

Building Knowledge Models for Agropedia Indica v 1.0 Requirements, Guidelines, Suggestions

Authors: Margherita Sini (FAO), Vimlesh Yadav (IITK)

Contributors: Claudio Baldassarre (FAO)

Revisers: Jeetendra Singh (IITK), TV Prabhakar (IITK)

Introduction	1
Requirements for Agropedia Indica	2
General guidelines.....	2
Guidelines for Maps	3
Guidelines for Concepts and Instances	3
Guidelines for Relationships	4
Suggestions for building KM	6
Good and Bad modelling.....	7
Case n. 1	7
Case n. 2	7
Case n. 3	8
Case n. 4	8
Case n. 5	8
Case n. 6	9
Case n. 7	10
Case n. 8	Error! Bookmark not defined.
Case n. 9	Error! Bookmark not defined.
Case n. 10: mathematical expressions.....	11
Management of synonyms	12

Introduction

This document is written with the purpose of giving indication on the development of Knowledge Maps (or Knowledge Models, KM) for the Agropedia Indica project version 1.0. Knowledge modelling refers to a way of structuring, acquiring and validating notions of real or abstract domains (representing knowledge) and storing it into models that are transferable across different information systems.

IITK developed the following models:

1. A generic model, acting as a top level agricultural ontology (foundational agricultural ontology) (Agropedia_crop);
2. A specific model about rice (Agropedia_rice), including:
 - a. pests (Agropedia_rice_pest);
 - b. disease (Agropedia_rice_disease);
3. A model on Pesticides (Agropedia_pesticide).

Each of the maps that should be generated (e.g. Litchi, Sugarcane, Cotton, Chickpea etc.) can be elaborated using different maps (e.g. generic chickpea map, chickpea pests, chickpea diseases, etc.).

Other models may include:

- a. pests;
- b. crop activities;
- c. geographical areas;
- d. experts;
- e. fertilizers;
- f. other as needed.

It is a good practice in this scenario not to produce duplicates (of models, of models' content). Multiple models will help the organization of the ontologies and the connections between them help will facilitate the reuse of the information (see figure 1 below).

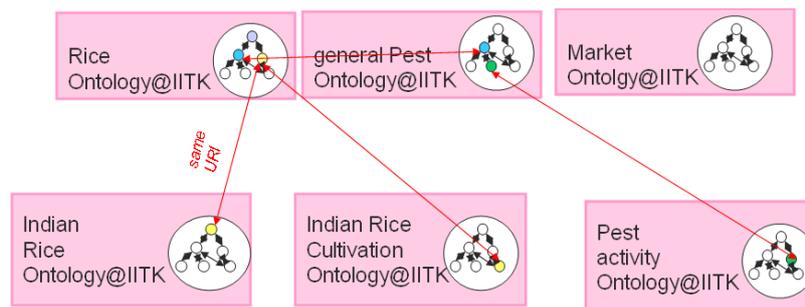


Figure 1: Reuse of concepts in different maps.

In this document we suppose users will use the CMap (Concept Ontology Editor, COE) tool. Note: same new features or guidelines have been introduced here and not during the Workshop at Pantnagar.

Requirements for Agropedia Indica

- The editor (the tool) used to build the KM should be simple, free and open source. The CMap (COE) tool has been chosen. Other editors may be used; in this case more complex information may be added to the KM (e.g. attributes).
- It should be possible to get cross-maps information.
- The purpose of the KM is to allow users (KVK specialists, experts, policy makers, advanced farmers, etc.) to visualize easily on the web such information (e.g. with hierarchical trees or a more advanced graphical navigation), and retrieve corresponding resources (e.g. documents, video files, audio files, web pages, etc.).
- Names of concepts and relationships should follow specific convention.
- Relationships should be used consistently.
- Concepts may/should be reused over maps (this creates linkages between maps). The name of concepts should be the same.
- KM should be developed in English. Translations in other languages (e.g. Hindi, Telegu) may be done after.

General guidelines

- With the CMap (COE) tool, at the minimum the knowledge can be represented as boxes. Each box can store a unique and well defined element (a concept or an instance).
- Boxes can be linked by arrows (these represent the relationships between the objects).

Guidelines for Maps

- It is suggested to have one top level agricultural KM only. Every map may have at least one link to this top level map, or to other intermediate maps. Relationships between concepts are propagated to all connected models (no need to repeat same relationships in different models).
- The editor who prepares a map must possess relevant prior knowledge (an expert on the topic).
- When developing a map the editor should refer to a particular question we seek to answer (the so called “competency questions”).
- Context will help to determine the hierarchical structure of the concept map (e.g. in a map about diseases, “bacterial leaf blight” is an instance of disease, but in a map about rice, this instance is connected to “basmati rice” with relationship “affect”).
- Link every instance to the corresponding concept (in the same or a different map).
- Use cross-links: relationships or links between concepts in different segments or domains of one or more concept maps.
- Quality: well organized, be consistent, qualitative data.
- Quantity: select a limited domain of knowledge; divide in multiple maps if needed.
- A concept map is always open to refinement and extension...

Guidelines for Concepts and Instances

- Identify the key concepts that apply to the domain.
- Use singular. Capitalize the concepts (e.g. “Manure”).
- Do not enter spaces in concept or instances names (e.g. “Rice_crop”).
- Organize you map hierarchically: general concepts at the top of the map and the more specific, less general concepts arranged hierarchically below.
- Label you concepts or instances in general with one word; sometimes multiple words if this can help clarifying the meaning of the concept (e.g. “Carbamate_fungicide” and/or “Carbamate_insecticide”, “Rice_virus_diseases” and/or “Potato_virus_diseases”); avoid “sentences” unless strictly needed or unless cannot be modelled differently; use English only.
- Use symbols as literals for concepts or instances (e.g. “P” hasChemicalSymbol for “Phosphorus”).
- Do not repeat concepts unless strictly needed; if needed reference to concepts already existing in other maps (using CMap links).
- Subclass or Instance? Keep in mind that if you can further specify types of an object, then it should be a class and not an instance. The final instances are elements that cannot be further detailed (example: a pesticide is a class, because you can specify several types of pesticides; the pesticide “Buprofezin” is an instance, because you cannot further specify any other sub-types of “Buprofezin”).
- Build individual concepts (e.g. do not create “seeds_and_sowing” but better to create two concepts “seeds” and “sowing” and relate them with appropriate relationships).

If you want to specify that an element is an instance you can also use this functionality in CMap (COE): select the concept then from the top level menu select “Edit” + “Change” + “Change to Individual”. The style of the instance will change to a rectangle.

Note: Traditionally the “is a” relationship is used for subclassOf; in CMap this is used as instanceOf.

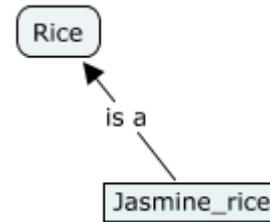


Figure 2: Different style for concepts and instances.

Guidelines for Relationships

- Use singular. All relationship names should be written starting with lower case and capitalizing other words, without any space. E.g. “isPerformedByMeansOf”.
- Connect two or more concepts or instances using linking words or phrases to form a meaningful statement (e.g. “Field_preparation” “usesProcess” “Harrowing”).
- Every concept could be related to every other concept: choose to identify the most prominent and most useful cross-links. If needed you can divide links in different maps.
- We will use only the following types of relationships from CMap (COE):
 - **are:** (subclassOf) links concepts in a hierarchical way (generic-object hasSubclass a-more-specific-object - example: “Herbicide” “**are**” “Pesticide”); use this when you can further specify types (e.g. you can list the different herbicides).
 - **is a:** (instanceOf) links every instance to its concept (example: instance “Brown_plant_hopper” “**is a**” “Plant_hoppers”); use this when you cannot further specify any sub-types of “Brown_plant_hopper”.
 - **Any other type of Property:** assign the name of this property possibly from the registry.
- Every arrow should have a name indicating the relationship.
- The names of relationships can be taken from the document provided during the workshop at Pantnagar University. IITK will prepare an online relationship registry at www.agropedia.net.
- Use relationships consistently (e.g. “usesProcess” should be used between techniques and process or methods).

Examples of “are” relationships

This relationship connects a subclass to a more generic class (e.g. connect “Rice” to “Crop”). The meaning of the arrow is the same as “isSubclassOf”.

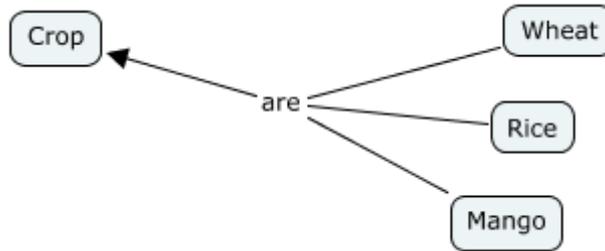


Figure 3: Example of “are” relationships in CMap (COE)

Note: there is no need to add another name to this relationship.

Note: the “are” relationship should be used consistently over a tree. E.g. in the example aside the lower element “Canidae” should be consistently the subclass of all its superclasses.

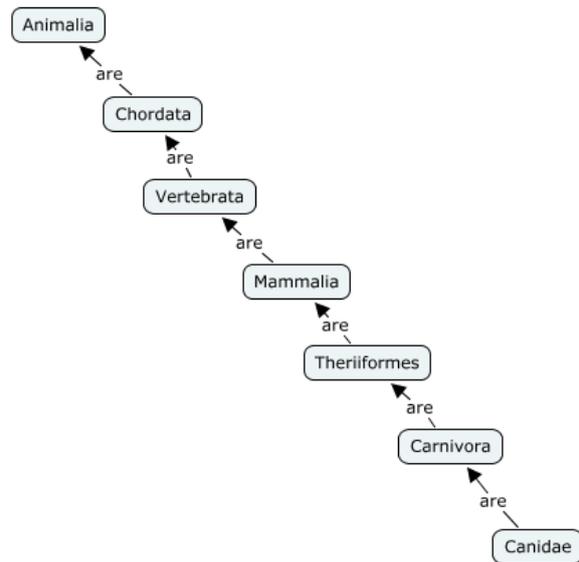


Figure 4: A series of subclasses.

So it should be true that “Canidae” is a subclass of “Theriiformes”, and subclass of “Mammalia”, and subclass of “Vertebrata”, etc. until the top level class.

If it is so, means the subclass relationship has been used correctly. See picture aside.

If in the hierarchy we have as top level crop and at the bottom a process.... the “are” relationship has not been used consistently, because a process cannot be considered a subclass of crop.

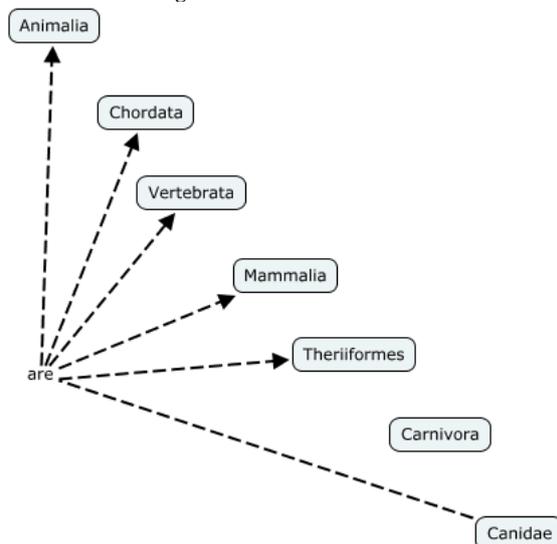


Figure 5: A correct use of subclassOf works for all concepts in the hierarchy.

Examples of “is a” relationships

This relationship connects an instance to its concept (e.g. connect “Basmati rice” to “Rice”). The meaning of the arrow is the same as “instanceOf”.

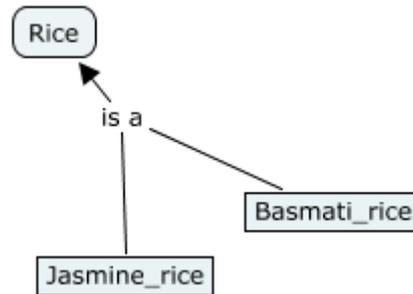


Figure 6: Example of “is a” relationships in CMap (COE)

Note: there is no need to add a name to this relationship.

Any other type of relationships

Relationships defined between 2 concepts, and can be used to relate 2 instances:

- from a concept (e.g. “Rice”) to another concept (e.g. “Pest”).
- from an instance (e.g. “Basmati_rice”) to instance (e.g. “Gundhi_bug”).

Examples:



Figure 7: relationship for an instance



Figure 8: relationships between instances

Suggestions for building KM

The model that you will build can be used for multiple purposes:

1. a generic map that can be used as guidelines for other more specific maps;
2. learn what is related to a concept in a specific context;
3. upload resources in Agropedia Indica, to find document for that concept;
4. be used to connect to information from maps from other experts;
5. to reply to a specific problem (e.g. “From which plants I can derive insecticides?”).

When building a new CMap (COE) we suggested above in the guidelines that it is better “to refer to particular questions we seek to answer (focus question)”.

Examples:

1. How do I obtain a patent for a specific variety of my crop?
2. Which agricultural experts I may contact for a problem on diseases?
3. What certifications exist for particular seeds?
4. How do I get a new certification?
5. etc...

In general expert may build specific maps for specific domains (e.g. a map on rice pests, a map on mango pests, on rice diseases, rice varieties, water management, nutrient management, post harvest management, weed management, etc.). In this case remember always to make connections to top level maps or other maps (see figure below).

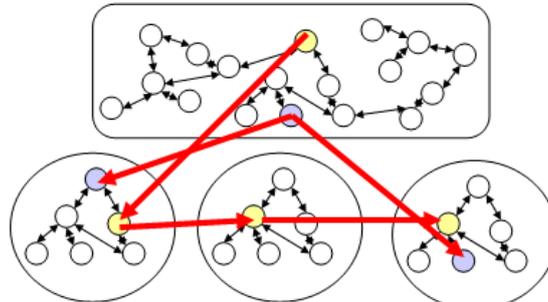


Figure 9: connections between different maps.

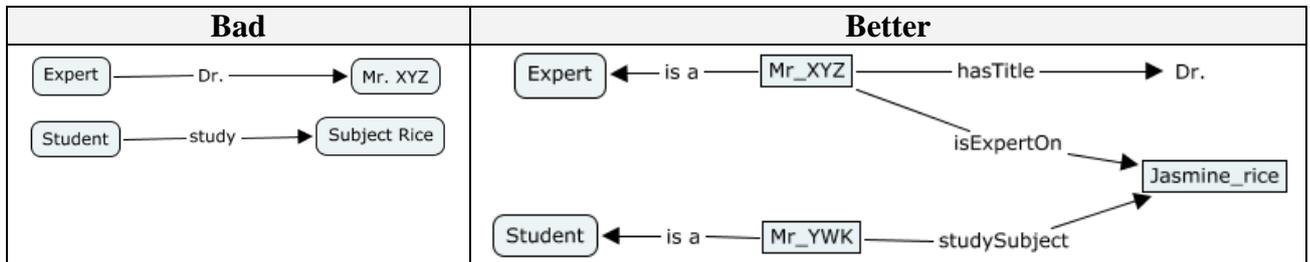
Good and Bad modelling

This section would like to explain some common mistakes we could do while modelling and solutions to avoid them. Please note the maps are not completed as they represent only the section useful to explain the examples.

NOTE: the indication “(ObjectProperty)” should be omitted in Agropedia v1.0.

Case n. 1

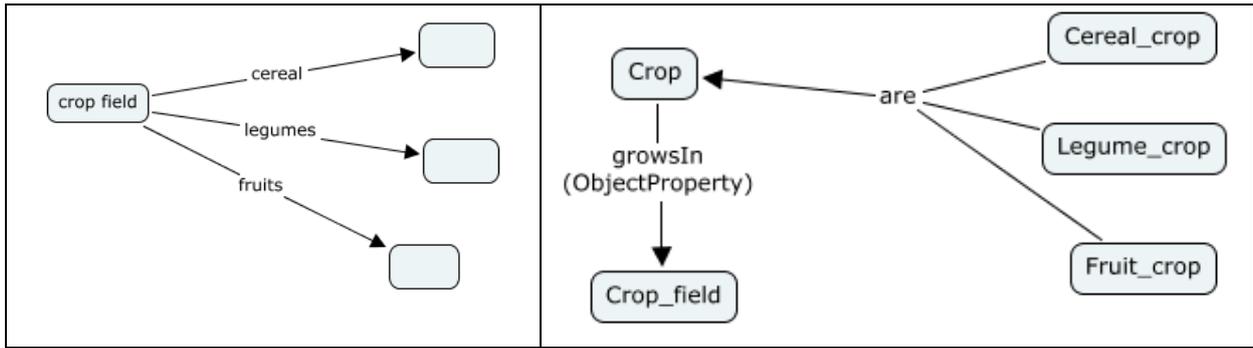
Representing a Map on people working in agriculture (experts, students of agricultural universities, etc.).



Case n. 2

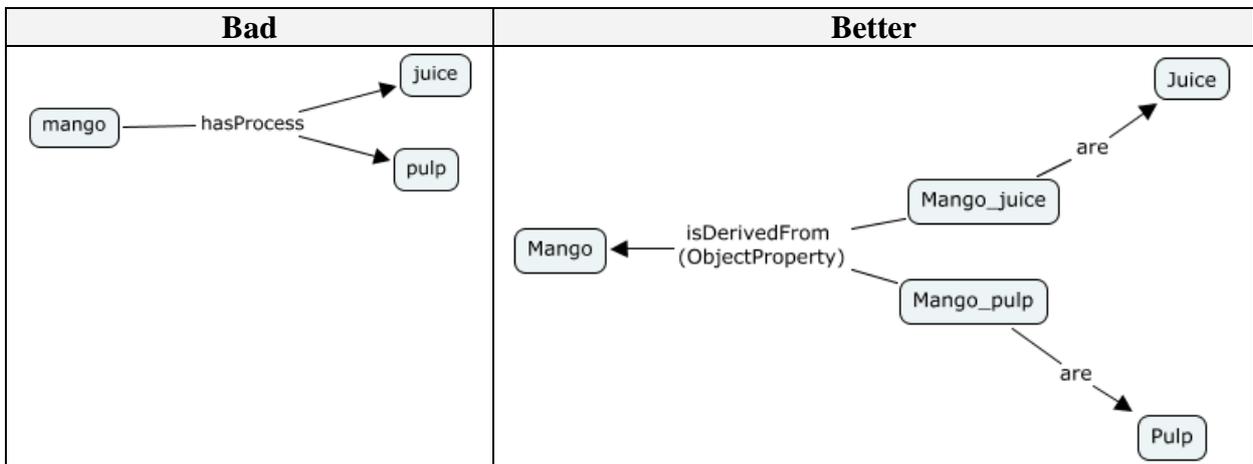
Representing different types of crops.





Case n. 3

Representing elements derived from a particular crop.



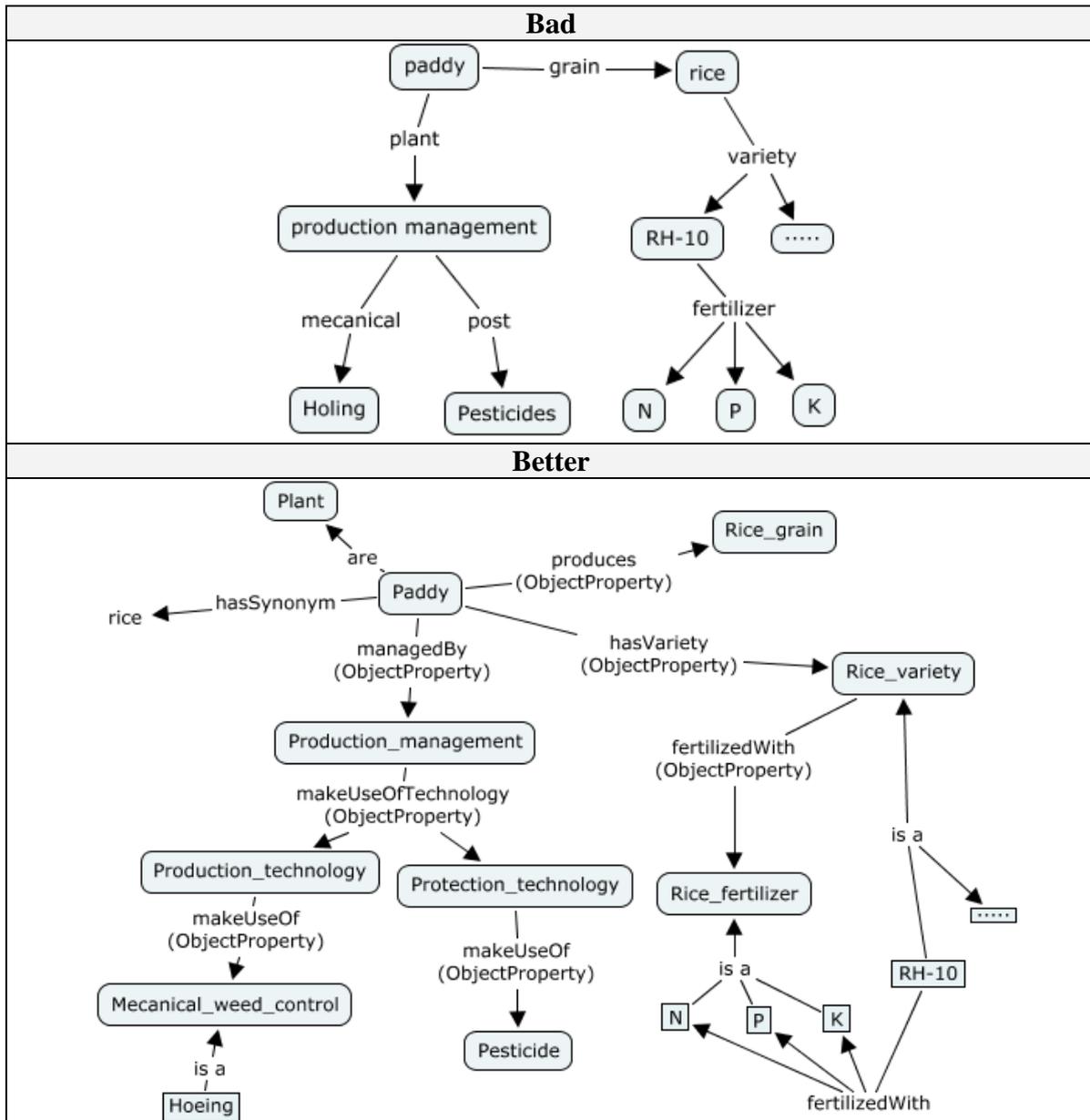
Case n. 4

Subtypes of diseases.



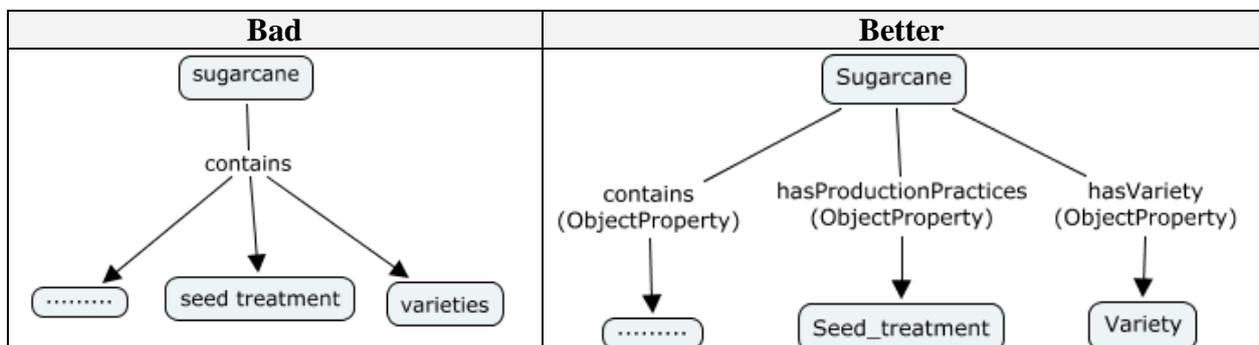
Case n. 5

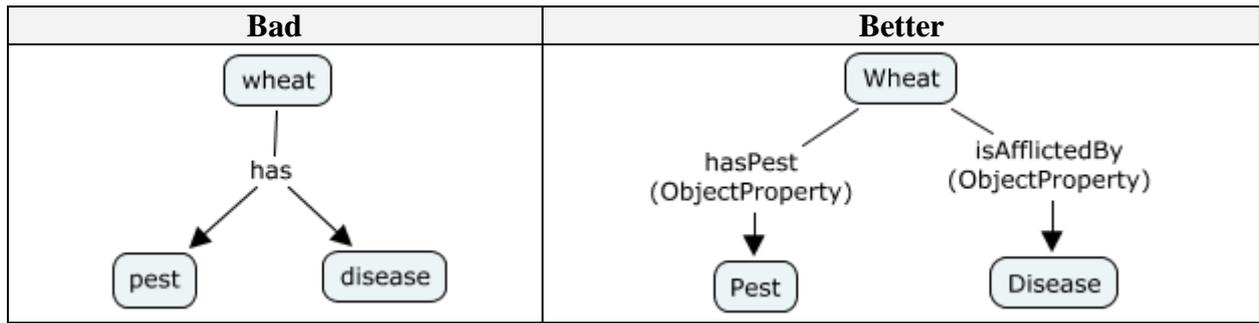
Representing a rice map.



Case n. 6

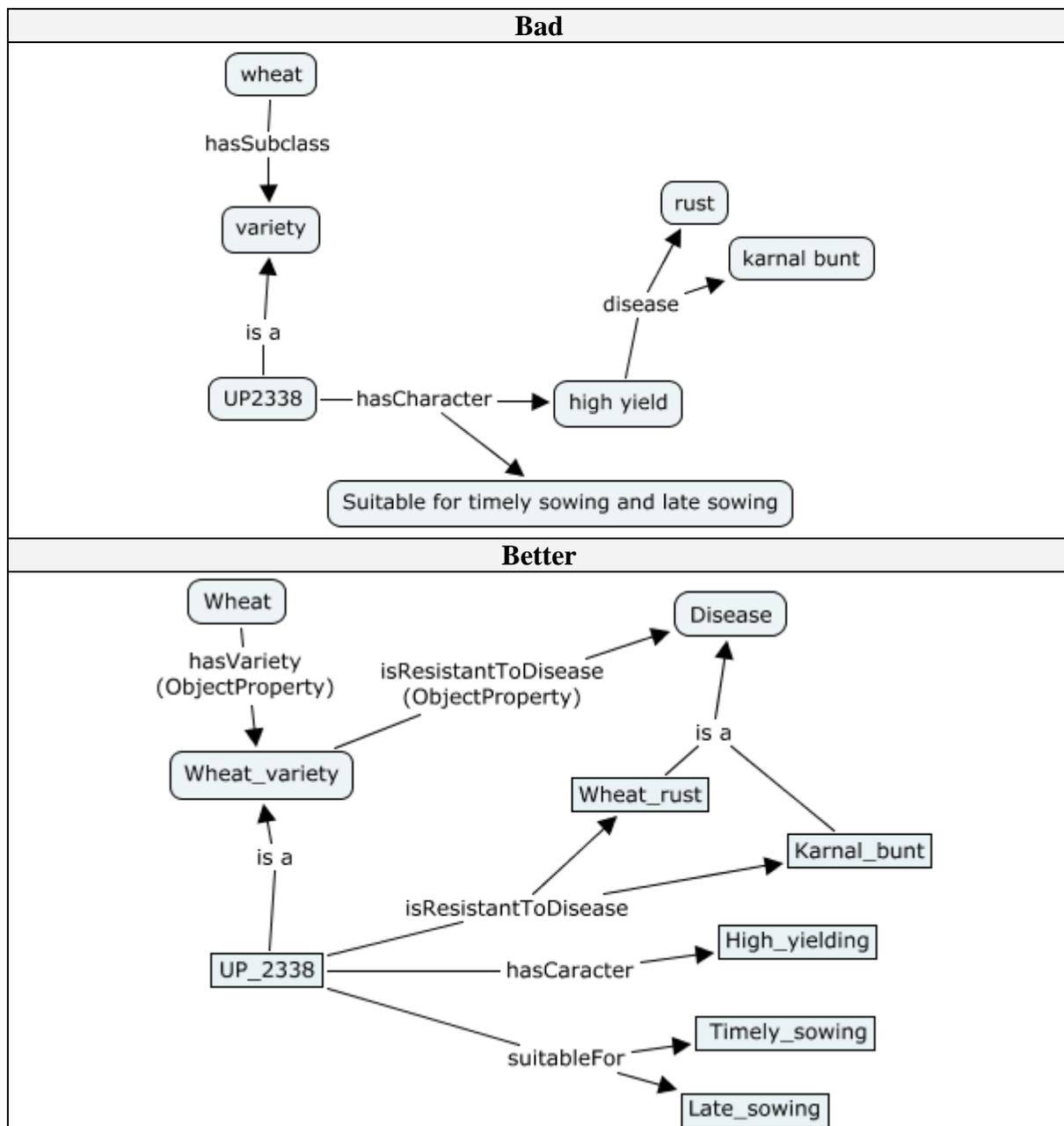
Representing different relationships.





Case n. 7

Representing a Wheat map.



Case n. 8: mathematical expressions

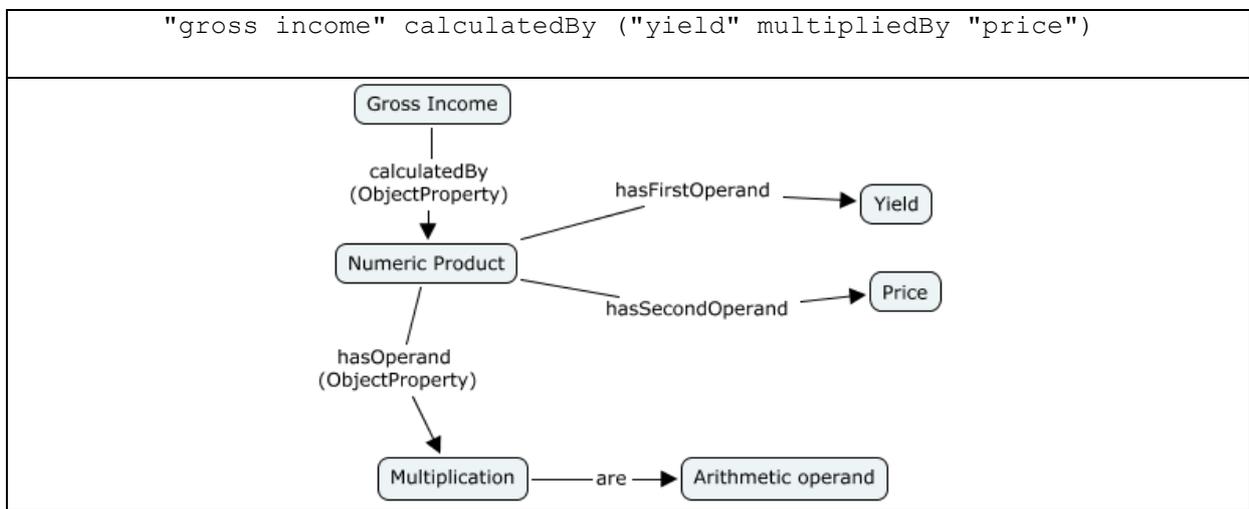
Express a mathematical expression, such as:

- "gross income" calculatedBy ("yield" multipliedBy "price")
- "net income" calculatedBy ("gross income" minus "cost product")

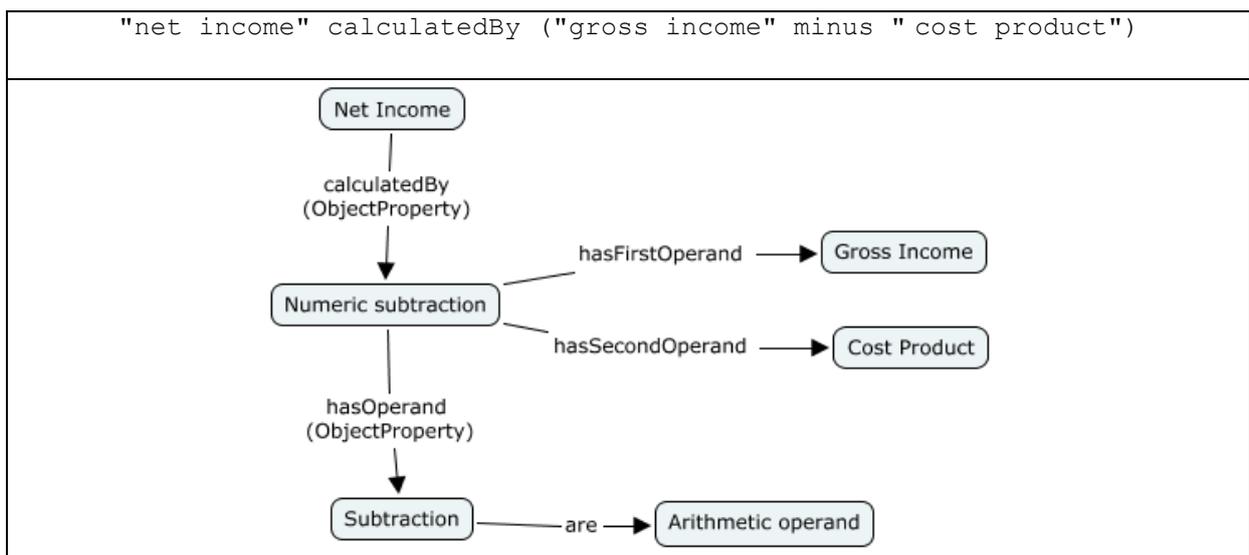
Note: OWL is not a language used for calculations, is only a representation language. But if we are able to express these mathematical expressions, we could make calculations using a query language such as SPARQL.

The idea is to create a concept that identify every calculation, and indicate the order of operands and the operator.

The two examples above will be represented as:



Similarly we have:



Management of synonyms

Synonyms for concepts or classes should be created in the following way:

- create a relationship called “hasSynonym” from the concept you want the synonym of (e.g. “Rice”);
- select the concept that appear and enter the synonym (e.g. “Paddy”);
- select the synonym and chose from the top menu “Edit” + “Change” + “Change to Literal”;

The result should be similar to the following:

