI in library. Specifically, “g” literal to gram entity. Part of the recommendation is the link to the vocabulary of the found matching entity which can be previewed as well as the entity itself. The user has a chance to inspect what the found term means and whether to use it

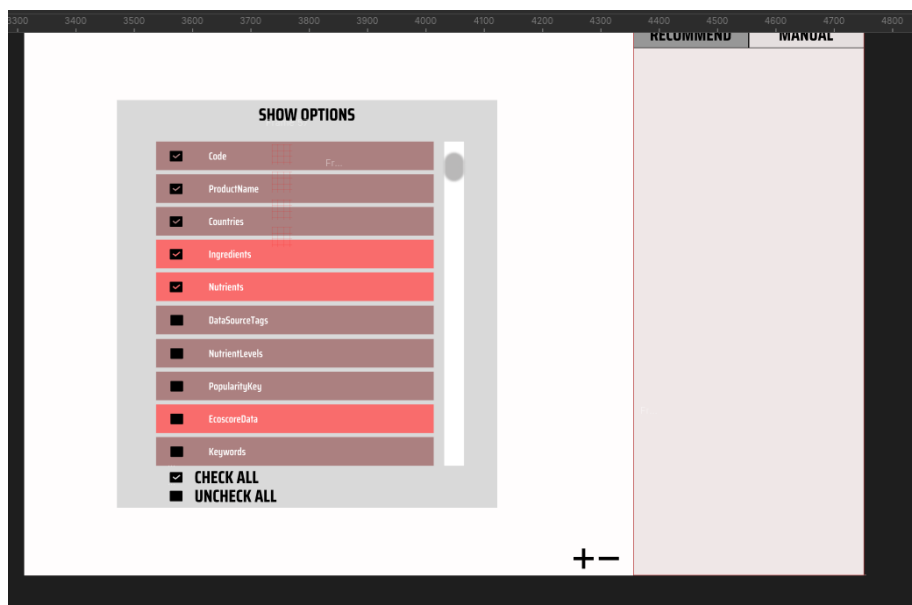


Figure 1: IMAGE EXAMPLE

or something else from the library.

If user proceeds with the recommendation, there is a detail view showing how the match was found (method) and values that were matched. This is very simple in this case, since the literals matched by string comparison on literals.

Proceeding with the recommendation shows the recommended change in the graphical model.

The following two frames then show the performed recommendation for “calciumUnit” and “energyUnit” where the process is the same but for “kcal”.

Now the user is satisfied with the model and wants to get the output rdf where the unmapped values should be part of new generated vocabularies. The tool detects all unmapped properties and provides a way to create new vocabularies and map them remaining unmapped properties to them. The first frame shows the unmapped properties in red.

The user adds