NeOn: Lifecycle Support for Networked Ontologies

Integrated Project (IST-2005-027595)

Priority: IST-2004-2.4.7 – "Semantic-based knowledge and content systems"

D2.4.3 Multilingual and Localization Support for Ontologies (v3)

Deliverable Co-ordinator: **Wim Peters**

Deliverable Co-ordinating Institution: University of Sheffield (USFD)

Other Authors: Mauricio Espinoza (UPM); Elena Montiel-Ponsoda (UPM); Margherita Sini (FAO)

This deliverable aims at providing further support for the localization of NeOn ontologies. The main contributions of this document are:
- a further extended version of the linguistic model for storing multilingual information in the *Linguistic Information Repository* (LIR);
- a set of networked ontologies for linguistic/terminological description and translation;
- evaluation results against FAO's AOS model;
- the automatic population of LIR with AOS linguistic data;
- a description of the latest version of the *LabelTranslator*, with focus on collaborative ontology localization.

Document Identifier:	NEON/2009/ D2.4.3/v1.7	Date due:	August 30, 2009
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	August 30, 2009
Project start date:	March 1, 2006	Version:	V1.7
Project duration:	4 years	State:	Final Version
		Distribution:	Public

NeOn Consortium

This document is a part of the NeOn research project funded by the IST Programme of the Commission of the European Communities by the grant number IST-2005-027595. The following partners are involved in the project:

Open University (OU) – Coordinator Knowledge Media Institute – Kmi Berrill Building, Walton Hall Milton Keynes, MK7 6AA United Kingdom Contact person: Martin Dzbor, Enrico Motta E-mail address: {m.dzbor, e.motta} @open.ac.uk	**Universität Karlsruhe – TH (UKARL)** Institut für Angewandte Informatik und Formale Beschreibungsverfahren – AIFB Englerstrasse 11 D-76128 Karlsruhe, Germany Contact person: Peter Haase E-mail address: pha@aifb.uni-karlsruhe.de
Universidad Politécnica de Madrid (UPM) Campus de Montegancedo 28660 Boadilla del Monte Spain Contact person: Asunción Gómez Pérez E-mail address: asun@fi.upm.es	**Software AG (SAG)** Uhlandstrasse 12 64297 Darmstadt Germany Contact person: Walter Waterfeld E-mail address: walter.waterfeld@softwareag.com
Intelligent Software Components S.A. (ISOCO) Calle de Pedro de Valdivia 10 28006 Madrid Spain Contact person: Jesús Contreras E-mail address: jcontreras@isoco.com	**Institut 'Jožef Stefan' (JSI)** Jamova 39 SI-1000 Ljubljana Slovenia Contact person: Marko Grobelnik E-mail address: marko.grobelnik@ijs.si
Institut National de Recherche en Informatique et en Automatique (INRIA) ZIRST – 655 avenue de l'Europe Montbonnot Saint Martin 38334 Saint-Ismier France Contact person: Jérôme Euzenat E-mail address: glesi.euzenat@inrialpes.fr	**University of Sheffield (USFD)** Dept. of Computer Science Regent Court 211 Portobello street S14DP Sheffield United Kingdom Contact person: Hamish Cunningham E-mail address: glesi@dcs.shef.ac.uk
Universität Koblenz-Landau (UKO-LD) Universitätsstrasse 1 56070 Koblenz Germany Contact person: Steffen Staab E-mail address: gle@uni-koblenz.de	**Consiglio Nazionale delle Ricerche (CNR)** Institute of cognitive sciences and technologies Via S. Martino della Battaglia, 44 – 00185 Roma-Lazio, Italy Contact person: Aldo Gangemi E-mail address: aldo.gangemi@istc.cnr.it
Ontoprise GmbH. (ONTO) Amalienbadstr. 36 (Raumfabrik 29) 76227 Karlsruhe Germany Contact person: Jürgen Angele E-mail address: angele@ontoprise.de	**Food and Agriculture Organization of the United Nations (FAO)** Viale delle Terme di Caracalla 1 00100 Rome Italy Contact person: Marta Iglesias E-mail address: marta.iglesias@fao.org
Atos Origin S.A. (ATOS) Calle de Albarracín, 25 28037 Madrid Spain Contact person: Tomás Pariente Lobo E-mail address: tomas.parientelobo@atosorigin.com	**Laboratorios KIN, S.A. (KIN)** C/Ciudad de Granada, 123 08018 Barcelona Spain Contact person: Antonio López E-mail address: alopez@kin.es

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed to the writing of this document or its parts:

USFD

UPM

FAO

Change Log

Version	Date	Amended by	Changes
1.1	10-07-2009	Wim Peters	Chapter 2
1.2	19-07-2009	Wim Peters	Chapter 3,4
1.3	22-07-2009	Mauricio Espinoza	Chapter 5
1.4	29-07-2009	Wim Peters	Integration and draft chapter 1
1.5	17-08-2009	Wim Peters	Addition of answer to EC comments on D2.4.2 Integration of comments provided by Elena Emontiel-Ponsoda
1.6	1-9-2009	Wim Peters	Finalization chapter 1
1.7	3-9-2009	Wim Peters	Chapter 6

Executive Summary

The previous deliverable in this series concentrated on the creation of the Linguistic Information Repository model, and localization assistance. With these in place, we have concentrate in this deliverable on the exploitation of the LIR model within the wider context of other models and user groups.

The two main themes addressed by this deliverable are the following:

- Interoperability between LIR and other models for linguistic/terminological description. This involves the following deliverable items:
    - a further extended version of the linguistic model for storing multilingual information in the Linguistic Information Repository (LIR);
    - a set of networked ontologies for linguistic/terminological description and translation;
    - evaluation results against FAO's AOS model
    - the automatic population of LIR with AOS linguistic data
- Collaboration between localizers using LabelTranslator**.** The deliverable contains a description of the latest version of the *LabelTranslator* plug-in for the NeOn Toolkit, with focus on collaborative ontology localization .

Table of Contents

List of tables

List of figures

1. Introduction

The LIR model has been designed in the previous deliverables D2.4.1 and D2.4.2 to be compatible with standards. In fact, it adopts a number of data categories for linguistic description from existing standards in order to guarantee interoperability with existing and proposed standards for the representation and integration of terminological and linguistic knowledge. For instance, the notion of *LexicalEntry* has been taken over from the Lexical Markup Framework (ISO[1] 24613, section 2.1.4), whereas the attribute *termType* has been borrowed from the ISO12620[2] set of terminological data categories.

The LIR was designed from the start with interoperability in mind. Compatibility with the incorporated elements from standard models is directly established by this incorporation strategy in the initial stage of the design of LIR.

Within the NeOn application areas, the LIR model is initially conceived as the hub within the network. The linguistic information covered by the LIR is sufficient for use case needs within NeOn, which are concept label selection and translation.

Constrained by its present coverage and usage within NeOn, the LIR can be seen as connected to a set of networked ontologies representing re-engineered versions of the standard representation formats. It interconnects the various standard descriptions for linguistic and terminological knowledge on the one hand, and ontological concepts on the other. It represents a core set of linguistic information units that cover a range of linguistic phenomena. Complementary lexical knowledge, inside or outside the direct scope of localization, should be obtained by navigating the ontologies networked together with the LIR.

Bearing in mind that the LIR model presents a sufficient, but non-exhaustive list of data categories for linguistic description, this deliverable addresses the need for interoperability between the LIR and standard models for linguistic and terminological description by mapping knowledge between these models and complementing the LIR with knowledge from these models. Our opinion is that interoperability as defined within a network of ontologies is the best way to go forward, because it allows us to re-use existing models, and is compliant with the NeOn philosophy of networked ontologies.

Another issue addressed in this deliverable is that the localization workflow developed so far in T2.4 only covers a single working scenario, where only one person performs all the major steps required for localizing an ontology to (and from) different natural languages. It needs to take collaborative aspects into account, if it wants to enable the localization workflow within organizations.

The structure of this deliverable is as follows:

Chapters 2 to 4 describe the selected standards and their integration into an informal network, which comprises the FAO AOS model for use case compatibility (see sections 2.2.1 and 4.3).

Chapter 5 introduces an advanced version of LabelTranslator that includes workflow-based model for a collaborative localization of ontologies

Chapter 6 draws conclusions and outlines future work for deliverable D2.4.4 in month 48.

[1] International Organization for Standardization

[2] http://www.ttt.org/oscar/xlt/webtutorial/

2. Networked ontologies for modelling linguistic and terminological knowledge

The embedding of LIR in a wider context of terminological and linguistic knowledge will ensure a maximum degree of interoperability and maximum authority and completeness of the captured knowledge by relying on multiple sources.

Linking the LIR to other resources also removes the need to re-implement knowledge that is already available, and make a first step in the direction of modularization of linguistic/terminological knowledge by selecting topic-based modules as extension of the LIR.

Furthermore, establishing an alignment between FAO's AOS model and the LIR has several advantages: it renders the two ontologies interoperable, and it allows automatic population of the LIR model with AOS elements, thus making AOS interoperable with the other ontologies in the network (see section 4.5).

Overall, lexical knowledge is expressed in various ways in terminological and linguistic resources. The nature and format of this knowledge is determined by internal factors, such as user needs and required level of adhesion to existing standards for the representation and integration of terminological and linguistic knowledge. Individual linguistic and terminological resources largely differ in the explicit linguistic information they expose, which may vary in format, content granularity and motivation (linguistic theories, intended users, purpose or system-oriented scope etc…) [3].

There are many proposed standards for enhancing the interoperability of encoded lexical knowledge. In this deliverable we mention some of the most important standardization initiatives, which capture terminological and linguistic information.

The task we set ourselves here is the integration of some of the standards relevant for localization and linguistic/terminological description into an ontology network, together with the LIR. In this network we will integrate FAO's AGROVOC ontological model (section 2.2.1), and LMM umbrella ontology for the linking of lexical and ontological knowledge (section 2.3).

The remainder of this chapter provides descriptions of each of the models that have been selected for integration into this network.

2.1 Translation Standards

Within a multilingual scenario, the correct and interoperable modelling of translational equivalence is crucial. Within the present version of the LIR this is modelled by means of the hasTranslation relation that links one LexicalEntry to another LexicalEntry. This relation indicates that one LexicalEntry is to be considered a translational variant of the other, and that the LIR entities involved is are semantically equivalent. The level of semantic equivalence is left underspecified. The hasTranslation relation offers a general level link between LexicalEntries, which is interoperable with existing translation standards, and extended by them.

The scope of research and development in localization and translation memory (TM) process development is very large, and many formats have been developed. Translation memories, also known as translation databases, are collections of entries where a source text is associated with its corresponding translation in one or more target languages. Some of the major formats of specific interest for localization and TM are TMX, XLIFF, and LMF (see below). There are many identical requirements for all the formats irrespective of the differences in final output. For example, all the formats aim at being user-friendly, easy-to-learn, and at reusing existing databases or knowledge. All these formats work well in the specific field they are designed for, but overall they lack a synergy. This is being addressed in the development of the MLIF standard (see below).

By aligning the standards mentioned above with the LIR we create a model that will make an increasing amount of translation data available for ontology localization.

Also, by concentrating on the interoperability between them it will be possible to freely exchange and compare elements from these different standards.

2.1.1 TMX

TMX (Translation Memory eXchange[3]) is the vendor-neutral open XML standard for the exchange of Translation Memory data created by Computer Aided Translation and localization tools. The purpose of TMX is to allow easier exchange of translation memory data between tools and/or translation vendors with little or no loss of critical data during the process. In existence since 1998, TMX is a certifiable standard format. TMX is developed and maintained by OSCAR (Open Standards for Container/Content Allowing Re-use), a LISA[4] (Localization Industry Standards Association) Special Interest Group.

The specifics of the TMX standard are available for free in the website[5], together with several related links, documents, articles and software tools.

Figure 2.1 below gives an example of a TMX file in XML format for translation variants in 6 languages. Figure 2.2 illustrates the organization of the subset of TMX classes that are most relevant to the modelling of translation relations. These are the following:

1. **TranslationUnit** (tu) contains the data for a given translation unit.
2. Attribute segType: "block", "paragraph", "sentence", or "phrase".
3. Attribute srcLang: Source language: specifies the language of the source text
4. **TranslationUnitVariant** (tuv) specifies text in a given language
5. Required attribute: xml:lang
6. **Context** describes the context of a TranslationUnit. The purpose of this context information is to allow certain pieces of text to have different translations depending on where they came from. The translation of a piece of text may differ if it is a web form or a dialog or an Oracle form or a Lotus form for example. This information is thus required by a translator when working on the file. Likewise, the information may be used by any tool proposing to automatically leverage the text successfully.
7. **Segment** an individual segment of translation-memory text in a particular language. It contains the text of the given segment. There is no length limitation to the content of a Segment element. All spacing and line-breaking characters are significant within a Segment element.
8. **Note** is used for comments. It has the attribute xml:lang.

[3] http://www.lisa.org/Translation-Memory-e.34.0.html

[4] http://www.lisa.org

[5] http://www.lisa.org/tmx/

```
<?xml version="1.0" encoding="UTF-8" ?>
    <!DOCTYPE tmx PUBLIC "-//LISA OSCAR:1998//DTD for Translation Memory eXchange//EN" "tmx14.dtd" >
<tmx version="1.4">
    <header adminlang="en" creationdate="20040731T164933Z" creationtool="Heartsome TM Server"
        creationtoolversion="1.0.1" datatype="xml" o-tmf="unknown" segtype="block" srclang="*all*"/>
    <body>
        <tu creationdate="20020919T004233Z" tuid="1091303313004">
            <tuv xml:lang="de">
                <seg>Xuan Zang - möglicherweise Chinas größter Übersetzer</seg>
            </tuv>
            <tuv xml:lang="en">
                <seg>Xuan Zang, Possibly China's Greatest Translator</seg>
            </tuv>
            <tuv xml:lang="es">
                <seg>Xuan Zang, probablemente el traductor más importante de China</seg>
            </tuv>
            <tuv xml:lang="it">
                <seg>Xuan Zang, probabilmente il più grande traduttore cinese</seg>
            </tuv>
            <tuv xml:lang="ko">
                <seg>현장법사(玄奘法師), 중국 최고의 번역가</seg>
            </tuv>
            <tuv xml:lang="zh-cn">
                <seg>玄奘，中国最伟大的翻译家</seg>
            </tuv>
        </tu>
    </body>
</tmx>
```

Fig. 2.1. Example of TMX file

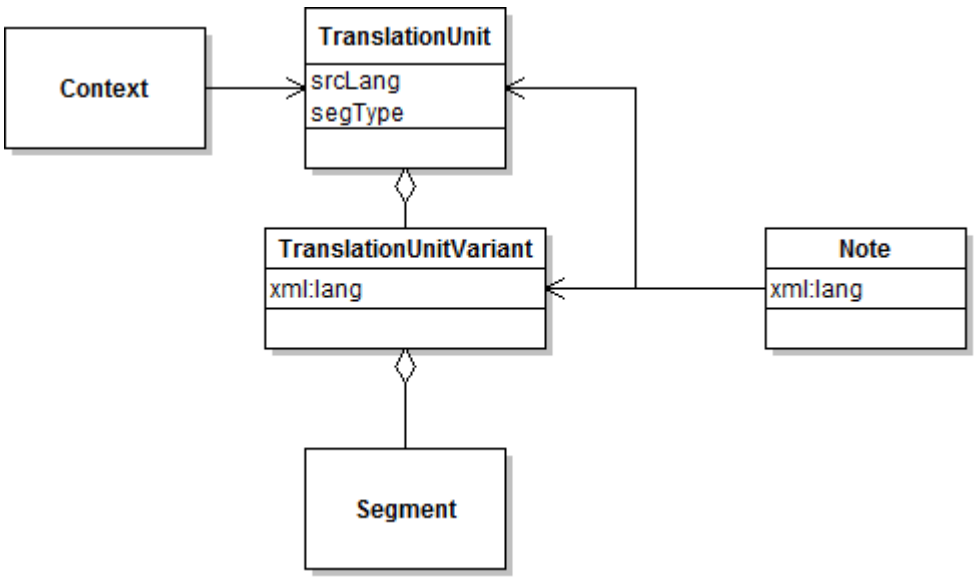

Fig.2.2. Structure of TMX fragment

2.1.2 XLIFF

The purpose of the OASIS XLIFF[6] standard is to define and promote, through extensible XML vocabularies, the adoption of a specification for the interchange of localizable software and document based objects and related metadata.

[6] http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html

XLIFF should be able to mark-up and capture localization information and interoperate with different processes and phases without loss of information. It should fulfill specific requirements of being tool-neutral. It should support the localization related aspects of internationalization and entire localization process. It also needs to support common software and content data formats. This should also provide an extensibility mechanism to allow the development of tools compatible with an implementer's proprietary data formats and workflow requirements.

1. **TransUnit**: contains (a set of) translational equivalences
2. **Source**: the source of the translation pair
3. **SegSource**: the translatable text, divided into segments
4. **Mrk**: Each segment is marked by means of the <Mrk> element with attribute **mType** set to the value "seg".
5. **Target**: the target of the translation pair; the attribute **Equiv-trans** indicates if the target language translation is a direct equivalent of the source text.
6. **Alt-Trans:** possible translations as Target instances

The example file below in figure 2.3 illustrates the bilingual part of an XLIFF document.

Figure 2.4 describes a selection from the XLIFF structural elements section in diagrammatical form.

```
<?xml version="1.0" encoding="UTF-8"?>
<xliff version="1.0" xml:lang="en">
    <file source-language="en" target-language="fr" datatype="winres" original="Sample1.rc">
        <header>
            <skl>
                <internal-file crc="64a2b9b0">
                    <!-- internal file information -->
                </internal-file>
            </skl>
        </header>
        <body>
            <group restype="dialog" resname="IDD_DIALOG1" coord="0;0;186;57" font="MS Sans Serif;8">
                <trans-unit id="1" restype="caption">
                    <source xml:lang="en">Title</source>
                    <target xml:lang="fr">Titre</target>
                </trans-unit>
                <trans-unit id="2" restype="label" resname="IDC_STATIC" coord="8;4;19;8">
                    <source xml:lang="en">&amp;Path:</source>
                    <target xml:lang="fr">&amp;Chemin :</target>
                </trans-unit>
                <trans-unit id="3" restype="check" resname="IDC_CHECK1" coord="8;40;41;10">
                    <source xml:lang="en">&amp;Validate</source>
                    <target xml:lang="fr">&amp;Valider</target>
                </trans-unit>
                <trans-unit id="4" restype="button" resname="IDOK" coord="129;7;50;14">
                    <source xml:lang="en">OK</source>
                    <target xml:lang="fr">OK</target>
                </trans-unit>
                <trans-unit id="5" restype="button" resname="IDCANCEL" coord="129;24;50;14">
                    <source xml:lang="en">Cancel</source>
                    <target xml:lang="fr">Annuler</target>
                </trans-unit>
            </group>
        </body>
    </file>
</xliff>
```

Fig. 2.3. Example of XLIFF

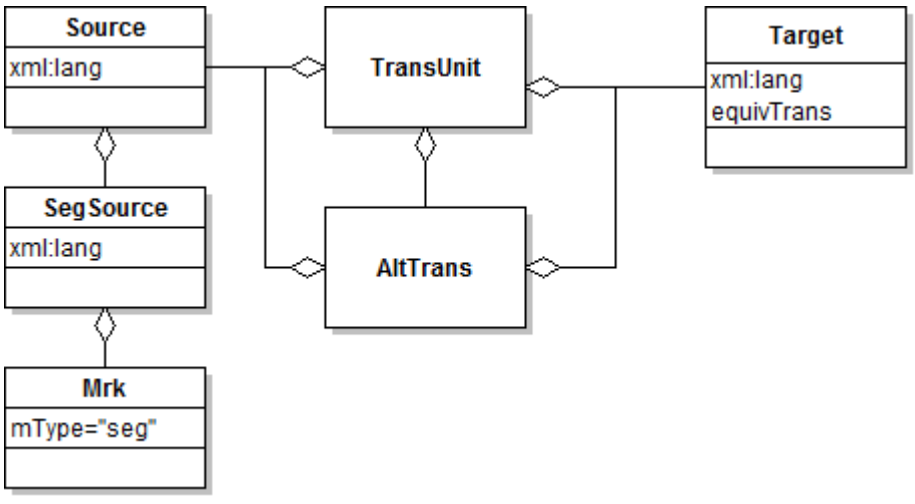

Fig. 2.4. Structure of XLIFF fragment

2.1.3 MLIF

Some of the major formats of specific interest for localization and translation have been listed above. In order to address the interoperability between them, the Multi Lingual Information Framework (MLIF) is a standard under development with the objective of providing a common platform for the existing standards. It provides a generic platform for modelling and managing multilingual information in various domains: localization, translation, multimedia, document management, digital library, and information or business modelling applications. MLIF provides a metamodel and a set of generic data categories for various application domains. MLIF also provides strategies for the interoperability and/or linking of models including (but not limited): XLIFF and TMX. It promotes the use of a common framework for the future development of several different formats. MLIF can be considered as a parent for all these formats, since all of them deal with multilingual data expressed in the form of segments or text units. They all can be stored, manipulated and translated in a similar manner.

MLIF will introduce a metamodel in combination with chosen data categories (ISO 12620), as a means of ensuring interoperability between several multilingual applications and corpora. MLIF deals with multilingual corpora, multilingual fragments, and the translation relations between them.

In order to ensure compatibility with this standard under development, we fit it into our network of translation ontologies. The reader needs to keep in mind that the structure illustrated by figure 2.5 represents a snapshot of the current version of MLIF.

According to the latest specifications, the MLIF core model has the following elements:

1. **MultiC** (Multilingual Component): groups together all variants of a given textual content.
2. **MonoC** (Monolingual Component): part of a multilingual component, containing information related to one language. Its attributes are the following: **languageIdentifier** contains an ISO639 code; **translationRole** determines whether the encompassing MonoC component corresponds to a source language or a target language in a translation process.
3. **SegC** (Segmentation Component): a recursive component allowing any level of segmentation for textual information. It has the following attributes: **segment** contains the segment string; **pos** denotes part of speech and **lemma** contains the citation/canonical form of the segment.

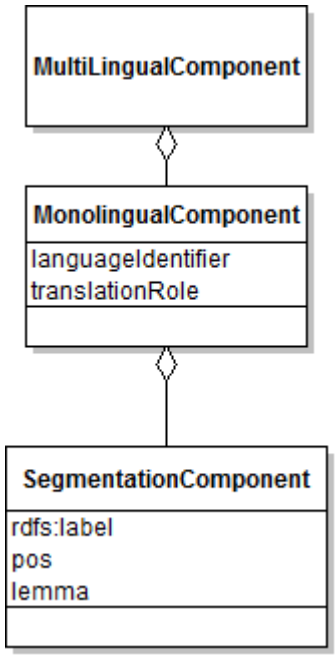

Figure 2.5. Structure of core MLIFF fragment

2.1.4 LMF

The LMF Multilingual notation package is a module, separate from the LMF core package described in section 2.2.2, which is devoted to multilingual notation, and can be used to represent bilingual and multilingual resources. The framework is based on the notion of SenseAxis. The multilingual package comes equipped with the possibility to define connections between a node in a lexicon (e.g. a SenseAxis instance) and knowledge representation systems, such as ontologies or fact databases, as well. This is allowed by the use of the InterlingualExternalRef class.

The core multilingual package contains the following elements.

1. **Sense Axis** is a class representing the relationship between different closely related senses in different languages and implements an approach based on the interlingual pivot. The purpose is to describe the translation of lexemes from one language to another. Optionally, a Sense Axis may refer to an external knowledge representation system where the appropriate equivalent can be found.
   Attributes: **id** and **label** (name of the relation, e.g. “synonymy”)

2. **Sense Axis Relation** is a class representing the relationship between two different Sense Axis instances. Attributes: **id** and **label** (name of the relation, e.g. “specialization”)

3. **Interlingual External Ref** is a class representing the relationship between a Sense Axis instance and an external interlingual knowledge organization system.
   The attributes **externalSystem** and **externalReference** are provided to refer respectively to the name(s) of the external system and to the specific relevant node in this given external system.

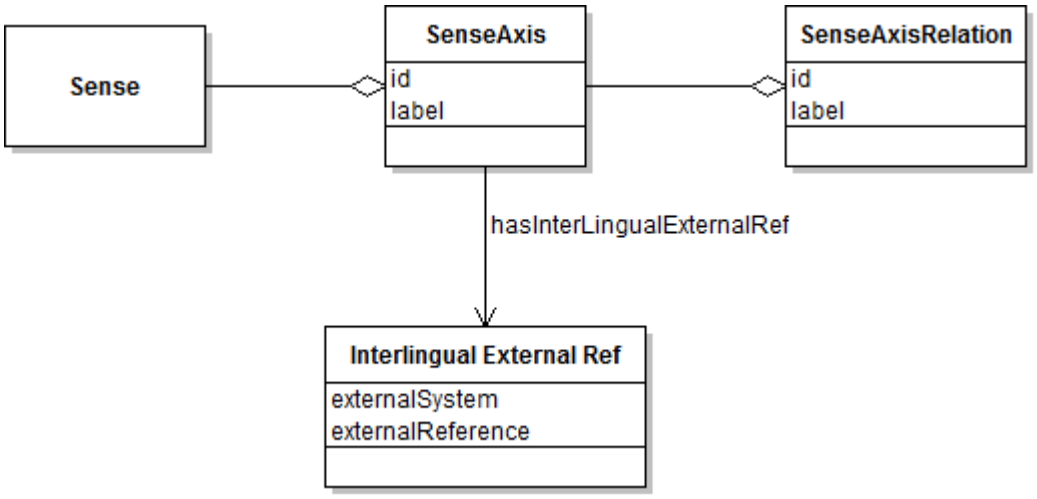

Figure 2.6. Structure of LMF multilingual module

2.1.5 Re-engineering of standards

In our modelling effort we focus on the multilingual part of the translation standards, which form the core subset of the total specification of each standard. Instead of re-engineering the complete specifications of these standards and best practice formats, we concentrate on those ontology elements that are pertinent to our purpose for describing translation and that have been described above. This means that we will only use the core part of the specifications that directly handle the translation process. Effectively, we modularize the descriptions for our networking purposes.

The standards contain more details in their specification. Several concepts from the standard specifications have been left out after having been judged not to form part of the core conceptual vocabulary that describes translation.

While re-engineering, we investigated the re-engineering patterns proposed in NeOn deliverable D2.2.2 [15]. None of them is applicable to the data structures contained in these standard descriptions. As a consequence, they all fall under the ad-hoc category of re-engineering patterns. However, a few patterns can be observed when re-engineering from the xml and xsd data structures.

XLIFF's xsd[7] suggests that the direct embedding of other xsd elements within the *xsd:sequence* element corresponds to a meronymic relation. For instance, the following embedding (represented by nesting) indicates that *target* and *source* are elements that are part of *trans-unit*.

```
<xsd:element name="trans-unit">
  <xsd:complexType>
    <xsd:sequence>
          <xsd:element name="target">
          <xsd:element name="source">
    </xsd:sequence>
  </xsd:complexType>
```

[7] http://schemas.liquid-technologies.com/Oasis/XLIFF/1.2/default.html?url=http://schemas.liquid-technologies.com/Oasis/XLIFF/1.2/xliff-core-1_2-strict_xsd.html

</xsd:element>

TMX seems to use the same pattern in its embedding operation within the dtd.

e.g.:

<!ELEMENT tu ((note|prop)*, tuv+) >

This describes that a translation unit (tu) consists of one or more translation unit variants (tuv).

The pattern derived from evidence from these two resources is the following:

If an entity that has a re-engineered counterpart is embedded within another entity, which has a re-engineered counterpart, then there is a meronymic relation between the two.

This meronymic relation can be implemented by means of the hasPart object property from the Dolce-ultra Light ontology[8], with the embedding element as the domain, and the embedded element as the range.

Because this pattern is corroborated by two resource models, we have proposed it as a candidate re-engineering pattern on the ontology design patterns portal[9].

2.1.6 Resulting ontologies

The ontologies that are the products of the re-engineering process described in this section are the following:

- TMX-core.owl
- XLIFF-core.owl
- MLIF-core.owl
- LMF-multilingual-module.owl

They are online available at: http://gate.ac.uk/gate-extras/neon/ontologies/, and fully listed in Appendix B-E.

2.2 Linguistic/terminological description

2.2.1 AOS

The AGROVOC Thesaurus was developed by the Food and Agriculture Organization (FAO) and the Commission of the European Communities in the early 1980s, and first published in 1982 in three languages: English, Spanish and French. It is defined as a multilingual structured and controlled vocabulary.

The AGROVOC thesaurus is the foundation that underpins the development of the Agricultural Ontology Service (AOS) project. Using the knowledge contained in vocabulary systems and thesauri such as AGROVOC, the AOS will be able to develop specialized domain-specific

[8] http://www.ontologydesignpatterns.org/ont/dul/DUL.owl

[9] http://ontologydesignpatterns.org

terminologies and concepts that will better support information management in the web environment. A key objective is to add more semantics to the thesaurus, for example, by expanding and better specifying the relationships between concepts.

The AGROVOC Concept Server[10] has remodelled the AGROVOC Thesaurus into an ontological format[11] (see figure 2.7).

2.2.1.1 Advantages of the LIR modelling choices for the AOS use case

As a use case, the interoperability between AOS and LIR is important in NeOn, and will therefore be addressed in this deliverable.

In this section we provide an answer to the comment of the review committee, regarding D2.4.2 section 2.3:

"It would be helpful to evaluate why the approach explained in Sec2.3 is actually better than the other two approaches in the FAO case study."

From a general methodological point of view, we identified three modelling options for the association of linguistic information with ontological concepts:

1. Including multilingual data in the ontology meta-model: localization at the terminological layer (D2.4.2 section 2.1)
2. The ontology meta-model with a mapping model: localization at the conceptual layer (D2.4.2 section 2.2)
3. Associating the ontology meta-model with a multilingual linguistic model: localization at the terminological and conceptual layers (D2.4.2 section 2.3)

The general benefits of keeping the ontology and the linguistic and multilingual information apart (i.e. choosing option 3) were already detailed in this section 2 of D2.4.2. Our objective there was to justify the modelling modality chosen within the NeOn project for the representation of multilingual information, which mainly relied on the association of the OWL meta-model with an external linguistic model, the *Linguistic Information Repository* or LIR.

In the same deliverable, we also made a first approach in the comparison of the LIR against traditional multilingual representations models on the basis of the AGROVOC Thesaurus example (section 4 in D2.4.2). Deficiencies of traditional resources such as thesauri or glossaries were highlighted, and we already exemplified how the LIR could overcome them.

With respect to the LIR's benefits specifically for AOS, we can make the following observations.

- The LIR permits the representation and inclusion of as much linguistic information as needed, without interfering with the conceptualization layer. For AOS this means that any future additions to the linguistic representation can be modelled directly in the LIR, and do not require an additional conceptual extension to the AOS ontology. This is preferable to option 1 above.
- Option 2 above requires localization at the conceptual level. In addition, complex interlingual mappings need to be established between language specific structures. Given the terminological provenance of its data, the AOS structure implies synonymy between the various lexicalizations of a concept in different languages, and does therefore not need the additional mapping complexity of this model, nor the considerable additional effort involved in a localizing activity according to option 2.

[10] http://naist.cpe.ku.ac.th/agrovoc/

[11] http://aims.fao.org/aos

- If required, If necessary for AOS purposes, the LIR allows the capturing of linguistic and conceptual divergences among different cultures and languages.
- The LIR's interoperability with other standards or representation schemas of lexical and terminological information enables access to many sources of lexical information for concept label localization. This assists in the choice of lexical material for the labelling of AOS concepts.
- The LIR permits complementation of linguistic coverage by its linkage to other linguistic representation models whenever the information needed for the final application is not covered in the LIR. This will cater for any future linguistic requirements of AOS versions.
- The LIR allows establishing links or relations not only among the linguistic elements within one language, but also among linguistic elements across languages (e.g. relation to the source of provenance of a certain term within one language, or translation relations across languages). Furthermore, whereas the AOS model allows some level of modulation of equivalence relations between terms within one language, the networked and extensible nature of LIR will cover the addition of any type of semantic relation between word senses a new version of the AOS model requires this.
- The LIR permits linguists or domain experts without ontology development expertise access to the linguistic information (terminological layer) in a distributed environment.

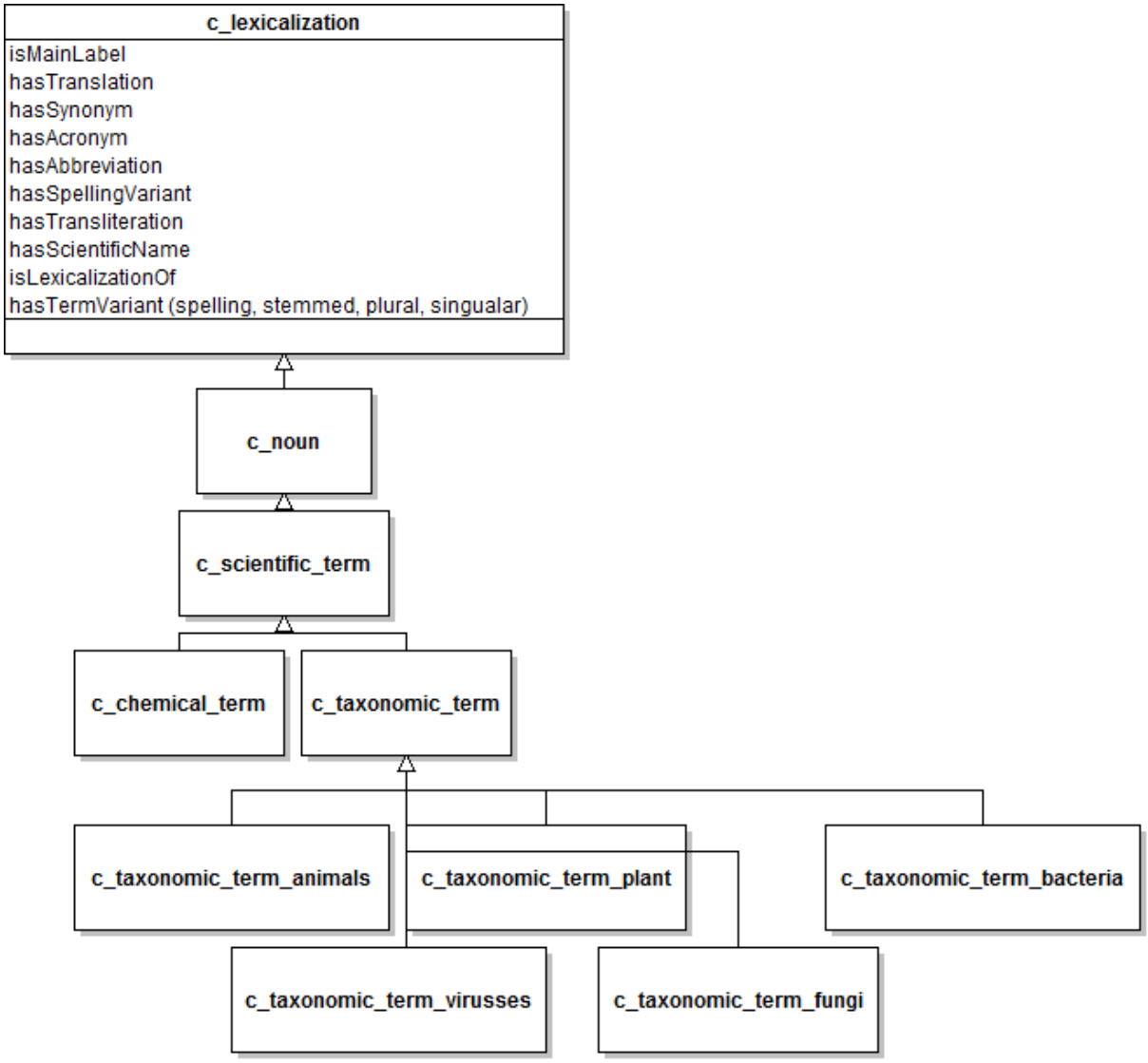

Figure 2.7. the AOS model

2.2.2 Lexical Markup Framework

The Lexical Markup Framework (LMF, ISO 24613:2008) is a model providing a common standardized framework for the description and representation of NLP lexicons. The goals of LMF are to provide a common model for the creation and use of such lexical resources, to manage the exchange of data between and among them, and to enable the merging of a large number of individual resources to form extensive global electronic resources.

LMF has been specifically designed to cover as many models of lexical representations as possible. It is therefore to be regarded as a meta-model, i.e a high-level specification for lexical resources defining the structural constraints of a lexicon. It is organised around two main components:

1. The core package, i.e. a structural skeleton to represent the basic hierarchy of information in a lexicon, under the form of core classes of objects and relations.

2. A set of modular extensions to the core package, i.e. additional classes and relations required for the description of specific types of lexical resources (see fig. 2.9). Available extensions include morphology, syntax, semantics, multilingual notations (see section 2.1.4), paradigm classes, multi-word expression patterns and constraint expressions.

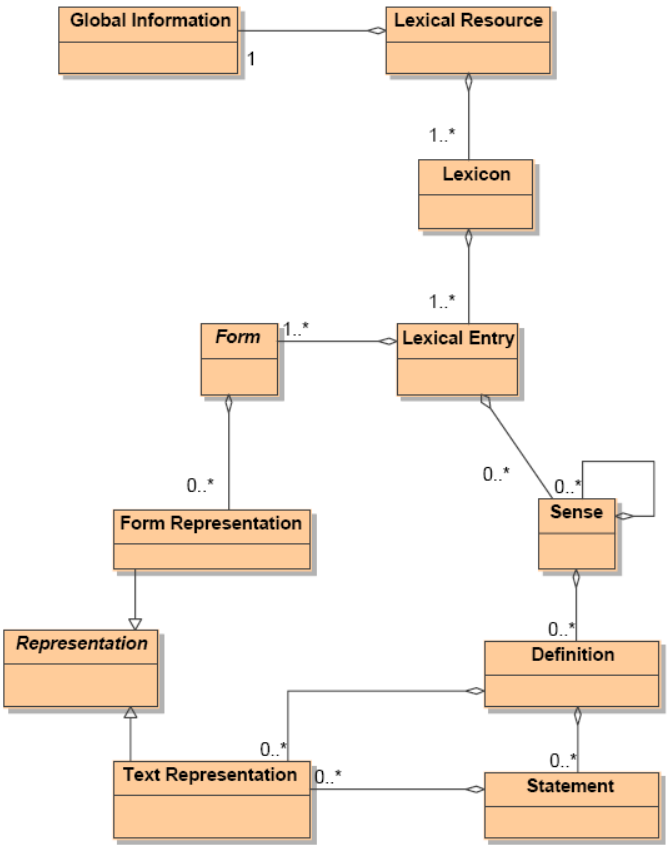

Figure 2.8. LMF core model

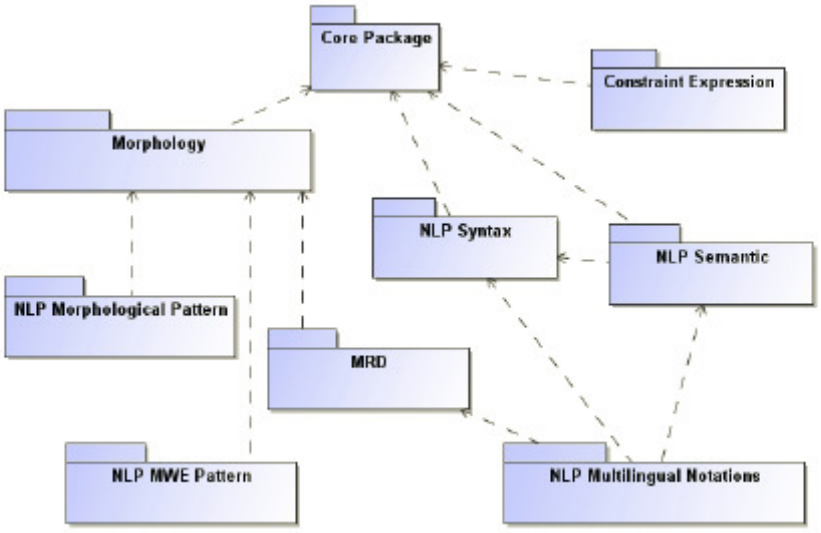

Figure 2.9. Dependencies between the LMF core and extension packages.

2.3 LMM

LMM (Linguistic Meta-Model) has been developed as an umbrella ontology for bridging lexical resources and ontologies [1]. The primary goal of LMM is to allow a simultaneous representation of multiple linguistic knowledge sources, allowing interoperability and an improved comprehension and exploitation of knowledge.

It provides a semiotic-cognitive representation of linguistic knowledge and grounds it in a formal semantics. LMM integrates linguistic knowledge sources, such as WordNet [9] and Framenet [7], as well as foundational ontologies, such as DOLCE [11] and its extensions, notably the Descriptions and Situations framework [10].

LMM encompasses elements from the ontologies mentioned above, where they are adapted to a semiotic perspective. In addition, LMM is already aligned with SKOS Core[12] (Simple Knowledge Organization Systems), which has been developed within the W3C framework, and provides a model for expressing the basic structure and content of concept schemes such as thesauri, classification schemes, subject heading lists, taxonomies, 'folksonomies', other types of controlled vocabulary, and also concept schemes embedded in glossaries and terminologies. This means that the alignment between LIR and LMM will provide at least a partial alignment with SKOS.

The most important feature of LMM is its ability to support the representation of different knowledge sources developed according to different underlying semiotic theories.

The LMM ontology consists of two modules:

LMM1[13] models the main semiotic notions by means of three classes: Reference, Meaning and Expression, formalizing the distinctions of the semiotic triangle [1] [8] (see figure 2.10).

The two layers of meaning and reference are connected to the language by means of the expression layer. According to [1], expressions are social objects produced by agents in the context of communicative acts. They are natural language terms, symbols in formal languages, icons, and whatever can be used as a vehicle for communication.

[12] http://www.w3.org/2004/02/skos/

[13] http://www.ontologydesignpatterns.org/ont/lmm/LMM_L1.owl

LMM2[14] contains, in addition to LMM1, specific linguistic constructs and references represented as subclasses of its InformationObject class (see figure 2.11). InformationObject itself is a subclass of Expression in the semiotic triangle. Because this ontology covers the same linguistic level as LIR, LIR is aligned to this ontology.

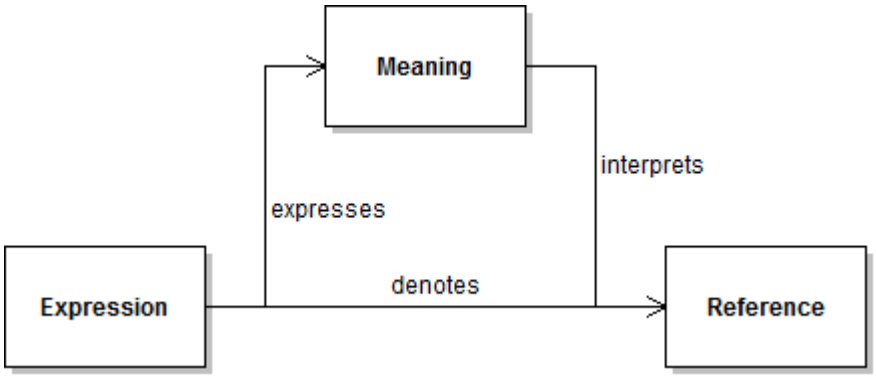

Figure 2.10. The semiotic triangle as represented in LMM1

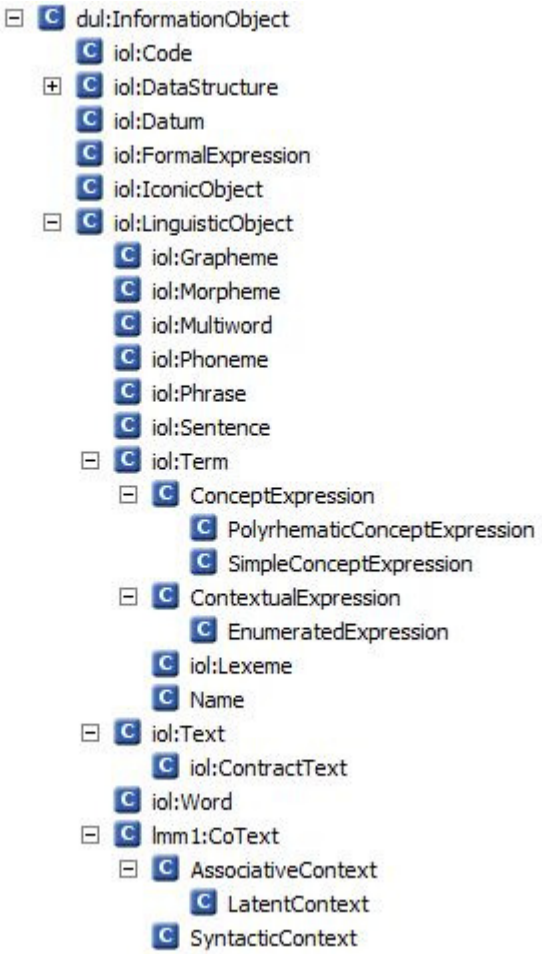

Figure 2.11. Linguistic/terminological information covered by LMM2

[14] http://www.ontologydesignpatterns.org/ont/lmm/LMM_L2.owl

3 Extension of LIR

This chapter describes a number of additions to the LIR model that are required by considerations of compatibility with standard translation, terminological and lexical models.

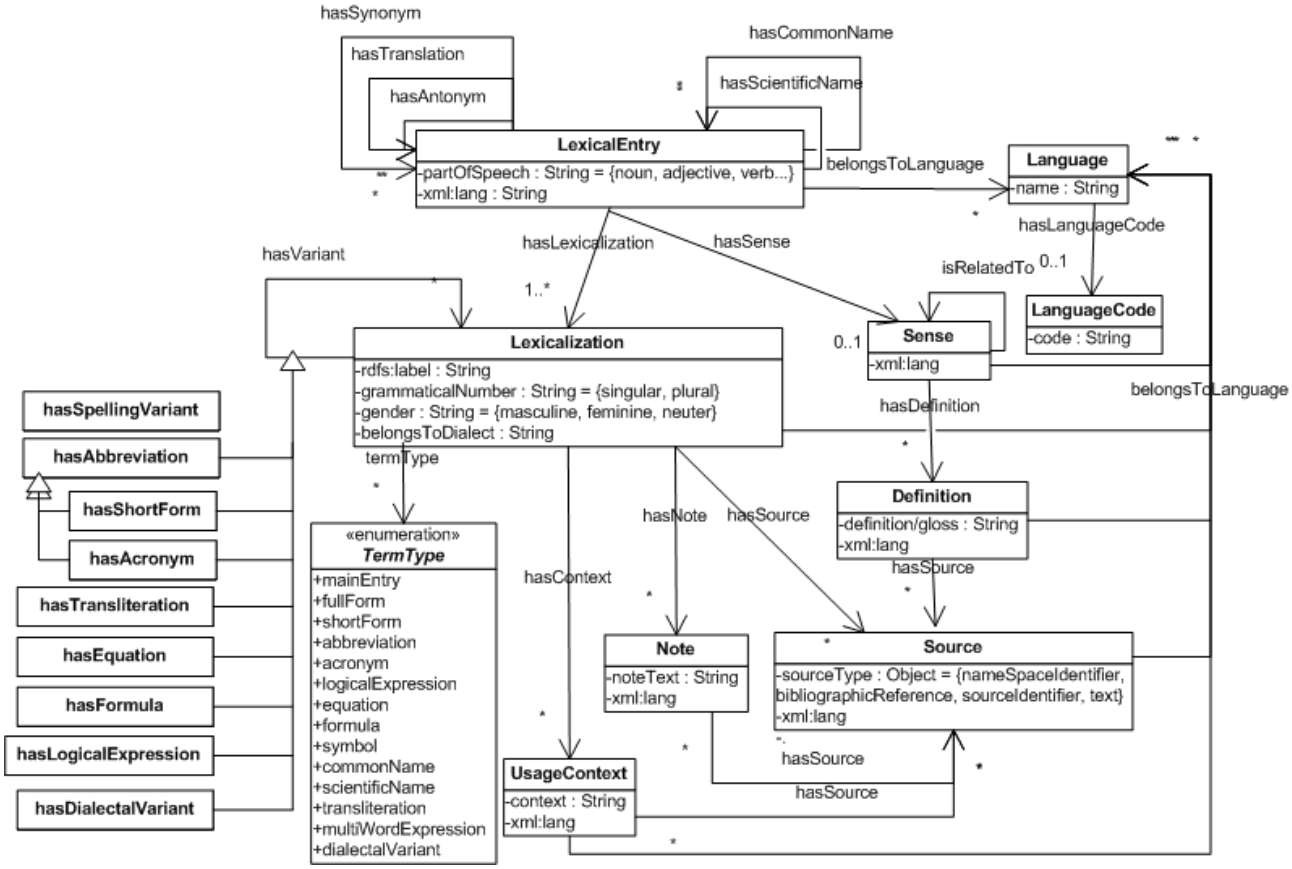

Figure 3.1. The D2.4.2 LIR model

In order to clarify where the additions fit in, the figure above illustrates the LIR model as defined in deliverable 2.4.2.

It has never been the aim of the LIR effort that it should cover the whole linguistic/terminological descriptive spectrum. The main requirement for the LIR is that it needs to cover a subset of linguistic description that is relevant for the NeOn purpose of providing a representational apparatus for multilinguality and linguistically closely associated notions, in order to associate this linguistic information with the labels that have been chosen for the ontology concepts.

Overall, the main requirement of representativity for the content of the LIR can be more specifically described as follows:

1. The LIR should contain a core set of linguistic data categories that is sufficient to cover general requirements of multilinguality and morpho-syntax, and the use cases in particular.
2. interoperability with other models should be as straightforward as possible

From a general perspective of linguistic coverage, exhaustiveness is not a requirement when there are standard models available that complement the information covered by the LIR. This information can be re-used and should not be created from scratch within the LIR. In effect, the LIR

does not cover phonetic, phonological, or pragmatic information, and only a small subset of syntactic. We aim mainly at good coverage in the areas of orthography and morpho-syntax.

These considerations of re-usability, interoperability and complementarity of linguistic/terminological descriptive classification point to a need for the creation of a set of networked ontologies for linguistic and terminological description. In order for the LIR to adequately fit into this network, it is first necessary to extend the LIR, according to the requirements above, with the additional concepts and properties, described in the following subsections.

The general methodology of re-using existing models as much as possible in a networked environment, while adding to the LIR when no standard or best practise solution is found, can, in principle, be applied to any task involving the capture of knowledge from both original as well as re-engieered linguistic and terminological resources.

3.1 Coverage of requirements I: generality of coverage

In the light of requirement 1 (generality of coverage) we decided on the following extensions, necessary to capture information from FAO's AOS model.

3.1.1 Language codes

Instead of referring to external URI's, the LIR now imports http://www.fao.org/aims/aos/languagecode.owl for easy and direct access to the different language codes.

3.1.2 Stem

The LIR so far lacks stemming information. This is remedied by

- introducing the Boolean data type property *stemmedForm*, which is a subproperty of termType

  domain: Lexicalization

- defining the *hasStemmedForm* object property

  domain: Lexicalization

  range: Lexicalization

inverse: stemmedFormOf

3.1.3 Semantic type

The hasSemanticType data property links a lexical entry to a high level terminological or lexical semantic class from any external taxonomy.

domain: LexicalEntry

range: anyURI

While this property will allow the inclusion of any external category, its main function is to cover the semantic types expressed in AOS as subclasses of c_noun:

- http://www.fao.org/aims/aos/agrovoc#c_scientific_term
- http://www.fao.org/aims/aos/agrovoc#c_chemical_term
- http://www.fao.org/aims/aos/agrovoc#c_taxonomic_term
- http://www.fao.org/aims/aos/agrovoc#c_taxonomic_term_animals
- http://www.fao.org/aims/aos/agrovoc#c_taxonomic_term_bacteria
- http://www.fao.org/aims/aos/agrovoc#c_taxonomic_term_fungi
- http://www.fao.org/aims/aos/agrovoc#c_taxonomic_term_plant
- http://www.fao.org/aims/aos/agrovoc#c_taxonomic_term_viruses

3.2 Coverage of requirements II: Maximizing interoperability with other standards

In the light of requirement 2 (interoperability with other models), we have opted for the creation of modules on the basis of the LMF specification, which constitute (parts of) extension packages describing particular phenomena. These modules are then imported by the LIR ontology.

3.2.1 Multi word expressions

Until now, multi word expressions are only broadly covered in LIR by means of the boolean data type property *multiWordExpression*.
We have decided to adopt the LMF specification, which models multi word units by creating a module that is part of its normative Morphology extension package. This module consists of the following classes:

- **List Of Components** is a class representing the aggregative aspect of a multiword expression. The List Of Components class is in a zero or one aggregate relationship with the Lexical Entry class. Each List Of Components instance should have at least two components.
  The mechanism can also be applied recursively, that is a multiword expression may be comprised of components that are themselves multiword expressions.
- **Component** is a class representing a reference to a lexical entry for each lexical component aggregated in a ListOfComponents class.

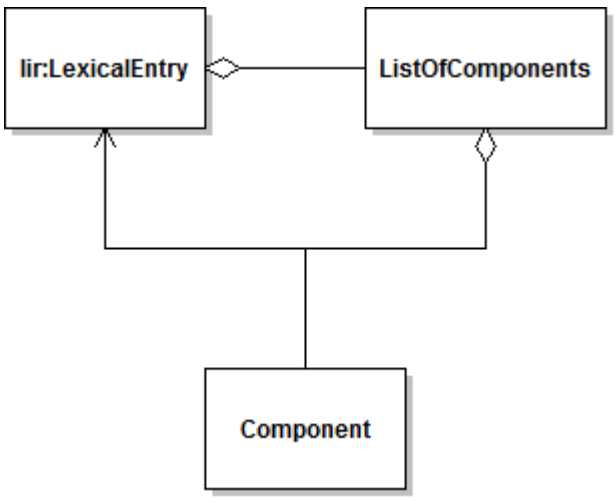

Figure 3.2. The LMF-derived list of components module integrated into LIR

The module is available from http://gate.ac.uk/gate-extras/neon/ontologies/Lmf-component-module.owl, and listed in Appendix E.

3.2.2 Lemma

As opposed to LIR, the LMF distinguishes the category Lemma, which is the citation form/canonical form that references the lexical entry it is associated with. It is described as follows: lemma; lemmatised form; canonical form (ISO 24613:2008): conventional form chosen to represent a lexeme."

In most cases the lemma is a word form of a certain type, such as singular form of a noun or infinitive form of a verb. This is modelled within the LIR at the level of Lexicalization, with the Boolean value of the data type property mainEntry set to "true".

In addition to the most common case of lemma representation by means of an existing word form, the LMF foresees abstract lemmas that do not orthographically fully reflect a linguistic unit. For example, in cases of homonymy, lemmas may express this by numbering: *bat1* (animal) and *bat2* (cricket instrument). Therefore, we introduce a new data type property *lemma* with a string value. This will take over the lemma identification function of mainEntry. String identity between the value of this property and the label of Lexicalization will enable the user to retrieve further typological information expressed by the data type properties associated with Lexicalization (e.g. singular/plural; fullForm).

3.2.3 Refinement of the synonymy relation

The hasSynonym relation is now further classified into subtypes, which brings the LIR into line with SKOS and AOS:

a. *hasSynonym* for full equivalence;

b. *hasNearSynonym* for fuzzy or partial overlap

c. *hasBroaderSynonym* or *hasnarrowerSynonym* (both taken from AOS) if there is an inclusion relation between two synonymous LexicalEntries.

4 Interoperability between (standard) terminological/linguistic descriptions

This section lists the alignments that combine the ontologies described above into a network between the LIR on the one hand, and the ontologies described in section 2 on the other. These alignments effectively constitute an ontology network for linguistic/terminological description.

4.1 Alignment relations between concepts within different networked ontologies

Various relation sets and formalisms have been defined to express alignments [14], amongst which c-owl [2], e-connections [23]. All these all constitute what we call *conceptual* alignments, in that they characterize the semantic nature of the mapping relation.

In the remainder of this section, these conceptual alignments between ontology elements will be informally expressed. Formal implementation will follow in deliverable 2.4.4 (M48) according to the selected formalism.

The relations we use for the various alignments in the next section are the following:

- IsEquivalentTo
- hasHypernym
- hasHyponym
- hasPart; partOf
- PartialOverlap

4.2 Alignments

The following sections informally define the alignments between the models described in section 2.

4.2.1 LIR – LMF

In order to link LIR with LMF, we have opted for creating LMF modules ourselves rather than modularizing the existing LMF ontology. The rationale behind this is that there is no authoritative LMF ontology yet. The present version[15] is a first step, but needs extension and adjustment. We have, however, maintained links with this version. Once the final version of LMF is available, the modular ontologies will be mapped back into it. Also, with respect to economy of representation, and the maximization of interoperability according to the NeOn notion of networked ontologies, any extension to the LIR model should take other models as much as possible into account by means of modular incorporation.

While LIR and LMF have a certain amount of overlap, LMF is more comprehensive in its linguistic coverage because of its modular extensions (see section 2.1.4 and 2.2.2). This is the reason why we adopted the strategy of identifying modules within the LMF according to the extension packages.

[15] Available from http://ontoware.org/frs/download.php/606/lmf.owl

On the other hand, LIR covers a number of data categories that are not specified in LMF, but have been sourced from other standards such as ISO12620. LMF is rather underspecified with respect to the number and nature of attributes it prescribes for a class. Therefore it is only possible to create most equivalences at class or relation level.

A comparison of LIR and the LMF core model yields the following informal correspondences:

Class to class

- Lir:LexicalEntry hasHypernym lmf:LexicalEntry (LMF core module)

  The difference between the two is that the LIR LexicalEntry represents a lexical unit: a unit of form and meaning expressed by the combination of a set of related word forms and zero or one single sense. LMF, on the other hand, defines its LexicalEntry concept as a lexeme (the minimal unit of language which has a semantic interpretation and is made up of one or more form-meaning composites called lexical units) The LMF lexeme.allows zero to many senses per LexicalEntry. Therefore lir:LexicalEntry is subsumed by lmf:LexicalEntry.
- Lir:Lexicalization isEquivalentTo lmf:WordForm (LMF morphological extension module)
- Lir:Sense isEquivalentTo lmf:Sense (LMF core module)
- Lir:Definition isEquivalentTto lmf:Definition (LMF core module)
- Lir:UsageContext isEquivalentTo lmf:Context (LMF MRD extension module)
- Lir:Source with attribute nameSpaceIdentifier isEquivalentTo lmf:MultilingualExternalReference with attribute externalReference (LMF Multilingual Extension module)
- Lir:Source with attribute nameSpaceIdentifier isEquivalentTo lmf:MonolingualExternalReference with attribute externalReference (LMF semantic package)

Attribute to attribute

- Lir:grammaticalNumber isEquivalentTo lmf: grammaticalNumber
- Lir:partOfSpeech isEquivalentTo lmf:partOfSpeech

Relation to class

- lir:isRelatedTo isEquivalentTo SenseRelation (LMF core module)
- Lir:hasStemmedForm isEquivalentTo lmf:Stem (LMF morphological extension)
- Lir:hasSynonym isEquivalentTo lmf:SenseRelation with label=”synonymy” (LMF core module)

Relation to class (complex)

- Lir:hasTranslation isEquivalentTo lmf:SenseAxis

Relation to attribute

- Lir:belongsToLanguage isEquivalentTo Language (LMF core module)

4.2.2 LIR – TMX core

Class to class

- lir:UsageContext isEquivalentTo Tmx:Context
- lir:Component isEquivalentTo tmx:Segment
- lir:Note isEquivalentTo tmx:Note
- lir:LexicalEntry isEquivalentTo tmx:TranslationUnitVariant

Relation to class (complex)

- lir:hasTranslation hasPartialOverlapWith tmx:TranslationUnit

Relation to attribute

- lir:belongsToLanguage isEquivalentTo tmx:srcLang

4.2.3 LIR-XLIFF core

Class to class

- lir:LexicalEntry hasPartialOverlapWith xliff:Source
- lir:LexicalEntry hasPartialOverlapWith xliff:Target
- lir:LexicalEntry hasPartialOverlapWith xliff:AltTrans
- lmf-component-module:Component isEquivalentTo xliff:Mrk with mType=”seg”
- lmf-component-module:ComponentList hasPartialOverlapWith xliff:SegSource

Relation to class (complex)

- lir:hasTranslation isEquivalentTo xliff:TransUnit

The lir:Lexicalization associated with the lir:LexicalEntry that is the domain of lir:hasTranslation isEquivalentTo xliff:Source.

The lir:Lexicalization associated with the lir:LexicalEntry that is the range of lir:hasTranslation isEquivalentTo xliff:Target.

Relation to attribute

- lir:belongsToLanguage isEquivalentTo xliff:languageIdentifier
- lir:hasSynonym isEquivalentTo xliff: equivTrans

4.2.4 LIR-MLIF core

Class to class

- Lir:LexicalEntry isEquivalentTo mlif:MonolingualComponent
- lmf-component-module:Component isEquivalentTo mlif:SegmentationComponent

Relation to class (complex)

- lir:hasTranslation isEquivalentTo mlif:MultiLingualComponent

The lir:Lexicalization associated with the lir:LexicalEntry that is the domain of lir:hasTranslation isEquivalentTo mlif:SegmentationComponent. This is part of a MonoLingualComponent with attribute translationRole=”source”.

The lir:Lexicalization associated with the lir:LexicalEntry that is the range of lir:hasTranslation isEquivalentTo mlif:SegmentationComponent. This is part of a MonoLingualComponent with attribute translationRole=”target”.

4.3 LIR-LMM

In general terms, LIR objects are subclasses of lmm:LinguisticObject, insofar they denote orthographic, morphosyntactic or morphological information.

Sense and Definition fall under lmm:Meaning.

The relation in LMM is LinguisticObject expresses Meaning.

Class to class

- Lir:LexicalEntry isEquivalentTo lmm2:ConceptExpression
- Lir:Lexicalization hasPartialOverlapWith lmm2:Grapheme
- Lir:Lexicalization hasHypernym lmm2:LinguisticObject
- Lir:UsageContext hasHypernym lmm2:Cotext
- Lir:Sense hasHypernym lmm2:Concept
- Lir:Definition hasHypernym lmm2:Concept
- Lir:Note hasHypernym lmm2:Text
- Lir:Source hasHypernym lmm2:InformationObject
- Lir:Language isEquivalentTo lmm2:NaturalLanguage
- Lir1:LanguageCode hasHypernym lmm2:Code
- lir:Component hasHypernym lmm2:InformationObject

Class with attribute to class

- Lir:Lexicalization with attribute multiWordExpression =”true” isEquivalentTo lmm:MultiWord
- Lir:Lexicalization with attribute multiWordExpression =”true” isEquivalentTo lmm:Phrase

Class to union of classes

- Lir:LexicalEntry hasHypernym lmm2:ConceptExpression and lmm2:Concept

Relation to class

- lir:isRelatedTo isEquivalentTo lmm2:RelatedMeaning

- lir:hasSynonym hasHypernym lmm2:RelatedMeaning
- lir:hasBroaderSynonym hasHypernym lmm2:RelatedMeaning
- lir:hasNarrowerSynonym hasHypernym lmm2:RelatedMeaning

4.4 LIR - AOS

FAO's AOS model is a complex terminological model that has its roots in existing terminological standards such as ISO12620 [16].

LIR also comprises terminological standard concepts from ISO12620, which makes the interoperability more straightforward where these standard data categories are used.

Class to class

- lir:LexicalEntry isEquivalentTo aos:c_lexicalization
- lir:LexicalEntry isEquivalentTo aos:c_scientific_term
- lir:LexicalEntry isEquivalentTo aos:c_chemical_term
- lir:LexicalEntry isEquivalentTo aos:c_taxonomic_term
- lir:LexicalEntry isEquivalentTo aos:c_taxonomic_term_viruses
- lir:LexicalEntry isEquivalentTo aos:c_taxonomic_term_bacteria
- lir:LexicalEntry isEquivalentTo aos:c_taxonomic_term_plant
- lir:LexicalEntry isEquivalentTo aos:c_taxonomic_term_fungi
- lir:LexicalEntry isEquivalentTo aos:c_taxonomic_term_animals
- lir:Definition isEquivalentTo aos:c_definition

Class with attribute to class

- lir:LexicalEntry with partOfSpeech="noun" isEquivalentTo aos:c_noun

Relation to relation (without explicitly linking the inverses)

- lir:hasLexicalEntry isEquivalentTo aos:hasLexicalization
- lir:hasSynonym isEquivalentTo aos:hasSynonym
- lir:hasBroaderSynonym isEquivalentTo aos:hasBroaderSynonym
- lir:hasNarrowerSynonym isEquivalentTo aos:hasNarrowerSynonym
- lir:isRelatedTo hasPartialOverlapWith aos:hasRelatedConcept
- lir:isRelatedTo hasPartialOverlapWith aos:isRelatedTypeOf
- lir:hasTranslation isEquivalentTo aos:hasTranslation
- lir:hasDefinition isEquivalentTo aos:hasDefinition
- lir:hasVariant isEquivalentTo aos:hasTermVariant

[16] http://www.ttt.org/oscar/xlt/webtutorial/

- lir:isDialectalVariantOf hasHypernym aos:hasTermVariant
- lir:hasSpellingVariant isEquivalentTo aos:hasSpellingVariant
- lir:hasScientificName isEquivalentTo aos:hasScientificName
- lir:mainEntry isEquivalentTo aos:isMainLabel
- lir:hasTransliteration isEquivalentTo aos:hasTransliteration
- lir:hasAbbreviation isEquivalentTo aos:hasAbbreviation
- lir:hasAcronym isEquivalentTo aos:hasAcronym
- lir:hasFormula isEquivalentTo aos:hasChemicalFormula
- lir:hasNote isEquivalentTo aos:hasScopeNote
- lir:hasSource isEquivalentTo aos:hasSourceLink
- lir:hasStemmedForm isEquivalentTo aos:hasStemmedForm
- lir:hasSource isEquivalentTo aos:takenFromSource

relation to attribute

- lir:transliteration isEquivalentTo aos:hasTransliteration
- lir:dialectalVariant hasHypernym aos:hasTermVariant
- lir:scientificName hasPartialOverlapWith aos:hasScientificName
- lir:transliteration hasPartialOverlapWith aos:isTransliterationOf
- lir:acronym hasPartialOverlapWith aos:hasAcronym
- lir:abbreviation hasPartialOverlapWith aos:hasAbbreviation
- lir:formula hasHyponym aos:hasChemicalFormula
- lir:grammaticalNumber hasHyponym aos:hasPlural
- lir:grammaticalNumber hasHyponym aos:hasSingular

4.4.1 Automatic alignment of AOS and LIR

Having manually established these correspondences we used them as a gold standard for an automatic alignment experiment using the Alignment Server plug-in for the NeOn Toolkit, produced by INRIA.

For the alignment experiment we used three algorithms. They are variants of one alignment method: fr.inrialpes.exmo.align.impl.method.StringDistAlignment [14].

Each of the three algorithms below automatically produces alignments, a relation for each alignment (e.g. “=”, “>”, “<”), and a confidence score for each alignment.

Stringdist, which applies the abovementioned method without any parameters.

It only compares the labels and says 1. if they are the same and 0. otherwise.

SubdistName: this algorithm applies the parameter:stringFunction = subStringDistance. It compares the labels of classes and returns a a similarity the ratio of the longest common substring over the size of the longest string.

SMOAName
This one applies the parameter:stringFunction = smoaDistance. It compares the entity labels with the SMOA method descibed in the paper:
"A String Metric For Ontology Alignment", published in ISWC 2005

Also, we used a fourth alignment algorithm, which runs outside the NeOn ToolKit:
Aroma is a terminological matcher. It uses annotations, property values, and the taxonomy (the subclass relation) [24].

4.4.2 Evaluation

In order to evaluate the alignments produced by the four algorithms described above, we used a manually created gold standard. This gold standard contains all possible alignments between AOS and LIR (72 pairs; see Appendix A).

Overall, we introduced a certain degree of leniency from the start in the evaluation process. Whenever a pair was scored in the gold standard with a core “2” (reasonable) or “3” (rather bad) (rather than with “1”: good), the pair was still used as a valid gold standard element.

The results from the alignment operations were compared against this gold standard in order to judge their performance. In order to produce a detailed picture of the alignment performance, we distinguished four evaluation types:

Strict: the number of valid alignment types determined by evaluation/the number of alignment pairs produces by the algorithm.

Lenient: this implies the reduction of the number of found alignment pairs by determining a threshold for the strength value, below which no produced alignment is deemed valid.

Then the division of the number of valid alignment types (determined by evaluation) by the total number of alignment pairs produced with a strength value greater than the threshold yield the semi-lenient score.

Strict Recall: the number of gold standard pairs found by the algorithm.

Lenient Recall: sometimes an algorithm identifies only a subset of the alignments that belong to the same conceptual type of alignment. For instance, the five pairings in the table below have been defined within the gold standard, which all pertain to the same conceptual alignment regarding transliteration. None of the applied algorithms found all five alignments. Only one algorithm found three, whereas the others only identified one.

Nevertheless, for this measure we decide not to penalize algorithms for not having a 100% hit ratio for all variations of the same conceptual alignment. Therefore, we identified 20 such conceptual alignments within the gold standard table, which lists all possible variations. We then used these 20 alignment types for the evaluation.

Table 4.1. Variations of relations denoting transliteration

AOS	LIR
hasTransliteration	transliteration
hasTransliteration	isTransliterationOf
isTransliterationOf	transliteration
hasTransliteration	hasTransliteration
isTransliterationOf	isTransliterationOf

The results are as follows:

Algorithm: StringDist

All confidence values are equal to 1, so we did not determine any threshold.

12 out of 12 were deemed correct (100%)

The recall against the gold standard: 12/72=16.6%

Identified conceptual alignment types of the gold standard is 9/20=45%.

Algorithm: SubdistName

Strict: There were 28 valid alignment pairs in the set of 169 pairs produced by this algorithm, which means a score of 28/169= 16.6%.

Lenient: Below confidence score 0.206 this algorithm produced no hits.

We used this threshold to trim the alignments by removing all produced alignments with a lower confidence score lower than this. Of a total of 146 alignments, 28 were valid (19.2%).

Strict Recall: The recall against the gold standard: 28/72=38.8%

Lenient:**Recall**:

The recall against the conceptual alignment types of the gold standard is 16/20=80%.

Algorithm: SMOAName

Strict: There were 37 valid alignment pairs in the set of 85 pairs produced by this algorithm, which means a score of 37/85= 43.5%.

Lenient: Below confidence score 0.5 this algorithm produced no hits. We used this threshold to trim the alignments by removing all produced alignments with a lower confidence score lower than this. Of a total of 85 alignments, 31 were valid (36.5%).

Strict Recall: The recall against the gold standard: 37/72=51.4%

Lenient Recall: The recall against the conceptual alignment types of the gold standard is 17/20 = 85%.

Algorithm: Aroma

Strict: There were 12 valid alignment pairs in the set of 19 pairs produced by this algorithm, which means a score of 12/19= 63%.

Lenient: Below confidence score 0.8 this algorithm produced no hits. We used this threshold to trim the alignments by removing all produced alignments with a lower confidence score lower than this. Of the resulting 17 alignments, 12 were valid (70.5%).

Strict Recall: The recall against the gold standard: 12/72=16.6%

Lenient Recall: The recall against the conceptual alignment types of the gold standard is 10/20 = 50%.

Table 4.2. Overall performance of all alignment algorithms

Algorithm	Strict	Lenient	Strict Recall	Lenient Recall
StringDist	100%	100%	16.6%	45%
SubdistName	16.6%	19.2	38.8%	80%
SMOAName	43.5%	36.5%	51.4%	85%
Aroma	63%	70.5%	16.6%	50%

The overall results listed in table 4.2 above show that there is a trade-off between success rate of alignment and breadth of coverage. The algorithms that only cover a small percentage of the gold standard (StringDist and Aroma) have a good precision while producing the alignments. They do score rather badly in terms of strict and lenient recall. This means that from a conceptual alignment point of view, they are not to be preferred over the the other two (SMOAName and SubdistName). Of these two SMOAName scores consistently higher, and is therefore considered to be the best algorithm from this test set.

4.5 Automatic LIR population from AOS ontologies

An integral part of this deliverable is the programmatic functionality to convert the linguistic and terminological information within an AOS ontology into LIR format. The conversion service – dubbed Lirificator – implements a number of explicit atomic conversion steps that encode e.g. the creation of new instances of LIR classes, object properties and data type properties on the basis of AOS ontology individuals.

Figure 4.1 below illustrates alignments between the two ontology models at class level.

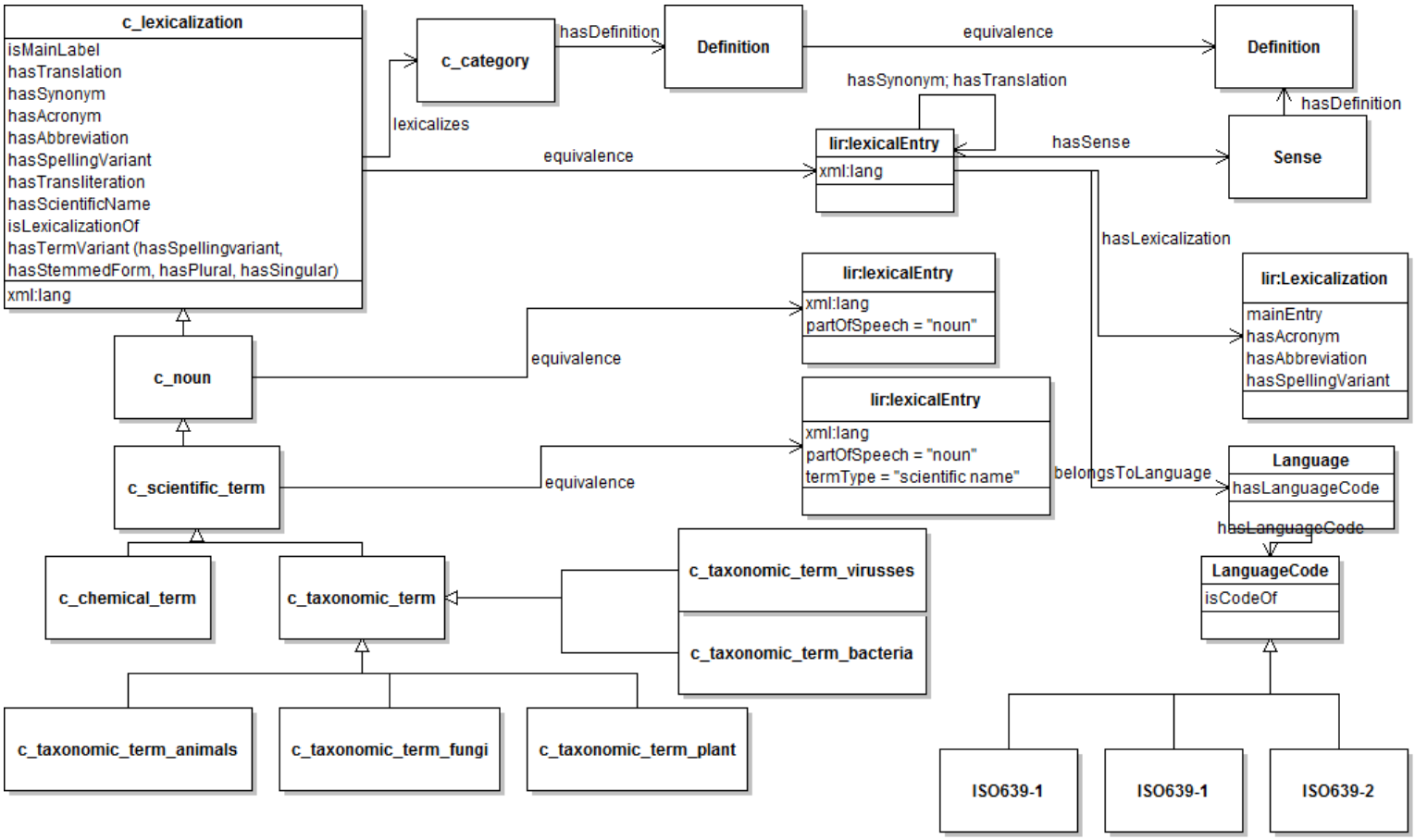

Figure 4.1. Alignment between LIR and AOS

The lirificator covers the following required steps for:

1. the conversion of both classes and properties from AOS to LIR;
2. the population of the LIR ontology above with AOS instances.

The left column lists the AOS ontology elements involved in this process.
The right column lists numbered population/conversion steps.

1. addition of name spaces from AGROVOC:
agrovoc <http://www.fao.org/aims/aos/agrovoc#>"
Asc <http://www.fao.org/aims/aos/Asc#>"
FaoPa <http://www.fao.org/aims/aos/FaoPa#>"

The left column lists the Agrovoc ontology elements involved in this process.
The right column lists numbered population/conversion steps.

for each instance of concept **c_noun (X)** or any of its descendants:

AGROVOC	LIR
	LEXICAL ENTRY 2. create instance of class LexicalEntry 3. copy value of xml:lang attribute of label of X as value of attribute xml:lang 4. create instance of object property isLexicalEntryOf domain:instance of class LexicalEntry (2) range: instance of AGROVOC class c_noun (X)

	5. create instance of data property partOfSpeech set value = "noun"
isLexicalizationOf domain: X range: c_category	6. create instance of object property hasLexicalEntry domain:c_category range: instance of class LexicalEntry (2)
	LEXICALIZATION 7. create instance of class Lexicalization 8. copy value of rdfs:label in AGROVOC as value of the rdfs:label attribute of (7)
Mainlabel="true/false"	9. create instance of data property mainEntry, with value "yes/no" corresponding to AGROVOC value 10. create instance of object property hasLexicalization domain: instance of LexicalEntry (2) range: instance of class Lexicalization (7) 11. create instance of object property isLexicalizationOf domain: instance of class Lexicalization (7) range: instance of LexicalEntry (2)
	SENSE 12. create instance of class Sense 13. create instance of object property hasSense domain: instance of LexicalEntry (2) range: instance of class Sense (12) 14. create instance of object property isSenseOf domain: instance of class Sense (12) range: instance of LexicalEntry (2)
if there is a "hasTranslation" relation between instances of c_lexicalization	TRANSLATION 15. create instance of object property hasTranslation domain: instance of LexicalEntry (2) range: instance of LexicalEntry
data property hasPlural	TERM VARIANTS 16. create instance of class Lexicalization 17. add label to 16 containing the value of the AGROVOC data property and the xml:lang from the AGROVOC c_noun concept (X) 19. create instance of object property hasPlural domain: instance of class Lexicalization (7) range: instance of class Lexicalization (16) 20. create instance of data property grammaticalNumber with domain = (16) and value "plural"
	21. create instance of object property hasSingular domain: instance of class Lexicalization (16) range: instance of class Lexicalization (7) 22. create instance of data property grammaticalNumber with domain = (7) and value = "singular"
data property hasSpellingVariant	23. create instance of class Lexicalization 24. add label to 23 containing the value of the AGROVOC data property and the xml:lang from the AGROVOC c_noun concept(X) 25. create instance of object property hasSpellingVariant

domain: instance of class Lexicalization (7)
range: instance of class Lexicalization (23)

26. create instance of object property hasSpellingVariant
domain: instance of class Lexicalization (23)
range: instance of class Lexicalization (7)

LANGUAGE

27. identify instance of class ISO639-1 with name equal to value of xml:lang in X's label

28. identify instance of object property http://www.fao.org/aims/aos/languagecode.owl#isCodeOf with domain = 27

29. create instance of object property belongsToLanguage
domain: 2
range: range of 28

30. create instance of object property belongsToLanguage
domain: 12 (instance of Sense)
range: range of 28

LEXICALENTRY SOURCE

31. create instance of class Source

32. create instance of data property nameSpaceIdentifier
domain: 31
value: url of X

32b. create instance of object property hasSource
domain: 2
range: 31

data property hasStemmedForm

33. create instance of Lexicalization

34. create instance of object property hasStemmedForm
domain: 7
range: 33

35. create instance of data property stemmedForm with boolean value = "yes"
domain: 33

35a. if there is a Lexicalization that is not yet the domain of the data property stemmedForm:
create instance of data property stemmedForm
Domain: Lexicalization
Boolean value=”no”

if concept == c_scientific term or any of its descendants

36. create instance of data property scientificName with value "yes"; else scientificName = “no”

36a Create instance of data property hasSemanticType
Domain: 2
Range: one of these URIs:

http://www.fao.org/aims/aos/agrovoc#{c_scientific_term;c_chemical_term;c_taxonomic_term; c_taxonomic_term_bacteria; c_taxonomic_term_fungi; c_taxonomic_term_animals; c_taxonomic_term_plant; c_taxonomic_term_viruses}

SYNONYMY

if there is a "hasSynonym" relation between instances of c_lexicalization or

37. create instance of object property hasSynonym
domain: instance of LexicalEntry (2)
range: instance of LexicalEntry (2)'

if two individuals are instances of the same concept

if there is a "hasBroaderSynonym" relation between instances of c_lexicalization

37a. create instance of object property hasBroaderSynonym
domain: instance of LexicalEntry (2)
range: instance of LexicalEntry (2)'

inverse: create instance of object property hasNarrowerSynonym
domain: instance of LexicalEntry (2)'
range: instance of LexicalEntry (2)

if there is a "hasBroaderSynonym" relation between instances of c_lexicalization

37b . create instance of object property hasNarrowerSynonym
domain: instance of LexicalEntry (2)
range: instance of LexicalEntry (2)'

inverse: create instance of object property hasBroaderSynonym
domain: instance of LexicalEntry (2)'
range: instance of LexicalEntry (2)

c_definition
hasDefinition
domain: c_category
range: c_definition

DEFINITION
38. create instance of Definition
39. create instance of object property hasDefinition
domain: 12
range:38

40. create instance of data property definitionText
domain: 38
value: c_definition

(hasSourceLink)
takenFromSource

DEFINITION SOURCE
41. create instance of class Source
42. create instance of data property nameSpaceIdentifier
domain: 41
value: value of takenFromSource attribute
43. create instance of object property hasSource
domain: 38
range: 41

hasScientificName

SCIENTIFIC NAME
44. create instance of object property hasScientificName
domain: 2
range: instance of LexicalEntry
44a. create instance of data property scientificName
Domain: instance of LexicalEntry
Value=yes

isScientificNameOf

44b. create instance of inverse property.

44c. create instance of data property scientificName
Domain: 7
Boolean value=yes

45. if there is a Lexicalization that is not yet the domain of a data property scientificName:
create instance of data property scientificName
domain: Lexicalization
boolean value=”no”

4.6 Structural alignments

Even if ontologies are conceptually equivalent, they often express their content in different ways, because they differ from each other in structural terms. For instance, a concept in one ontology can be expressed with a concept and an attribute in the other.

This structural mismatch has necessitated a step-wise conversion from AOS to LIR (discussed in the previous section).

In order to follow a more structured approach to these conversion problems, we have investigated an alternative way to express the informal alignments listed above at a higher level of generalization.

The grouping of the alignments defined in sections 4.2 to 4.4 form an initial typology that reflects structural differences. [17] defines a principled approach to the creation of a typology of alignment patterns against which this empirically defined typology of structural mappings can be mapped. The patterns identified in [17] have been collected into a library[17].

To recapitulate, the empirically identified typology on the basis of the informal alignments described in section 4.2 to 4.4 is the following:

1. Class to class
2. Class with attribute to class
3. Class to union of classes
4. Relation to relation
5. Relation to class
6. Relation to attribute
7. Attribute to attribute

Comparing this set of structural correspondences with the library of identified patterns yields the following results:

1. Class to class	EquivalentClass
2. Class with attribute to class	reverse of ClassByAttributeValue
3. Class to union of classes	ClassUnionCorrespondence
4. Relation to relation	SimpleRelationCorrespondence
	Sub-Super-RelationCorrespondence
	EquivalentRelationCorrespondence
5. Relation to class	
6. Relation to attribute	
7. Attribute to attribute	EquivalentAttributeCorrespondence

The pattern library covers five out of seven patterns. Patterns relating object properties (relations) to a class or attribute are still needed in order to cater for the alignments of the networked ontologies covered in this deliverable.

[17] http://www.omwg.org/TR/d7/patterns-library/

Future work will involve the formalization of the alignments intro these patterns, and the proposal of new patterns to cover the correspondence types 5 and 6.

5 Collaborative Ontology Localization

Ontology Localization is an emergent research topic to support the construction of multilingual ontologies. In previous works [5, 17], we have proposed the technological and methodological aspects to support this activity. However, the implications of localization in a concrete organization have not been examined to a satisfactory extent yet.

In this section, we propose an advanced version of LabelTranslator, our system to localize ontology terms into different natural languages, which differs from previous versions in that it uses a workflow-based model for a collaborative localization of ontologies. The current version of the system uses a centralized client-server architecture in which the server maintains the current state and full history of all localized ontologies.

5.1 Introduction

Multilinguality in ontologies is nowadays demanded by institutions worldwide with a huge number of resources available in different languages. Thus, for example, both use case partners, FAO[18] and the Spanish Pharmaceutical Industry, have expressed the need for semantically structuring the information they have in different natural languages.

Conscious of this problem, we have identified Ontology Localization as one of the activities in the ontology network development process to support the construction of multilingual ontologies. The Neon Glossary of activities [18] defines ontology localization as the activity that consists in *adapting an ontology to a particular language and culture*. Additionally, in [19] we have characterized the ontology localization problems and the different strategies for representing multilingual information in ontologies and for solving translation problems. Also, with the aim of guiding users in the development of multilingual ontologies, we have proposed some methodological guidelines.

Regarding ontology localization tools, the LabelTranslator [20,21,22], used in NeOn, is a system whose goal is to automatically localize an ontology to different natural languages. While this first prototype gave us an insight in the potential of this paradigm, it lacked a collaborative scenario. Basically, the previous implementation of LabelTranslator provides a single working scenario, where only one person performs all the major steps required for localizing an ontology to (and from) different natural languages. This scenario is feasible only in some cases. However, it is very difficult for a person to update all the linguistic information associated with a particular concept. We believe that this process will be done by different people from different countries at different time instants. Thus, the maintenance cycle for each language should be done separately. To address these limitations, we decided to add a workflow-based model for the collaborative localization of ontologies in distributed environments and describe the components required to support it.

In this deliverable we present our approach for the management of collaborative ontology localization in a distributed scenario by means of a workflow. In the first place, we analyze the case study at FAO as illustrating scenario of a collaborative localization process. As a result of that, we derive a set of features to support the process. Then, we focus on two common challenges found in early-stage ontology localization projects and analyse how to overcome them. To address these challenges cost-efficiently, we introduce a new approach. Our contribution also includes the implementation of the proposed approach.

The rest of this section is structured as follows. In section 5.2 we analyze the collaborative scenario at FAO and derive a set of characteristics to support a collaborative localization process. In section 5.3 we introduce our approach for the collaborative ontology localization on the features

[18] Food and Agricultural Organization.

derived in the previous section and present our implementation that provides the technological support to the presented models and methods.

5.2 Requirements for Collaborative Localization Process

In this section we present the most relevant requirements to support a collaborative ontology localization based on the analysis of the process (i.e. workflow) typically followed by organizations in the development and localization of ontologies. For our analysis, we considered existing processes for collaborative localization used in international institutions. We also took into consideration similar works in the state of the art, specifically in the field of software localization. As case study, we focused on the collaborative localization process followed at FAO for localizing different types of resources such as thesauri or ontologies. Finally, we discuss the features of these workflows and identify the core components of a workflow for collaborative ontology localization.

5.2.1 Use Cases for Collaborative Ontology Localization

AGROVOC thesaurus. AGROVOC is a multilingual thesaurus under development at FAO since 1983 (see also section 2.2.1). Originally, it was available only in printed copy, but since 2000 it is available online. The development of the thesaurus was done by several experts who met regularly in order to discuss how to enrich it. The collaboration was therefore achieved by face-to-face meetings. Nowadays, the localization (translation) is done in each partner country. Experts meet and organize themselves for the development of a version of the thesaurus in the national language. FAO provides guidelines[19] and tools for this task. Generally, FAO users can make use of the FAO AGROVOC current maintenance tool[20], but some countries also have their own tools. Once the national version is completed, it is sent to FAO for inclusion in the master copy of the thesaurus and for its online publishing. Figure 5.1 illustrates the localization workflow of the AGROVOC thesaurus.

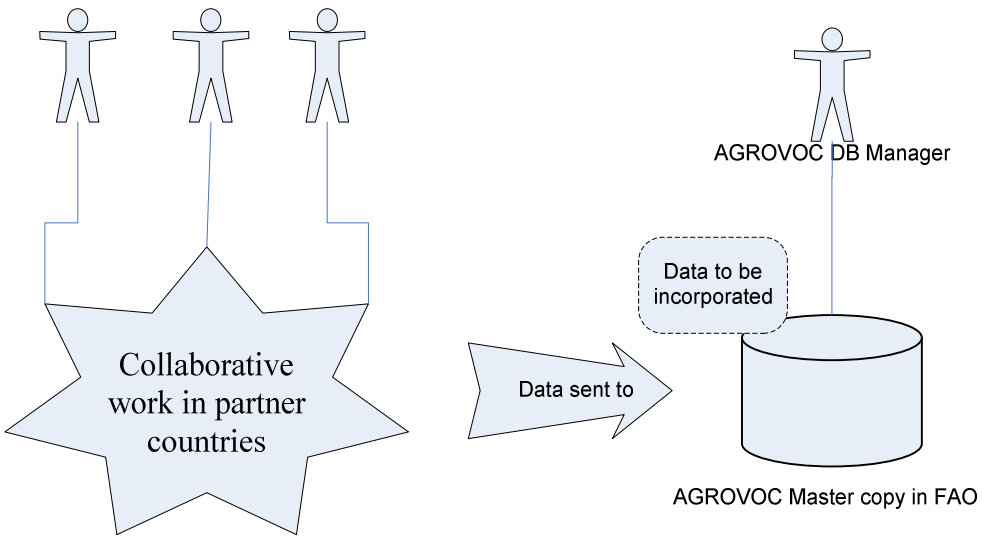

Figure 5.1. Localization workflow of AGROVOC thesaurus

[19] ftp://ftp.fao.org/gi/gil/gilws/aims/publications/papers/agrovoc_translation_guidelines.pdf
[20] http://www.fao.org/aims/tools_thes.jsp

Currently, new terms in several languages can also be added to AGROVOC by direct suggestions from users all over the world. Suggestions in multiple languages arrive to FAO through a specific web-form or per email. Suggestions are then evaluated and committed to the ontology by the FAO AGROVOC team (see Figure. 5.2).

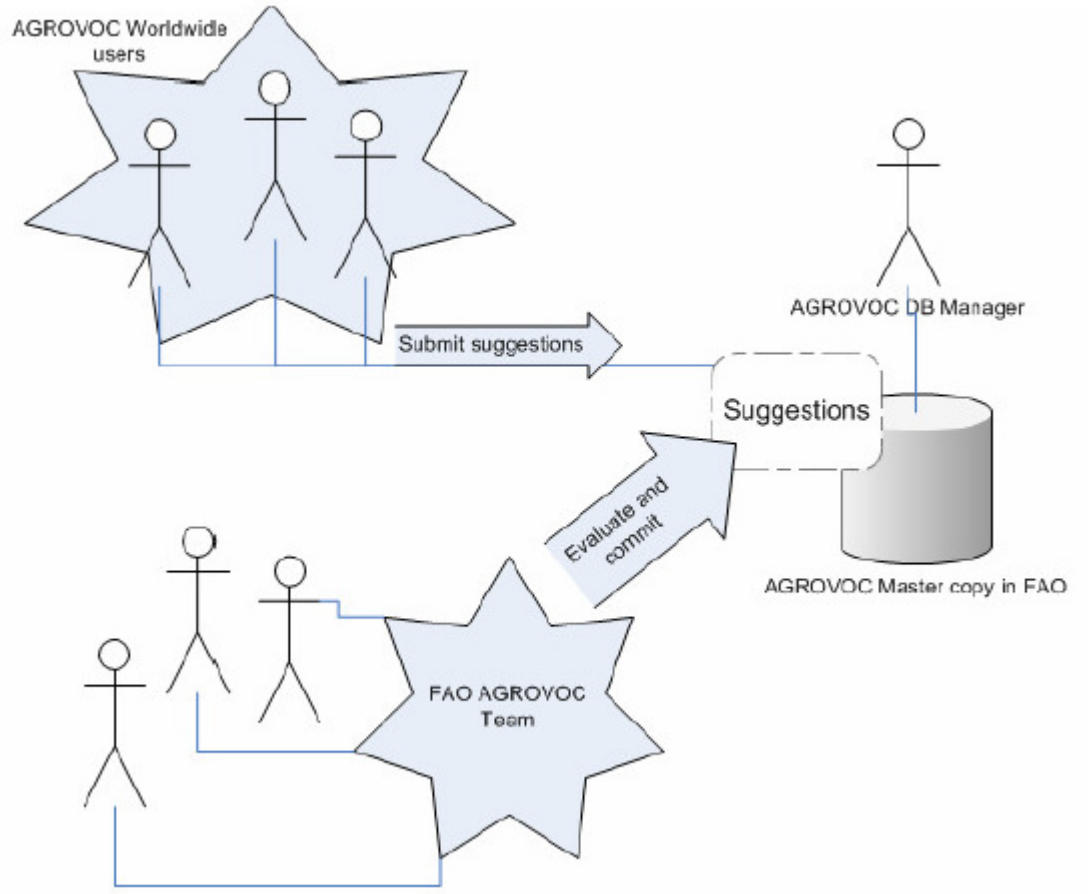

Figure 5.2. Current Workflow for Collaborative Localization of AGROVOC thesaurus

During this year, FAO has promoted a virtual conference, the so-called the AGROVOC E-conference. This has been recognized as an effective method for further collaborations between AGROVOC experts and users all over the world. This technique did not require any working mission or trips by participants, with the consequent save of money and time, and allowed asynchronous participations, giving in that way time to users to reply when and from where was more convenient for them. In this case, collaboration is achieved thorough Internet as a mean for communication and organization of the work.

We have seen how both activities, creation and localization of the thesaurus, involved many actors who needed strict collaboration for the sequential actions that are involved:

- First, the creation of terms involves many domain experts, whose task is to identify which the best term to use is, and which terms have to be created as non-descriptors;
- Then, the translation of terms involves discussions between translators and domain experts in order to evaluate and define the better translations of terms;

Therefore, in the collaborative aspects of the creation and localization of an ontology we need to consider:

- collaboration over different steps performed by different people, and

- collaboration among several participants for every single step

AGROVOC Concept Server (ontology). A different use case for collaborative localization used at the FAO is the AGROVOC Concept Server, and ontology developed on the basis of the AGROVOC thesaurus. The development of the AGROVOC Concept Server Workbench was based on: i) the need of making the development and maintenance of the AGROVOC thesaurus more collaborative and especially more direct for users without the intermediate actions of FAO staff, and ii) the idea to convert AGROVOC into a more complete structure allowing the representation of more information (such as additional linguistic information, or the ability to have multiple translations for a specific term, etc.).

AGROVOC Concept Server Workbench allows collaborative and distribute management of the new restructured AGROVOC. Figure 5.3 shows the new workflow. The new collaborative features are summarized in the following:

- AGROVOC editors all over the world can have direct access to a unique and homogenized maintenance tool;
- changes are immediate and there is no need to wait for FAO actions (eventually only validations of proposed editions);
- all users can immediately see and benefit from other users contributions;
- the cycle of adding data to AGROVOC and reuse it in their systems is more immediate, because once the data has been inserted in the system, and eventually validated, it becomes immediately available for remote access through web services or can be immediately downloaded (while currently publications of the updated AGROVOC are done every three months online or in the ftp area).

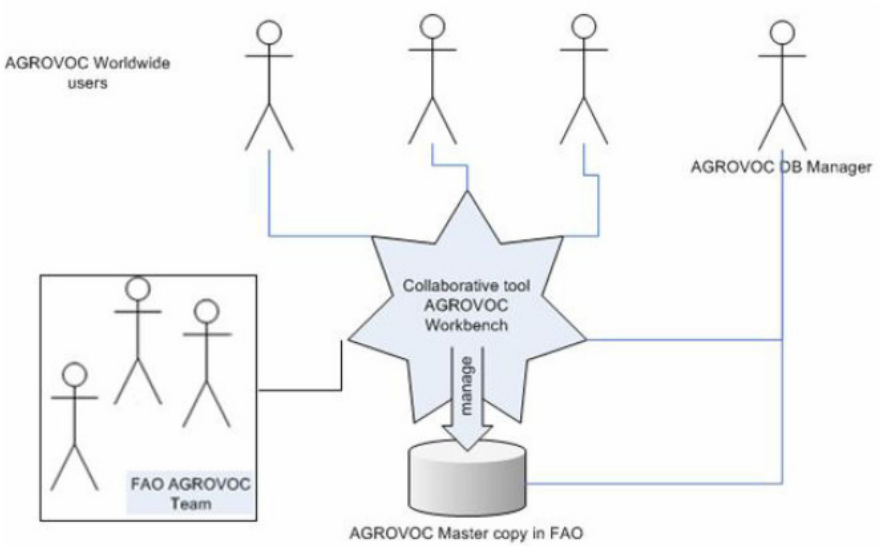

Figure 5.3. Workflow for Collaborative Localization of AGROVOC Ontology

5.2.2 Main Features for Supporting Collaborative Ontology Localization

In this section we discuss the main features of the workflows described in the previous section. Then, we identify the core components to support collaborative ontology localization. Some of the components that we introduce in the following are similar to the ones that have been already

identified in related areas (e.g software localization). However, in this work, we extend these features to coordinate efficiently the collaborative ontology localization. When appropriate, we illustrate the different features using FAO use cases.

Human-centred workflows. The main common thread for the process that we described in the previous sections is that many steps in these workflows require human actions. Human-centred workflows are different from service workflows that combine software services for automatic execution. Furthermore, we believe that most activities in the workflows should be described in terms of the steps that a proposed translation must go through in order to be incorporated into the ontology, or in terms of the states that an ontology label (to be localized) has to go through before it is published.

User management and provenance of information. With multiple users contributing to the localization of an ontology, it is critical for users to understand where information is coming from. Thus, users must be able to see how project participants reach consensus on ontology label translations, who can perform translations, who can comment on them, when ontology label translations become public and so on.

Our main contributions to the user management and provenance of information is the integration and implementation of a workflow process to manage the sequence of translation/review/edit tasks, providing the status of tasks and processes, and notifying participants the changes in state, the new work, or other information.

Collaborative Workflow Support. As we have described in the use cases of FAO, a typical localization project involves several steps that extend far beyond the translation process itself. To start with, more people are involved in the process. In fact, on most large localization projects today, localization is a collaborative effort, where the number of users participating in localization ranges from a handful to a couple of dozens.

With larger groups of users contributing to ontology localization, we believe that it is necessary define appropriate workflows, strategies and an infrastructure to support the process that coordinates the collaborative ontology localization within an organizational setting. Basically, we have identified three actors: the localization manager, translators, and reviewers that support the localization of the ontology.

The Ontology Localization Workflow that we propose is designed to support all aspects of the translation and ontology localization project. Figure 5.4 below shows a schematic view of the workflow:

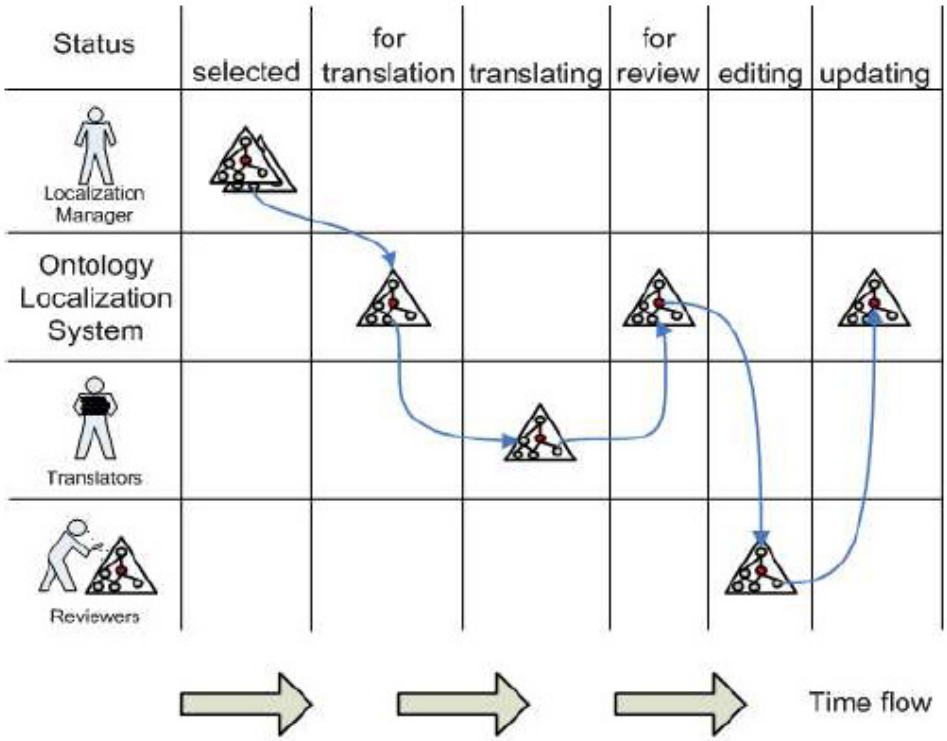

Figure 5.4. Workflow process used to localize an ontology

- An ontology is passed to the Localization Manager for localization.
- The Localization Manager selects the ontology labels to be localized and sends them for translation.
- A translator downloads the selected labels to be localized and (s)he performs the translation task.
- The same translator uploads the translated ontology labels and sends them for review.
- The reviewer downloads the translated labels and checks for possible errors.
- Finally, the Localization System updates all the linguistic information related to each localized label.

Our contributions to collaborative ontology localization can be summarized in the following points:

- Design of a formal model based on collaborative workflows for the representation of the process usually followed by different organizations.
- Development of architecture for supporting the task assignment (customizable workflows) of the collaborative ontology localization.
- Development of an interface that allows users to perform the different tasks of the localization process collaboratively.

Flexible Localization Management. The most obvious element of complexity in the use cases that we described in the previous section is the number of parties involved in it and their geographic distribution (see the typical localization scenario in Figure 5.5). The geographic

distribution of the parties is motivated by the fact that localization in most cases requires in-country reviews to check the content of the translation.

Managing a large number of parties presents a lot of challenges. However, in this work we focus only on two challenges commonly found in international organizations: communication and version tracking.

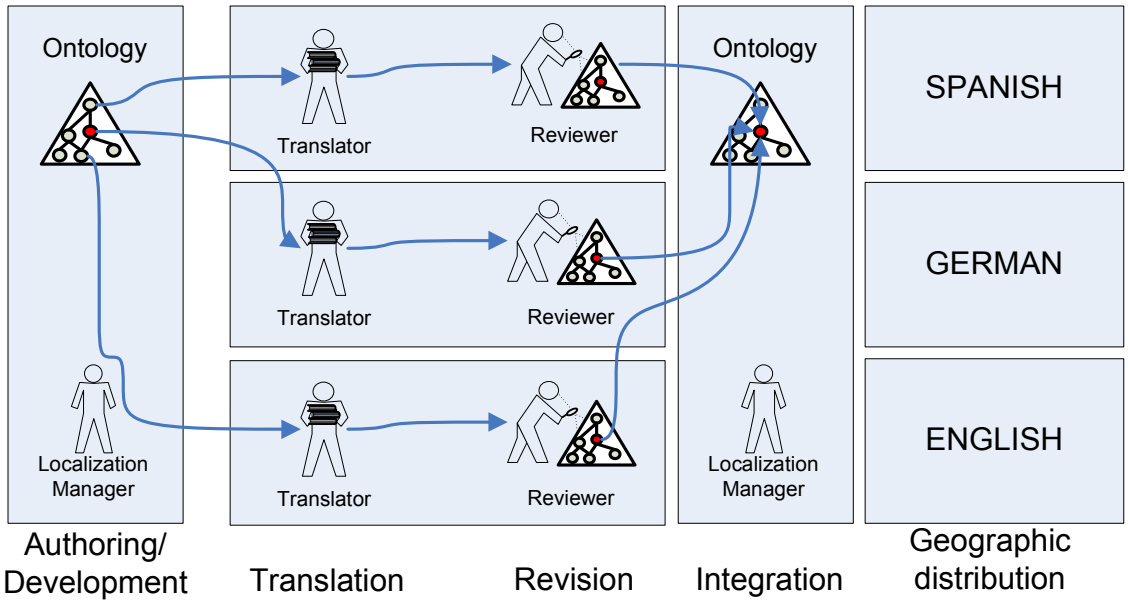

Figure 5.5. Typical Localization Project

Thus, for example, in order to reach consensus on a specific action to be performed in the AGROVOC thesaurus, the most used mechanism is email exchange among AGROVOC experts. Because it is simple and readily available, e-mail is usually the primary tool for communicating, which includes handing off projects and files that contain versions of documents, as well as tracking status.

However, the use of e-mail frequently results in confusion over which version of which document is current, the status of each document, and who is currently doing what, all of which leads to inefficiency and waste of time. The second main source of complexity is due to the high number of documents that need to be distributed across project participants. This situation is even further complicated when changes are made to the original texts during the course of the project. These late modifications need to be introduced manually into the translation chain and lead to the existence of multiple versions of the same set of files, which in turn leads to frequent errors and substantial management overhead. In the case of the FAO, a unique and homogenized maintenance tool for the collaborative management of AGROVOC is employed.

Basically, in this work we adopt a similar approach that uses a workflow model integrated with a content management component for providing greatest flexibility and power. Our contributions about providing an efficient mechanism to localization management can be summarized in the following points:

- Design of an architecture for managing versions of localized ontologies, controlling ontology access (through some form of check in/check out and file locking), and enabling remote or distributed access.

- Design and implementation of a common repository for all the components of the ontology localization work. The repository is used to manage both work in progress as well as completed or archived localized ontologies.

The flexibility and wide coverage of the ffeatures above described are the main characteristics of our approach to support collaborative ontology localization. This will be described in more detail in Section 5.3.

5.3 Architecture of Collaborative LabelTranslator System

In this section we present our solution to support collaborative ontology localization, and how it meets the aforementioned requirements. Figure 5.6 illustrates the main components of the Collaborative LabelTranslator system. Firstly, we introduce the components that provide the foundations to represent the required information in our solution. Secondly, we present the implementation support.

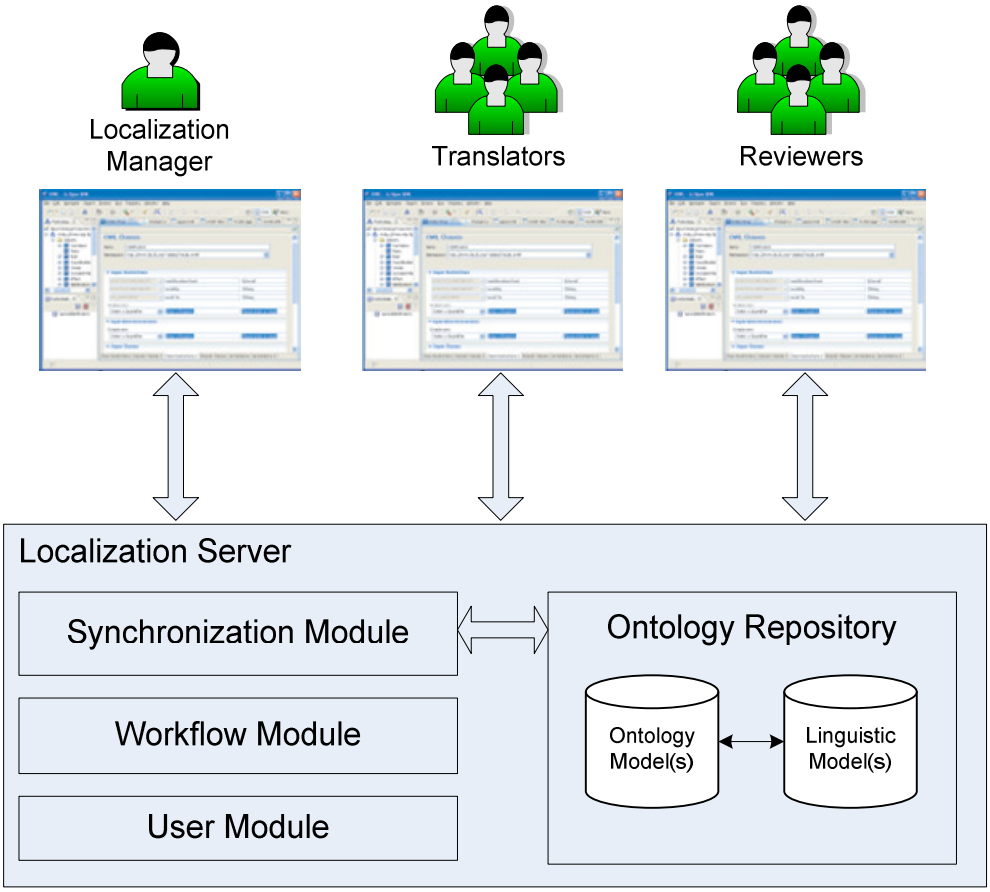

Figure 5.6. A Typical Localization Project

5.3.1 Ontology Repository

The Collaborative LabelTranslator has an ontology repository that contains all the ontologies that users can localize in the collaborative mode. LabelTranslator follows the current trend in the integration of multilinguality in ontologies, which suggests the suitability of keeping ontology knowledge and linguistic (multilingual) knowledge separately and independently. Thus, the

ontology repository relies on the combination of two independent modules, the ontological and the linguistic one.

In D2.4.2 we presented the module for managing the conceptual knowledge and the linguistic knowledge by means of synchronization techniques. Hence, in this deliverable we briefly highlight the main features. Addition of new terms in the ontology, or deletion of an existing term is controlled by a mechanism of change synchronization. In the NeOn ToolKit, an advanced change tracking based on Resource Delta2 is able to capture changes even when ontological terms have changed their position within the ontology model. By adopting this feature, our system can accurately identify the minimal set of changes needed to adjust the structure of the linguistic model, a critical first step to ensure that a matching change is made in the localized ontology. Figure 5.7 illustrates the localization management used in our system to synchronize the conceptual and linguistic information. In the following we analyze the process in more detail, describing the actions performed by each actor of our scenario.

- *Ontology Expert (Localization Manager).* (S)he is responsible for editing the changes in the ontology model. All the changes executed in each user session are stored in the server as a new version. The types of changes that our system can manage are the following: changes of the label content (e.g., ontology label rename) and ontology structure changes (e.g., delete or add operations). For each case, LabelTranslator stores the type of operation executed and its additional information (e.g., the name of the renamed label). This information is used in our system to synchronize the conceptual and linguistic information.
- *Translators or Reviewers (Linguist experts)*. Linguist experts in a specific target language are responsible for performing the localization process (to translate or to review ontology labels). Notice that this process always uses the last version of an ontology. When the linguist needs to update the linguistic model (LM), our system tries to synchronize both models, performing the following actions: (1) obtaining the current version of the LM to be updated, (2) extracting the last version of the changes in the ontology model (OM) from which the last localization was taken (normally the one with the same number as the LM), (3) performing all the actions of the file of changes in the LM, and (4) updating the LM version in the server repository.

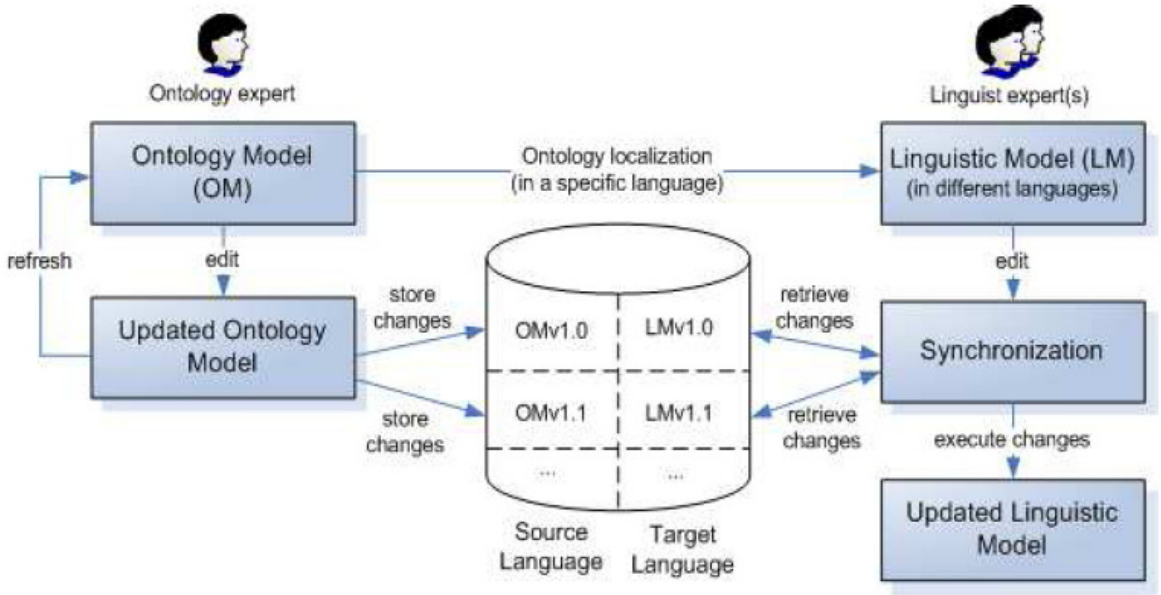

Figure 5.7. A Typical Localization Project

5.3.2 Workflow Module

The workflow module describes the work scenario used to support the localization of an ontology. Each workflow is associated with a set of initialization parameters, source and target languages, and a partially ordered set of activities or states.

Our system supports two scenarios: single and collaborative scenario. In the first scenario, there is only a person impersonating the project manager, translator, and reviewer all at the same time. As such, (s)he will have to perform all organizational and translation tasks. While in the collaborative scenario, there is a team sharing the translation work. This requires a higher volume of organisational work. In this case, one of the participants should be appointed to assume the responsibility of project management.

Furthermore, based on the analysis of the requirements presented in the previous section, our solution considers the localization workflow at two levels: ontology term and ontology label translation level. Although the workflows can be used independently of the underlying ontology model, the specific set of ontology terms depends on the ontology model. In our approach, we are mainly considering the OWL ontology model, in which an OWL ontology consists of a set of axioms and facts. Facts and axioms can relate to classes, properties or individuals, and hence that is the set of ontology elements we are taking into account.

- *Ontology Term status*. Each ontology term (to be localized) has a status that is automatically assigned by the system. Possible values are:

Status	***Description***
In use	Ontology term is in its normal state
New	Ontology term has been added to the ontology since creation
Changed	Either the context value or the original value of the Ontology term have changed
Unused	Ontology term is not to be found in the ontology any more

- *Ontology Label Translation status.* Each label translation has a status. Possible values are:

Status	**Description**
Not translated	No translation is given
Auto translated	Ontology label has been automatically translated using the translation algorithm provided by the tool
Translated	Ontology label has been translated but not verified yet
For review	Ontology label has been translated but should be reviewed
Complete	Translation has been marked completed. Use this when you have checked the translation and found it to be final.

Note that during the collaborative workflow, actions are performed either implicitly or explicitly. For instance, when a user updates (i.e. modifies) an ontology label, he does not explicitly perform an update action. In this case the action has to be captured from the user interface and recorded

when the ontology is saved. In contrast, Reviewers explicitly approve/reject proposed translations and the action is recorded immediately when performed.

5.3.3 User Module

This component allows managing the profile of each participant of the localization activity. The module records information about the skills of each participant (source and target languages), and describes the roles, operations and policies that apply to a certain ontology.

LabelTranslator uses this module for checking the users' credentials at login time, and for determining whether a user is allowed to perform a certain operation based on the policies of the ontology to be localized. A user can play several roles in the localization of an ontology. For example, a user Elena can play the role of Translator and of Reviewer.

Key benefits of the User Module include:

- *Improved Project Staffing.* The user module allows Localization Managers to see all the information related with a participant (e.g. language skills). This saves time and allows for better decisions when staffing an ontology localization project.
- *Shared User Database Across Ontology Projects***.** The user module provides a particular benefit for ontology projects that need to localize several ontologies. The module allows different ontology projects maintaining a single user database, which allows to share users in different projects.

5.3.4 Implementation

Our approach has been implemented within the NeOn Toolkit, an extensible ontology engineering environment based on Eclipse, by means of a set of plugins and extensions. In the scenario shown in the Figure 5.6, the team of ontology users (localization manager, translators, and reviewers) will be collaboratively localizing the ontology following a well defined process (i.e. workflow). The following characteristics describe the environment:

- There is only one copy of the ontology (ontology model) which is stored in a central server. Additionally, each ontology user has its own copy located at his/her PC.
- Ontology users are working in a distributed manner (i.e. they are physically located at different PCs).
- Ontology users are working concurrently.
- Each ontology user uses his/her own NeOn Toolkit installation and connects to the central server.
- Each ontology user specifies his/her credentials (e.g. name and role) in the NeOn Toolkit.
- The translations of an ontology label from all translators are stored at one specific place.
- Each NeOn Toolkit has configured that label translations are stored at a specific location (i.e. a specific localization server) and connects to it.

Configuration of machines

- One PC is configured as the server. This PC has to be running the Localization Server.
- The PCs used by the ontology users are only running a NeOn Toolkit installation configured as described above.

User Interface

The graphical user interface of the plugin is composed of different views (see figure 5.8) for the configuration of the ontology localization process and the presentation of the results. The ontology Navigator in Figure 5.8 is locates on the left side of the main view. It contains all created/imported ontology projects. Each ontology project can contain one or more ontologies to be localized. An ontology configured for localization contains all ontology terms that have been added to the ontology localization project.

The localization view is located on the middle of the main view. The view is used to enter ontology label translations and show all elements of the item (e.g.: classes object properties or data properties) that has been selected in the project tree. Each ontology term is located in its own row. The localization view contains several shortcuts that make work faster, and are enabled according to user profile.

The filter view is located on the right side of the localization view. It contains several check box that user can check and uncheck to modify what items are shown in the localization view.

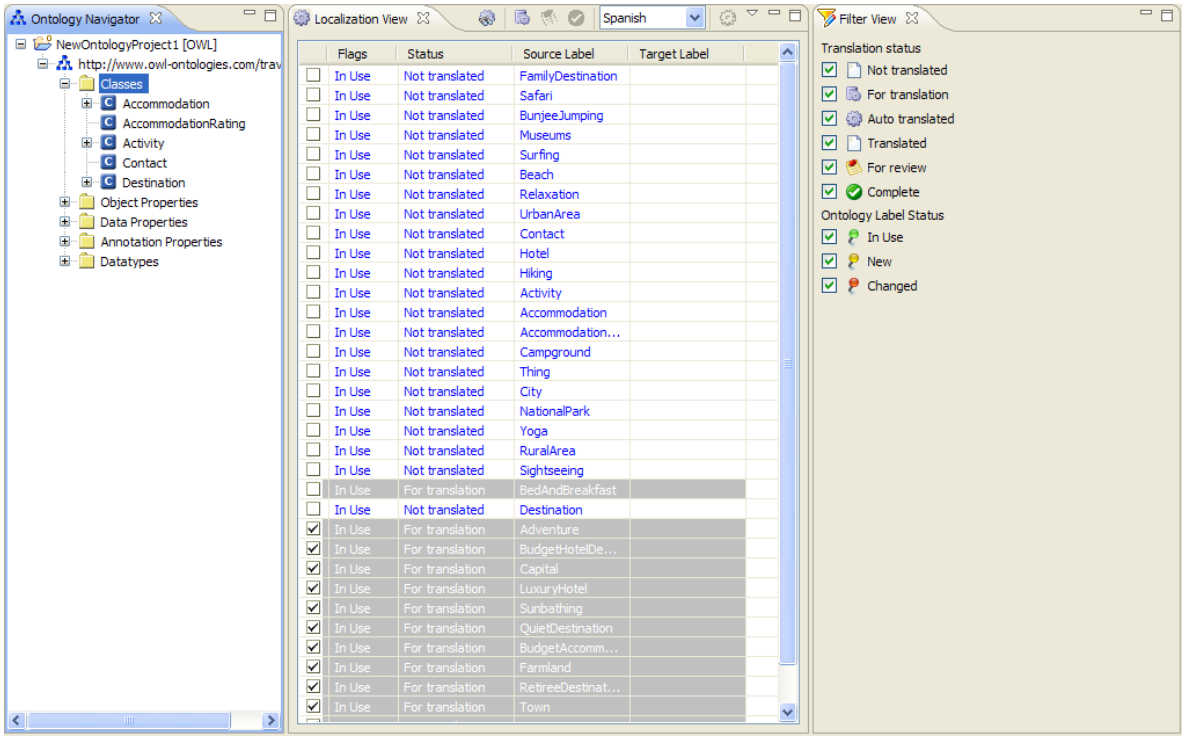

Figure 5.8 A Typical Localization Project

6 Conclusions and Future Work

In this deliverable we have argued that the LIR is not a stand-alone model. It is intricately related to existing standard models describing linguistic and terminological knowledge. We have re-engineered some of these models, more precisely standard models for translation memory and the LMF standard model for linguistic representation. We have described the embedding of the LIR into this network.

In order to maintain interoperability with FAO's AOS model, we have integrated this model into this network by aligning it with the LIR. Also, we have implemented an automatic conversion of linguistic information in AOS format to LIR representation.

Furthermore, the localization process makes use of LabelTranslator. We have presented additional functionality to LabelTranslator addressing workflow management for collaborative localization.

Future work within NeOn will include:

1. The embedding of additional standard models into the ontology network:
    - ISOCAT data category registry

the ISO Technical Committee 37, "Terminology and other Language and Content Resources"[21], is developing a Data Category Registry (DCR) [0,0]. This registry will provide a reusable set of (standardized) data denoting linguistic concepts that cover a range of linguistic domains. The concepts in the DCR can be referenced to from all sorts of tools and resources. Therefore, the DCR acts as an intermediate between those tools and resources. is foreseen that researchers can seamlessly access and combine resources and services offered by various service and repository centres.

    - LexInfo [25] This model adopts LMF mainly for morphological and syntactic processing. Linking LIR to this model will enhance the coverage and reusability of linguistic description complementary to LIR, and define interoperability through LMF overlap.
    - LexOnto [26] focuses on representing subcategorization frames as well as their mapping to ontological structures. This alignment will also provide linguistic information complementary to LIR's coverage.

2. The formalization of the informal alignment relations defined in section 4. This will entail the choice of relation types, which should describe at least the C-Owl relations equivalence, hypernymy, hyponymy and partial overlap [0]. Further, the link to more fine-grained alignment language, as initiated in section 4.6, will be worked out further.

[21] http://en.wikipedia.org/wiki/ISO/TC37

References

[1] Picca, D., Gangemi, A., and Gliozzo, A. (2008). LMM: an OWL Metamodel to Represent Heterogeneous Lexical Knowledge. In Proc. of the International Conference on Language Resources and Evaluation (LREC), Marrakech, Morocco. ACL.

[2] Bouquet, P. Giunchiglia, F. van Harmelen, F. Serafini, L. and Stuckenschmidt, H. (2004) *Contextualizing Ontologies*. Journal of Web Semantics 1 (4), 2004.

[3] Pazienza, M.T. and A. Stellato, A, Clustering of terms from translation dictionaries and synonyms lists to automatically build more structured Linguistic Resources. 6th Language Resources and Evaluation Conference (LREC 2008). Marrakech (Morocco). May 2008.

[4] Montiel-Ponsoda, E. and Peters, W. (coordinators): Multilingual Ontology Support. NeOn Project Deliverable 2.4.1 (2007)

[5] Montiel-Ponsoda, E. and Peters, W. (coordinators): Multilingual Ontology Support. NeOn Project Deliverable 2.4.2 (2008)

[6] Francopoulo, G, George, M. Calzolari, N. Monachini, M. Bel, N. Pet, M. Soria, C.: Lexical Markup Framework (LMF) In: Proc. of the International Conference on Language Resources and Evaluation (LREC) , Genoa, Italy (2006)

[7] Collin F. Baker, Charles J. Fillmore, and John B. Lowe (1998), The berkeley framenet project. In Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics.

[8] Charles Sanders Peirce. 1958. Collected Papers of Charles Sanders Peirce. MIT Press, Cambridge, Mass.

[9] C. Fellbaum (ed.) *WordNet: An Electronic Lexical Database*. Cambridge: The MIT Press. 1998.

[10] A. Gangemi. 2008. Norms and plans as unification criteria for social collectives. Journal of Autonomous Agents and Multi-Agent Systems, 16(3).

[11] A Gangemi, N Guarino, C Masolo, and A Oltramari. 2002. Sweetening ontologies with dolce. Proceedings of EKAW, Jan.

[12] N. Ide, L. Romary. *A Registry of Standard Data Categories for Linguistic Annotation*. Proceedings of the 4th Language Resources and Evaluation Conference (LREC 2004), Lisbon, 2004.

[13] M. Kemps-Snijders, M. Windhouwer, P. Wittenburg, and S.E. Wright. *ISOcat: Corralling Data Categories in the Wild*. Proceedings of the 6th Language Resources and Evaluation Conference (LREC 2008) Marrakech (Morocco). May 2008.

[14] J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer-Verlag Berlin Heidelberg, 2007.

[15] Villazón-Terrazas, B. (coordinator), Methods and Tools Supporting Re-engineering, NeOn Project Deliverable 2.2.2 (2008)

[16] M. C.Suárez-Figueroa (coordinator). Revision and Extension of the NeOn Methodology for Building Contextualized Ontology Networks. Neon Project Deliverable 5.4.2 (2009).

[17] F. Scharffe, J. Euzenat, and D. Fensel. *Towards design patterns for ontology alignment*. In R.L. Wainwright and H. Haddad (eds.): Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil, March 2008: 2321-2325.

[18] M. C. Suárez-Figueroa and A. Gómez-Pérez. First attempt towards a standard glossary of ontology engineering terminology. In Proc. of 8th International Conference on Terminology and Knowledge Engineering (TKE'08), 2008.

[19] M. Espinoza, E. Montiel-Ponsoda, and A. Gómez-Pérez. Ontology localization. In The Fifth International Conference on Knowledge Capture (K-CAP'09), California, USA, 2009.

[20] M. Espinoza, A. Gómez-Pérez, and E. Mena. Enriching an ontology with multilingual information. In Proc. of 5th European Semantic Web Conference (ESWC'08), Tenerife, (Spain), June 2008.

[21] M. Espinoza, A. Gómez-Pérez, and E. Mena. Labeltranslator - automatically localizing an ontology. In Proc. of 5th European Semantic Web Conference (ESWC'08), Tenerife, (Spain), June 2008.

[22] M. Espinoza, A. Gómez-Pérez, and E. Montiel-Ponsoda. Multilingual and localization support for ontologies. In Proc. of 6th European Semantic Web Conference (ESWC'09), Heraklion, (Greece), June 2009.

[23] Grau,B.C., Parsia, B. and Sirin, E., Combining OWL ontologies using E-connections, Journal of Web Semantics, Vol. 4, No. 1, pages 40-59. 2006.

[24] Jerome David, J., Guillet, F., Briand, H. (2007), Association Rule Ontology Matching Approach. Int. J. Semantic Web Inf. Syst., 2007: 27~49

[25] Buitelaar, P., Cimiano, P., Haase, P., Sintek, M. (2009), Towards Linguistically Grounded Ontologies.
In: proceedings of ESWC 2009, pp. 111-125

[26] Cimiano, P. Haase, P. Herold, M. Mantel, M. Buitelaar, P. (2007), LexOnto: A Model for Ontology Lexicons for Ontology-based NLP
In: proceedings of Ontolex2007, Busan, Korea

Appendix A: FAO evaluation gold standard

This table contains all possible alignments between the AOS and the LIR models. It illustrates the gold standard used in the alignment experiment described in section 4.3.

Column “evaluation-score”: the alignments are scored according to the conceptual match between the members:

1: good; 2: reasonable; 3: rather bad

Column “equivalence-type” indicates whether the pair members are fully equivalent or whether they are linked through a subsumption relation. The following symbols are used:

“=”full equivalence

"<" (subclassOf)

">" (superclassOf)

The numbers in column “group” represent the conceptual alignment types discussed in section 4.3.2.

gold_standard_list					
id	**aos**	**lir**	**evaluation_score**	**equivalence_type**	**group**
1	hasLexicalization	hasLexicalEntry	1	=	1
2	isLexicalizationOf	isLexicalEntryOf	1	=	1
3	hasSynonym	hasSynonym	1	=	2
4	hasBroaderSynonym	hasSynonym	1	<	2
5	hasNarrowerSynonym	hasSynonym	1	<	2
6	hasRelatedConcept	isRelatedTo	1	=	3
7	isRelatedTypeOf	isRelatedTo	1	<	3
8	hasTranslation	hasTranslation	1	=	4
9	hasTranslation	isTranslationOf	1	=	4
10	hasDefinition	hasDefinition	1	=	5
11	isDefinitionOf	isDefinitionOf	1	=	5
12	hasTermVariant	hasVariant	1	=	6
13	hasTermVariant	dialectalVariant	1	>	6
14	hasTermVariant	isDialectalVariantOf	1	>	6
15	hasSpellingVariant	hasSpellingVariant	1	=	6
16	hasScientificName	hasScientificName	1	=	8

gold_standard_list					
id	aos	lir	evaluation_score	equivalence_type	group
17	hasScientificName	scientificName	1	=	8
18	isMainLabel	mainEntry	1	=	9
19	hasTransliteration	hasTransliteration	1	=	10
20	hasTransliteration	transliteration	1	=	10
21	isTransliterationOf	transliteration	1	=	10
22	hasAbbreviation	hasAbbreviation	1	=	11
23	hasAbbreviation	abbreviation	1	=	11
24	isAbbreviationOf	abbreviation	1	=	11
25	isAcronymOf	isAcronymOf	1	=	12
26	isAcronymOf	acronym	1	=	12
27	hasAcronym	acronym	1	=	12
28	hasAcronym	hasAcronym	1	=	12
29	hasChemicalFormula	formula	1	<	13
30	hasChemicalFormula	hasFormula	1	<	13
31	hasChemicalFormula	isFormulaOf	1	<	13
32	hasScopeNote	hasNote	1	=	14
33	hasScopeNote	noteText	1	=	14
34	hasSourceLink	hasSource	1	<	15
35	c_lexicalization	LexicalEntry	1	=	16
36	c_noun	LexicalEntry	1	<	16
37	c_scientific_term	LexicalEntry	1	<	16
38	c_chemical_term	LexicalEntry	1	<	16
39	c_taxonomic_term	LexicalEntry	1	<	16
40	c_taxonomic_term_viruses	LexicalEntry	1	<	16
41	c_taxonomic_term_bacteria	LexicalEntry	1	<	16
42	c_taxonomic_term_plant	LexicalEntry	1	<	16
43	c_taxonomic_term_fungi	LexicalEntry	1	<	16
44	c_taxonomic_term_animals	LexicalEntry	1	<	16
45	c_definition	Definition	1	=	17

gold_standard_list					
id	aos	lir	evaluation_score	equivalence_type	group
46	hasStemmedForm	hasStemmedForm	1	=	18
47	hasPlural	grammaticalNumber	1	<	19
48	hasSingular	grammaticalNumber	1	<	19
49	takenFromSource	hasSource	1	=	20
50	isTransliterationOf	isTransliterationOf	1	=	10
51	hasTransliteration	isTransliterationOf	1	=	10
52	isAbbreviationOf	isAbbreviationOf	1	=	11
53	isAbbreviationOf	hasAbbreviation	1	=	11
54	hasScientificName	isScientificNameOf	1	=	8
55	isScientificNameOf	hasScientificName	1	=	8
56	isScientificNameOf	isScientificNameOf	1	=	8
57	isDefinitionOf	hasDefinition	1	=	5
58	hasDefinition	isDefinitionOf	1	=	5
59	hasTermVariant	hasSpellingVariant	1	>	6
60	c_lexicalization	Lexicalization	2	<	1
61	hasImageSource	hasSource	3	<	20
62	hasRelatedTerm	isRelatedTo	2	=	3
63	hasRelatedType	isRelatedTo	1	=	3
64	isLexicalizationOf	isLexicalizationOf	3	>	1
65	isScientificNameOf	scientificName	1	=	8
66	hasLexicalization	hasLexicalization	3	>	1
67	isLexicalizationOf	isLexicalizationOf	1	>	1
68	hasDefinition	definitionText	3	<	5
69	hasStemmedForm	hasShortForm	3	=	18
70	hasSynonym	isSynonymOf	1	=	2
71	hasTermVariant	hasDialectalVariant	2	>	6
72	isAbbreviationOf	isAbbreviationFor	1	=	11

Appendix B: TMX-core.owl

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
   <!ENTITY b 'http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#'>
   <!ENTITY kaon2 'http://kaon2.semanticweb.org/internal#'>
   <!ENTITY owl 'http://www.w3.org/2002/07/owl#'>
   <!ENTITY owlx 'http://www.w3.org/2003/05/owl-xml#'>
   <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
   <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
   <!ENTITY ruleml 'http://www.w3.org/2003/11/ruleml#'>
   <!ENTITY swrl 'http://www.w3.org/2003/11/swrl#'>
   <!ENTITY swrlb 'http://www.w3.org/2003/11/swrlb#'>
   <!ENTITY swrlx 'http://www.w3.org/2003/11/swrlx#'>
   <!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
]>

<rdf:RDF
   xml:base="http://www.gate.ac.uk/gate-extras/ontologies/TMX-core.owl"
   xmlns="http://www.gate.ac.uk/gate-extras/ontologies/TMX-core.owl"
   xmlns:kaon2="http://kaon2.semanticweb.org/internal#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:owlx="http://www.w3.org/2003/05/owl-xml#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:ruleml="http://www.w3.org/2003/11/ruleml#"
   xmlns:swrl="http://www.w3.org/2003/11/swrl#"
   xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
   xmlns:swrlx="http://www.w3.org/2003/11/swrlx#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

<owl:Ontology rdf:about=""/>

<owl:ObjectProperty rdf:about="#associatedWith">
   <rdfs:range rdf:resource="#TranslationUnit"/>
   <rdfs:range rdf:resource="#TranslationUnitVariant"/>
</owl:ObjectProperty>
```

```
<owl:DatatypeProperty rdf:about="#segType">
   <rdfs:domain rdf:resource="#TranslationUnit"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#srcLang">
   <rdfs:domain rdf:resource="#TranslationUnit"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#segType">
   <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#srcLang">
   <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:about="#associatedWith">
   <rdfs:domain rdf:resource="#Note"/>
   <rdfs:domain rdf:resource="#Context"/>
</owl:ObjectProperty>

<owl:Class rdf:about="#Segment"/>

<owl:ObjectProperty rdf:about="&b;hasPart">
   <rdf:type rdf:resource="&owl;TransitiveProperty"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&b;isPartOf">
   <rdf:type rdf:resource="&owl;TransitiveProperty"/>
   <owl:inverseOf rdf:resource="&b;hasPart"/>
</owl:ObjectProperty>

<owl:Class rdf:about="#Context"/>

<owl:Class rdf:about="#Note"/>
```

<owl:Class rdf:about="#TranslationUnit"/>

<owl:Class rdf:about="#TranslationUnitVariant"/>

<rdfs:Datatype rdf:about="&xsd;string"/>

</rdf:RDF>

Appendix C: XLIFF-core.owl

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY a 'http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#'>
    <!ENTITY kaon2 'http://kaon2.semanticweb.org/internal#'>
    <!ENTITY owl 'http://www.w3.org/2002/07/owl#'>
    <!ENTITY owlx 'http://www.w3.org/2003/05/owl-xml#'>
    <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
    <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
    <!ENTITY ruleml 'http://www.w3.org/2003/11/ruleml#'>
    <!ENTITY swrl 'http://www.w3.org/2003/11/swrl#'>
    <!ENTITY swrlb 'http://www.w3.org/2003/11/swrlb#'>
    <!ENTITY swrlx 'http://www.w3.org/2003/11/swrlx#'>
    <!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
]>

<rdf:RDF
    xml:base="http://www.gate.ac.uk/gate-extras/ontologies/XLIFF-core.owl"
    xmlns="http://www.gate.ac.uk/gate-extras/ontologies/XLIFF-core.owl"
    xmlns:kaon2="http://kaon2.semanticweb.org/internal#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:owlx="http://www.w3.org/2003/05/owl-xml#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:ruleml="http://www.w3.org/2003/11/ruleml#"
    xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
    xmlns:swrlx="http://www.w3.org/2003/11/swrlx#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

<owl:Ontology rdf:about=""/>

<owl:ObjectProperty rdf:about="&a;hasPart">
    <rdf:type rdf:resource="&owl;TransitiveProperty"/>
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:about="&a;isPartOf">
   <rdf:type rdf:resource="&owl;TransitiveProperty"/>
</owl:ObjectProperty>

<owl:Class rdf:about="#Source"/>

<owl:Class rdf:about="#SourceLanguage"/>

<owl:Class rdf:about="#TransUnit"/>

<owl:Class rdf:about="#TargetLanguage"/>

<owl:Class rdf:about="#Target"/>

<owl:ObjectProperty rdf:about="&a;isPartOf">
   <owl:inverseOf rdf:resource="&a;hasPart"/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:about="#equivTrans">
   <rdfs:domain rdf:resource="#Target"/>
   <rdfs:range rdf:resource="&xsd;boolean"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#languageIdentifier">
   <rdfs:domain rdf:resource="#Source"/>
   <rdfs:domain rdf:resource="#Target"/>
   <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<rdfs:Datatype rdf:about="&xsd;boolean"/>

<rdfs:Datatype rdf:about="&xsd;string"/>

</rdf:RDF>
```

Appendix D: MLIF-core.owl

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY b 'http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#'>
    <!ENTITY kaon2 'http://kaon2.semanticweb.org/internal#'>
    <!ENTITY owl 'http://www.w3.org/2002/07/owl#'>
    <!ENTITY owlx 'http://www.w3.org/2003/05/owl-xml#'>
    <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
    <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
    <!ENTITY ruleml 'http://www.w3.org/2003/11/ruleml#'>
    <!ENTITY swrl 'http://www.w3.org/2003/11/swrl#'>
    <!ENTITY swrlb 'http://www.w3.org/2003/11/swrlb#'>
    <!ENTITY swrlx 'http://www.w3.org/2003/11/swrlx#'>
    <!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
]>

<rdf:RDF
    xml:base="http://gate.ac.uk/gate-extras/neon/ontologies/mlif-core.owl"
    xmlns="http://gate.ac.uk/gate-extras/neon/ontologies/mlif-core.owl#"
    xmlns:kaon2="http://kaon2.semanticweb.org/internal#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:owlx="http://www.w3.org/2003/05/owl-xml#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:ruleml="http://www.w3.org/2003/11/ruleml#"
    xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
    xmlns:swrlx="http://www.w3.org/2003/11/swrlx#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

<owl:Ontology rdf:about=""/>

<owl:Class rdf:about="#MultiLingualComponent"/>

<owl:Class rdf:about="#MonoLingualComponent"/>
```

```
<owl:DatatypeProperty rdf:about="#languageIdentifier">
   <rdfs:domain rdf:resource="#MonoLingualComponent"/>
   <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#translationRole">
   <rdfs:domain rdf:resource="#MonoLingualComponent"/>
   <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:Class rdf:about="#SegmentationComponent"/>

<owl:DatatypeProperty rdf:about="#segment">
   <rdfs:domain rdf:resource="#SegmentationComponent"/>
   <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#pos">
   <rdfs:domain rdf:resource="#SegmentationComponent"/>
   <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#lemma">
   <rdfs:domain rdf:resource="#SegmentationComponent"/>
   <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:about="&b;hasPart"/>

<owl:ObjectProperty rdf:about="&b;isPartOf"/>

<owl:ObjectProperty rdf:about="&b;hasPart">
   <owl:inverseOf rdf:resource="&b;isPartOf"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&b;isPartOf">
   <rdfs:domain rdf:resource="#SegmentationComponent"/>
```

```
    <rdfs:domain rdf:resource="#MonoLingualComponent"/>
    <rdfs:range rdf:resource="#MonoLingualComponent"/>
    <rdfs:range rdf:resource="#MultiLingualComponent"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&b;hasPart">
    <rdfs:domain rdf:resource="#SegmentationComponent"/>
    <rdfs:domain rdf:resource="#MonoLingualComponent"/>
    <rdfs:range rdf:resource="#MonoLingualComponent"/>
    <rdfs:range rdf:resource="#MultiLingualComponent"/>
</owl:ObjectProperty>

<owl:Class rdf:about="#MonoLingualComponent">
    <rdfs:comment xml:lang="en">part of a multilingual component, containing information related to one language. Its attributes are the following: languageIdentifier contains an ISO639 code; translationRole determines whether the encompassing MonoC component corresponds to a source language or a target language in a translation process.</rdfs:comment>
</owl:Class>

<owl:Class rdf:about="#MultiLingualComponent">
    <rdfs:comment rdf:datatype="&xsd;string">groups together all variants of a given textual content.</rdfs:comment>
</owl:Class>

<owl:Class rdf:about="#SegmentationComponent">
    <rdfs:comment rdf:datatype="&xsd;string">a recursive component allowing any level of segmentation for textual information. It has the following attributes: rdfs:label contains the segment string; pos denotes part of speech and lemma contains the citation/canonical form of the segment.</rdfs:comment>
</owl:Class>


<rdfs:Datatype rdf:about="&xsd;string"/>


</rdf:RDF>
```

Appendix E: LMF-multilingual-module.owl

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
   <!ENTITY b 'http://lexonto.ontoware.org/lmf#'>
   <!ENTITY kaon2 'http://kaon2.semanticweb.org/internal#'>
   <!ENTITY owl 'http://www.w3.org/2002/07/owl#'>
   <!ENTITY owlx 'http://www.w3.org/2003/05/owl-xml#'>
   <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
   <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
   <!ENTITY ruleml 'http://www.w3.org/2003/11/ruleml#'>
   <!ENTITY swrl 'http://www.w3.org/2003/11/swrl#'>
   <!ENTITY swrlb 'http://www.w3.org/2003/11/swrlb#'>
   <!ENTITY swrlx 'http://www.w3.org/2003/11/swrlx#'>
   <!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
]>
<rdf:RDF
   xml:base="http://gate.ac.uk/gate-extras/neon/ontologies/lmf-multilingual-module.owl"
   xmlns="http://gate.ac.uk/gate-extras/neon/ontologies/lmf-multilingual-module.owl#"
   xmlns:kaon2="http://kaon2.semanticweb.org/internal#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:owlx="http://www.w3.org/2003/05/owl-xml#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:ruleml="http://www.w3.org/2003/11/ruleml#"
   xmlns:swrl="http://www.w3.org/2003/11/swrl#"
   xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
   xmlns:swrlx="http://www.w3.org/2003/11/swrlx#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

<owl:Ontology rdf:about=""/>

<owl:Class rdf:about="#SenseAxis"/>

<owl:Class rdf:about="#SenseAxisRelation"/>

<owl:Class rdf:about="#InterlingualExternalRef"/>
```

```
<owl:DatatypeProperty rdf:about="#id">
   <rdfs:domain rdf:resource="#SenseAxis"/>
   <rdfs:domain rdf:resource="#SenseAxisRelation"/>
   <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#name">
   <rdfs:domain rdf:resource="#SenseAxis"/>
   <rdfs:domain rdf:resource="#SenseAxisRelation"/>
   <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#externalSystem">
   <rdfs:domain rdf:resource="#InterlingualExternalRef"/>
   <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#externalReference">
   <rdfs:domain rdf:resource="#InterlingualExternalRef"/>
   <rdfs:range rdf:resource="&xsd;anyURI"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:about="#hasInterlingualExternalRef">
   <rdfs:domain rdf:resource="#SenseAxis"/>
   <rdfs:range rdf:resource="#InterlingualExternalRef"/>
</owl:ObjectProperty>

<owl:Class rdf:about="#InterlingualExternalRef">
   <owl:equivalentClass rdf:resource="&b;MultilingualExternalRef"/>
</owl:Class>

<owl:Class rdf:about="#SenseAxis">
   <owl:equivalentClass rdf:resource="&b;SenseAxis"/>
</owl:Class>

<owl:Class rdf:about="#SenseAxisRelation">
   <owl:equivalentClass rdf:resource="&b;SenseAxisRelation"/>
```

</owl:Class>

<owl:Class rdf:about="#InterlingualExternalRef">

<rdfs:comment xml:lang="en"><![CDATA[Interlingual External Ref is a class representing the relationship between a Sense Axis instance and an external interlingual knowledge organization system.

The attributes externalSystem and externalReference are provided to refer respectively to the name(s) of the external system and to the specific relevant node in this given external system.

]]></rdfs:comment>

</owl:Class>

<owl:Class rdf:about="#SenseAxis">

<rdfs:comment xml:lang="en"><![CDATA[Sense Axis is a class representing the relationship between different closely related senses in different languages and implements an approach based on the interlingual pivot. The purpose is to describe the translation of lexemes from one language to another. Optionally, a Sense Axis may refer to an external knowledge representation system where the appropriate equivalent can be found.

Attributes: id and label (name of the relation, e.g. â€œsynonymyâ€)

]]></rdfs:comment>

</owl:Class>

<owl:Class rdf:about="#SenseAxisRelation">

<rdfs:comment xml:lang="en">Sense Axis Relation is a class representing the relationship between two different Sense Axis instances. Attributes: id and label (name of the relation, e.g. â€œspecializationâ€)</rdfs:comment>

</owl:Class>

<owl:Class rdf:about="&b;MultilingualExternalRef"/>

<owl:Class rdf:about="&b;SenseAxis"/>

<owl:Class rdf:about="&b;SenseAxisRelation"/>

<rdfs:Datatype rdf:about="&xsd;anyURI"/>

<rdfs:Datatype rdf:about="&xsd;string"/>

</rdf:RDF>