From thesauri to Ontologies: A short case study in the food safety area in how ontologies are more powerful than thesauri

From thesauri to RDFS to OWL

Boris Lauser
Food and Agriculture Organisation of the United Nations
Rome, Italy

Keywords: Thesauri, Ontologies, RDFS, OWL, KAON, Protégé

This short case study will show on the basis of a simple example taken from the Food Safety area, how ontologies differ from thesauri. The example will start with showing an extract from the AGROVOC[1] thesaurus and exploring the information that can be extracted from here. We will then develop this example further in order to show growing functionality and expressive power first in RDFS and finally in OWL ontologies.

1. Knowledge modelling in thesauri

Figure 1 shows a graphical representation of a few terms and their relationships taken from AGROVOC.

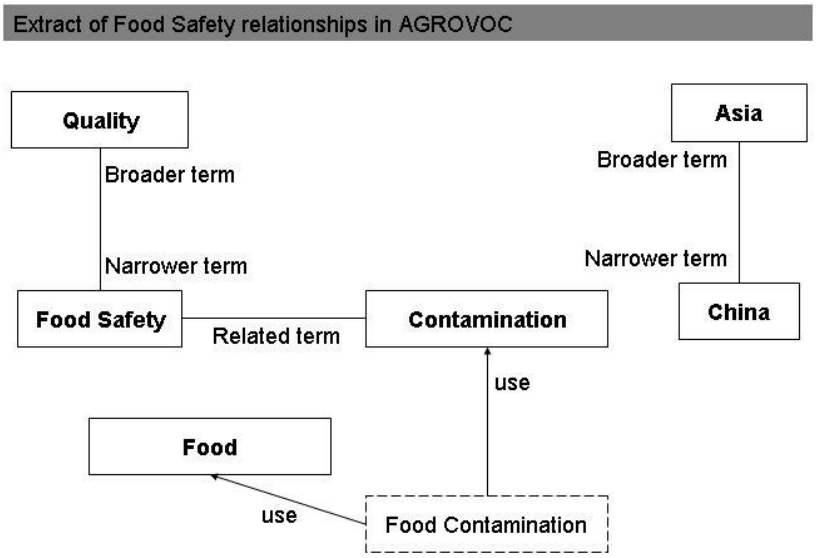

Figure 1: Thesaurus example on Food Safety taken from AGROVOC

[1] AGROVOC is a multilingual, agricultural thesaurus, developed and maintained by the Food and Agriculture Organization (FAO). It is accessible online at http://www.fao.org/agrovoc.

Basically, all the possible modelling paradigms (the most important) of thesauri are represented in the above example. Let's have a closer look at the information we can get out of this example:
The example shows the term "Food Safety" as a narrower term of "Quality", which means that "Food Safety" is generally regarded as a specification of "Quality". It is further linked via the generic 'related term' relationship to the term "Contamination" which means that in some way, "Food Safety" has to do with "Contamination". The relationship between "Contamination", "Food" and "Food Contamination" expresses synonymy, i.e. the term "Food" should be used in combination with "Contamination" in order to express "Food Contamination". In thesauri this relationship is mainly used for indexing purposes, indicating that instead of using "Food Contamination" for indexing a document, the other two terms should be used in combination. Furthermore, China is shown as a narrower term of Asia.

Obviously, the information that can be extracted from the above example is very limited to static and rather unspecific information. There is no indication on how "Food Safety" is affected by "Contamination", neither is there a consistent application of the broader/narrower term relationship. It simply indicates that in some way one term is more specific than another. The screen shot in Figure two shows the full extract of the term "Food Safety" taken from AGROVOC. The application of relationships shown here leave much space for individual interpretation.

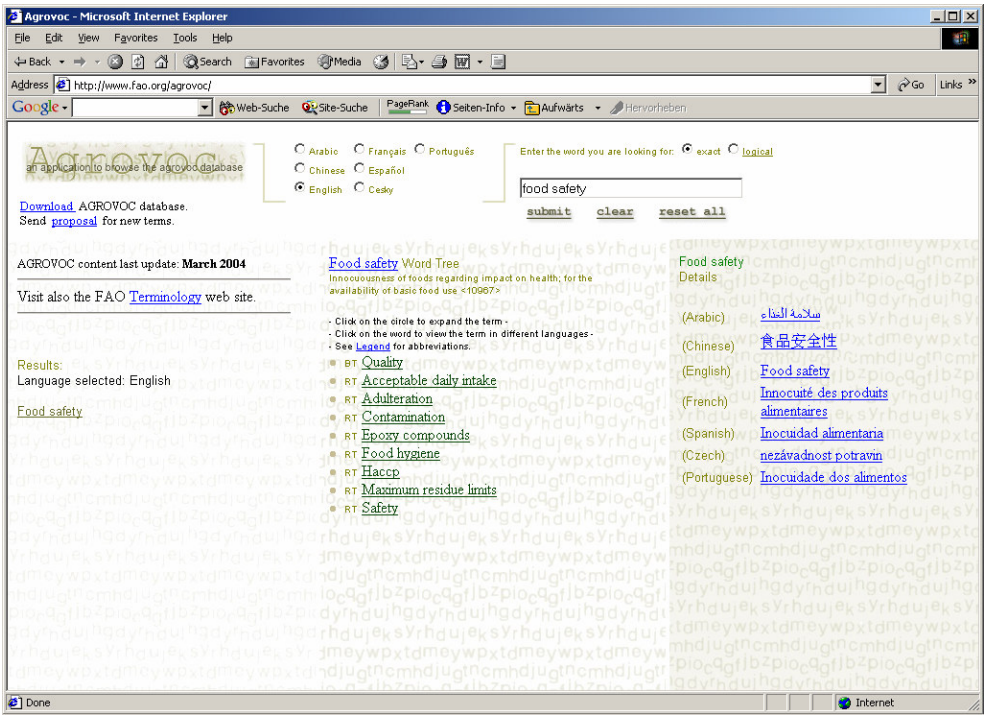

Figure 2: The term "Food Safety" in the AGROVOC thesaurus

Now what, if you want to talk about specific food items in specific countries and want to retrieve information about their food safety? In a thesaurus, as seen in the above examples, you can only model unspecified relations on a conceptual level. No notion of

specific instances exist, neither a way to express that one specific instance is contaminated and another is not. The following section will show how ontologies modelled in RDFS can address these issues and provide the necessary modelling principles for more precise representations.

2. Ontologies in RDFS

While in thesauri only terms are related, RDFS introduces the notions of **concepts** and **instances**. A concept describes an entity on an abstract level with generic properties, whereas an instance is an actual representation of this concept with specific values of these properties. Figure 3 shows a screenshot of a conceptual ontology modelled in RDFS using the OIModeler, one of the KAON[2] tools.

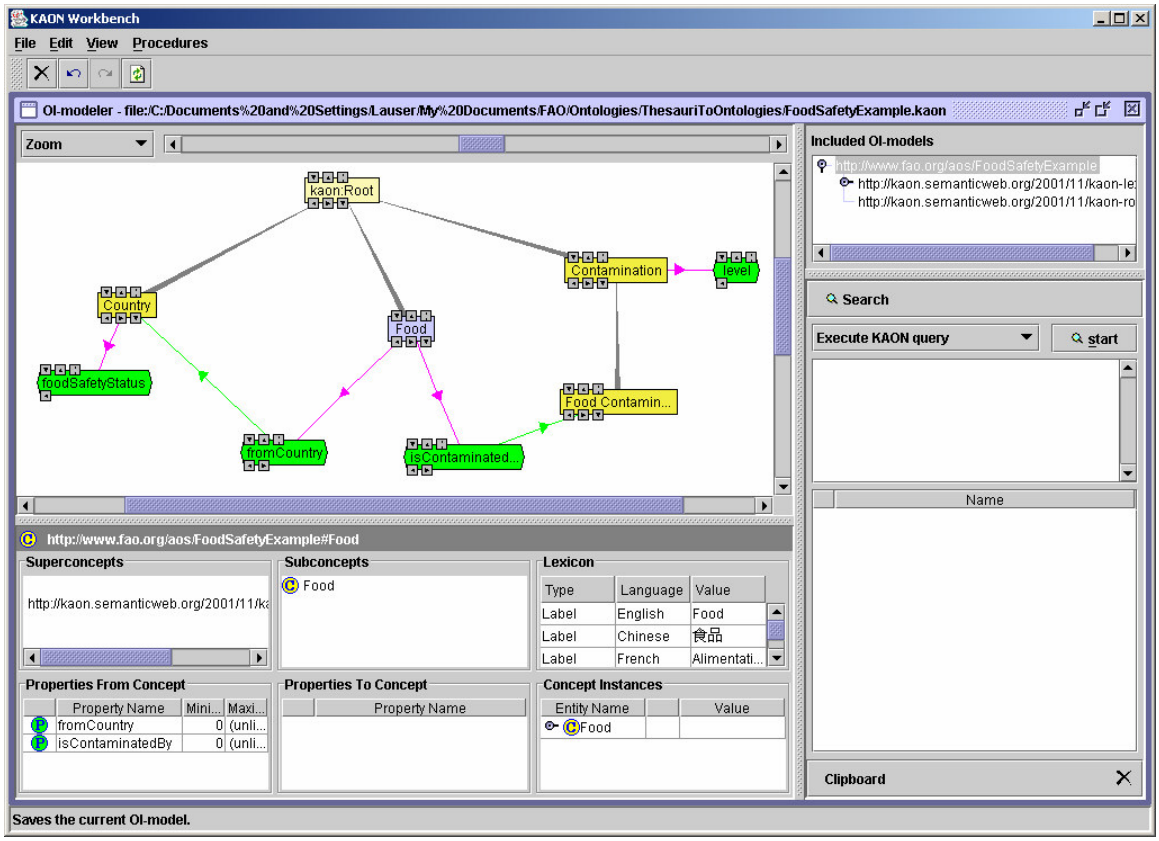

Figure 3: Mini example of a conceptual model of an RDFS ontology on Food Safety modelled in KAON

In an RDFS ontology, concepts are basically ordered in a sub-concept / super-concept hierarchy, similar to a broader term / narrower term hierarchy in thesauri. The top of the hierarchy is given by the Root node (here kaon:Root). In our example there are basically three upper level concepts “Country”, “Food” and “Contamination”. “Food Contamination” is a sub-concept of “Contamination”. A sub-concept is, however, much

[2] KAON is an ontology tool environment developed at the University of Karlsruhe, Germany. The model is based on RDFS with some proprietary extensions. The software is freely available at: http://kaon.semanticweb.org.

more specific than a narrower term relationship mainly due to the important paradigm of **inheritance**. A sub-concept inherits all properties and relationships from its super-concept. In our example, the concept "Contamination" has the property "level", meaning that "Contamination" of something is determined by its level of contamination. We don't have to reapply this property on the level of "Food Contamination". It simply gets inherited, because "Food Contamination" is a sub-concept of "Contamination".
Besides this attribute-like property "level" our little example shows two properties that connect concepts: "fromCountry", meaning that a particular food comes from a particular country and "isContaminatedBy", meaning that a particular food is contaminated by a food contamination. As opposed to thesauri, where we are limited to the simple 'broader term', 'narrower term' and 'related term' relationships, these relationship/property[3] types are limitless in ontologies. I can therefore more specifically describe the relationship that holds between two concepts.
In ontologies, such a conceptual model is now accompanied by a **lexicon**, binding terms to entities (concepts, properties, etc.). In Figure 3 you can see that the concept food has three labels in three different languages. Other lexical information that can be attached to a concept consists of synonyms, word stems or definitions and descriptions. Whereas in thesauri the notions of terms and concepts are not well defined, in an ontology conceptual and lexical information is clearly separated, rendering it less ambiguous.

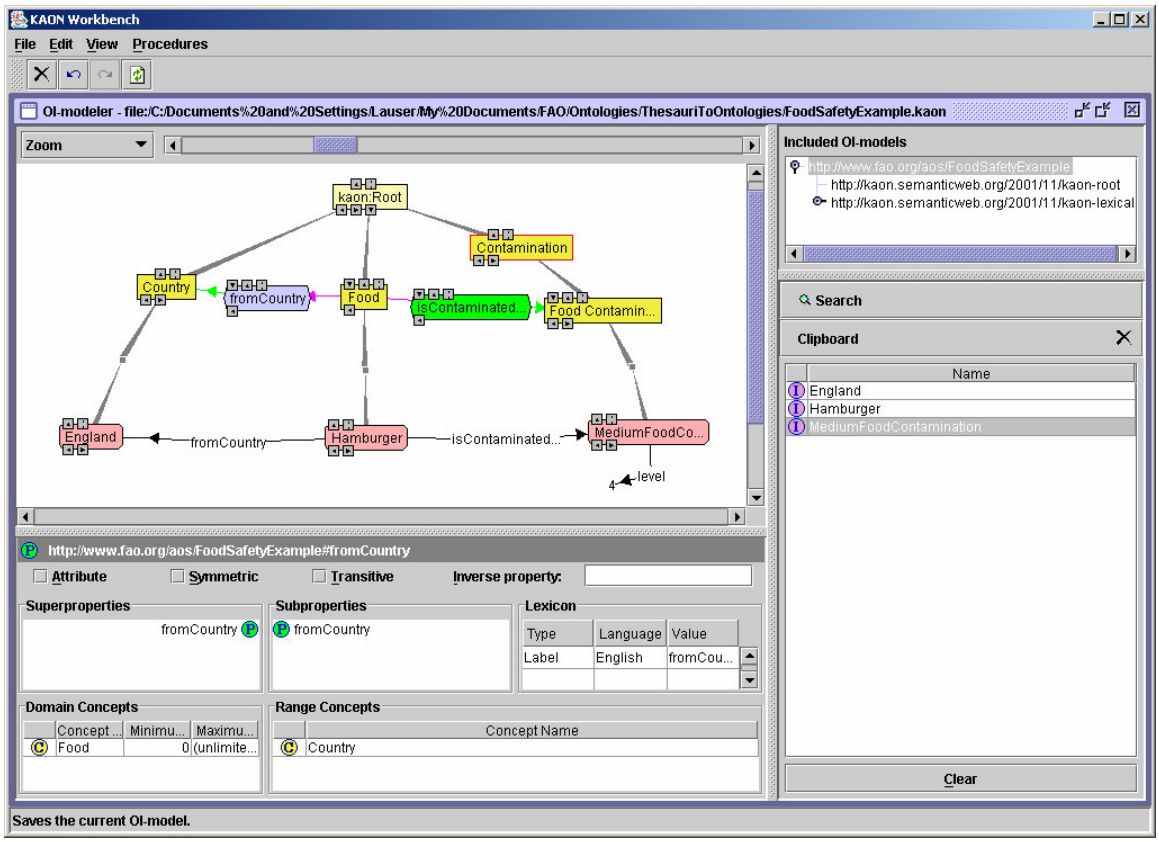

Figure 4: Instances in an RDFS ontology modelled in KAON

[3] In this document we use property and relationship synonymously.

Another important difference is the introduction of **instances** in ontologies. If you would like to express in a thesaurus for example that a specific food, let's say a 'Hamburger' comes from 'England', you could only create these two new terms as narrower terms of country and food respectively and then relate them using a 'related term' relationship. However, this does not give you any further information or meaning. Figure 4 shows the two new items as instances, interrelated with an instance of the 'fromCountry' property. The specific instance 'Hamburger' of the concept "Food" 'comesFrom' the country England and is contaminated by a 'MediumFoodContamination', which in fact is an instance of the concept "Food Contamination" with the level of contamination 4 (assuming that there might be levels of contamination on a range from 1 to 5 for example). In a thesaurus, this level of specificity would not be achievable with the modelling principles it offers.

The example has shown, that an RDFS ontology offers considerably more powerful modelling principles than a thesaurus does, the main ones are:

- The notion of Concepts vs. instances
- Separation of entities and terms
- Specification of concepts via properties and attributes
- The ability to create infinite, more specific and semantically richer properties
- The inheritance paradigm.

However, RDFS has restrictions. Imagine in the example you want to express that a Food Contamination is a Contamination with level > 3. RDFS does not give you the opportunity to do so. In fact you could create any instance of "Food Contamination" and assign any level and there would be no mechanism to check or restrict this on the class level. Furthermore, there is no cardinality constraint in RDFS, making it impossible to restrict the number of instances that a certain concept can have. These and other issues are resolved in the new OWL (Web Ontology Language), a recent recommendation released by the W3C[4].

3. Ontologies in OWL

OWL adds more vocabulary for describing concepts and properties. One of the most important features of OWL is **overriding** properties at the class level using restrictions. In our example before in RDFS we could not model that Food Contamination is a Contamination with levels >= 3. In OWL this is now possible simply by including a restriction on the class level of "Food Contamination". For this purpose we first introduce a new class "Contamination Levels" and populate it with 5 instances {1,2,3,4,5} that define the different levels. On the class level of "Food Contamination" we now add a restriction stating that all values of the "level" property (which is already inherited from "Contamination") must be one of the values {3,4,5}. Figure 5 shows the class "Food Contamination" with its restriction on the property "Level". The example OWL ontology has been modelled in Protégé[5], an ontology editor which recently released an OWL plug in. The code of the sample ontology is attached in Appendix B.

[4] http://www.w3.org/TR/owl-features/.
[5] http://protege.stanford.edu.

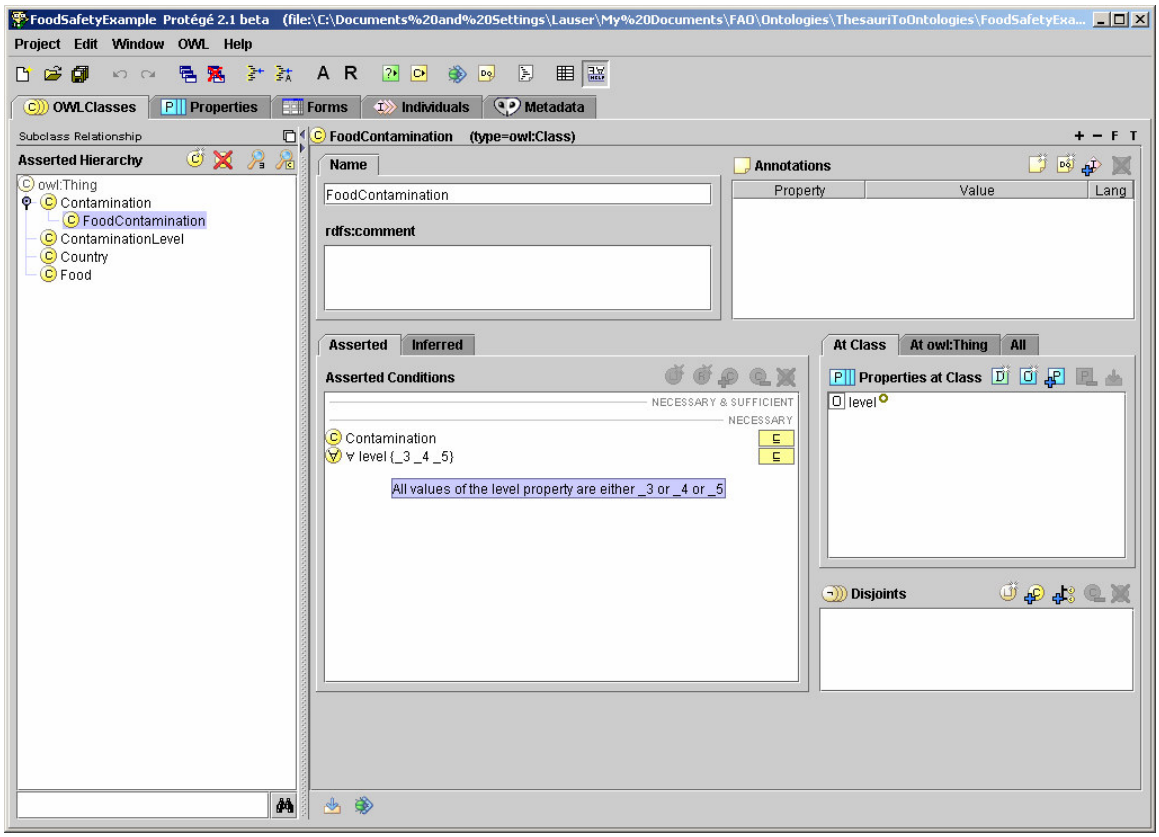

Figure 5: Modelling class level restrictions in OWL using Protégé

In RDFS there is no mechanism that would prevent from creating an instance of "Food Contamination" with a level 2 or lower. In Protégé on the other hand, having added the OWL restriction on the class level, trying to create such an instance will raise an exception as shown in Figure 5.
These class level restrictions finally are a powerful instrument that can be used for inference, since reasoners can use such restrictions to infer knowledge and deduct links that are not modelled explicitly. Other useful restrictions in OWL include **cardinality**, i.e. to model that a property must have at least or at most x values.

Besides class level restrictions, the OWL web ontology language introduces several more additional features over RDFS, some of the most important are:

- Equality:
  In OWL you can explicitly state that two entities are equal. This is especially helpful in the case of integrating or merging ontologies. In RDFS you could not state for example that the two instances 'Burger' and 'Hamburger' actually refer to the same physical entity. In OWL you can so this using the equivalence statement.
- Versioning:
  RDFS did not provide any ontology versioning and metadata information. This is extremely important especially with evolving ontologies. As ontology engineering and maintenance is an ever ongoing process, OWL introduced the feature of

adding metadata and versioning information to keep track of ontology development over time.

- Data Types
  In RDFS no data types as we know from programming languages (String, Integer, Float, etc.) could be used for attributes of concepts. In order to close this gap, OWL introduces the use of such standard predefined data types.

OWL provides a few more features than the ones mentioned here and we do not want to go into the details of all of them within this overview. For a full list of OWL features, refer to the online OWL web ontology language guide[6].

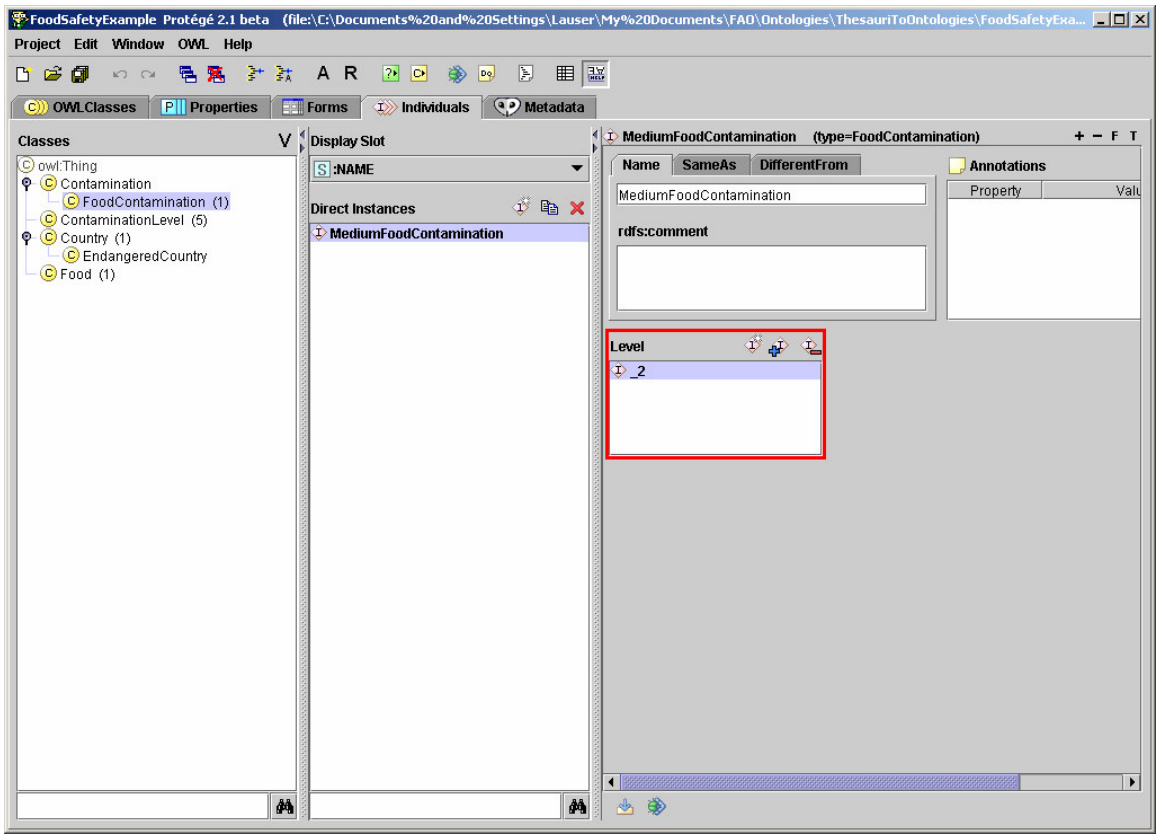

Figure 5: Enforcement of class level restrictions in OWL

It becomes obvious in the above last example that such modelling paradigms in the case of OWL provide powerful opportunities for knowledge inference. In the example above, one could imagine to have a country food safety alert system that categorized a country into an alert status, if a certain number of food items are contaminated with a certain level. The creation of such tools has not been possible with the simple knowledge that can be extracted from a thesaurus, but becomes only possible exploring the powerful features, ontology modelling provides.

[6] http://www.w3.org/TR/owl-guide/.

Summary

In this short case study we have shown how thesauri differ from RDFS and OWL ontologies. We have shown on a simple example the growing expressiveness of the languages from thesauri to RDFS to finally OWL evolving from a simple thesaurus model to a more complex OWL ontology model. Whereas thesauri provide only very basic modelling paradigms and no knowledge can be extracted from a thesaurus except mere simple keyword relationships, OWL allows for powerful knowledge modelling and inference by offering a vast variety of features. Moreover, thesauri are mostly stored in proprietary databases without standard exchange formats or procedures, whereas OWL is a W3C recommendation, encoded in XML and therefore allows for perfect interoperability on the web.

Appendix A: RDFS sample Food Safety ontology

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [
   <!ENTITY kaon 'http://kaon.semanticweb.org/2001/11/kaon-lexical#'>
   <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
   <!ENTITY a 'http://www.fao.org/aos/FoodSafetyExample#'>
   <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
]>

<rdf:RDF xml:base="http://www.fao.org/aos/FoodSafetyExample"
   xmlns:kaon="&kaon;"
   xmlns:rdf="&rdf;"
   xmlns:a="&a;"
   xmlns:rdfs="&rdfs;">

<rdfs:Class rdf:ID="Food-Contamination">
   <rdfs:label xml:lang="en">Food Contamination</rdfs:label>
   <rdfs:subClassOf rdf:resource="#Contamination"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Food">
   <rdfs:label xml:lang="en">Food</rdfs:label>
   <rdfs:label xml:lang="zh">é£Ÿå“□</rdfs:label>
   <rdfs:label xml:lang="fr">Alimentation</rdfs:label>
</rdfs:Class>
<rdfs:Class rdf:ID="Contamination">
   <rdfs:label xml:lang="en">Contamination</rdfs:label>
</rdfs:Class>
<rdfs:Class rdf:ID="Country">
   <rdfs:label xml:lang="en">Country</rdfs:label>
</rdfs:Class>
<rdf:Property rdf:ID="fromCountry">
   <rdfs:label xml:lang="en">fromCountry</rdfs:label>
   <rdfs:domain rdf:resource="#Food"/>
```

```
    <rdfs:range rdf:resource="#Country"/>
</rdf:Property>
<rdf:Property rdf:ID="isContaminatedBy">
    <rdfs:label xml:lang="en">isContaminatedBy</rdfs:label>
    <rdfs:domain rdf:resource="#Food"/>
    <rdfs:range rdf:resource="#Food-Contamination"/>
</rdf:Property>
<rdf:Property rdf:ID="foodSafetyStatus">
    <rdfs:label xml:lang="en">foodSafetyStatus</rdfs:label>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
    <rdfs:domain rdf:resource="#Country"/>
</rdf:Property>
<rdf:Property rdf:ID="level">
    <rdfs:label xml:lang="en">level</rdfs:label>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
    <rdfs:domain rdf:resource="#Contamination"/>
</rdf:Property>
<a:Food rdf:ID="Hamburger">
    <rdfs:label xml:lang="en">Hamburger</rdfs:label>
    <a:fromCountry rdf:resource="#England"/>
    <a:isContaminatedBy rdf:resource="#MediumFoodContamination"/>
</a:Food>
<a:Country rdf:ID="England">
    <rdfs:label xml:lang="en">England</rdfs:label>
</a:Country>
<a:Food-Contamination rdf:ID="MediumFoodContamination"
    a:level="4">
    <rdfs:label xml:lang="en">MediumFoodContamination</rdfs:label>
</a:Food-Contamination>

</rdf:RDF>
```

Appendix B: OWL sample Food Safety ontology

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns="http://a.com/ontology#"
    xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://a.com/ontology">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
  </owl:Ontology>
  <owl:Class rdf:ID="Contamination"/>
  <owl:Class rdf:ID="FoodContamination">
```

```
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty>
     <owl:ObjectProperty rdf:about="#level"/>
    </owl:onProperty>
    <owl:allValuesFrom>
     <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
       <ContaminationLevel rdf:ID="_3"/>
       <ContaminationLevel rdf:ID="_4"/>
       <ContaminationLevel rdf:ID="_5"/>
      </owl:oneOf>
     </owl:Class>
    </owl:allValuesFrom>
   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Contamination"/>
</owl:Class>
<owl:Class rdf:ID="Food"/>
<owl:Class rdf:ID="ContaminationLevel"/>
<owl:Class rdf:ID="Country"/>
<owl:ObjectProperty rdf:ID="level">
  <rdfs:domain rdf:resource="#Contamination"/>
  <rdfs:range rdf:resource="#ContaminationLevel"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="fromCountry">
  <rdfs:range rdf:resource="#Country"/>
  <rdfs:domain rdf:resource="#Food"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="contaminatedBy">
  <rdfs:domain rdf:resource="#Food"/>
  <rdfs:range rdf:resource="#FoodContamination"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="foodSafetyStatus">
  <rdfs:domain rdf:resource="#Country"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<ContaminationLevel rdf:ID="_2"/>
<Country rdf:ID="England"/>
<FoodContamination rdf:ID="MediumFoodContamination">
  <level rdf:resource="#_3"/>
</FoodContamination>
<ContaminationLevel rdf:ID="_1"/>
<owl:Restriction>
  <owl:hasValue rdf:resource="#MediumFoodContamination"/>
```

```
    <owl:onProperty rdf:resource="#contaminatedBy"/>
  </owl:Restriction>
  <Food rdf:ID="Hamburger">
    <fromCountry rdf:resource="#England"/>
    <contaminatedBy rdf:resource="#MediumFoodContamination"/>
  </Food>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 1.1 beta, Build 104)
http://protege.stanford.edu -->
```