Reengineering Thesauri for New Applications: the AGROVOC Example

Dagobert Soergel*, Boris Lauser, Anita Liang, Frehiwot Fisseha, Johannes Keizer and Stephen Katz
*College of Library and Information Services, University of Maryland, College Park
Email: dsoergel@umd.edu
Food and Agriculture Organization (FAO) of the United Nations,
Library & Documentation Systems Division, 00100 Rome, Italy
Email: {frehiwot.fisseha; boris.lauser; anita.liang; johannes.keizer; stephen.katz}@fao.org

Abstract

Existing classification schemes and thesauri are lacking in well-defined semantics and structural consistency. Empowering end users in searching collections of ever increasing magnitudes with performance far exceeding plain free-text searching (as used in many Web search engines), and developing systems that not only find but also process information for action, requires far more powerful and complex knowledge organization systems (KOSs). The paper presents a conceptual structure and transition procedure to support the shift from a traditional KOS towards a full-fledged and semantically rich KOS. The proposed structure also complies with other interoperability approaches like RDFS and XML in the Web environment. AGROVOC, a traditional thesaurus developed and maintained by the Food and Agriculture Organization (FAO) of the United Nations, serves as a case study for exploring the reengineering of a traditional thesaurus into a fully-fledged ontology. We start the process of developing an inventory of specific relationship types with well-defined semantics for the agricultural domain and explore the rules-as-you-go approach to streamlining the reengineering process.

1. From thesauri to rich ontologies

1.1. The problem

Empowering end users in searching collections of ever increasing magnitudes with performance far exceeding plain free-text searching (as used in many Web search engines), and developing systems that not only find but also process information for action, require considerably more powerful - and complex - knowledge organization systems (KOS) than the classification schemes and thesauri that currently exist. Such systems must serve the following functions, among others:

- Improved user interaction with the KOS on both the conceptual and the term level for improved query formulation and subject browsing, and for more user learning about the domain.
- Intelligent behind-the-scenes support for query expansion, both concept expansion and synonym expansion, within one language and across languages.
- Intelligent support for human indexers and automated indexing/categorization systems.
- Support for artificial intelligence and semantic Web applications.

All of these functions require semantic relations that are more expressive and nuanced than the few rudimentary categories and relationships found in traditional thesauri and classifications.

A typical scenario in information retrieval illustrates some of the shortcomings of current free-text search engines such as Google. A farmer is interested in finding out about *rice* and starts a search by entering the string '*rice*'. The results returned in response to the query immediately indicate several problems. First, because the system performs the search based on the actual text string entered rather than on an interpretation of the meaning of the string, many irrelevant results are retrieved. This occurs because the query term itself is ambiguous (i.e. *rice* can refer to the grain, to the university in Houston, or to the name of an author, among others). Further, there are millions of results with no apparently meaningful arrangement. To find something of possible relevance, the user may need to click and scan page after page of the retrieved results. Finally, the user is stuck with the results that have been retrieved; to find other related resources, such as *rice cultivation*, the user must start from the beginning again and formulate a different query, despite the fact that the new query corresponds to concepts related to the original query. The problem becomes evident: The biggest challenge in information retrieval is **concept identification** in a specific **domain of interest**!

In contrast, in a semantics-driven information retrieval system, the system would recognize, i.e. "understand", that the string '*rice*' was ambiguous; it would then request clarification from the user as to which of the possible meanings was intended. Only then, after the user disambiguated the term, would the system execute the search. The system would then retrieve only those resources that had been semantically marked up (through manual or automatic indexing) with the concept of *rice*, no matter what words or even languages are used in the resources to refer to *ric*e. Moreover, because the system is semantically rich, it not only presents results that are based on understanding the user's request, it also offers related concepts the user might not have thought of initially. Based on a <*hasPest*> relation, the system could display such concepts as *rice weevil* and *rice moth*. Searching on these latter concepts could in turn lead to concepts on pesticides used on rice, and so on. The system could retrieve not only information directly pertinent to the user's query but also help the user explore and clarify the information need and find useful related information. In this scenario, a KOS has two functions: assisting the user with exploring the topic of the query, and supporting intelligent automatic indexing (metadata assignment) through statistical and syntactic-semantic analysis and "understanding" of text; both functions require a KOS with a rich and precisely defined semantic structure.

To accomplish these and other more sophisticated tasks, the new KOS must marry the conceptual structure of full-fledged ontologies - well-structured hierarchies of concepts connected through a rich network of detailed relations that support concept retrieval and reasoning - with the terminological richness of good thesauri. While existing KOSs do not provide the full set of precise concept relations needed for reasoning, existing KOSs, both large and small, represent much intellectual capital. This paper explores the question of how this intellectual capital can be put to use in constructing full-fledged KOSs.

Please follow the steps outlined below when submitting your manuscript for the Proceedings of the 5th AOS Workshope. The Workshop Proceedings are printed using camera-ready papers prepared for printing by their authors. [NOTE: These written instructions serve as a representative sample of how your finished paper should look when printed on your home or office printer.]

1.2. The relationship of traditional KOS to ontologies

Reengineering thesauri, classification schemes, etc., into ontologies means building on the information contained in them and refining that information as needed. Consider the relationships given in the **ERIC Thesaurus** (ERIC = Educational Resources Information Center) with those given in a hypothetical ontology as shown in Table 1.

Table 1: Statements and rules of a hypothetical ontology versus the information given in the ERIC thesaurus (broader term (BT), related term (RT))

Eric Thesaurus	**Hypothetical ontology**
reading instruction BT instruction RT reading RT learning standards **reading ability** BT ability RT reading RT perception	**Statements:** **reading instruction** <*isa*> instruction <*hasDomain*> reading < *governedBy*> learning standards **reading ability** <*isa*> ability <*hasDomain*> reading <*supportedBy*> perception

	Rule 1 **Instruction in a domain should consider ability in that domain:** **X *shouldConsider* Y** **IF** X <*isa (type of)*> instruction **AND** X <*hasDomain*> W **AND** Y <*isa*> ability **AND** Y <*hasDomain*> W yields: : The designer of *reading instruction* should also consider *reading ability.* **Rule 2** **X *shouldConsider* Z** **IF** X <*shouldConsider*> Y **AND** Y <*supportedBy*> Z yields: The designer of *reading instruction* should also consider *perception.*

The inferences given rely on the detailed semantic relationships given in the ontology. But the ERIC thesaurus gives only some poorly defined broader term (BT) and related term (RT) relationships. These relationships are not differentiated enough to support inference.

For another example, consider the hypothetical ontological relationships and rules we could formulate with these relationships in an example taken from the AGROVOC thesaurus (described in detail in section 2) in Table 2.

Table 2: AGROVOC relationships compared with more differentiated relationships of a hypothetical ontology (narrower term (NT), broader term (BT))

AGROVOC	Hypothetical Ontology
Undifferentiated hierarchical relationships in AGROVOC **milk** NT cow milk NT milk fat **cow** NT cow milk **Cheddar cheese** BT cow milk	Differentiated relationships in an ontology **milk** <*includesSpecific*> cow milk <*containsSubstance*> milk fat **cow** <*hasComponent*> cow milk* **Cheddar cheese** <<*madeFrom*> cow milk
	Rule 1 **Part X <*mayContainSubstance*> Substance Y** **IF** Animal W <*hasComponent*> Part X **AND** Animal W <*ingests*> Substance Y **Rule 2** **Food Z <*containsSubstance*> Substance Y** **IF** Food Z <*madeFrom*> Part X **AND** Part X <*containsSubstance*> Substance Y

In the context of food and nutrition it makes eminent sense to consider milk and egg as parts of an animal since their nutritional value and safety depend on the nature of the animal and the feed it ingests just as do skeletal meat and organ meat. This is an example of careful definition of relationships.

From the statements and rules given in the ontology, a system could infer that *Cheddar cheese <containsSubstance>milk fat* and, if cows on a given farm are fed mercury-contaminated feed, that *Cheddar cheese* made from milk from these cows *<mayContainSubstance>mercury*. But the present AGROVOC Thesaurus (described in detail in section 2) gives only narrower term/broadr term (NT/BT) relationships without differentiation.

The **limitations of existing KOS** can be summarized as follows:

- **Lack of conceptual abstraction:** thesauri and other traditional KOSs are collections of terms (generic or domain-specific), ordered in a polyhierarchic lattice structure or a monohierarchic tree structure and interlinked with some very broad and basic relationships. The distinction between a concept (meaning) and its lexicalizations (words) is not made consistently, if at all, in such a system, and as such it does not reflect the ways humans understand the world in terms of meaning and language.
- **Limited semantic coverage**: most thesauri do not differentiate concepts into types (such as *living organism*, *substance*, or *process*) and have a very limited set of relationships between concepts, distinguishing only between hierarchical relationships, i.e. NT/BT, and associative relationships, i.e. RT. These very rudimentary relationships are not powerful enough to guide a user in meaningful information discovery on the Web or to support inference. They do not reflect the conceptual relationships that people know and that can be used by a system to suggest concepts for expanding the query or making it more specific. Examples:
  - The relation between *cow* and its part *cow milk* is expressed as NT rather than the more semantically expressive relation <*hasComponent*>, so a user who wants to expand the query hierarchically (search for all concepts narrower than *cow* as well) could not distinguish between searching for all cow parts or searching for all varieties of cow;
  - the relation between *mad cow disease* and the animal it afflicts, *cow*, is expressed using RT instead of the more semantically precise relation <*afflicts*>, so the user could not easily assemble a list of all cow diseases and search for recent occurrences;
  - *mad cow disease* and one of its symptoms *anorexia* would also be related using RT rather than the more semantically expressive relation <*hasSymptom*>.
- The concept relations provided by most thesauri force all relations into the two broad categories, hierarchical and associative. Too often the semantic relationships captured in this way are ambiguous and poorly defined. The generalization/specialization relations defined in most thesauri are not adequately developed to be of use for semantic description and discovery of Web resources. Thus there is a need for a richer and more powerful set of relationships.
- **Lack of consistency**: since the relationships in thesauri lack precise semantics, they are applied inconsistently, both creating ambiguity in the interpretation of the relationships and resulting in an overall internal semantic structure that is irregular and unpredictable. Many of the NT/BT hierarchical relationships could, for example, be resolved to the non-hierarchical RT relationship, and vice versa.
- **Limited automated processing**: traditionally thesauri were designed for indexing and query formulation by people and not for automated processing. The ambiguous semantics that characterizes many thesauri makes them unsuitable for automated processing.

To overcome these limitations and enable more powerful searching and intelligent information processing, especially as such capabilities can be made more widely available through the Web, traditional KOSs must be reengineered into KOSs that contain domain concepts linked through a rich network of well-defined relationships and a rich set of terms identifying these concepts. A concept can be represented by many different terms (words or phrases) in multiple languages. This paper refers to terms as **lexicalizations** of a concept. One term can identify several concepts (homonymy) and one concept can have multiple synonymous terms. A concept is conveyed by all its lexicalizations, the domain it occurs in, and by its **relationships** to other concepts. In addition, valid rules and constraints need to be specified to provide additional generalizations over sets of related concepts and to support inference. These systems must also be converted to machine-processable formats based on Web technologies like XML which tag the vocabularies in a standardized way.

In contrast to traditional KOSs, ontologies provide conceptual abstraction and differentiated relationships. Ontologies specifically separate concepts from lexicalizations and thereby better reflect the structure of human understanding of a domain. In ontologies, the semantics are developed through ensuring that each concept within the domain is uniquely and precisely defined and by specifying elaborated relationships among the concepts. The relationships in an ontology are explicitly named and developed with specification of rules and constraints so that they reflect the context of the domain for which the knowledge is modeled.

Given their more precise and unambiguous semantics, ontologies allow further knowledge to be inferred from the knowledge explicitly represented in the ontology. The new (implicit) knowledge could be derived by applying generalization or transitivity rules, the level of applicability of which is limited in a poorly defined KOS like a traditional thesaurus. This added knowledge in the ontology makes it powerful when employed for intelligent information processing. Although there is a huge cost involved in moving from thesauri to ontologies, there is an expectation that the added power of consistency, precision, and completeness will be worth the investment even though reliable numbers on the return on investment (ROI) of ontology development are hard to come by.

1.3. Potential benefits of future generation KOSs

For emerging KOSs to satisfy user needs, they must improve both information organization and retrieval in a way that was not possible with traditional KOSs. The following potential benefits are expected from such systems:

- **Unique identifiers and formal semantics:** the explicit definition of concepts and relations in an ontology allows a unique identifier to be assigned to each concept. As each concept and relation is explicitly defined as a unique entity, the ontology lends itself to semantic formalization.
- **Internal consistency:** another benefit of explicit semantics is the achievement of internal structural consistency in the expression of knowledge due to the possibility of applying integrity constraints.
- **Interoperability:** clear semantics enables interoperability among different KOSs since corresponding concepts within different KOSs would have the same unique identifier, irrespective of the actual lexicalizations used to express those concepts. Semantic interoperability promotes sharing and reuse of knowledge.
- **Greater information integration:** interoperability among different KOSs makes it possible for machines to recognize and analyze intended meaning of terms from disparate vocabularies. This is possible by using structured meta-information and formal knowledge description such as agreed-upon metadata schemas, controlled domain vocabularies, and taxonomies. The ability to integrate terminologies from different sources maximizes the value of investment made in the ontology.
- **Inferencing capability:** new KOSs have the potential for expressing knowledge beyond what is present in the structure of the system. Unlike traditional KOSs where both concepts and relations are underspecified and very few, if any, axiomatic rules exist, the facts (concepts and relations) and rules that can be derived from an ontology have the expressive capabilities that allow for reasoning.
- **Automated information processing:** new KOSs create improved potential to discover relevant information from different sources by exploring patterns and filtering information using conceptual connections represented in the ontology. This enables question-answering from one or more databases or, using natural language processing (see next bullet), from text.
- **Natural language processing (NLP) support:** offers the possibility of providing a direct reply to a search question that is expressed in natural language, using the enhanced relationships and semantics in an ontology, instead of only returning a list of relevant documents.
- **Search query understanding:** using NLP and semantic processing, a system can understand a query posed in natural language, determine the concepts involved and, where useful, create a Boolean query.
- **Concept-based search:** an ontology can provide context-aware search capabilities specific to the area of interest.
- **Integrated information search/browse support:** text mining on the Web (Web mining) through meaning-oriented access, dynamic organization of information with the possibility for cross-domain links are feasible with emerging KOSs.

- **Search query expansion:** the enhancement, extension, and disambiguation of user query terms become possible with the addition of enriched domain- and context-specific information.

To be an effective tool to facilitate information categorization, integration and retrieval, ontologies should be multilingual, domain-specific, and cross disciplinary at the same time. For maximum application potential they should be developed in a non-proprietary, application-independent, and machine-processable format to ensure interoperability among different systems.

1.4. The process of reengineering: the rules-as-you-go approach

Reengineering a thesaurus into an ontology entails refining thesaurus relationships, a laborious process. The steps in the process are:

1. Define the ontology structure
2. Fill in values from one or more legacy KOS to the extent possible
3. Edit manually using an ontology editor:
    1. make existing information more precise
    2. add new information

Step 1 is addressed in section 3, which gives an overall conceptual model at a high level of abstraction, and section 4, which begins the process of defining a set of relationship types for the food and agriculture domain by examining relationships in AGROVOC as to their relationship types.

Step 3 is the most laborious. We have plans to streamline this process by implementing intelligent conversion using a "**rules as you go**" approach. The idea is as follows: The KOS editor watches out for patterns; based on these patterns the editor formulates rules that can be applied immediately to all subsequent similar cases as illustrated in the following:

1. An editor has determined that
   cow NT *cow milk* should become *cow* <*hasComponent*> *cow milk*
2. She recognizes that this is an example of the general pattern
   animal <hasComponent> *milk* (or, even more general *animal* <hasComponent> *body part*)
3. Given this pattern, the system can derive automatically
   goat NT *goat milk* should become *goat* <hasComponent> *goat milk*
   since goat is an animal and goat's milk ends with the word milk and thus can be seen to be a type of milk.

To automate this approach even more, we plan to build an inventory of patterns such as *animal* <*hasComponent*> *body part*, augmented by an ontology that specifies the concepts of type *animal* (*cow, goat, sheep, horse, chicken*, etc.) and the concepts of type *body part* (*skeletal meat part, liver, bone, milk, egg*, etc.). This information would be drawn from AGROVOC itself and other sources, such as Langual, UMLS, and even WordNet. The system can then detect the applicability of these patterns, at least once it saw one example transformed by an ontology editor. The ontology editors will add to the pattern inventory incrementally.

These patterns are a special type of constraint. Other constraints can be formulated and used to limit the options presented to the human editor as thesaurus relationships are refined. The bases for such constraints are the thesaurus relationships, on the one hand, and the entity types of the concepts involved, on the other. Table 3 shows some examples of constraints based on thesaurus relationships.

Table 3: Some relationship constraints

Thesaurus Relationships	Possible ontology relationships
NT/BT	<*hasMember*> / <*memberOf*> <*includesSpecific*> / <*isa*> <*hasComponent*> / <*componentOf*> <*spatiallyIncludes*> / <*spatiallyIncludedIn*> etc.
RT	<*similarTo*> <*growsIn*> / <*EnvironmentForGrowing*> <*treatmentFor*> / <*treatedWith*>? <*hasMember*> / <*memberOf*> etc. Note that the RT relationship often transforms into relationships that are not symmetric. Note further that in a well constructed thesaurus, an RT should not resolve into an <*isa*> relationship. However, reality shows that the RT relationship has been applied to express this relationship. This can be taken as another proof for the weak definition of relationships in many thesauri.

This inventory will constrain the available choices when manually refining a thesaurus relationship to a more specific ontology relationship. Of course, an authorized ontology editor can override such constraints and thereby update the relationship table. As a relationship has been added or refined the inverse relationship is automatically added or refined.

2. AGROVOC: a multilingual agricultural thesaurus

This section describes the AGROVOC Thesaurus and further illustrates the problem of semantic under-specification.

2.1. Background

AGROVOC is a multilingual, structured, controlled vocabulary/thesaurus designed to cover concepts and terminology in agriculture, forestry, fisheries, food and related domains (e.g. environment). It was developed by the Food and Agriculture Organization (FAO) of the United Nations and the Commission of the European Communities in the early 1980s to describe documents and other information resources in a controlled language for indexing and searching. It contains approximately 16,500 descriptors and 10,000 non-descriptors.

AGROVOC is available online and for download in the five FAO official languages (English, French, Spanish, Chinese and Arabic). It is translated into other national languages such as Czech, Danish, German, Italian, Polish, Portuguese, Slovak and Thai.

2.2. Applications and related terminologies

AGROVOC is used for controlled-vocabulary indexing and searching globally and in various systems throughout the FAO. Systems where AGROVOC is used include:

- the AGRIS/CARIS network, an international information system for indexing and retrieval since 1986, coordinated by FAO;
- domain-specific documentation centers around the world;
- indexers, librarians and translators working in the global food and agriculture sector.

Within FAO:

- Electronic Information Management Services (EIMS)
- WAICENT information finder
- The FAO library catalogue online

AGROVOC coexists as one knowledge organization system next to numerous others in the agricultural domain. Among the most important ones at the FAO are:

- FAO Terminology
- ASFA (Aquatic Sciences and Fisheries Abstracts) Thesaurus
- Fisheries Glossary
- One Fish Glossary

There are a number of other thesauri in the food and agricultural sector, developed by other institutions, such as

- US National Agricultural Library (NAL) Thesaurus,
- Commonwealth Agricultural Bureau Incorporate (CABI) thesaurus,
- Langual thesaurus providing an easily accessible hierarchy with 14 facets.

These thesauri basically all follow the same conceptual structure, which will be discussed in the following section. Nevertheless, we will see that, although all these thesauri use basically the same conceptual model, the information contained in them can differ substantially.

2.3. Conceptual structure of AGROVOC

AGROVOC follows a traditional thesaurus approach. It is a collection of terms, definitions, and term relationships. As is the case with most thesauri, a small, standard, non-adaptable set of relationship types is applied to interlink terms.

2.3.1 Equivalence relationships

USE: Since thesauri have been primarily developed for the purpose of indexing and retrieval, this relationship indicates that any term preceding the USE relation should be replaced, for the purposes of indexing documents and formulating queries, by the term following the USE relation. The relationship usually (but not always) expresses synonymy between two terms.

USED FOR (UF): This is the inverse of USE and indicates that term A is USED FOR term B for indexing purposes.

2.3.2 Hierarchical relationships

Narrower Term (NT): if X is a NT of Y, then X is narrower in some sense than Y. For example, *milk* NT *cow milk, grain* NT *rice*.

Broader Term (BT): if Y is a BT of X, then X is broader than Y; for example *cow milk* BT *milk*, *rice* BT *grain*. BT is the inverse of NT.

Given these rather unspecific definitions, BT and NT relationships can be applied to express generic relations, meronymic relations, instantiations, and many others (see section 4).

2.3.3 Associative relationships

Related Term (RT): the thesaurus conceptual model contains the RT relationship to express any kind of associative relationship between two terms that is not a hierarchical relationship. This relationship is hence very ambiguous in that it is the default for all other relationships.

Hierarchical (NT, BT) and associative (RT) relationships are relationships between concepts. In the thesaurus, these exist only between descriptors. Following a traditional thesaurus approach, AGROVOC distinguishes between **descriptors** and **non-descriptors** (often referred to imprecisely as preferred terms and non-preferred terms). The rationale behind this is that only a descriptor should be used when referring to the concept (for example, for indexing and retrieval); each descriptor uniquely and unambiguously designates a concept. A non-descriptor must not be used for indexing or retrieval; it is linked through a USE cross-reference to the corresponding descriptor that must be used instead. There are no relationships from one non-descriptor to another.

2.3.4 Scope notes

Many descriptors in AGROVOC have a scope note, which can be a definition of a term, a history note, instructions to the indexer or searcher, or simply a comment. The purpose is to provide the user with more detail about the term and its usage.

2.3.5 Top level structure

Currently AGROVOC has more than 1500 top-level terms, i.e. descriptors which do not have a broader term, making it cumbersome to access the thesaurus from a top-level approach and browse through the hierarchy. Superimposed on AGROVOC is the AGRIS categorization scheme; it has more than 100 top-level categories, ordered in a shallow two-level hierarchy. AGROVOC descriptors are mapped to the second level of AGRIS categories. For example, the AGROVOC descriptor *fish farms* is mapped to the AGRIS category *aquaculture production* which is a subcategory of *fishery and aquaculture*. Thus the AGRIS categorization scheme provides high-level organization for information that has been tagged with AGROVOC descriptors.

2.4. Semantic problems of AGROVOC

Given its minimalist conceptual structure, AGROVOC (as other thesauri) has a number of **semantic flaws**. In the following we will use examples to point out the major drawbacks of the current system and develop the rationale for the shift towards a more powerful, expressive, and unambiguous conceptual model.

2.4.1 Ambiguous descriptor to non-descriptor relationship

In AGROVOC, as indicated, USE/UF covers synonyms and formal variants. In addition, the relation also links quasi-synonyms and very specific narrower terms, which the AGROVOC defines as any of the following:

1. "two concepts considered sufficiently alike to be identified by one descriptor"
2. "a concept and its opposite"
3. "more specific concepts encompassed by one descriptor"

Definition 1 deals with semantically very **closely related**, yet separate concepts (so that the terms designating such concepts would not be true synonyms), such as

famine
UF hunger

Definition 2 involves concepts on **opposite** ends of a scale or otherwise in opposition to each other. (With a few exceptions, the terms designating such concepts are antonyms). Example:

hydrophilicity
UF hydrophobicity

Definition 3 indicates that USE/UF can also express a **hierarchical relationship**, for example:

biological competition
UF interspecific competition
UF intraspecific competition

where the fine distinction between *interspecific competition* and *intraspecific competition* is deemed unnecessary for retrieval and therefore abandoned in favor of the more general category.

2.4.2 Ambiguous hierarchical definitions

The BT/NT relationship used to build up the hierarchy is very ambiguous; it lumps together several different types of relationships as the following examples show:

2.4.2.1 <includesSpecific> relationship (*erythrocytes* are a specific kind of *blood cell*):

blood cells
NT erythrocytes
NT leukocytes

2.4.2.2 <hasComponent> relationship (*blood* contains as a component *blood cells*):

blood
NT blood cells

2.4.2.3 The following example shows clearly the discrepancies between different thesauri that apply the ambiguously defined modeling principles:

AGROVOC and **CABI**:
water
NT ice
NT water vapor
...
NT fresh water
NT drinking water

But **ASFA:**
water
RT ice
RT water vapor
...
NT fresh water

Water vapor and *ice* are *phases of water* while *fresh water* and *drinking water* are *kinds of water,* so in AGROVOC and CABI hierarchical relationships lump together several different semantic relationships. For retrieval this is generally useful (a search for *water* should generally find documents on *ice* as well), but for more differentiated retrieval a user may want to ask for *water in all phases* or for *all kinds of water.* There are many other purposes of semantic processing that need more differentiated relationships. In ASFA the *phase* relationship is treated as a RT, an example of how grouping relationships may lead to inconsistency. Note, by the way, that neither thesaurus includes the concept *liquid water*, which is logically necessary if *water* means *water in any phase.*

There are many more examples in AGROVOC where the currently used BT/NT relationship is used to describe different relationships. The most obvious ones have been identified and are used in our proposal below in section 4.

2.4.3 Ambiguous associative relationships

Like the BT/NT relationships, the associative RT relationships can be refined into more specific relationships. Some examples are given below.

2.4.3.1 ***<hasMember>*** **relationship** (*Anglophone Africa* <hasMember> *Botswana*)

Anglophone Africa

RT Botswana
RT Gambia
RT Ghana
RT Kenya
RT Lesotho
...

2.4.3.2 ***<causes>*** **relationship** (*bleaching* <*causes*> *discoloration*):

bleaching

RT discoloration

2.5. The need for reengineering AGROVOC into an ontology

The examples above indicate clearly the ambiguous nature of the relationships in AGROVOC. With respect to future information retrieval and intelligent processing needs, where it will be necessary to combine different KOS in order to give access to different information systems, it becomes evident that a more rigid structure is required. A reassessment of AGROVOC (as well as other thesauri) to transform its UF, NT, BT, and RT relationships into unambiguously defined relationships and hierarchical order will provide the first step towards solving the problem of ambiguity and inconsistency in information description and retrieval.

3. Conceptual model: combining thesauri and ontologies

This section introduces a conceptual model that provides the necessary structure to create precise semantics to facilitate the transition from traditional thesauri to ontologies. Figure 1 shows the high level conceptual model we propose. Its chief characteristic is a clear separation of the concept level, the term or lexicalization level, and the string level. Present thesauri give a more or less muddled representation of information about concepts and information about terms. The proposed structure allows for a clear separation of concept information and term information. This model owes much to the structure of the UMLS.

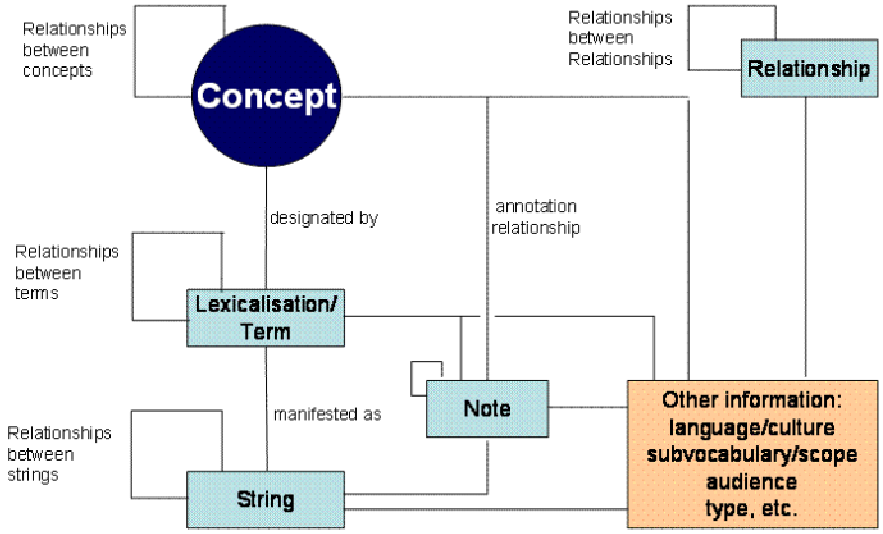
Relationships between concepts
Concept
Relationships between Relationships
Relationship
designated by
annotation relationship
Relationships between terms
Lexicalisation/ Term
Note
Other information: language/culture subvocabulary/scope audience type, etc.
manifested as
Relationships between strings
String

Figure 1. Conceptual model for combining thesauri and ontologies

3.1. The basic model

The following is just the broad outline of the model. Many more types of information could be added. In any event, we consider the model extensible. On the other hand, not all applications will use all features of the model. For example, our model provides for relationships between notes (for example, as hypertext links). This is not possible in all environments but very useful in some. Our intent is to present a framework that can be used for the simplest thesaurus or the most complex and rich ontology in a format that communicates equally to thesaurus and ontology editors with a background in information science, artificial intelligence, or linguistics.

- A **concept** encapsulates meaning.
- A concept can be represented or designated by one or more linguistic expressions, namely **terms** or lexicalizations which can be single words or multi-word phrases (or composite words in agglutinative languages).
- A term, in turn, can take variant forms (singular/plural, variations in case, spelling variants, abbreviations, acronyms); so just as a concept can have many lexical representations, a term can have many **string** manifestations.

Each concept, term, and string can be assigned an identifier, preferably a Unique Resource Identifier (URI); for concepts, UMLS uses Concept Unique Identifiers (CUI), while the Topic Map Standard uses unique subject identifiers. Using unique concept identifiers allows for unambiguous reference to concepts, as opposed to often ambiguous terms. Concepts can furthermore be assigned notations (such as class numbers in the Dewey Decimal Classification; notations are also called term numbers); notations can be used to maintain a logical, meaningful sequence in hierarchical displays.

Concepts take center stage in our proposed thesaurus/ontology information model; accordingly, relationships between concepts are central. Concepts are arranged in **hierarchies** and have **additional relationships** to other concepts in the network; a hierarchy can be defined on any weak ordering relationship including *isa*, *part-whole*, *spatial containment*, etc. (the relationship must be transitive and not symmetric, but must have an existing inverse relationship, for example <*componenttOf*> is the inverse relationship of <*hasComponent*>). There are many other relationship types, such as <*causes*> ; a scheme of relationship types needs to be defined for the domain of the respective thesaurus. One source for finding relationship types is the detailed analysis of concept relationships present in the thesaurus that is to be reengineered into a richer ontology (see section 4). Each concept should be assigned to an entity type or facet, such as *process*, *function*, *substance*, *living organism* (see, for example, the semantic types in the UMLS Semantic Network); the type of a concept constrains its participation in relationships.

A concept is designated or represented by one or more **lexicalizations** or **terms** in one or more languages; this is the linkage between the concept level and the term level. For examples see Table 4.

Table 4. Concepts, terms, strings (concept and term numbers are fictitious and used only for illustration)

Concept ID	Term ID	Strings manifesting the term
AGROVOC:C316301	AGROVOC: T657210	bovine spongiform encephalopathy, BSE
"	AGROVOC: T657211	mad cow disease, Mad Cow Disease, MCD
"	AGROVOC: T734567	encephalopathy spongiforme bovine, ESB
"	AGROVOC: T734566	maladie de la vache folle, MVF
"	AGROVOC: T700345	encefalopatia espongiforma bovina, EEB
"	AGROVOC: T700346	enfermedad de la vaca loca, EVL
AGROVOC:C014593	AGROVOC: T187953	plow, plows, plough, ploughs, plogh [a frequent misspelling]
"	AGROVOC: T498001	charrue
"	AGROVOC: T498002	materiel de labour

If a term is a homonym (designates more than one concept), several disambiguated terms are introduced. The homonym is linked to each of the disambiguated terms, and each disambiguated term is linked to the corresponding concept. Two terms designating the same concept are called synonyms. Conversely, if one does not agree that concepts per se exist, one can simply view "concept" as a convenient shorthand for an equivalence class of terms that are linked by the <*hasSynonym*> relationship, such as the synsets in Wordnet. A KOS may select a preferred term as the term used to represent the concept or it may make that choice dependent on the audience (for example, veterinarians versus farmers).

Terms can be connected through many **relationships** such as <*hasSynonym*> (with <*hasScientificName*> as a special case), <*hasAntonym*>, <*hasCognate*> (term in a different language from the same root), and <*hasTranslation*>. One might think that the *synonym* and *translation* relationships are not needed since all terms linked to the same concept would be synonyms or translations. However, two terms may be linked to the same concept yet be used in different contexts, i.e. they are not strict synonyms. If a concept has linked to it several English terms and several French terms, it is not true that just any of the French terms is a good translation for a given English term (see the examples in Table 4). Another example of term-specific relationships is <*hasAntonym*>. For example, *big* and *small* designate opposite concepts but are not antonyms. (The antonym pairs are *big* versus *little* and *large* versus *small*; see Wordnet.)

Finally, a term is manifested in one or more **strings**,as shown in Table 4. Strings can be connected through relationships such as <*hasCaseVariant*>, <*hasSpellingVariant*>, <*hasAbbreviationOrAcronym*>, <*pluralOf*> / <*singularOf*>, which are all subordinate of a broader relationship <*hasStringVariant*>. A term can be seen as a convenient shorthand for an equivalence class of strings that are linked by the <*hasStringVariant*> relationship. A KOS may select a preferred variant as the string used to represent the term or it may make that choice dependent on the audience (as in British versus US spellings). A string, especially an acronym, may belong to several terms, in which case it needs to be disambiguated.

In addition, a concept, a lexicalization/term, a string, or a relationship type can have several types of **notes** (definitions, usage notes, comments, image, etc.) in different languages (in the case of multilingual thesauri). Just like concepts and terms, notes can be related to each other through relationships such as <*hasTranslation*>, <*hasSimplifiedVersion*>, <*hasOtherDefinition*>, or any other type of hyperlink. Many other pieces of information about terms can be added, for example, case frames for verbs (in case the verb has a case frame different from the case frame for the corresponding action concept) or register (see below) or whether the term is the preferred term for the concept. Administrative data will be accommodated as well.

Relationship types themselves can form **relationship hierarchy** (i.e. a relationship of relationships), in which more generic relationships are further up in the hierarchy than more specific relationships, for example, <c*omponentOf*> is a specific kind of <*partOf*> relationship.

Why define concepts, terms, and strings as separate entity types?

First, each of these entity types takes different types of information. Conceptual relationships and other information are associated with concepts. Linguistic information, such as part of speech and how a term combines with other terms into sentences, usage, or information on etymology, are associated with terms. Information such as that a string is an acronym is associated with terms. Usage information may sometimes be associated with strings; for example, lay people may commonly use a slang abbreviation while professionals use the full string. Definitions are primarily associated with concepts but may also be associated with terms.

Second, this distinction avoids confusion. In a standard thesaurus like AGROVOC, for each concept that is to be used in indexing and searching, a preferred term, and for that term a preferred string, is selected; this string is the descriptor. Non-descriptors are linked only to descriptors, not among themselves. As a result, *BSE*, *mad cow disease*, and *MCD* [which we made for illustration] are all linked to *bovine spongiform encephalitis* as synonyms (or, in some thesauri, as synonym and as abbreviations). But the information that *BSE* belongs with *bovine spongiform encephalitis* and *MCD* with *mad cow disease* is lost. Furthermore, if decisions on terms are made (for example, omitting *mad cow disease* as a non-scientific name), these decisions should apply to all term variants, in the example *MCD* as well.

3.2. Model extensions

As was mentioned above, many more types of information could be added to concepts, terms, strings, notes, and relationships. For example, we might specify an audience (general lay public, K-12 students by grade level, university students, experts), a subject domain, a scope (as in Topic Maps), or a specially selected subset of concepts and terms to be used for a given application, or all concepts and terms taken from a given source.

Scopes could be defined in many ways. For example, one might define a scope as the conceptual system embedded and expressed in a language (whereas the link from terms and notes to language simply refers to the surface form). Consider the conceptual system underlying Walpiri (an Australian indigenous language); one of its noun classifiers includes *women*, *fire*, and *dangerous things* (Lakoff 1987). A native speaker of English would find this classifier and the corresponding <*isa*> relationships very curious. Thus one would introduce the category and the <*isa*> relationships with a scope of the Walpiri conceptual system. (By the way, the relationship between these relationships makes sense in the context: fire is dangerous; fire is sometimes started by or anyway related to the sun; the gender of the noun for sun is female). Many such problems, if more subtle, occur in thesauri for international use.

A **subvocabulary** can be extracted using any type information about concepts, terms, strings, and relationships that is available in the thesaurus. Thus one could extract as subvocabulary

- a subset that was selected for a given application;
- all strings that are acronyms (for an acronym reference);
- all scientific names;
- all entries for taxonomic entities for the entire range of living things or for a given large taxon such as insects;
- terms suitable for a given audience.

Each of these subvocabularies provides a specific view on the entire KOS for a given purpose. In online implementations such subvocabularies can be created on the fly or defined as views for certain user groups. But subvocabularies can also be printed or exported (for example, a subvocabulary extracted as the personal KOS of a researcher who maintains an information retrieval system on his or her own computer).

3.3. Limitations

The separation into the concept layer and the term layer is appealing for its simplicity and elegance but it is somewhat of an oversimplification. Terms, particularly terms in different languages, rarely mean exactly the same thing. So the question arises as to when to map two terms to the same concept - and possibly explain shades of meaning and associations in the definition of each term that complements the definition of the concept - and when to create two closely related concepts, possibly under one broader concept. Our model permits any type of relationship between terms. Thus it is possible to introduce conceptually motivated relationships between terms that more accurately reflect the reality of language than the mapping of terms to "concepts". These two representations of conceptual information can coexist within the same system.

3.4. Implementation

All relationships from all layers (concept, term, string) can be stored in the same format within a database. The type of each element should be explicitly given to enable integrity constraints (so that the relationship <*hasSpellingVariant*> is not allowed between two concepts, for example). A concept can be identified by a URI or other number (cleanest solution) or by its preferred term in the base language of the thesaurus (the term being typed as preferred). Likewise, a term can be identified by a URI or other number (cleanest solution) or by the preferred string (the term being typed as preferred). The same holds for strings. The main difference with implementations in most existing thesaurus management software is that relationships between non-descriptors are allowed. Thoughts for an XML/RDF schema for KOS data are presented in the Appendix.

3.5. Related approaches

The proposed conceptual model integrates well with standardization approaches regarding Web technologies like RDFS. The proposed structure shows all aspects of a proposed RDFS-compatible Thesaurus Interchange Format by Matthews *et al*. (2002), which will appear as a W3C note. The proposal is being done in the context of the SWAD-Europe project. The Appendix presents another approach for representing ontology and thesaurus data in XML/RDFS.

4. The AGROVOC case: exploring conceptual relationships in the agricultural domain

The model we introduced has no restrictions on potential relationships to be applied. The model is extensible, and any possible specific relationships can be included. We carried out a preliminary linguistic and conceptual analysis of AGROVOC and found a set of relationships; most of them are well known (but it is important to know that they are needed in the food and agriculture domain), some of them add new nuances. Table 5 lists relationship types found in AGROVOC or otherwise proposed here, and subsections 4.1 - 4.3 give an explanation and examples for some of these relationship types; others appear in examples throughout the paper. This section is not in any way intended as a complete list of relationship types; it merely gives examples to illustrate the additional information and clarity of conceptual structure that can be conveyed through more specific relationships. Much more work, including comparison, is needed to converge on a set of relationships to replace the currently used thesaurus relationships BT, NT, RT, USE and UF in a reengineering of AGROVOC.

Table 5: Concept relationships: Examples

X, Y are concepts
Isa
X <includesSpecific> Y / Y <*isa*> X
X <*inheritsTo*> Y / Y <*inheritsFrom*> X
Holonymy / meronymy (the generic whole-part relationship)
containsSubstance> Y / Y <*substanceContainedIn*> X
X <*hasIngredient*> Y / Y <*ingredientOf*> X
X <*madeFrom*> *Y / Y* <*usedToMake*> *X*
X <*yieldsPortion*> Y / Y <*portionOf*> X
X <*spatiallyIncludes*> Y / Y <*spatiallyIncludedIn*> X
X <*hasComponent*> Y / Y <*componentOf*> X
X <*includesSubprocess*> Y / Y <*subprocessOf*> X
X <*hasMember*> Y / Y <*memberOf*> X

Further relationship examples (some from (Schmitz-Esser 1999)
X <*causes*> Y / Y <*causedBy*> X
X <*instrumentFor*> Y / Y <*performedByInstrument*> X
X <*processFor*> Y / Y <*usesProcess*> X
X <*beneficialFor*> Y / Y <*benefitsFrom*> X
X <*treatmentFor*> Y / Y <*treatedWith*> X
X <*harmfulFor*> Y / Y <*harmedBy*> X
X <*hasPest*> Y / Y <*afflicts*> X
X <*growsIn*> Y / Y <*growthEnvironmentFor*> X
X <*hasProperty*> Y / Y <*propertyOf*> X
X <*hasSymptom*> Y / Y <*indicates*> X
X <*similarTo*> Y / Y <*similarTo*> X
X <*oppositeTo*> Y / Y <*oppositeTo*> X
X <*hasPhase*> Y / Y <*phase*Of> X
X <*growsIn*> Y / Y <*EnvironmentForGrowing*> X
X <*ingests*> Y / Y <*ingestedBy*> Y

4.1. The logical generic relationship

4.1.1 **X <includesSpecific> Y / Y <*isa*> X** (implies X <*inheritsTo*> Y / Y <*inheritsFrom*> X)

This is the standard generic relationship between concepts. It can be used for hierarchical inheritance (but is not the only relationship for this purpose). Hierarchical inheritance is useful within the thesaurus/ontology to streamline the writing and presentation of definitions and scope notes and for inheriting relationships. Examples (all NT in AGROVOC unless stated otherwise):

chemical soil types <includesSpecific> saline soils
bovine spongiform encephalopathy <includesSpecific> spongiform encephalopathy
cells <includesSpecific> blood cells
blood cells <includesSpecific> leukocytes
leukocytes <includesSpecific> lymphocytes
lymphocytes <includesSpecific> T-lymphocytes UF in AGROVOC

4.2. The part-whole family of relationships

There are several relationships that fall under the part-whole umbrella. Some of these can be displayed in a hierarchical format. However, the direction of the hierarchy, and the direction of hierarchical inheritance, is sometimes from part to whole and sometimes from whole to part. All relationships are shown starting from the whole first.

4.2.1 X <containsSubstance> Y / Y <substanceContainedIn> X and X <hasIngredient> Y / Y <ingredientOf> X

Y is the material or substance of which X is made by nature (<*containsSubstance*>) or by man (<*hasIngredient*>). Y loses its identity once it is incorporated into X. There is no implication of an "all and only" relation with respect to the composing substance, where Y is the sole substance making up X; but a subsequent refinement could make this distinction. In AGROVOC, this relationship appears as BT/NT or RT; quite often, it does not appear at all. Some examples are given below:

blood <*containsSubstance*> blood gases
blood <*containsSubstance*> blood lipids
blood <*containsSubstance*> blood proteins
blood <*containsSubstance*> blood cells (borderline case, see comment in 4.2.4)

All NT in AGROVOC

cocoa beverages <*hasIngredient*> cocoa powder

RT in AGROVOC

4.2.2 X <yieldsPortion> Y / Y <portionOf> X

X <*yieldsPortion*> Y describes a relation between a mass X and a piece Y taken from the mass, for example,

roots <*yieldsPortion*> cuttings RT in AGROVOC

(Note: the example assumes *root cuttings,* but AGROVOC refers to any kind of *cutting*.

4.2.3 X <spatiallyIncludes> Y / Y <spatiallyIncludedIn> X

This relation is used for objects with spatial extent. It holds when X is an inalienable part of Y, identifiable but not inherently separable. These include body parts and geographical locations.

Asia <*spatiallyIncludes*> East Asia NT in AGROVOC

Transitivity is also a feature of spatial relations. If X <*spatiallyIncludes*> Y and Y <*spatiallyIncludes*> Z, then X <*spatiallyIncludes*> Z.

Asia <*spatiallyIncludes*> East Asia
East Asia <*spatiallyIncludes*> China
→ Asia <*spatiallyIncludes*> China

4.2.4 Y <*hasComponent*> Y / Y <*componentOf*> X

This relationship holds when X is a part of Y that retains its identity as an object even when built into the whole. In addition, each X must be enumerable or nameable, i.e. not part of a mass. Examples:

plough **<***hasComponen*t**>** ploughshare   RT in AGROVOC
woody plant **<***hasComponent***>** plant anatomy (i.e. parts of plants)   RT in AGROVOC
nucleus **<***hasComponen*t**>** chromosome   NT in AGROVOC

There are cases when it is hard to decide when to use <*containsSubstance*> and when to use **<***hasComponen*t**>**; *blood* <?> *blood cell* is such a borderline case. We decided on *blood* **<***hasComponen*t**>** *blood cell* because each blood cell is a recognizable and separate entity. But one could equally strongly argue for *blood* <*containsSubstance*> *blood cell* because a blood cell is part of a mass and cannot be distinguished from others within the mass; it has at most a very weak identity as an object.

4.2.5 X <includesSubprocess> Y / Y <subprocessOf> X

There are many processes in AGROVOC which could be linked using this relation, for example,
ATP cycle <*includesSubprocess*> phosporylation   RT in AGROVOC

4.2.6 X <*hasMember*> Y / Y <*memberO*f> X

X <*hasMember*> Y indicates a relation of membership within a collective or group or organization.

Francophone Africa <*hasMember*> Benin   RT in AGROVOC
biotope <*hasMember*> plant   not in AGROVOC
pesticide crops <*hasMember*> Artemisia absynthium   RT in AGROVOC
Note that transitivity is not a characteristic of membership relations.

4.3. Further relationship examples (some from Schmitz-Esser 1999)

4.3.1 X <*causes*> Y / Y <*causedBy*> X

Examples
overgrazing <*causes*> desertification   RT in AGROVOC
Serpulina hyodysenteriae <*causes*> swine dysentery
preharvest sprouting <*causes*> crop losses

4.3.2 X <instrumentFor> Y / Y <performedByInstrument> X

This relation expresses the fact that concept X is instrumental to achieve, as a result, concept Y. Example:

plough <*instrumentFor*> ploughing   RT in AGROVOC

The instrument considered may be one applied by a living being (man, animal) or a machine or a system. The sense of the relation points to the result achieved by the use of the instrument.

4.3.3 X <*processFor*> Y / Y <*usesProcess*> X

This is a case where X indicates a process involved in Y. Examples:

soil injection <*processFor*> fertilization   RT in AGROVOC
gonadectomy <*processFor*> sterilization [of organisms]   BT in AGROVOC

4.3.4 X <beneficialFor> Y / Y <benefitsFrom> X

fertilization ***<beneficialFor>*** crop yield   Not found in AGROVOC

4.3.5 X <treatmentFor> Y / Y <treatedWith> X

Pentosan polysulphate **<***treatmentFor***>** bovine spongiform encephalopathy   Not in AGROVOC

4.3.6 X <*harmfulFor*> Y / Y <*harmedBy*> X

Found in AGROVOC only indirectly. For example, from:

preharvest sprouting <*causes*> crop losses

one can conclude

preharvest sprouting <*harmfulFor*> crop yield

4.3.7 X <growsIn> Y / Y <growthEnvironmentFor> X

Halophytes <*growsIn*> saline soils   RT in AGROVOC

4.3.8 X <*hasProperty*> Y / Y <*propertyOf*> X

fertilization <*hasProperty*> application rate   RT in AGROVOC
blood circulation <*hasProperty*> blood pressure   NT in AGROVOC

4.3.9 X <*similarTo*> Y / Y <*similarTo*> X

bovine spongiform encephalopathy <*similarTo*> Creutzfeld-Jakob syndrome   RT in AGROVOC

4.3.10 X <*oppositeTo*> Y / Y <*oppositeTo*> X

crop losses <*oppositeTo*> crop yield   Not in AGROVOC

4.3.11 Concluding comment

This preliminary analysis shows that AGROVOC contains many relationships that can be made more specific in a reengineering project but that some relationship types are missing so relationships need to be added from scratch. Also it is hardly possible to compile a complete inventory of relationship types. Thus, a generic relationship *Related Term* should still be kept to express residual relationships; such relationship instances should be accompanied by a note that specifies the meaning or intention of the relationship. This will facilitate a later deduction of new relationship types from these residual relationships.

5. Exploring the rules-as-you-go approach for the case of AGROVOC

We explored the applicability of the rules-as-you-go approach to transforming AGROVOC into an ontology. We looked for examples of patterns that could be used. The results are shown in Table 6.

Table 6: Examples for the rules-as-you-go approach

Pattern: plant <*growsIn*> soil type

Rice RT moist soil → rice <*growsIn*> moist soil

Pattern: geographical entity <*spatiallyIncludedIn*> geographical entity

Benin BT West Africa → Benin <*spatiallyIncludedIn*> West Africa

Pattern: geographical entity <*isa*> geographical entity

Benin RT Francophone country → Benin <*isa*> Francophone country

Pattern: body part <*containsSubstance*> (substance | small particle)

blood NT blood gas → blood < *containsSubstance* > blood gas

blood NT blood cell → blood < *containsSubstance* > blood cell

From this exploration it appears that the approach is promising. On the other hand, the rules-as-you-go approach is not error-free; results must be checked by an ontology editor. In some cases, it may not be possible to define a rule.

Another difficulty is illustrated by the concept of *Francophone Africa*. Does this term refer to the set of Francophone African countries, in which case it refers to a *type of geographical entity*, or does it refer to the area (not necessarily contiguous) covered by these countries, in which case it refers to a *geographical entity* (just as *West Africa*).

UF (Used For) may refer to a synonymous or quasi-synonymous term or to a narrower concept (generally *includesSpecific*). Since the terms on both sides of the UF most often refer to concepts that have the same entity type, it is difficult to formulate a rule. One might use an external knowledge source, such as a large dictionary, to detect synonymity and treat the other cases as *includesSpecific*, always subject to editor verification.

The rules-as-you-go approach implies that the reengineering effort should start with the top-most concepts so that entity types and patterns can be detected early.

6. Implications and further work

On the basis of the ideas presented in this paper, we plan to undertake the reengineering of AGROVOC into an ontology system that will also serve the functions of a traditional thesaurus. This involves creating a complete inventory of domain-relevant entity types and relationship types, which we will base on further analysis of AGROVOC and related vocabularies and on existing inventories, such as the UMLS Semantic Network. We tend towards using a frame representation of relationships, as presented in Slaughter and Soergel (2003).

Based on the encouraging results of our exploration of the applicability of the rules-as-you-go approach to AGROVOC, we plan to develop a system that streamlines the reengineering process. We expect that this system will drastically reduce the effort required and thus make the reengineering effort feasible.

The resulting knowledge base will enable

- improved user interaction with the vocabulary for improved query formulation and more user learning about the domain;
- intelligent behind-the-scenes support for query expansion;
- intelligent support for human indexers and automated indexing/categorization systems;

- support for artificial intelligence and semantic Web applications for agriculture, food processing, and food safety.
-

The approach will be incremental, starting with providing a proof of concept through pilot application that demonstrates that the expected benefits will in fact result. This will provide the basis for planning and efficient implementation of the large-scale effort to reengineer AGROVOC into a full-fledged ontology. We hope this effort can be carried out in a cooperative, distributed, and coordinated environment to promote active participation and ultimately use of a widely accepted Knowledge Organization System for the domain of food and agriculture.

Acknowledgements

We particularly thank all the participants of an expert workshop on the AGROVOC thesaurus and especially all the users and developers for their valuable feedback provided over the years. Their feedback contributed in great part to this proposal. We also thank the anonymous reviewers for their helpful comments.

References

Lakoff, George (1987) *Women, fire, and dangerous things* (University of Chicago Press)

Links to RDF and XML thesaurus formats and related topics http://www.w3c.rl.ac.uk/SWAD/thes_links.htm

Matthews, B.M., Miller, K. and Wilson, M.D. (2002) "A Thesaurus Interchange Format in RDF". Submitted to the *Semantic Web Conference 2002*

Schmitz-Esser, W. (1999) "Thesaurus and Beyond: An Advanced Formula for Linguistic Engineering and Information Retrieval". *Knowledge Organization*, Vol. 26, No. 1, 10-22

Schmitz-Esser, W. (1999) "Gedankenraumreisen - neue Thesaurusstrukturen, multimedial präsentiert, machen Anregung, Spielen, Lernen, Finden möglich für jedermann". In *Proc. DGI-Jahrestagung*, Hamburg (Frankfurt/M.: DGI), pp. 347-353

Slaughter, Laura and Soergel, Dagobert (2003) "How physicians' answers relate to health consumers' questions". *Proc. 66th ASIST Annual Meeting*, Long Beach, CA, Oct. 19-22 (Medford, NJ: Information Today), pp. 28-39

Appendix: Towards an XML/RDF specification for KOS

Proposals and schemes for encoding KOS data range from general schemes, such as RDF (with the OWL extensions) and the Topic Map standard to much simpler and specific schemes for encoding thesaurus data as specified in ISO 2788, which inherit all the limitations of this standard and are of equally limited usefulness for the new tasks ahead. Something in between is needed. After some preliminary remarks on the functions of such a standard, we make a proposal that is parsimonious but allows for encoding very rich data. For this reason, it may seem a bit opaque at first. The first author will be happy to entertain any questions or comments.

Functions to be served by standards for machine-readable KOS

1. Input of KOS data into programs / transfer of thesaurus data from one program into another

1.1 Format for original input files (but XML difficult for that, use a more user-friendly format, such as inputting a hierarchy with levels specified by the number of dots at the beginning of a line)
1.2 Transfer from one KOS development program to another
1.3 Transfer from a KOS development program to an information system that uses a KOS for authority control, query expansion (synonym and /or hierarchic), display/browse/search, or other purposes
1.4 Transfer from a KOS development program to a KOS display / browse / search program

2. Querying KOSs and viewing results (for example, using Z39.50)
2.1 By people
2.2 By systems to use data from external KOSs for query term expansion, etc.

3. Identifying specific terms/concepts in specific KOSs
This requires rules for URIs that uniquely identify specific term/concept records in specific KOSs. Probably requires some sort of name resolution service (such a KOS registry)
3.1 Links from one KOS to another
3.2 Indexing terms/concepts in the metadata for an object, or any other reference to a term/concept in a text/object

Elements of an XML KOS data specification

This schema is parsimonious yet allows the recording of many types of data. It gives enough information to derive a full XML specification.

This specification assumes that data from each source are grouped, so that source attribution is not needed for each element; otherwise the structure would be much more complex. This works for a communications format but not for an internal database format.

The term itself is indicated in a relationship of type TERM. This allows for terms in multiple languages for the same concept and simplifies the schema since elements in term would be the same as in relationship target.

Addition of the scope element was inspired by the Topic Map Standard.

Most schemes advanced for KOS data hard-code the permissible relationship types as tags. This makes it very hard to introduce new relationship types. The scheme proposed here is based on a more elegant principle: it simply provides a generic syntax for recording relationships and makes the relationship type a data element recorded in an appropriate tag. The scheme needs a method (not given here) for indicating a relationship set defined elsewhere and used within the source or for defining a relationship set for the source. A relationship must specify the relationship types and domain and range for each. RDFS could be used for this specification (RDF object classes are entities, RDF properties are relationship types). A system processing data organized in this scheme would process a relationship instance as follows: look up the relationship type in the relationship set and get the proper entity type for domain and range, respectively. Then check the entity type in the source (domain) slot and the target (range) slot to verify agreement with the restrictions. The scheme is neutral as to how concepts, terms, and relationship types are identified. The identifiers could be URIs, system-assigned numbers, character strings representing codes or character strings representing terms.

Default is minOccurs="1" maxOccurs="1"
Source (minOccurs="0" maxOccurs="unbounded")
Pointer to or definition of relationship set used
Unit: Concept or term or group of terms (minOccurs="0" maxOccurs="unbounded")
Unique identifier
    Type of concept/term [from list of values, to include facetHead]
Hierarchy position (minOccurs="0" maxOccurs="unbounded")
Hierarchical level
Class number / notation
Scope for which this concept/term holds (minOccurs="0" maxOccurs="unbounded")
Relationship (minOccurs="0" maxOccurs="unbounded")
Relationship type
Relationship target
/* See below for structure. */
Relationship strength (minOccurs="0" maxOccurs="1")
Audience level /* Of this relationship */ (minOccurs="0" maxOccurs="unbounded")

Perspective /* Of this relationship */ (minOccurs="0" maxOccurs="unbounded")

Scope for which this relationship holds (minOccurs="0" maxOccurs="unbounded")

Relationship, added information (minOccurs="0" maxOccurs="unbounded")

/* This could be a scope note explaining the relationship, an image illustrating the relationship, another term, etc. */

Type of added information /* Relationship types might be reused here. */

Relationship target

Audience level /* Of this piece of info. */ (minOccurs="0" maxOccurs="unbounded")

Perspective /* Of this piece of information */ (minOccurs="0" maxOccurs="unbounded")

Where relationship target has this structure (unifying term, text, images, multimedia document)

Relationship target

Type

/* Includes types of terms (descriptor, other preferred term, non-preferred term and types of texts and other documents, may be an elaborate hierarchy. */

Target value (a term or a document)

Term

Term variant (minOccurs="0" maxOccurs="unbounded")

Type of variant

/* Such as Preferred Spelling, other SPelling, ABbreviation, Full Term. */

Term form (complete term or Stem plus suffix)

Complete term

Stem plus suffix

Stem

Suffix

Document

Language (zero to many, exactly one for terms)

Audience level /* Of this relationship target */ (minOccurs="0" maxOccurs="unbounded")

Perspective /* Of this relationship target */ (minOccurs="0" maxOccurs="unbounded")

Scope for which this/term holds (minOccurs="0" maxOccurs="unbounded")