Database Models and Data Formats

DELIVERABLE NR. 1 / WP NR. 2

Version 3.2
Date 06 07 2009

Carlo Aliprandi, Federico Neri – SYNTHEMA
Andrea Marchetti , Francesco Ronzano, Maurizio Tesconi – CNR IIT
Claudia Soria, Monica Monachini – CNR ILC
Piek Vossen – VUA/IRION
Wauter Bosma – VUA
Eneko Agirre, Xabier Artola, Arantza Diaz de Ilarraza,
German Rigau, Aitor Soroa - EHU

Knowledge Yielding Ontologies for Transition-based Organization

ICT 211423

Grant Agreement No.	**ICT 211423**
Project Acronym	**KYOTO**
Project full title	**Knowledge Yielding Ontologies for Transition-based Organization**
Technologies	
Funding Scheme	**FP7 – ICT**
Date latest version Annex I	**19-12-2007**
Project Coordinator	**Prof. Dr. Piek T.J.M. Vossen VU University Amsterdam Tel. + 31 (0) 20 5986466 Fax. + 31 (0) 20 5986500 Email: p.vossen@let.vu.nl**
Project website	**http://www.kyoto-project.eu/**
Deliverable Document Number	**D2.1**
Status	**Draft**
Security (distribution level)	**Public**
Contractual date of delivery	**August 31, 2008**
Actual date of delivery	
Type	**Report**
WP contributing to the deliverable	**WP2**
WP responsible	
Authors	**Carlo Aliprandi, Federico Neri –SYNTHEMA Andrea Marchetti , Francesco Ronzano, Maurizio Tesconi – CNR IIT Claudia Soria, Monica Monachini –CNR ILC Piek Vossen – VUA/IRION Wauter Bosma – VUA Eneko Agirre, Xabier Artola, Arantza Diaz, German Rigau, Aitor Soroa – EHU**
EC project officer	**Werner Janusch**
Keywords	**XML data format, TMF, SEMAF, OWL/KIF, FACTAF, KAF**
Abstract	**The deliverable describes data structure and XML formats that have been investigated and defined for data representation of linguistic and semantic resources underlying the KYOTO system.**

Table of Content

List of Figures

1 Introduction

This deliverable describes data structure and XML formats that have been investigated and defined for data representation of linguistic and semantic resources underlying the KYOTO system.

As Kyoto will be an open and public system, the consortium agreed to adopt standard XML specifications for all the data formats that will encode and represent data.

For all the different databases that will be implemented and all the processes that will make use of the databases the partners started to investigate the availability of XML standards, coming from the research community and international standard organizations. It was clearly identified that the KYOTO system will rely on different text annotation tasks (or Linguistic Annotators):

Terms representation
Morpho-syntax annotations
Semantic annotations
Facts annotations

For each given Linguistic Annotator a data format has been defined, starting from existing standards:

1. Terms annotation
ISO standard LMF (Lexical Markup Framework) was investigated.
ISO standard TMF (Terminological Markup Framework) was investigated.

2. Morpho-syntax annotation
MAF and SYNAF were investigated.
MAF is an ISO reference format for the representation of Morphologic annotations and low level Syntactic annotations.
SYNAF is an ISO reference format for the representation of high level Syntactic annotations.

3. Semantic annotation
A new format was defined, KAF, as satisfactory standard were not found in literature.

4. Facts annotation
A new format was defined, FACTAF, as satisfactory standard were not found in literature.

We also agreed to standardize the format for Wordnets and Ontologies, as the use of a shared common format for representing the various resources available in the Consortium was judged essential.

5. Wordnets
As the standardization of lexical resources is in a very mature stage, the consortium approved adoption of LMF with minor modifications.

6. Ontologies

Two different candidates, OWL and KIF were investigated.

This deliverable is structured as follows. In the next section, we shortly describe the system architecture of KYOTO. In the following sections we describe in details the specific format of each annotation task and related resources. In the final section we provinde details of the functionalities for the semantic search, which is implemented on top of the annotated resources.

2 Overview of the KYOTO system

The global architecture of the KYOTO system is given in Figure 0. User of the KYOTO system can upload their document to the server. These documents are processed to by some basic linguistic processors that apply mostly a structural linguistic analysis. The output of these processors has to be a representation of morph-syntactic structure of the text.

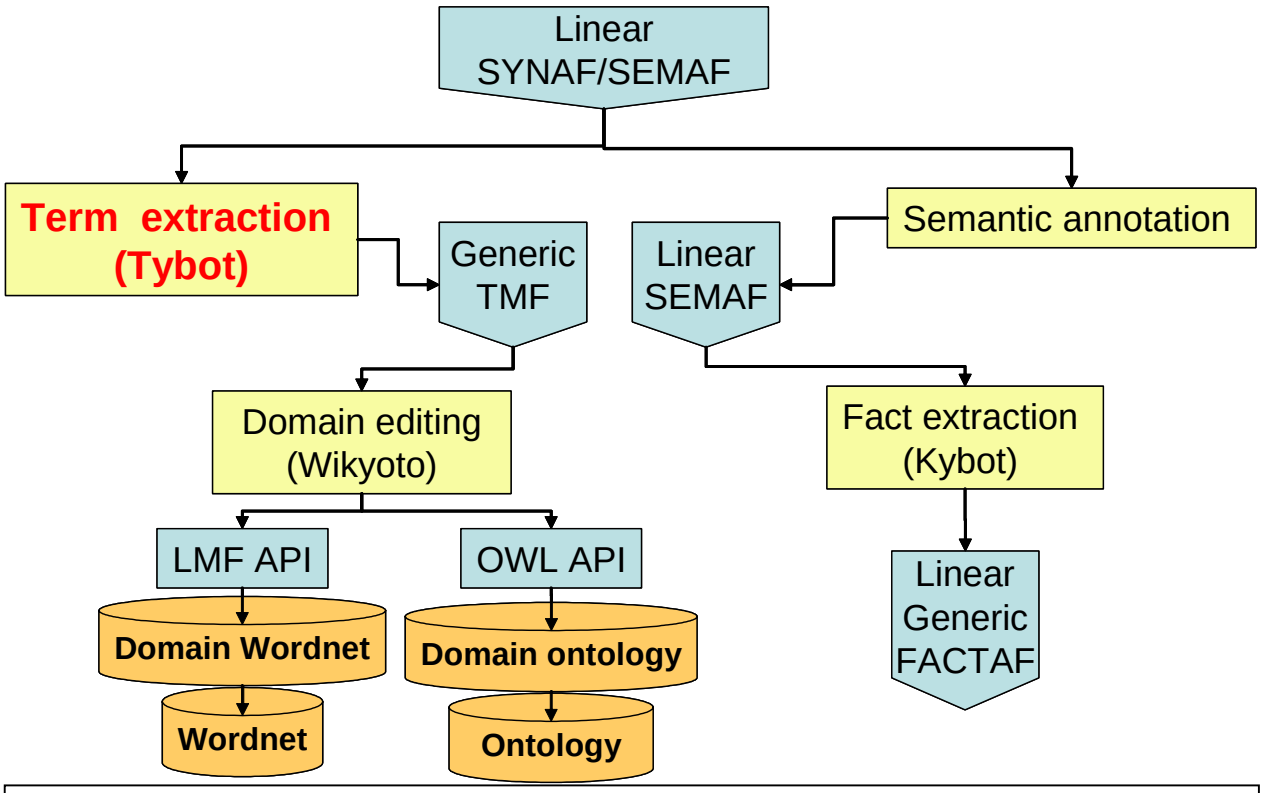

Figure 0: Simplified system overview KYOTO

The representation of the text in KYOTO is defined in the KYOTO KAF structure. KYOTO is a multilayered representation of the text as it occurs as a sequence of words. The following layers are distinguished:

1. Sequence of sentences with words.
2. Sequence of terms
3. Sequence of constituent chunks
4. Sequence of syntactic roles
5. Sequence of semantic roles
6. Collection of facts, focused around events

Each of these layers is interconnected through identifiers so that each level of analysis can be related to the next level. Terms are extracted from levels 1, 2, 3, and 4. This is before a semantic interpretation is done of the text. The terms are extracted by so-called Tybots: term extracting robots. Tybots typically first rely on the structural

properties of the text. Additionally, structural data are interpreted as semantic relations.

The resulting structure is a so-called Term hierarchy (Vossen 2008). A term hierarchy uses so-called parent relations from term to term to group semantically related terms together. For example, the terms "climate change" and "temperature change" will be both be grouped to the term "change", and "rapid climate change" as a child to "climate change". A hierarchy of terms represents many possibilities to abstract from individual terms and individual occurrences of terms.

The term data structure thus represents an abstraction from the actual corpus. We also say that it is a generic data structure, whereas KAF represents a linear or sequential data structure of occurrences of terms. In a generic structure a term is listed only once with pointers to the actual occurrences in the text. Properties of the term are based on all the occurrences and need to be stated only once. In the case of a linear annotation, each individual occurrence of a term is considered separately and as being unique.

However, the term structure is not expected to be the final lexical and knowledge structure that represents the most abstract representation of a term and concept. This level is represented in the domain wordnet and ontology. In order to achieve this, the term structures are validated by users who are experts in the domain. Through the validation process, the initial data structure is confirmed and generalized to a maximal level. The details of this process are described in various KYOTO working papers on the Wikyoto system. Most important here is that the term structures represent a level of abstraction in between the corpus of text and the lexicon and ontology.

3 Terms Annotation

We branded the terms annotation task and the related agent "Tybot": term extracting robot.
Tybots assume that the text is is tokenized to sentences and sentences are processed morpho-syntactically. The output of this process is represented in the KAF notation that is described later in the document.

The term extraction is then done in several steps (see also Morin and Jacquemin 1999, Vossen 2001, Morin and Jacquemin 2004, Basili et al. 2007, Vossen 2008):

1. extraction of potential terms using the morpho-syntactic structure
2. statistical selection of salient terms
3. external sources
4. semantic interpretation

3.1 Structural analysis

Within the NP, each word form is lemmatized, where compounds are split and multiwords are detected using a rich lexicon of existing word forms. From these NPs, we extract candidate terms according to the following principles:

1. Words that are the syntactic head of an NP or VP, e.g.: *card*, *grasslands*, *grass-lands*, *accelerate*
2. Word combinations that include the syntactic head, e.g.: *grass lands*, *forests montane grasslands*, *grasslands in tropical area*, *rapid climate change*, *climate change.*.
3. The head of a compound: *land* as the head of *grass-lands* and the single word *grassland*.

Using this heuristics, we over-generate many multiword terms for from NPs, where the head of the NP is combined with one ore more modifiers or with one or more PPs. These terms are further filtered after we have build a term hierarchy.

All terms that normalize to the same basic form are grouped together as variants of the same term. Note that this is not just lemmatization but also includes variants such as *grass lands*, *grass-lands* and *grasslands*. Further normalization can be achieved through synonym-detection (using wordnet or text-based analysis, e.g abbreviations and full forms).

The heads of the terms represent the parents in the hierarchy. This structure thus represents a tree with the non-decomposable words as the tops. The less tops the better and the more branching subtrees the better. We can use graph-features as a measurement of the quality of the structure, e.g. unary-branching nodes are bad and

can be removed. Below are a few examples of such subtrees extracted from the living planet document with over-generated terms. We can see here that we get intermediate levels in the branches due to terms extracted from phrases that drop a single modifier, e.g. "rapid" from "rapid transition to sustainability" generates the term "transition to sustainability":

transition:0:0
  transition to sustainability:3:2
    rapid transition to sustainability:1:1
land:7:5
  grassland:21:5
    montane grasslands:2:1
      forests montane grasslands:1:1
      savannahs montane grasslands:1:1
    subtropical grasslands:3:3
    temperate grasslands:2:1
      savannahs temperate grasslands:1:1
      scrub temperate grasslands:1:1
loss:4:4
  loss of natural habitat to agriculture:2:1
  loss of biodiversity:2:2
    permanent loss of biodiversity:1:1

account:0:0
  footprint accounts:4:1
    national footprint accounts:2:1
    ecological footprint accounts:1:1
    future footprint accounts:1:1

The numbers behind the terms indicate the frequency and the number of documents in which they occur. These trees are pruned using the following heuristics: unary-branching nodes that occur only embedded in a larger term are removed and its children are moved up. This rule validates the marked intermediate levels in the above tree. In the case of the *transition* tree, "transition to sustainability" is a unary-branching node but has a frequency of 3, which is two more than "rapid transition to sustainability". The same holds for "loss of biodiversity". In the case of "grasslands", "temperate grasslands", "moderate grasslands" and "footprint accounts", there is no unary-branching. The node represents a nice grouping of sub-terms even if there is independent frequency of the term, as is the case for "footprint account". Other potential terms, such as "permanent loss" and "rapid transition", are removed from the tree because they result in unary-branching nodes without independent frequency.

Using this method a more compact tree can be obtained. However, it is not possible to decide on the status of the leaf-terms in the hierarchy. For this it is always necessary to consult an expert in the field to finally decide.

In addition to the parent-relation, we also list all the structural relations that apply to all the occurrences of the term. These involve the modifiers of the term, PP-relations, possessive constructions, subject and object relations. Frequency of patterns is stored and where possible we try to abstract and generalize, e.g. passive and active sentences can result in the same structural relations.

In Table 1, an example is given of the type of structural relations that can be extracted for the term footprint. In addition to the structural relation, a semantic relation and the semantic type of the related concept are given here as well.

Table 1: Structural patterns related to "footprint", based on the Living_planet document

footprint:51:31:121	**Examples**	**Structure**	**Role**	**Type**
NP+VP				
NP+PP	footprint of citizens:2:2:2	of	Attribute of	People
	footprint of nuclear electricity:2:2:2	of	Attribute of	Power
	footprint of nuclear power:2:2:2	of	Attribute of	Power
	footprint of country:2:2:2	of	Attribute of	Country
	footprint in region:2:2:2	in	Location	Region
	total ecological footprints of cropland:2:2:2	of	Attribute of	Land
	total ecological footprint of nation:2:2:2	of	Attribute of	Country
	total footprint of asia-pacific:2:2:2	of	Attribute of	Country
	average footprint in low-income countries:2:2:2	in	Location	Country
	average footprint in high-income countries:2:2:2	in	Location	Country
PP of another NP	countries with highest total footprints:2:2:2	with	Attribute	Country
	2003 total national footprints as proportion of global footprint:2:2:2	proportion of	Relation	Amount
	average per person footprint:2:2:2	per	Unit	Person
	reductions for regional footprints:2:2:2	for	Object	ReduceProcess
Possessive's	country's ecological footprint:2:2:2	's	Attribute of	Country
S	energy footprint shows:2:2:2	Subject	Agent	Show
	factors shape ecological footprint:2:2:2	Object	Patient	Modify
	to allocate the national per capita footprint to consumption categories	Object	Patient	Allocate
Compound	consumption footprint:2:2:2	Modifier	Attribute of	Consumption
	energy footprint: 2:2:2	Modifier	Attribute of	Energy
	footprint documents:2:2:2	Head	Signaling	Document
	footprintnetwork:20:8:8	Head	Whole-of	Network
Adj+N	2003 ecological footprint:2:2:2	Modifier	Time	2003
	ecological footprint:16:14:14	Modifier	Qualifier	Ecological
	global footprint:4:4:6	Modifier	Location	Global
	resulting national footprints:2:2:2	Modifier	Caused-by	Result
		Modifier	Location	National
	total ecological footprint:8:8:8	Modifier	Quantifier	Total

This table gives an idea about the types of generalizations that can be made. PPs with "in" dominantly express a location and PPs with "of" indicate the source of which the "footprint" is an attribute. Modifiers constrain the concept to time or place. etc.

3.2 Statistical analysis of terms

For the statistical filtering, the frequency and distribution of the term is compared with the frequency and distribution in a reference corpus. The reference corpus is based on

a wide diversity of websites of companies. The reason for choosing companies is that the extracted terms should be company products. For each term in the data structure, we then check the proportion of websites in which it is found. This results in a salience value for each term. Words that occur relatively frequent on the website that is the source of the term, but occur on only a small proportion of the reference websites are considered to be salient. Typically, words such as *home page* or *webmaster* will thus be unsalient and can be excluded from the term hierarchy with a proper threshold.

The following formula is used for calculating the salience:

Salience = normFreq * normRef

where ***normFreq*** is the normalized frequency of terms on the website and ***normRef*** is the normalized number of websites on which the term occurs in the reference corpus. These are calculated as follows:

normFreq = $nTermFrequency^{nWords}$ / nPages
normRef = 1-(($nWebsites^{nWords}$) / (referenceCorpusSize))

The ***normFreq*** is the absolute frequency (nTermFrequency) normalized by the size of the website (nPages), where nWords represents the number of words a term consists of. The nWords power enforces the frequency of multiwords. For ***normRef*** we take the proportion of the reference corpus where the term was found. Again, multi words are enforced by the power of the number of words (nWords).

3.3 External sources

For each term, we can get defining phrases (Hirst 1997) and actual definitions. For this we can use the uploaded documents, the domain collection, Google snippets or Wikipedia.

3.4 Semantic interpretation of terms

Structural relations need to be translated into semantic relations. This involves two different processes:

1. Assigning synsets to the words that match with Wordnet
2. Interpreting structural syntactic relations as semantic roles and features

The first can be seen as a special type of WSD. We do not need to consider the individual terms and their unique occurrences in the text, but we can disambiguate a term given all the information and data that we have available for the complete branch in which it occurs. This is shown in Figure 1 for the term subtree headed by "land":

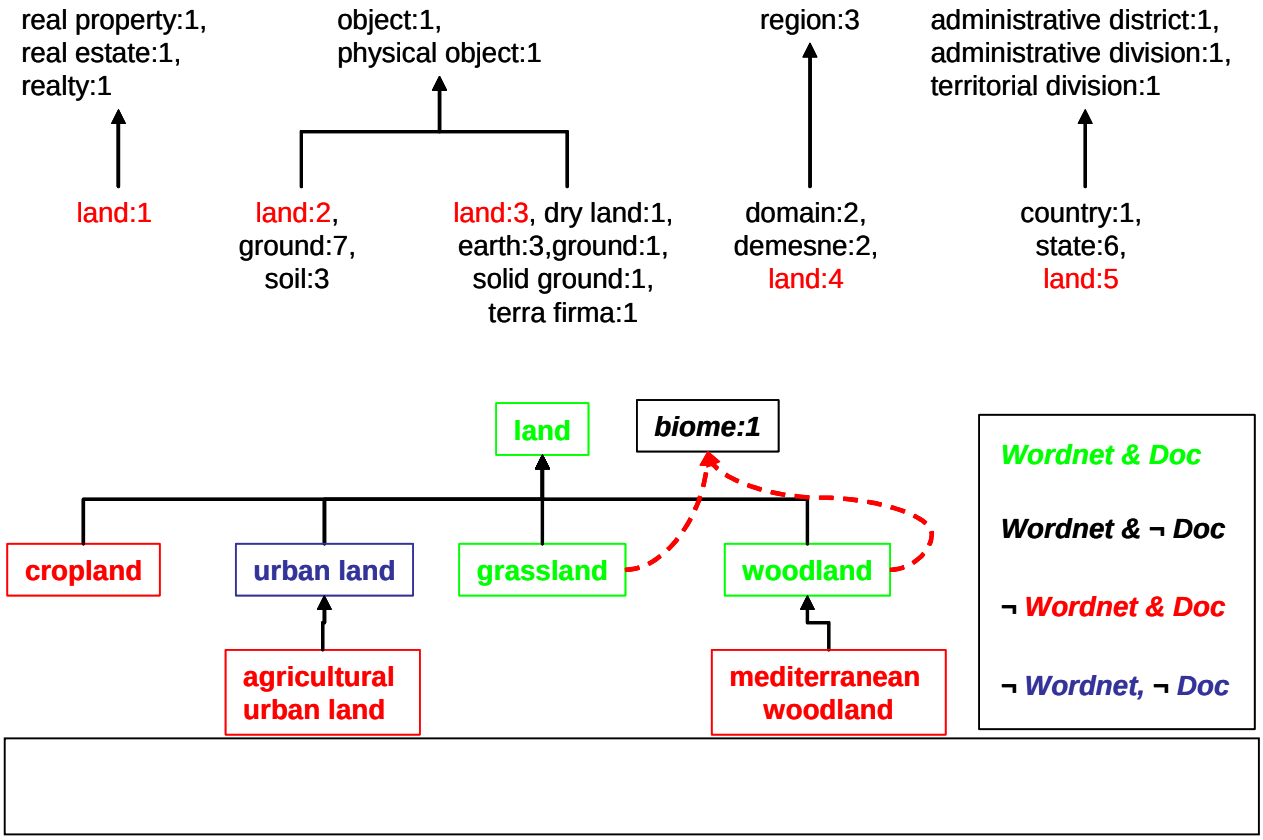

Figure 1 -Mapping of term branches to Wordnet synsets

The term subtree below "land" has four different colors depending on the matching of the word with wordnet:

1. Words that occur in the document as a term and in Wordnet (green), e.g. "land" and "grassland".
2. Words that occur in the document as a term but are not in Wordnet (red), e.g. "cropland".
3. Words that do not occur as a term (see criteria above) and also do not exist in Wordnet (blue), "urban land".
4. Words that do not occur in the document but do occur in Wordnet (black), e.g. "biome"

The last two categories are there only for adding further hierarchical levels to the tree. For assigning synsets, we only need to look at the words that are also entries in Wordnet but we can nevertheless use the full tree-structure to find the most appropriate meanings.

The noun "land" has many different meanings in Wordnet of which 5 are shown here. The WSD module needs to choose the most appropriate meaning of land given the terms and their contextual relations that are below land in the tree. Various traditional WSD techniques can be used among which Conceptual Density. In this case, we get an interesting situation, since two of the sub-terms, "grassland" and "woodland", are monosemous and have "biome" as a parent rather than "land". Since these concepts

are not exclusive, the system can suggest both as the hyperonyms, although it still needs to score most appropriate meaning of "land".

For the interpretation of the structural relations, we already saw that we can come to specific interpretations given the constraints of the concepts. These constraints can come from the associated Wordnet or any ontology linked to it (FrameNet, SUMO, etc.). The fact that "footprint" in Table 1 is typed as an attribute (*the degree of usage of natural resources*) determines to a large extent the possible relations. The automatic role and relation detection can proceed in the same way as for assigning synsets. It can interpret the relations of all terms grouped in a subtree.

Further details about the processes will be worked out in the project. So far it gives an idea about the type of output structures we expect initially. In the next section, we will describe the output format in more detail.

3.5 The Term Markup Format

TMF stands for Terminological Markup Framework (ISO 16642 2001), an international standard designed in the framework of the ISO initiatives to support the creation and use of computer applications for terminological data and exchange of such data between different applications. TMF can be described as a meta-model consisting of two levels of abstraction.

1) The most abstract level is the *meta-model*, (abstract and conceptual data model level) which supports analysis and design of terminological data at a very general level.
The terminological data model comprises the following structural nodes:

- **TDC**, the Terminological Data Collection, the top level container for all the information contained in a terminology system. This is used as a container for other containers;
- **GIS** and **CI**, respectively, Global Information Section and Complementary Information, used to contain external administrative data (the title of the file, the institution originating the file, address, copyright …) or reference to contextual links to related text corpora;
- **TE**, the terminological entry, i.e. the term assigned to a concept. It can contain one or more language sections, depending on whether the termbase is mono- or multi-lingual.
- **LS**, Language Section, containing all the Term Sections for a terminological entry
- **TS**, the Term Section, where information about terms is held. It usually contains a single term used to designate a concept as well as any other information (e.g. definitions, contexts …)
- **TCS**, the Term Component Section, where information about the components of a term are described. For some languages it could be useful to include information about individual words used to construct a multiword term.

2) The second level is the *data model level*, which enables the designer of a terminological data collection to make particular choices according to his/her particular needs. A specific implementation of the meta-model for terminology markup expressed in XML is called a Terminological Markup Language, **TML**.

The abstract meta-model with the various structural nodes can be instantiated in XML (cf. the DTD available on the Twiki site) by means of a the generic element <struct> which can be recursively expressed. Each structural node can be identified by means of a type attribute whose possible values can be the identifiers of the levels in the meta-model, TDC, GIS, TE, CI, LS, TS, TCS.

A complete description of TMF can be found in the document available on the Kyoto Twiki site:

http://www2.let.vu.nl/twiki/pub/Kyoto/WP02:SystemDesign/TMF_ISO16642_160802.pdf .

We decided that the second level of the TMF representation as proposed is not so useful for the project. Almost all information that we need to represent for the terms in our system need to be represented in the form of so-called *brack* and *feat* elements, for example:

```
<brack>
        <feat type="sem"/>
                <feat type="synsets" orig="urn:wordnet1.7"/>
                <brack>
                        <feat type="source">EHU-WSD1</feat>
                        <feat type="synset">ENG20-00180570-n</feat>
                        <feat type="weight">0.80</feat>
                </brack>
                <brack>
                        <feat type="source">EHU-WSD1</feat>
                        <feat type="synset">ENG20-00290564-n</feat>
                        <feat type="weight">0.30</feat>
                </brack>
</brack>
```

Instead of these structures, we propose more condensed solutions, which is easier for processing and storage:

```
<semanticMatch type="senseAlt" orig="urn:wordnet1.7">
        <sense source="EHU-WSD1" sensecode="ENG30-00180570-n" weight="0.80"/>
        <sense source="EHU-WSD1" sensecode="ENG30-00290564-n" weight="0.30"/>
</semanticMatch>
```

Another motivation is that our data level is not stable and specific to the type of heuristics that we will develop during the project. Nevertheless, we will deliver convertors (XSLT) at the end of the project that can transform any KYOTO-LMF into proper LMF. We will explain the structure of KYOTO-LMF in the next section.

For KYOTO TMF, we adopted the top-level of TMF proposal. The top-level structures are:

```
<tmf>
<struct type="TE" id="t001">
<languageCoding>ISO 639-3</languageCoding>
<languageLetterCoding>ENG</languageLetterCoding>
<termDomain>environment</termDomain>
<treeProfile/>
<struct type=”LS”></struct>
<struct>
```

```
</tmf>
```

We foresee that our term structures are language specific and therefore include a specification of the language as a general property. We also provide a label for the user-domain (termDomain) that is manually assigned to identify the domain. In addition we provide a so-called treeProfile. This is an automatic domain classification of the complete term tree. An example of the expanded element looks as follows:

```
<treeProfile>
<microWorld score="0.88">Topography</microWorld>
<microWorld score="0.72">Finance</microWorld>
<microWorld score="0.7">Bio</microWorld>
</treeProfile>
```

The microWorld element holds the domain labels above a certain threshold and a score for each domain. This is used to match the term branches in the tree in terms of relevance or coherence with the overall classes of the tree hierarchy.

The struct LS then holds the term data. It consists of a list of other struct elements of the type TS with has an identifier attribute for each term:

```
<struct type="TS" id="t66"></struct>
```

Each term data structure then consists of the following KYOTO elements:

```
<normalizedTerm/> <!—unifies different variants of the term -->
<partOfSpeech/> <!—part of speech of the term, should be the same for all variants -->
<preferredForm/> <!—canonical form for representation purposes -->
<forms/> <!—list of forms and pointers to positions in SemAF files -->
<parentData/> <!—structural parent relation for establishing the term tree -->
<termStatistics/> <!—statistics at the term level -->
<termProfile/> <!—domain classification of the tree branch that includes the term -->
<sources/> <!—List of type of sources from which the term is derived -->
<semanticRelations/> <!—definitions and semantic relations to other resources -->
<structuralRelations/> <!—list of structural relations that the term occurs in -->
```

All these elements are specific for KYOTO and not represented as standard TMF. The normalizedTerm is only used for internal purposes. It may be omitted if the term identifier is used. The preferredForm is only used for labelling the term for the user in an interface. Canonical forms can be the shortest form or the most frequent form.

The form element is used for listing all the different appearances of the term in the source documents with pointers to the locations in the KAF notation. The elements are of the type termFormData as the next example shows:

```
<termFormData id="tf_2" frequency="6">
        <termForm>climate changes</termForm>
        <span docId="124">
                <span from="w24" to="w43"/>
                <span from="w123" to="w125"/>
                <span from="w5627" to="w5628"/>
        </span>
                <span docId="7824">
                <span from="w24" to="w43"/>
                <span from="w123" to="w125"/>
                <span from="w5627" to="w5628"/>
        </span>
```

```
</termFormData>
```

The *termFormData* has a identifier for the specific variant and a total frequency derived from the collection of source documents. It furthermore has two subelements: *termForm* and a list of *spans*. The *termForm* has the actual form and the spans contain the pointers to the KAF structures. There is a separate *span* element for each document, identified by the docId attribute. Each span element contains a pointer to the beginning and end word in the document that represents an occurrence. So the form "climate change" has 3 ocurrences in document 124, ranging from word w24 to w43, w123 to 125, w5626 to w5628. Other possible form for "climate change" can be inflected variants ("climat changes"), case variants ("Climate Change"), and synonyms of we decide to detect these (e.g. "Climatic changes").

The parentData structure contains a parentTerm and a list of other alternative parents, e.g.:

```
<parentData>
    <parentTerm target="t13">change</parentTerm>
    <wikiCategory source="http://en.wikipedia.org/wiki" date="2008-06-20">Climate change feedbacks
        and causes</wikiCategory>
    <wikiCategory source="http://en.wikipedia.org/wiki" date="2008-06-20">Global warming
        </wikiCategory>
    <wikiCategory source="http://en.wikipedia.org/wiki" date="2008-06-20">History of
        climate</wikiCategory>
    <wikiCategory source="http://en.wikipedia.org/wiki" date="2008-06-20">Carbon
        finance</wikiCategory>
    <wikiCategory source="http://en.wikipedia.org/wiki" date="2008-06-20">Climate and weather
        statistics</wikiCategory>
</parentData>
```

The parentTerm can be empty, in case of the top term in the tree, or ot must contain a pointer to another term in the data structure. The target attribute identifies that term and the value gives the normalizedTerm of the parent. Both can be used to build up the tree structure. We als listed here categories that are derived from Wikipedia using the wikiCategory element. Other parent relations can be derived from other external resources that are exploited during the project.

The termStatistics element has a flat set of subelements, e.g.:

```
<termStatistics>
        <documentNumber>5</documentNumber>
        <termFrequency>13</termFrequency>
        <termSalience>0.04</termSalience>
        <termConnectivity>13</termConnectivity>
        <cumulativeFrequency>18</cumulativeFrequency>
        <cumulativeDocumentNumber>5</cumulativeDocumentNumber>
        <termSiblings>3</termSiblings>
</termStatistics>
```

The *documentNumber* gives the number of source documents from which the term is derived. The *termFrequency* gives the total number of occurrences in these documents. Both numbers are the sum of statistics of each term variant. The *termSalience* was discussed in the previous section and has a value between 0 and 1. The *termConnectivity* represents the number of connections of a term class in the tree, also including structural contextual relations. Well-connected nodes are more important

than poorly connected nodes. The *cumulativeFrequency* and *cumulativeDocumentNumber* give the sum of the *termFrequency* and *documentNumber* of all descendants below and term in the term hierarchy. Finally, *termSiblings* gives the number of terms that share the same parent.

The *termProfile* is similar to the *treeProfile* except that it has an additional attribute *profileMatch* that indicates the overlap of *microWorld* values across the *termProfile* and the *treeProfile*.

```
<termProfile profileMatch="0.69">
        <microWorld score="0.88">Geography</microWorld>
                <microWorld score="0.75">Finance</microWorld>
                <microWorld score="0.73">Metereology</microWorld>
                <microWorld score="0.7">Society</microWorld>
</termProfile>
```

The sources element is a list with types of source sections from which the term is extracted. The score attribute indicates the proportion to which a term was found in the particular section. The scores thus should add up to 1.00.

```
<sources>
        <termSource score="0.2">TOC</termSource>
        <termSource score="0.8">BODY</termSource>
</sources>
```

The *semanticRelations* element contains the mapping to external semantic resources and the definitions. We provided here two examples of a *termDefinition*, one extracted from Wikipedia and one from Google-snippets. In addition, we listed two cases of a mapping: one to Wordnet and one to SUMO. The Wordnet mappings are weighted and have a similar structure as in KAF. Multiple matches can be provided with different weights. The source attribute refers to the software module that was to generate the mapping. Something similar can be done for assigning a SUMO label to the term (or any other resource to which we want to link).

```
<semanticRelations>
        <definitions>
                <termDefiniton source="http://en.wikipedia.org/wiki/Climate_change" date="2008-06-
                20">Climate change is any long-term significant change in the "average weather" that a
                given region experiences. Average weather may include average temperature, precipitation
                and wind patterns. It involves changes in the variability or average state of the
                atmosphere over durations ranging from decades to millions of years. These changes can
                be caused by dynamic process on Earth, external forces including variations in sunlight
                intensity, and more recently by human activities.</termDefiniton>
                <termDefiniton source="googleSnippets" date="2008-06-20">factors such as climate
                changes affecting our oceans</termDefiniton>
                <termDefiniton source="googleSnippets" date="2008-06-20">environniental problems
                such as climate changes or acid rains</termDefiniton>
                <termDefiniton source="googleSnippets" date="2008-06-20">global environmental issues
                such as climate changes</termDefiniton>
                <termDefiniton source="googleSnippets" date="2008-06-20">environmental changes
                such as climate changes</termDefiniton>
                <termDefiniton source="googleSnippets" date="2008-06-20">related activities such as
                climate changes and changes in land use pattern explanatory events such as climate
                changes</termDefiniton>
```

```
            <termDefiniton source="googleSnippets" date="2008-06-20">all kinds of other
            geographical data such as climate changes, plant growth, radiation, rainfall, forest
            fires</termDefiniton>
        </definitions>
        <semanticMatch type="senseAlt" orig="urn:wordnet1.7">
            <sense source="EHU-WSD1" sensecode="ENG30-00180570-n" weight="0.80"/>
            <sense source="EHU-WSD1" sensecode="ENG30-00290564-n" weight="0.30"/>
        </semanticMatch>
        <semanticMatch type="ontologyAlt" orig="urn:sumo">
            <ontology source="EHU-WSD1" class="Process" weight="0.65"/>
            <ontology source="EHU-WSD1" class="NaturalProcess" weight="0.70"/>
            </semanticMatch>
 </semanticRelations>
```

The final element is structuralRelations, which is also a list. A *structuralRelation* groups all occurrences of a morpho-syntatic pattern in which a term occurred. The attributes indicate the syntactic role (*syntacticRole*) and the semantic role (*semanticRole*) that is expressed by the pattern. An optional attribute is active (true or false) which is only relevant for syntactic relations. The values for the *semanticRole* and *syntacticRole* attributes depend on the definition in KAF.

The direction and dependency of the relations are important. The *structuralRelation* can be used for simple bi-gram relations or for dependency relations. We listed both examples below. The subelements are *syntaxElement* and *termFormData*. The *syntaxElement* can get as a value any word that labels the relation, e.g. propositions. The termFormData again contain the term identifier and a frequency attribute. Repeated contexts are thus lumped together and frequency can be used to determine weight or relevance of relations. In addition to the *termForm*, we have the elements deps and *termContext*. The deps element is a list of dependency structures in the KAF files from which the structure is derived. This is similar to the span element we have seen above. The dependency structure of KAF is however a higher level of annotation. The element *termContext* is just a simpler way to represent the context, in case the dependency layer is not available.

```
<structuralRelations>
        <structuralRelation syntaxRole="leftnp" semanticRole="">
            <syntaxElement/>
            <termFormData id="tf_23" frequency="1">
                <termForm>sayan</termForm>
                <deps docId="1824">
                    <dep from="t3" to="t4"/>
                </deps>
                <termContext/>
            </termFormData>
        </structuralRelation>

        <!-- NP to the left of the term as PP -->
        <structuralRelation syntaxRole="pp_left_np" semanticRole="CAUSE">
            <syntaxElement>of</syntaxElement>
            <termFormData id="t597" frequency="1">
                <termForm>impact</termForm>
                <deps docId="124">
                    <dep from="t13" to="t15"/>
                </deps>
                <termContext>impact of climate change</termContext>
            </termFormData>
        </structuralRelation>
```

```
        <!-- Term is the subject of the main verb in an ACTIVE sentence -->
        <structuralRelation syntaxRole="subj" active="true" semanticRole="AGENT">
                <syntaxElement/>
                <termFormData id="t11" frequency="1">
                        <termForm>cause</termForm>
                        <deps docId="124">
                                <dep from="t56" to="t63"/>
                        </deps>
                        <termContext>climate change causes a decline of biodiversity</termContext>
                </termFormData>
        </structuralRelation>

        <!-- Term is the subject of the main verb in a PASSIVE sentence -->
        <structuralRelation syntaxRole="subj" active="false" semanticRole="PATIENT">
                <syntaxElement/>
                <termFormData id="t11" frequency="1">
                        <termForm>cause</termForm>
                        <deps docId="124">
                                <dep from="t356" to="t359"/>
                        </deps>
                        <termContext>climate change is caused by an increase in industrial
                        activity</termContext>
                </termFormData>
        </structuralRelation>

        <!-- Term is modified by an adjective or adverb-->
        <structuralRelation syntaxRole="mod" semanticRole="ATTRIBUTE">
                <syntaxElement/>
                <termFormData id="t111" frequency="1">
                        <termForm>rapid</termForm>
                        <deps docId="124">
                                <dep from="t356" to="t359"/>
                        </deps>
                        <termContext>rapid climate changes</termContext>
                </termFormData>
        </structuralRelation>
</structuralRelations>
```

4 Morpho-syntax Annotation

4.1 MAF

In Natural Language Resource Management, the morpho-syntactic annotation phase assigns to each document segment one or more *tags* providing *morpho-syntactic* information about the *part of speech* (noun, adjective, verb, ...), *morphological* and *grammatical* features (such as number, gender, person, mood, verbal tense, ...) and possibly other specific linguistic properties.

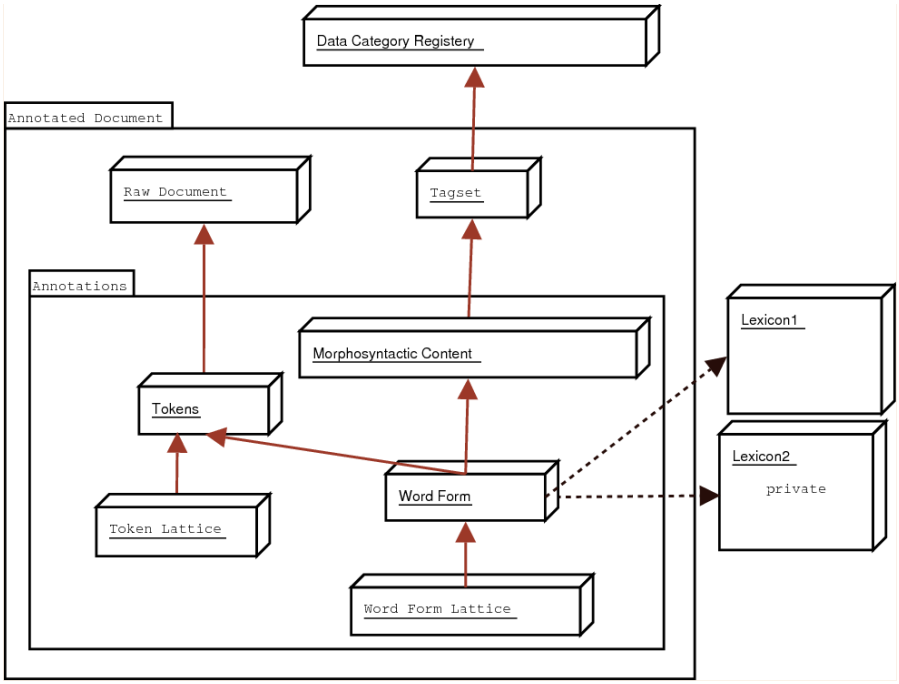

Figure 2 - Simplified view of MAF model

Figure 2 presents a simplified view of the proposed meta-model for morpho-syntactic annotations, while Figure 3 presents a more formal view based on UML. An annotated document is formed by a raw original document and a set of annotations. The annotations are carried by *word forms* covering zero, one or more segments or *tokens* of the original document. A word form may reference a lexicon entry and provides information about its underlying lemma and inflected form. The morpho-syntactic content attached to a word form is expressed by *feature structures* following the guidelines of one or more *tagsets*.

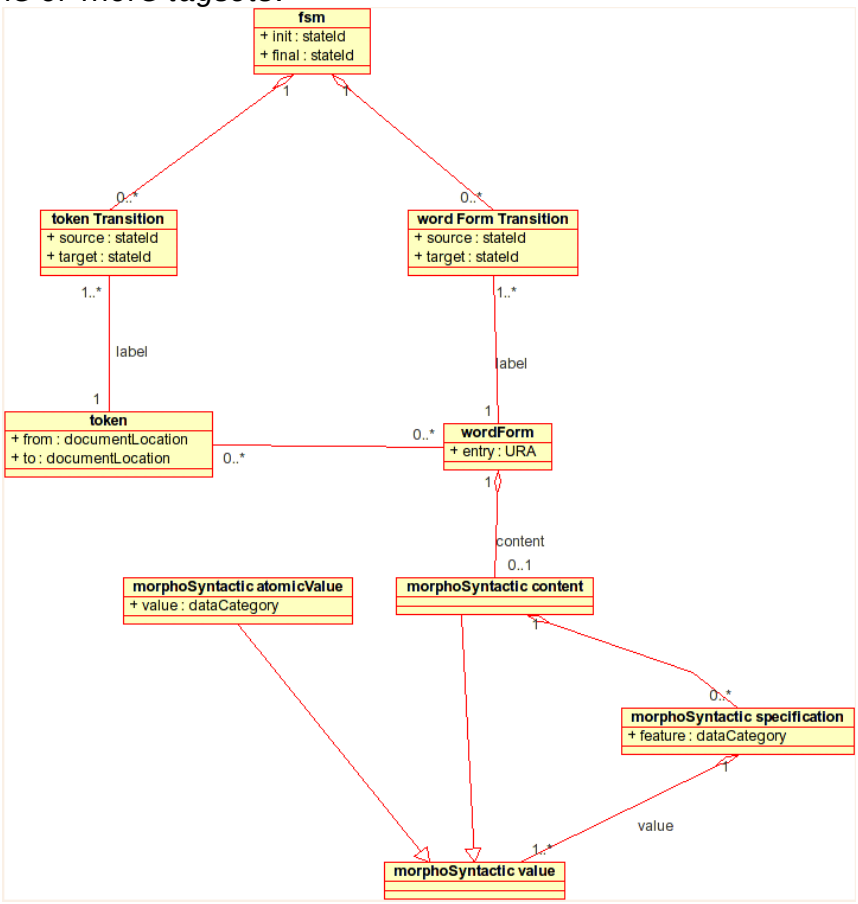

Figure 3 - UML rapresentation of MAF model

4.1.1 Segmentation

Morpho-syntactic annotations are carried by segments, called *tokens*, present in the document flow, but this does not imply that the resulting segmentation corresponds to a sequence of adjacent segments partitioning the original document. The element token is used to represent segments of the original document that, roughly speaking, follow typographical, morphological, or phonological boundaries. In example,

> *"The Living Planet Report indicates that our reliance on fossil fuels to meet our energy needs continues to grow and that climate-changing emissions now make up 48 per cent of our global footprint"*

may be displayed as

```
<token id="t1" from="" join="no">The</token>
<token id="t2" from="" join="no">Living</token>
<token id="t3" from="" join="no">Planet</token>
```

```
<token id="t4" from="" join="no">Report</token>
<token id="t5" from="" join="no">indicates</token>
<token id="t6" from="" join="no">that</token>
<token id="t7" from="" join="no">our</token>
<token id="t8" from="" join="no">reliance</token>
<token id="t9" from="" join="no">on</token>
<token id="t10" from="" join="no">fossil</token>
<token id="t11" from="" join="no">fuels</token>
<token id="t12" from="" join="no">to</token>
<token id="t13" from="" join="no">meet</token>
<token id="t14" from="" join="no">our</token>
<token id="t15" from="" join="no">energy</token>
<token id="t16" from="" join="no">needs</token>
<token id="t17" from="" join="no">continues</token>
<token id="t18" from="" join="no">to</token>
<token id="t19" from="" join="no">grow</token>
<token id="t20" from="" join="no">and</token>
<token id="t21" from="" join="no">that</token>
<token id="t22" from="" join="no">climate-changing</token>
<token id="t23" from="" join="no">emissions</token>
<token id="t24" from="" join="no">now</token>
<token id="t25" from="" join="right">make</token>
<token id="t26" from="" join="no">up</token>
<token id="t27" from="" join="no">48</token>
<token id="t28" from="" join="right">per</token>
<token id="t29" from="" join="no">cent</token>
<token id="t30" from="" join="no">of</token>
<token id="t31" from="" join="no">our</token>
<token id="t32" from="" join="no">global</token>
<token id="t33" from="" join="no">footprint</token>
```

Tokens address segments of the original document but also provide a level of possible abstraction. The non mandatory attributes form, transcription, transliteration may be used to perform this abstraction, providing, for instance, the phonetic transcription of a speech segment, the roman transliteration of some Cyrillic word, the expansion of an abbreviation, the correction of a typographical error, or the choice of a normalized form in presence of variations. In example,

```
<token form="etcetera" id="t1">etc.</token>
<token form="tzar" id="t2">csar</token>
```

In addition, two tokens may overlap, for instance to denote an agglutinated or contracted form (for instance, in French, "des" may be seen as a contraction for "de les").

4.1.2 Word Forms as linguistic units

The segments identified by token elements are used to anchor word forms, that may generally be associated by an attribute entry to a lexical entry in a lexicon. Words forms are also characterized by a part of speech as well as morphological and grammatical properties expressed by feature structures. Immediate information about the lemma and inflected forms may also be attached with the attributes lemma and form. In particular, the attribute form is useful when the inflected form attached to the word form does not coincide with the content attached to the covered tokens, because, for instance, of spelling corrections. A token may be associated to more than one word form and, conversely, a word form may cover more than one token.

```
<token id="t1" from="" join="no">The</token>
<wordForm lemma="the" tag="pos.det" tokens="t1"/>
```

```
<token id="t2" from="" join="no">Living</token>
<wordForm lemma="living" tag="pos.adj" tokens="t2"/>
<token id="t3" from="" join="no">Planet</token>
<wordForm lemma="planet" tokens="t3">
 <fs>
    <f name="pos"><symbol value="noun"/></f>
    <f name="number"><symbol value="singular"/></f>
  </fs>
</wordForm>
<token id="t4" from="" join="no">Report</token>
<wordForm lemma="report" tokens="t4">
 <fs>
    <f name="pos"><symbol value="noun"/></f>
    <f name="number"><symbol value="singular"/></f>
  </fs>
</wordForm>
<token id="t5" from="" join="no">indicates</token>
<wordForm lemma="indicate" tokens="t5">
 <fs>
    <f name="pos"><symbol value="verb"/></f>
    <f name="mood"><symbol value="indicative"/></f>
    <f name="tense"><symbol value="present"/></f>
    <f name="person"><symbol value="third"/></f>
    <f name="number"><symbol value="singular"/></f>
  </fs>
</wordForm>
<token id="t6" from="" join="no">that</token>
<wordForm lemma="that" tag="pos.c" tokens="t6"/>
<token id="t7" from="" join="no">our</token>
<wordForm lemma="our" tag="pos.det" tokens="t7"/>
<token id="t8" from="" join="no">reliance</token>
<wordForm lemma="reliance" tokens="t8">
 <fs>
    <f name="pos"><symbol value="noun"/></f>
    <f name="number"><symbol value="singular"/></f>
  </fs>
</wordForm>
<token id="t9" from="" join="no">on</token>
<wordForm lemma="on" tag="pos.prep" tokens="t9"/>
<token id="t10" from="" join="no">fossil</token>
<wordForm lemma="fossil" tokens="t10">
 <fs>
    <f name="pos"><symbol value="adjective"/></f>
    <f name="number"><symbol value="singular"/></f>
  </fs>
</wordForm>
<token id="t11" from="" join="no">fuels</token>
<wordForm lemma="fuel" tokens="t11">
 <fs>
    <f name="pos"><symbol value="noun"/></f>
    <f name="number"><symbol value="plural"/></f>
  </fs>
</wordForm>
<token id="t12" from="" join="no">to</token>
<wordForm lemma="to" tag="pos.prep" tokens="t12"/>
<token id="t13" from="" join="no">meet</token>
<wordForm lemma="meet" tokens="t13">
 <fs>
    <f name="pos"><symbol value="verb"/></f>
    <f name="mood"><symbol value="infinitive"/></f>
  </fs>
```

```
</wordForm>
<token id="t14" from="" join="no">our</token>
<wordForm lemma="our" tag="pos.det" tokens="t14"/>
<token id="t15" from="" join="no">energy</token>
<wordForm lemma="energy" tokens="t15">
 <fs>
    <f name="pos"><symbol value="noun"/></f>
    <f name="number"><symbol value="singular"/></f>
  </fs>
</wordForm>
<token id="t16" from="" join="no">needs</token>
<wordForm lemma="need" tokens="t16">
 <fs>
    <f name="pos"><symbol value="noun"/></f>
    <f name="number"><symbol value="plural"/></f>
  </fs>
</wordForm>
<token id="t17" from="" join="no">continues</token>
<wordForm lemma="continue" tokens="t17">
 <fs>
    <f name="pos"><symbol value="verb"/></f>
    <f name="mood"><symbol value="indicative"/></f>
    <f name="tense"><symbol value="present"/></f>
    <f name="number"><symbol value="singular"/></f>
    <f name="person"><symbol value="third"/></f>
  </fs>
</wordForm>
<token id="t18" from="" join="no">to</token>
<wordForm lemma="to" tag="pos.prep" tokens="t18"/>
<token id="t19" from="" join="no">grow</token>
<wordForm lemma="grow" tokens="t19">
 <fs>
    <f name="pos"><symbol value="verb"/></f>
    <f name="mood"><symbol value="infinitive"/></f>
  </fs>
</wordForm>
<token id="t20" from="" join="no">and</token>
<wordForm lemma="to" tag="pos.coord" tokens="t20"/>
<token id="t21" from="" join="no">that</token>
<wordForm lemma="that" tag="pos.c" tokens="t21"/>
<token id="t22" from="" join="no">climate-changing</token>
<wordForm lemma="climate-changing" tokens="t22">
 <fs>
    <f name="pos"><symbol value="prop"/></f>
    <f name="number"><symbol value="singular"/></f>
  </fs>
</wordForm>
<token id="t23" from="" join="no">emissions</token>
<wordForm lemma="emission" tokens="t23">
 <fs>
    <f name="pos"><symbol value="prop"/></f>
    <f name="number"><symbol value="plural"/></f>
  </fs>
</wordForm>
<token id="t24" from="" join="no">now</token>
<wordForm lemma="now" tag="pos.adv" tokens="t24">
<token id="t25" from="" join="no">make</token>
<wordForm lemma="make" tokens=" t25">
 <fs>
    <f name="pos"><symbol value="verb"/></f>
    <f name="mood"><symbol value="indicative"/></f>
```

```
      <f name="tense"><symbol value="present"/></f>
      <f name="number"><symbol value="plural"/></f>
      <f name="person"><symbol value="third"/></f>
</fs>
</wordForm>
<token id="t26" from="" join="no">up</token>
<wordForm lemma="up" tag="pos.prep" tokens="t26">
<token id="t27" from="" join="no">48</token>
<token id="t28" from="" join="right">per</token>
<token id="t29" from="" join="no">cent</token>
<wordForm lemma="48 per cent" tag="pos.np" tokens="t27 t28 t29"/>
<token id="t30" from="" join="no">of</token>
<wordForm lemma="of" tag="pos.prep" tokens="t30"/>
<token id="t31" from="" join="no">our</token>
<wordForm lemma="our" tag="pos.det" tokens="t31"/>
<token id="t32" from="" join="no">global</token>
<wordForm lemma="global" tokens="t32">
 <fs>
    <f name="pos"><symbol value="adjective"/></f>
    <f name="number"><symbol value="singular"/></f>
  </fs>
</wordForm>
<token id="t33" from="" join="no">footprint</token>
<wordForm lemma=" footprint tokens="t33">
 <fs>
    <f name="pos"><symbol value="noun"/></f>
    <f name="number"><symbol value="singular"/></f>
  </fs>
</wordForm>
```

The example shows both the simplest case of relationship between tokens and word forms, when a word form covers a single token, as well as the handling of more complex cases, as the identification of compound words covering several adjacent tokens (see 4.1.2.2).

4.1.2.1 Words from lexicon

A word form is a linguistic unit carrying morpho-syntactic properties. Generally, a linguistic unit may be characterized by a label corresponding to an entry if some lexicon. This identification is materialized by the attribute entry, whose content should express a reference (an URN) to the lexicon entry.

```
<token id="t0">climate</token>
<token id="t1">change</token>
<wordForm lemma="urn:lexicon:en:climate change" tokens="t0 t1"/>
```

The notion of " lexicon entry" is outside the scope of MAF. A reference to a lexicon entry is therefore not precisely defined but, in first approximation, should correspond to an URN (Uniform Resource Name). It should be noted that one may wish to reference lexicons "sub-entries" for polysemous entries or for compound forms.

4.1.2.2 Compound word forms

The structure of compound forms (including multi-word expressions) may be expressed using nested word forms, therefore providing information about the subparts even when none is available for the whole. In fact, note that in the following example "climate change" can be recognised as a multiple words expression

```
<token id="t0">climate</token>
<token id="t1">change</token>
<wordForm lemma="climate change" tokens="t0 t1"/>

<token form="Geburtstag " id="t1" join="right">Geburtstags</token>
```

```
<token form="Geschenk" id="t2" join="right">ges chenk</token>
<token form="Papier " id="t3">papier</toke>
<wordForm tokens="t1 t2 t3">
      <wordForm entry="urn:lexicon:de:geburstag" lemma="geburstag"
           tokens="t1"/>
      <wordForm entry="urn:lexicon:de:geschenk" lemma="geschenk"
           tokens="t2"/>
      <wordForm entry="urn:lexicon:de:papier" lemma="papier"
           tokens="t3”/>
</wordForm>
```

4.1.3 Morpho-syntactic content

A word form may be completed by a morpho-syntactic content defining its linguistic nature and its grammatical function in its current context. This content is expressed using Feature Structures, following the recommendation of ISO 24610 Part 1 document on “Feature Structure Representation” (FSR). In first approximation, a feature structure may attach one or several (possibly complex) values to linguistic properties (i.e., noun to part of speech, present to tense, indicative to mood, etc).

```
<token id="t23" from="" join="no">emission</token>
<wordForm lemma="emission" tokens="t23">
 <fs>
    <f name="pos"><symbol value="noun"/></f>
    <f name="number"><symbol value="singular"/></f>
  </fs>
</wordForm>
```

4.1.3.1 Compact morpho-syntactic tags

FSR provides ways for the compact representation of feature structures, by relying on libraries naming feature values and feature specifications (a feature specification being a pair formed by a feature and a value). These names may be used in wordForm attribute tag to get compact tags, following a standard practice in the NLP community.

```
<token id="t23" from="" join="no">emission</token>
<wordForm   tokens="t23"
            entry="urn:lexicon:en:emission"
            tag="pos.noun num.sing"/>
```

The generic way provided by FSR to use libraries is illustrated by the following example, with the attribute feats of element fs:

```
<!-- A feature value library -->
<fvLib n="French morpho values">
     <symbol xml:id="noun" value="noun"/>
     <symbol xml:id="sing" value="singular"/>
     <symbol xml:id="plu" value="plural"/>
     <symbol xml:id="masc" value="masculine"/>
     <symbol xml:id="fem" value="feminine"/>
</fvLib>
<!-- A feature specification library -->
<fLib>
   <f xml:id="pos.n" name="pos" fVal="noun"/>
   <f xml:id="num.s" name="number" fVal="sing"/>
   <f xml:id="num.p" name="number" fVal="plu"/>
   <f xml:id="gen.f" name="gender" fVal="fem"/>
   <f xml:id="gen.m" name="gender" fVal="masc"/>
</fLib>
```

With such a library, following FSR rules, one may write:

```
<wordForm lemma="climate_change" tokens="t1">
      <fs feats ="pos.n num.s "/>
</wordForm>
```

or, equivalently, by using attribute tag, one may write:

```
<wordForm tokens="t1 t2"
          lemma="climate_change"
          tag ="pos.n num "/>
```

Disjunctive values are allowed by FSR and may also be simplified, following the same mechanism:

```
<!-- A feature value library -->
<tagset>
  <fvLib>
     <vAlt xml:id="first.third">
          <symbol value="first"/>
          <symbol value="third"/>
     </vAlt>
     <symbol xml:id="verb" value="verb"/>
     <symbol xml:id="sing" value="singular"/>
  </fvLib>
  <!-- A feature specification library -->
  <fLib>
    <f xml:id="pers.13" name="pers" fVal="first.third">
    </f>
    <f xml:id="pos.v" name="pos" fVal="verb"/>
    <f xml:id="num.s" name="number" fVal="sing"/>
  </fLib>
</tagset>
<!-- Annotated document -->
<token  id="t0">porte</token>
<wordForm tokens="t0"
          entry="urn:lexicon:fr:porter"
          tag="pos.v pers.13 num.s"/>
```

4.1.4 Handling ambiguities

Ambiguities naturally arise when handling natural language, especially for automatically produced annotations. Ambiguities may occur at various levels and, therefore, MAF proposes several alternatives to cope with ambiguities as simply as possible.

4.1.4.1 Word form Content Ambiguities

The FSR proposal provides several ways to represent ambiguities, for instance at the level of feature values. These mechanisms may be used to handle the ambiguities occurring within the morpho-syntactic content of a word-form. For instance, the French inflected verb form "mange" (to eat) is ambiguous between the 1st and 3rd persons, and this ambiguity can be captured by the **vAlt** element present in FSR:

```
<token  id="t0">mange</token>
<wordForm tokens="t0" entry="urn:lexicon:fr:manger">
   <fs>
     <f name="pos"><symbol value="verb"/></f>
     <f name="aux"><symbol value="avoir"/></f>
     <f name="mood"><symbol value="indicative"/></f>
```

```
        <f name="tense"><symbol value="present"/></f>
        <f name="person">
             <vAlt>
                <symbol value="first"/>
                <symbol value="third"/>
             </vAlt>
        </f>
        <f name="number"><symbol value="singular"/></f>
      </fs>
  </wordForm>
```

A compact tag notation can still be used by registering most frequent cases of ambiguities in FSR libraries.

```
<token id="t0">mange</token>
<wordForm tokens="t0"
          entry="urn:lexicon:fr:manger"
          tag="pos.v aux.avoir mood.i tense.p pers.13 num.s"/>
```

4.1.4.2 Lexical Ambiguities

Ambiguities between different lexical entries for a same sequence of tokens can be handled by the element **wfAlt**:

```
<token id="t0">porte</token>
<wfAlt>
  <wordForm tokens="t0" entry="lexicon:porte"  tag="pos.n ..."/>
  <wordForm tokens="t0" entry="lexicon:porter" tag="pos.v ..."/>
</wfAlt>
```

4.1.4.3 Structural Ambiguities

For instance, the French textual sequence "*fer à cheval*" (horse shoe) can still be decomposed into several readings ("[horse shoe]", "[iron] [on horse]", "[iron] [of] [horse]"), giving the following lattice representation:

```
<token id="t1">fer</token>
<token id="t2">à</token>
<token id="t3">cheval</token>
<fsm init="S0" final="S3">
   <transition source="S0" target ="S3">
      <wordForm  tokens="t1 t2 t3"
                 entry="urn:lex:fr:fer_%E0_cheval"
                 lemma="fer_à_cheval"/>
   </transition>
   <transition source="S0" target ="S1">
      <wordForm entry="urn:lex:fr:fer" tokens="t1"/>
   </transition >
   <transition source="S1" target ="S2">
      <wordForm tokens="t2"
                entry="urn:lex:fr:%E0" lemma="à"/>
   </transition>
   <transition source="S2" target ="S3">
      <wordForm  tokens="t3" entry="urn:lex:fr:cheval"/>
   </transition>
   <transition source="S1" target ="S3">
      <wordForm  tokens="t2 t3"
                 entry="urn:lex:fr:%E0_cheval" lemma="à_cheval"/>
   </transition>
</fsm>
```

The linguistic units "fer à cheval", "fer", "à", "cheval", and "à cheval" correspond to minimal syntagmatic units that can be annotated. Additional information could be added to edges such as probabilities.

4.2 SYNAF

The Syntactic Annotation Framework (SynAF) is a high level model for representing the syntactic annotation of textual documents. SynAF has been built on the MAF (Morpho-Syntactic Framework) proposal. MAF is dealing with the morpho-syntactic annotation of specific segments of textual documents. The morpho-syntactic annotation framework is about *part of speech* (noun, adjective, verb, etc.), *morphological* and *grammatical* features (such as number, gender, person, mood, verbal tense). SynAF is about the annotation of the syntactic constituency of such morpho-syntactically annotated fragments and the syntactic dependency relations existing between those morpho-syntactically annotated fragments.

This makes possible *t*o represent linguistic constituencies like Noun Phrases (NP), which describe a structured sequence of morpho-syntactically annotated items, or to represent dependency relations, like head-modifier relation, finding out simple taxonomies. The dependency information can exist between morpho-syntactically annotated items within a phrase (an adjective is the modifier of the head noun within an NP) or describe a specific relation between syntactic constituents at the clausal and sentential level (i.e. an NP being the "subject" of the main verb of a clause or sentence).

4.2.1 The SynAF diagram

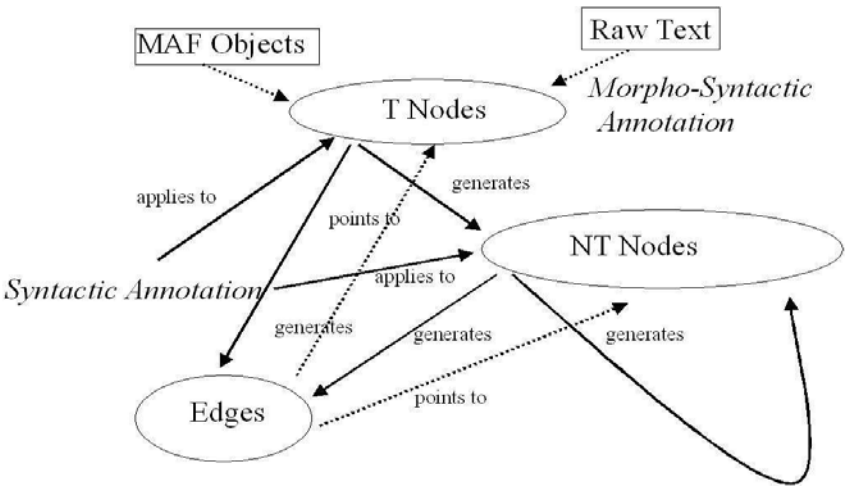

Figure 4 - SYNAF metamodel

4.2.1.1 T Nodes class

The *t_nodes* class represents the terminal nodes of a syntax tree, mostly consisting of morpho-syntactically annotated words, but empty elements are allowed. The *t_nodes* are defined over a *span*, a pair of points identifying a segment of the document submitted to syntactic annotation. This can be a multiple span (for accounting for discontinuous constituents). The t_nodes are labeled with syntactic categories valid for the word level.

4.2.1.2 NT Nodes class

The *nt_nodes* class represents the non-terminal nodes of a syntax tree, mostly consisting of *t_nodes* and *nt_nodes*, but empty elements are allowed. The *nt_nodes* are also defined over a (possibly multiple) *span.* The *nt_nodes* are labeled with syntactic categories valid at the phrasal level and higher (clausal, sentential).

4.2.1.3 Edges class

The *Edges* class represents the dependency relation between nodes (both terminal and nonterminal nodes). The dependency relation is a binary one and consists of a label name and a pair of source and target nodes.

4.2.1.4 Syntactic Annotation class

The *Syntactic Annotation* class represents the application of syntactic information to MAF annotated input. It can be either a manual or an automatic application. When syntactic annotation is applied to nodes (non-terminal or terminal), then it generates either a new (nonterminal) node or a dependency edge.

4.2.2 Data Categories for SynAF

4.2.2.1 Constituency

Constituency_labels	Meaning
AA	superlative phrase with am (for German)
AP	adjective phrase
AVP	adverbial phrase
CAC	coordinated adposition
CAP	coordinated adjective phrase
CAVP	Coordinated adverbial phrase
CCP	Coordinated complementiser
CH	Chunk (non-recursive constituent)
CNP	Coordinated noun phrase
CO	coordination
CPP	Coordinated adpositional phrase
CVP	Coordinated verb phrase (nonfinite)
CVZ	Coordinated infinitive with zu (for German)
NP	noun phrase
PN	proper noun
PP	adpositional phrase (prepositional and postpositional phrases
S	Sentence
VP	verb phrase (non-finite)
VZ	infinitive with zu (for German)

4.2.2.2 Dependency

In the following we present the candidate data categories for dependency structures (the labels of edges in the annotation graph).
mod: indicates the word introducing the dependent in a head-modifier relation
mod(of,gift,book) the gift of a book
mod(by,gift,Peter) the gift of a book by Peter
mod(of,examination,patient) the examination of the patient
mod('s,doctor,examination) the doctor's examination of the patient

subj: indicates the subject in the grammatical relation Subject-Predicate. The relation between a predicate and its subject; where appropriate, the **initial_gr** indicates the syntactic link between the predicate and subject before any GR-changing process.
subj(arrive,John,_) John arrived in Paris
subj(employ,Microsoft,_) Microsoft employed 10 C programmers
subj(employ,Paul,obj) Paul was employed by Microsoft

csubj, xsubj, ncsubj: The Grammatical Realtions (RL) s **csubj** and **xsubj** may be used for clausal subjects, controlled from within, or without, respectively. **ncsubj** is a non-clausal subject.
xsubj(win,require,_) to win the America's Cup requires heaps of cash

dobj: Indicates the object in the grammatical relation between a predicate and its direct object.
dobj(read,book,_) read books

iobj The relation between a predicate and a non-clausal complement introduced by a preposition; type indicates the preposition introducing the dependent.
iobj(in,arrive,Spain) arrive in Spain
iobj(into,put,box) put the tools into the box
iobj(to,give,poor) give to the poor

obj2: The relation between a predicate and the second non-clausal complement in ditransitive constructions.
obj2(head,dependent)
obj2(give,present) give Mary a present
obj2(mail,contract) mail Paul the contract

4.2.3 Example

The first example shows the constituency annotation and the second one the related dependency annotation.

```
<frase id="0" morfofile="sole.morph026" rs="Presentato un report ambientale del
WWF.">
  <nodo tipo="F3">
    <nodo tipo="SV3" id="0">
      <foglia lemma="presentare" href="mw_001"/>
      <nodo tipo="COMPT" id="1">
        <nodo tipo="SN" id="2">
          <foglia lemma="un" href="mw_002"/>
          <foglia lemma="report" href="mw_003"/>
          <nodo tipo="SA" id="3">
            <foglia lemma="ambientale" href="mw_004"/>
          </nodo>
          <nodo tipo="SPD" id="4">
            <foglia lemma="di" href="mw_005"/>
            <nodo tipo="SN" id="5">
              <foglia lemma="WWF" href="mw_006"/>
            </nodo>
          </nodo>
        </nodo>
      </nodo>
    </nodo>
    <foglia lemma="." href="mw_008"/>
  </nodo>
</frase>

<frase id="0" morfofile="sole.morph026" rs="Presentato un report ambientale del
WWF">
      <partec partec_id="partec_000"
            lemma="presentare" modo="part_pass"
            href="mw_001"/>
      <partec partec_id="partec_001" lemma="report"
            definitezza="-" href="mw_003"/>
      <partec partec_id="partec_002" lemma="ambientale"
            href="mw_004"/>
      <partec partec_id="partec_003" lemma="WWF"
            definitezza="+" introdep="di" href="mw_006"/>
      <relfunz relazione_funzionale="mod"
            partec1_id="partec_001" partec2_id="partec_002"
            relfunz_id="r_000"/>
      <relfunz relazione_funzionale="mod"
            partec1_id="partec_001" partec2_id="partec_003"
            relfunz_id="r_001"/>
      <relfunz relazione_funzionale="mod"
            partec1_id="partec_001" partec2_id="partec_000"
            relfunz_id="r_003"/>
</frase>
```

4.3 Conclusions

We decided to remove MAF and SYNAF from the system design. Instead of that, we added to the KAF format some syntactic layers, thus representing among the different KAF levels also the Morphological and syntactic levels.

Basic motivation for that were that MAF is not finalized and complete, and that current documents are not consistent. Moreover SYNAF contains a lot of information that we do not need and, embedding representation of data into the original text documents, it complicates the representation and manipulation of information.

5 Semantic Annotation

KAF (Kyoto Annotation Framework), formerly called SEMAF in older project documentation, has been defined to be used within the KYOTO project. KAF aims to provide a reference format for the representation of syntactic and semantic annotations.

KAF comprises several annotations over a text at different syntactic levels (tokens, words, lemmas, terms,) and semantic levels (synsets, roles, quantifier detection, temporal relations, etc), and adopts a stand off strategy for annotating the source text:

- elements are used for grouping linguistic elements.
- Linguistic annotations of a particular level always spans elements of previous levels.
- Linguistic annotations of different levels are not mixed.

We will describe the annotation levels in turn, using the sentence "John taught mathematics 20 minutes every Monday in New York." as a running example, using version 0.5 of KAF[1], as described in [33].

5.1 Root element

All KAF documents have a root element <KAF> which has the following attributes:

- xml:lang: language identifier, as described in [0]
- **doc:** The identifier for the source document.

Ex:

```
<KAF xml:lang="en" doc="KYOTO_3_3012">
<!--- ... --->
</KAF>
```

5.2 Word forms

After tokenization step, all word forms are annotated within the <text> element, and each form is enclosed by a <wf> element.

[1] The consortium continuously updates the KAF format, improving its features.
To keep track of the different revisions it is possible to access the Kyoto Subversion repository, where also the KAF DTD and the KAF XML Schema are maintained.
The address to retrieve all KAF related files and to download the latest KAF version is:
https://kyoto.let.vu.nl/svn/kyoto/trunk/doc/user/KAF

<wf> elements have the following attributes:

- wid: the unique id for the word form.
- sent: sentence id of the token (optional)
- para: paragraph id (optional)
- page: page id (optional)
- xpath: in case of source xml files, the xpath expression identifying the token (optional)

Ex:

```
<text>
  <wf wid="w1">John</wf>
  <wf wid="w2">taught</wf>
  <wf wid="w3">mathematics</wf>
  <wf wid="w4">20</wf>
  <wf wid="w5">minutes</wf>
  <wf wid="w6">every</wf>
  <wf wid="w7">Monday</wf>
  <wf wid="w8">in</wf>
  <wf wid="w9">New</wf>
  <wf wid="w10">York</wf>
  <wf wid="w11">.</wf>
</text>
```

5.3 Terms

Terms refer to previous word forms (and groups multi word forms) and attach lemma, part of speech, synset and name entity information.

<term> elements have the following attributes:

- tid: unique identifier
- type: type of the term. Currently, 3 values are possible:
  - open: open category term
  - close: close category term
  - entity: term is a named entity
- lemma: lemma of the term
- pos: part of speech

The first letter of the pos attribute must be one of the following:

N common noun
R proper noun
G adjective
V verb
P preposition
A adverb
C conjunction
D determiner
O other

more complex pos attributes may be formed by concatenating values separated by a dot ".". For example, in Basque we have “V.ADI.SIN” for simple verbs or “V.ADI.KON” for complex verbs.

- `netype`: if the term is a named entity, the type of the entity (only if type=”entity”)

`<sense>` elements have the following attributes:

- `sensecode`: code of wordnet synset
- `confidence`: confidence weight of the association

Ex:

```
<terms>
  <term tid="t1" type="entity" lemma="John" pos="N" netype="person">
    <span>
      <target id="w1"/>
    </span>
  </term>
  <term tid="t2" type="open"  lemma="teach" pos="V">
    <span>
      <target id="w2"/>
    </span>
    <senseAlt>
    <sense sensecode="EN-17-00861095-v" confidence="0.80"/>
    <sense sensecode="EN-17-00859568-v" confidence="0.20"/>
    </senseAlt>
  </term>
  <term tid="t3" type="open"  lemma="mathematics" pos="N">
    <span>
      <target id="w3"/>
    </span>
    <senseAlt>
    <sense sensecode="EN-17-04597590-n" confidence="1.0"/>
    </senseAlt>
  </term>
  <term tid="t4" type="entity" lemma="20" pos="Z" netype="number">
    <span>
      <target id="w4"/>
    </span>
  </term>
  <term tid="t5" type="open" lemma="minute" pos="N">
    <span>
      <target id="w5"/>
    </span>
  </term>
  <senseAlt>
    <sense sensecode="EN-17-12621100-n" confidence="0.80"/>
    <sense sensecode="EN-17-12631889-n" confidence="0.06"/>
    <sense sensecode="EN-17-12630443-n" confidence="0.01"/>
    <sense sensecode="EN-17-11241911-n" confidence="0.01"/>
    <sense sensecode="EN-17-05339359-n" confidence="0.01"/>
    <sense sensecode="EN-17-04316149-n" confidence="0.01"/>
  </senseAlt>

  <term tid="t5" type="close" lemma="every" pos="D">
    <span>
      <target id="w6"/>
    </span>
  </term>

  <term tid="t6" type="entity" lemma="Monday" pos="N" netype="date">
```

```
      <span>
        <target id="w7"/>
      </span>
      <senseAlt>
      <sense sensecode="EN-17-12557842-n" confidence="1.0"/>
      </senseAlt>
    </term>
    <term tid="t7" type="close" lemma="in" pos="P">
      <span>
        <target id="w8"/>
      </span>
    </term>
    <term tid="t8" type="entity" lemma="New_York" pos="N" netype="location">
      <span>
        <target id="w9"/>
        <target id="w10"/>
      </span>
    </term>
  </terms>
```

5.4 Dependencies

Dependencies represent dependency relations among terms. Each dependency is represented by an empty <dep> element and span previous terms. <dep> element have the following attributes:

- from: term id of the source element
- to: term id of the target element
- rfunc: relational function. One of:
  - mod: indicates the word introducing the dependent in a head- modifier relation.
    Ex:
    mod(by,gift,Peter) the gift of a book by Peter
    mod(of,examination,patient) the examination of the patient
  - subj: indicates the subject in the grammatical relation Subject-Predicate.
    Ex:
    subj(arrive,John,_) John arrived in Paris
    subj(employ,Microsoft,_) Microsoft employed 10 C programmers
    subj(employ,Paul,obj) Paul was employed by Microsoft
  - csubj, xsubj, ncsubj: The Grammatical Realtions (RL) s csubj and xsubj may be used for clausal subjects, controlled from within, or without, respectively. ncsubj is a non-clausal subject.
    Ex:
    xsubj(win,require,_) to win the America's Cup requires heaps of cash
  - dobj: Indicates the object in the grammatical relation between a predicate and its direct object.
    Ex:
    dobj(read,book,_) read books
  - iobj: The relation between a predicate and a non-clausal complement introduced by a preposition; type indicates the preposition introducing the dependent.
    Ex:
    iobj(in,arrive,Spain) arrive in Spain
    iobj(into,put,box) put the tools into the box
    iobj(to,give,poor) give to the poor
  - obj2: The relation between a predicate and the second non-clausal complement in ditransitive constructions.

Ex:
obj2(head,dependent)
obj2(give,present) give Mary a present
obj2(mail,contract) mail Paul the contract

Ex:

```
<deps>
  <!-- subj(teach, John) -->
  <dep from="t1" to="t2" rfunc="subj" />
  <!-- dobj(teach, Mathematics) -->
  <dep from="t3" to="t2" rfunc="dobj" />
  <!-- iobj(teach, New_York) -->
  <dep from="t8" to="t2" rfunc="iobj" />
</deps>
```

5.5 Chunks

Chunks are noun or prepositional phrases, spanning terms.
<chunk> elements have the following attributes:

- cid: unique identifier
- head: the chunk head's term id
- phrase: typo of the phrase

Valid values for the phrase elements are one of the following:

NP	noun phrase
VP	verbale phrase
PP	prepositional phrase
S	sentence
O	other

- case (optional): declension case

```
<chunks>
  <!-- John -->
  <chunk cid="c1" head="t1" phrase="NP">
    <span>
      <target id="t1"/>
    </span>
  </chunk>
  <!-- taught -->
  <chunk cid="c2" head="t2" phrase="V">
    <span>
      <target id="t2"/>
    </span>
  </chunk>
  <!-- Mathematics -->
  <chunk cid="c3" head="t3" phrase="NP">
    <span>
      <target id="t3"/>
    </span>
  </chunk>
  <!-- 20 minutes -->
  <chunk cid="c5" head="t5" phrase="NP">
    <span>
      <target id="t4"/>
      <target id="t5"/>
```

```
    </span>
  </chunk>
  <!-- every -->
  <chunk cid="c6" head="t6" phrase="R">
    <span>
      <target id="t6"/>
    </span>
  </chunk>
  <!-- in New York -->
  <chunk cid="c8" head="t9" phrase="PP">
    <span>
      <target id="t8"/>
      <target id="t9"/>
    </span>
  </chunk>
</chunks>
```

5.6 Events

Events provide event information, including roles, spanning chunks. The specific semantics of <event> elements is defined in [1].
<events> elements have the following attributes:

- eid: unique identifiers
- span: chunk id of the main event
- lemma: lemma of the event
- pos: part of speech
- eiid:
- class: event class
- tense:
- aspect:
- polarity:

Ex:

```
<events>
  <event eid="e1" span="c2" lemma="teach" pos="V" eiid="ei1" class="OCCURRENCE"
         tense="PAST" aspect="NONE" polarity="POS">
    <roles>
      <role cid="c1" role="agent"/>
      <role cid="c2" role="subject"/>
      <role cid="c3" role="location"/>
      </roles>
  </event>
</events>
```

5.7 Quantifiers

Quantifiers are annotated within <quantifiers> element. Normally, they are further used for specifying relations. The specific semantics of <quantifier> elements is defined in [1], the main difference being that on KAF quantifiers refer to chunks.

<quantifier> elements have the following attributes:

- qid: unique identifier
- span: chunk id of quantifier

Ex:

```
<!-- every -->
<quantifiers>
  <quentifier qid=”q1” span=”c6”/>
</quantifiers>
```

5.8 Time expressions (timex)

Time expressions are annotated within <timexs> element. The specific semantics of <timex> elements are defined in [1], the main difference being that on KAF quantifiers refer to chunks.

Ex:

```
<!-- 20 minutes every monday -->
<timexs>
  <timex3 texid="tex1" type="DURATION" value="P20TM">
    <span>
      <target id="c5"/>
    </span>
  </timex3>

  <timex3 texid="tex2" type="SET" value="xxxx-wxx-1" quant="EVERY">
    <span>
      <target id="c7"/>
    </span>
  </timex3>

  <tlink timeID="tex1" relatedToTime="tex2" relType="IS_INCLUDED"/>
  <tlink eventInstanceID="ei1" relatedToTime="tex1" relType="SIMULTANEOUS"/>
</timexs>
```

5.9 General relations

General relations are annotated within the <relations> element. There are two types of relations elements, <qrelation> and <trelation>, for specifying relations among quantifiers or time expressions, respectively. The trelation and qrelation elements semantics are defined in [1].

6 Fact Annotation

Within the Kyoto project, the goal of *concept extraction* is to acquire generic domain knowledge - knowledge which is true under any circumstances. Specific knowledge (so-called `facts') is extracted by means of fact mining. Facts generally refer to instances rather than classes of processes and concepts. A fact is an assertion of something which may or may not happen (can be true or false) at a particular place and time, given one or more entities.

FactAF is the representation format of facts that was designed for Kyoto Purposes. FactAF was defined using existing standards where possible.

6.1 Related work

FactAF draws heavily from previous work. We present examples consistently in XML. XML is chosen because of its flexibility with respect to using different types of annotation. However, XML is not usually essential and other formats may be usable or preferred in specific cases.

We make a distinction between *linear annotation* and *generic annotation* of text. A linear annotation consists of tags in the text. If the tags are removed, the original unannotated text is recovered. The following is an example of a linear annotation of the sentence, ``Temperate and tropical species populations declined by around 30 per cent overall from 1970 to 2003''.

```
<term>Temperate and tropical species populations</term>
<process>declined</process>
by <quantity>around 30 per cent</quantity> overall
<period from="1970" to="2003">from 1970 to 2003</period>.
```

In contrast, a generic annotation is a representation of generic knowledge in the text. This may require reordening, etc. The annotation is separated from the text. The following is an example of a generic annotation.

```
<process type="decline">
 <participant role="pantient" quantity="around 30 per cent">
  temperate and tropical species populations</participant>
 <period from="1970" to="2003">from 1970 to 2003</period>
</process>
```

Hybrid solutions are possible, e.g. when generic annotation elements refer to linear annotations. The following is an example of a hybrid linear/generic annotation.

```
<text>
 <term id="1">Temperate and tropical species populations</term>
 <process id="2">declined</process> by <quantity id="3">around
 30 per cent</quantity> overall <period id="4" from="1970"
 to="2003">from 1970 to 2003</period>.
</text>
<process type="decline">
 <participant role="pantient" term="1" quantity="3"/>
 <time period="4"/>
</process>
```

The remainder of this section describes some existing standards and resources relevant for fact annotation.

6.1.1 Linear annotation of time and events: SemAF

SemAF is an ISO XML layer of linear semantic annotation of text. SemAF is a representation format of time and events, and relations between them. An event occurs in time but it may take anything from a point in time to an extended period of time. The following is an example of a SemAF-annotated sentence (SemAF part 1, page 16): ``John taught 20 minutes every Monday.''

```
John
<event eid="e1" eiid="ei1" class="OCCURRENCE" pos="VERB"
 tense="PAST" aspect="NONE" polarity="POS">
taught
</event>
<timex3 tid="t1" type="DURATION" value="P20TM">
20 minutes
</timex3>
<timex3 tid="t2" type="SET" value="xxxx-wxx-1" quant="EVERY">
every Monday
</timex3>
<tlink timeID="t1" relatedToTime="t2" relType="IS_INCLUDED"/>
<tlink eventInstanceID="ei1" relatedToTime="t1"
 relType="SIMULTANEOUS"/>
```

A SemAF annotation provides an interpretation of expressions of events and time, but more is needed for representation of facts. Essential for an event to become a meaningful fact is the participants who are involved in the event. SemAF makes no effort to relate events to their participants. For instance, in the above example, *John* is not tagged or related to the event of teaching.

The concept of *teaching* provides a frame which must be filled in in order to move from the concept to a fact. Teaching happens at a particular place, at a particular time. There will be somebody who does the teaching, and somebody who is being taught. If we know all of this, we have a fact instantiating the concept of teaching.

6.1.2 Template-based knowledge representation: FrameNet

A process has a set of parameters valid for that specific class of processes. For instance, if there is a declination process, there may also be something which declines, a rate at which it declines, etc. These parameters are likely to be found in the text, but their syntactic and lexical realization may vary. The FrameNet project aims to capture frames, i.e. templates consisting of a process or object and possible sets of parameters (or elements in FrameNet terminology). In addition, FrameNet relates frames to their possible realizations in free text. FrameNet imposes constraints on which frames or frame elements are realized by means of which lexical units.

```
<text>
 <term id="1">Populations</term> of <term id="2">terrestrial
 species</term> <event id="3">declined</event> by <quantity
 id="4">about 30 per cent on average</quantity>
 <time id="5">between 1970 and 2003</time>.
</text>
```

```
<frame type="Change_position_on_a_scale" event_id="3">
 <element role="attribute" term_id="1"/>
 <element role="item" term_id="2"/>
 <element role="difference" quantity_id="4"/>
 <element role="time" time_id="5"/>
</frame>
```

In Kyoto, an attractive feature of frames is that they formalize the type of elements that may fill frame slots. Filled frames carry meaning. However, the FrameNet project has no ambition to facilitate reasoning with frame elements. FrameNet prescribes that a `Change_position_on_a_scale` frame may have a `difference` element, an `initial_value` element and a `final_value` element, but there are no constraints on the relation between these values.

FrameNet does provide clues for automatic frame extraction by defining relations between lexical units and frames. If the text does not provide all necessary frame elements explicitly, a partial frame may be completed by inferencing, but how this is done would have to be determined by an external inferencing mechanism.

Another limitation of FrameNet is that it is available for few languages. The frame definitions themselves may be largely language independent, but relations between frames and lexical units certainly are not.

6.1.3 Ontology: Sumo+Milo

The Sumo project (Suggested Upper Merged Ontology) has generated a set of logical expression, representing the meaning of concepts such as `Decreasing`. Milo is a mid-level ontology which is the glue between Sumo and domain ontologies. Sumo and Milo concepts are mapped to WordNet synsets.

Example: in WordNet, one of the senses of the noun *decline* is defined as ``change toward something smaller or lower.'' This sense is linked to the Sumo concept `Decreasing`. Sumo defines this concept as follows:

```
(=>
 (and
   (instance ?DECREASE Decreasing)
   (patient ?DECREASE ?OBJ))
 (exists (?UNIT ?QUANT1 ?QUANT2)
   (and
     (holdsDuring
       (BeginFn
         (WhenFn ?DECREASE))
       (equal
         (MeasureFn ?OBJ ?UNIT) ?QUANT1))
     (holdsDuring
       (EndFn
         (WhenFn ?DECREASE))
       (equal
         (MeasureFn ?OBJ ?UNIT) ?QUANT2))
     (lessThan ?QUANT2 ?QUANT1))))
```

This definition translates to the following (from ontologyportal.org):

- if a process is an instance of decreasing and a real number is a patient of process

- then there exist an unit of measure a constant quantity and constant quantity so that real number unit of measure(s) is equal to constant quantity holds during the beginning of the time of existence of process and real number unit of measure(s) is equal to constant quantity holds during the end of the time of existence of process and constant quantity is less than constant quantity

In Kyoto, Sumo and Milo may be useful for reasoning with text. For instance, if a textual statement provides an initial value, the Sumo expression can be used to infer that the final value is less than that. If the text also provide a difference (e.g. decline by 30 per cent), the final value can be derived, provided the Sumo definition is extended to introduce the notion of *difference*.

In combination with FrameNet, Sumo or an extended Sumo-like ontology could be used to derive the value of missing frame elements. Most likely, the Kyoto project requires a to be developed domain specific extension of Sumo. The ambition of Kyoto to develop a reusable application implies that a domain specific ontology requires effort from users (and functionality in the Wikyoto).

6.2 Fact extraction in Kyoto

Frames (as defined by FrameNet) and ontologies are complementary and may both provide useful knowledge in Kyoto. Frames link processes and their attributes with their lexical realizations; ontologies provide relations between them which can be used for inferencing. Fact extraction is responsible for detecting frames and filling frame slots (correctly associating frames with elements). A frame may span multiple sentences. To find a frame's elements, we may rely on a linear annotation of concepts and other textual items. SemAF could play a role here. Explicit associations between frames and their lexical realizations considerably ease the process of detecting frames. These associations are part of FrameNet.

Some frame elements may be left unspecified or implicit in a frame. If a frame element is not specified, we have a *partial fact*. Partial facts are common and should not be ignored - it might be the case that a particular attribute is not relevant in the original context and is therefore left unspecified. Implicit elements are elements that need inferencing to be associated with an appropriate frame slot.

The combination of frames and ontologies may be powerful, but which resources are candidates for use in Kyoto? Unfortunately, no known resource provides both frames and an inference mechanism satisfactorily. We could use for instance FrameNet and Sumo+Milo - since both resources are intended as general-purpose resources they have much overlap, but they are not aligned and creating a mapping between them is not trivial. Alternatives are to choose for FrameNet and then produce inference rules within Kyoto, or to choose for Sumo+Milo and then produce the mapping to lexical units within Kyoto.

Option 1: FrameNet: Starting from FrameNet as a formalization of which parameter sets are valid for which processes, we may have to define new frames for the domain knowledge we need to extract. This will have to be done using the Wikyoto. In addition, we will extend frames with a formalization of their attributes and the relation between them.

Option 2: Sumo+Milo: Sumo+Milo define knowledge which could be used for inferencing in Kyoto, but the textual realization of this knowledge remains largely unspecified. In order to do information extraction properly, we need to implement this missing link between text and knowledge. For a particular process such as *Decreasing*, Sumo+Milo do specify which arguments the process may take - in this case an object (*a patient*), a unit, and two quantities. This can be used to generate a frame-like structure. The relation between frames and lexical units are then based on user interaction, i.e. the Wikyoto.

In this scenario, fact extraction will consist of the following steps:

1. frames are automatically extracted from the ontology (option 2);
2. the user selects an existing frame or defines a new domain frame;
3. the user specifies which frame attributes are relevant and how they are realized in text: always (option 2); or in case of a new frame only (option 1);
4. the user defines relations between frame attributes for automatic inferencing: always (option 1); or in case of a new frame only (option 2);
5. frame slots are filled with annotated text items automatically;
6. frame slots are filled by means of automatic inferencing;
7. repeat step 6.

The list above shows that in either case, the same type of user interaction must be implemented (which are probably our greatest challenge). This means that the choice for a particular resource should be motivated by the usefulness of the resource rather than on a fundamental decision between frames or ontology.

6.3 Fact representation in Kyoto: FactAF

A FactAF annotation consists of two parts. Part one is the text with a linear semantic annotation of expressions of events (SemAF), time (SemAF), terms, quantifiers, and relations between them. The second part is a set of facts whose evidence is in the text. Each fact consists of a FrameNet-like representation of a process and its arguments. Some arguments apply to any process, such as location and time. A process (e.g. *cover*) in FactAF corresponds to an event in SemAF. Annotation of expressions of time (e.g. *now*) and events is done in accordance with SemAF. A process' argument may be a term (e.g. *Europe*), a quantifier (e.g. *about 4.4%*) or a relation between terms (e.g. *non-EU Europe excluding the Russian Federation*) or quantifiers. The following is a FactAF sample.

```
<text>
 <term tid="1">Wetlands</term>
 <semaf:event eid="1">provide</semaf:event>
  <quantifier qid="1">multiple</quantifier>
  <term tid="2">social, economic and environmental benefits</term>,
  <trelation rid="1" reltype="inclusive" arg1="2" arg2="3">for
   example</trelation>
  <term tid="3">water flows regulation</term>.
</text>
<facts>
 <fact fid="1" frame="Provide" confidence="1">
  <process eid="1"/>
  <arg aid="1" role="agens" confidence=".8"/>
  <arg aid="2" role="patient" qid="1" rid="1"/>
 </fact>
```

```
</facts>
```

The linear annotation is in the *text* element in the above annotation. The *text* element contains running text in which segments are semantically tagged. Each term, quantifier, etc. receives a unique identifier which can be referred to. The facts are in the *facts* element, following the tagged text. Constituents of each fact refer to tags in the text by means of their identifiers. For instance, the fact with `fid=``3''` specifies a process, one argument, a time and a location. The process refers to the event with `eid=``3''`; the argument refers to the relation with `rid=``4''`; the time refers to the expression of time with `xid=``1''`; and the location refers to the term with `tid=``10''`.

Note that the time and place are left implicit in the other two facts. However, they may be derived from meta information of the source, such as the publication date (in this case, 2003) and the scope of the document (in this case, Europe's environment). Although not done in the example above, filling in these empty slots using meta information may be desirable. FactAF supports representation of partial facts so that facts can be built up iteratively.

Also, the degree to which elements in the linear annotation are parsed may vary. For instance, TimeML could be used, which allows for very rich representation of expressions of time. If an expression cannot be parsed, it may just be annotated as an expression of time to which a frame may refer, e.g. `<time>recently</time>`.

Essential is the representation of confidence. Strict inferencing may make little difference, while loose reasoning could overgenerate facts. The confidence value distinguishes reliable facts and attributions from less reliable (but possibly useful) ones.

7 Wordnets

7.1 Description of KYOTO-LMF representation format

The format for representing WordNets inside the Kyoto project (henceforth, "Kyoto-LMF wordnet format") adopted as a reference LMF (Lexical Markup Framework), version 16. Major lexical objects and the general framework are the same. Kyoto wordnet format deviates from standard LMF only regarding the way to handle data categories: in LMF, these are represented by means of attribute-value pairs that are instantiated as separate XML elements. In Kyoto-LMF wordnet format we decided to represent the same information by means of XML attributes and values instead of nested elements. In this respect, the Kyoto-LMF wordnet format has to be seen as an LMF dialect. This decision is motivated on the basis of better parsing efficiency.
For instance, this is a snapshot in standard LMF:

```
<Synset id="ENG-16-06060223-n">
        <feat att="baseConcept" val="1"/>
        <feat att="author" val="piek"/>
        <feat att="date" val="2008-05-12"/>
        <Definition>
         <feat att="gloss" val="bla bla"/>
        <Statement>
                        <feat att="example" val="bla bla"/>
```

```
                </Statement>
        </Definition>
        <SynsetRelation targets="ENG-16-06056130-n">
                <feat att="relType" val="has_hyperonym"></feat>
                <feat att="cs" val="99"></feat>
                <feat att="status" val="yes"></feat>
                <feat att="source" val="whatsoever"></feat>
                <feat att="author" val="german"></feat>
                <feat att="date" val="2008-05-12"/>
        </SynsetRelation>
        <MonolingualExternalRef>
                <feat att="externalSystem" val="SUMO"></feat>
                <feat att="externalReference" val="PoliticalProcess"></feat>
                <feat att="relType" val="at"></feat>
                <feat att="author" val="monica"></feat>
                <feat att="date" val="2008-05-27"/>
        </MonolingualExternalRef>
</Synset>
```

This is the corresponding translation in the Kyoto dialect:

```
<Synset id="ENG-16-06060223-n" baseConcept="1">
        <meta author="piek" date="2008-05-12"/>
        <Definition gloss="bla bla">
                <Statement example="bla bla"/>
        </Definition>
        <SynsetRelations>
                <SynsetRelation targets="EU-16-06056130-n" relType="has_hyperonym">
                        <meta author="german" date="2008-05-12" status="yes"
                        source="whatsoever" confidenceScore="99"/>
                </SynsetRelation>
        </SynsetRelations>
        <MonolingualExternalRefs>
                <MonolingualExternalRef externalSystem="SUMO"
                externalReference="PoliticalProcess" relType="at">
                        <meta author="monica" date="2008-05-27"/>
                </MonolingualExternalRef>
        </MonolingualExternalRefs>
</Synset>
```

7.2 Description of KYOTO representation format

7.2.1 LexicalResource

LexicalResource is the root element, as in LMF. It has three children:

- GlobalInformation
- Lexicon
- SenseAxes

A lexical resource can contain more than one lexicon, and inter-lingual correspondences are grouped in a section, separated from the lexical resources proper, containing only inter-lexicon correspondences.

LexicalResource				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
			GlobalInformation	1..1
			Lexicon	1..*
			SenseAxes	0..1

7.2.2 GlobalInformation

This is used to record general information about the lexical resource. The attribute "label" is a free text field.

GlobalInformation				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
label	<free text>	optional		

Example:

```
<GlobalInformation label="Proposal for Kyoto-internal WordNet representation"/>
```

7.2.3 Lexicon

This element contains a monolingual resource. Attribute 'languageCoding' has "ISO 639-3" as a fixed value. We recommend use of the standardized 3-letter language coding (e.g. eng, nld) for specifying the value of attribute 'language'. Attributes 'owner' and 'version' are used to declare copyright holder and resource version, respectively. 'label' is an optional attribute for recording any additional information that may be needed.
The *Lexicon* element has two child elements, *LexicalEntry* and *Synset*.

Lexicon				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
languageCoding	<free text>	fixed	LexicalEntry	1..*

label	<free text>	optional	Synset	0..*
language	<free text>	required		
owner	<free text>	required		
version	<free text>	required		

Example:

```
<Lexicon languageCoding="ISO 639-3" label="English Wordnet 1.6, Meaning" language="eng"
owner="Princeton" version="1.6">
```

7.2.4 LexicalEntry

This element is a container for representing a lexeme in a lexicon. A *LexicalEntry* element can contain one lemma and zero to many different senses. It has one attribute: 'id' (a unique identifier).

LexicalEntry				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
id	ID	optional	Meta	0..1
			Lemma	1..1
			Sense	0..*

Example:

```
<LexicalEntry id="Department_of_Justice">
```

7.2.5 Meta

The element Meta is used to encode administrative information. Attributes are:

- author
- date
- source: expresses the originating database/system. It is typically associated with *SynsetRelation* elements.
- status: a key expressing editing status of the parent element. Possible values are empty (=not confirmed), false (wrong to be deleted) or true (confirmed as ok) and sometimes yes (confirmed as ok).
- confidenceScore: a numeric value indicating the degree of certainty about a given element. Typically, it is specified for *SynsetRelation* and *MonolingualExternalRef* elements.

Meta				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
author	<free text>	optional		
date	<free text>	optional		
source	<free text>	optional		
status	empty, false, true, yes	optional		
confidenceScore	<free text>	optional		

Example:

```
<SynsetRelation targets="EU-16-00403152-n" relType="gloss">
        <Meta author="monica" date="2008-05-27" status="false" source="whatsoever"
        confidenceScore="0.3"/>
</SynsetRelation>
```

7.2.6 Lemma

This element represents a word form chosen by convention to designate the lexical entry.
Attribute 'writtenForm' is added in case the id of *LexicalEntry* is numerical and it takes Unicode strings as values. Attribute 'partOfSpeech' is attributed to *Lemma*, in conformance with LMF, and takes as its value the part-of-speech value that is in general specified for a synset.

Lemma				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
writtenForm	<free text>	required		
partOfSpeech	<free text>	required		

Example:

```
<Lemma writtenForm="Department_of_Justice" partOfSpeech="N"></Lemma>
```

7.2.7 Sense

This element represents one meaning of a lexical entry. For wordnet representation, it represents the variant (or literal) of a synset. Attribute 'id' must be specified according to the convention used in wordnet, i.e. word_sense#nr. Attribute 'synset' takes as its value the ID of the synset to which the sense belongs. The element *Sense* can contain zero to one *Meta* elements and zero to one *MonolingualExternalRefs* elements.

Sense				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
id	ID	required	Meta	0..1
synset	IDREF	required	MonolingualExternalRefs	0..1

Example:

```
<Sense id="Department_of_Justice_1" synset="ENG-16-06060223-n">
        <MonolingualExternalRefs>
                <MonolingualExternalRef externalSystem="Wordnet3.0"
                externalReference="department_of_justice%1:14:00::"/>
        </MonolingualExternalRefs>
</Sense>
```

7.2.8 MonolingualExternalRefs

This is a bracketing element for grouping together all *MonolingualExternalRef* elements (see below). It must contain at least one of them.

MonolingualExternalRefs				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
			MonolingualExternalRef	1..*

Example:

```
<MonolingualExternalRefs>
        <MonolingualExternalRef externalSystem="Wordnet 3.0"
        externalReference="department_of_justice%1:14:00::"/>
</MonolingualExternalRefs>
```

7.2.9 MonolingualExternalRef

This element can be used to encode any reference or correspondence to an external resource or database. Its use is defined by slightly different conventions according to the particular parent element in which it appears. For instance, when occurring as a child of the *Sense* element, it can be used to express mapping between a sense and its correspondent in another lexical resource[2]. In the particular case of English WordNet it can also serve as a representational device to express SenseKey value.
When occurring inside the representation of the *Synset* element, then *MonolingualExternalRef* allows to i) encode reference to the domain; ii) express one or more links to an ontological system[3]; iii) encode synset mappings between different versions of WordNet.
The *MonolingualExternalRef* element has two required attributes, 'externalSystem' and 'externalReference', and the optional attribute 'relType'. The required attributes are used to express, respectively, the name of the external resource and the particular identifier or node. Possible values of the 'externalSystem' attribute are, for instance, 'domain', 'SuperSense', 'SUMO', 'TCO' (= Top Concept Ontology), and 'WordNet3.0' (for recording SenseKey values). A list of values
The attribute 'relType' serves to specify relations with nodes in SUMO ontology. Possible values are "at", "plus", "equal".

MonolingualExternalRef				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
externalSystem	<free text>	required	Meta	0..1
externalReference	<free text>	required		
relType	at, plus, equal	optional		

Example:

```
<MonolingualExternalRef externalSystem="Domain" externalReference="administration">
        <Meta author="monica" date="2008-05-27"/>
</MonolingualExternalRef>
```

[2] For example, see the Dutch instantiation, where linking to Cornetto database is encoded in this way.
[3] See again the Dutch and English instantiation, where linking to SUMO ontology is specified.

```
<MonolingualExternalRef externalSystem="Domain" externalReference="law">
        <Meta author="monica" date="2008-05-27"/>
</MonolingualExternalRef>
<MonolingualExternalRef externalSystem="SuperSense" externalReference="act">
        <Meta author="monica" date="2008-05-27"/>
</MonolingualExternalRef>
<MonolingualExternalRef externalSystem="SUMO" externalReference="PoliticalProcess"
relType="at">
        <Meta author="monica" date="2008-05-27"/>
</MonolingualExternalRef>
<MonolingualExternalRef externalSystem="TCO" externalReference="Agentive"/>
<MonolingualExternalRef externalSystem="TCO" externalReference="Purpose"/>
<MonolingualExternalRef externalSystem="TCO" externalReference="Social"/>
<MonolingualExternalRef externalSystem="TCO" externalReference="UnboundedEvent"/>
```

7.2.10 Synset

This element encodes information about a WordNet synset. A *Synset* element can link senses of different *LexicalEntry* instances within the same part of speech. Attributes for this element are the following:

- id: a unique identifier. The agreed syntax is "language code-version-id-pos tag"
- baseConcept: values for the baseConcept attribute will be numerical (1, 2, 3) which correspond to the BaseConcept sets

Synset elements can contain zero to one *Meta*, zero to one *Definition*, one *SynsetRelations* and one *MonolingualExternalRefs* bracketing elements.

Synset				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
id	ID	required	Meta	0..1
baseConcept	1,2,3	required	Definition	0..1
			SynsetRelations	1..1
			MonolingualExternalRefs	1..1

Example:

```
<Synset id="ENG-16-06060223-n" baseConcept="1">
        <Meta author="piek" date="2008-05-12"/>
        <Definition gloss="bla bla">
                <Statement example="bla bla"/>
        </Definition>
        <SynsetRelations>
                <SynsetRelation targets="EU-16-06056130-n" relType="has_hyperonym">
                        <Meta author="german" date="2008-05-12" status="yes"
                        source="whatsoever" confidenceScore="99"/>
                </SynsetRelation>
                <SynsetRelation targets="EU-16-06060479-n" relType="has_mero_part">
                        <Meta author="german" date="2008-05-12" status="true"
                        source="whatsoever" confidenceScore="99"/>
                </SynsetRelation>
                <SynsetRelation targets="EU-16-00403152-n" relType="gloss">
                        <Meta author="monica" date="2008-05-27" status="false"
                        source="whatsoever" confidenceScore="0.3"/>
```

```
                </SynsetRelation>
            </SynsetRelations>
            <MonolingualExternalRefs>
                <MonolingualExternalRef externalSystem="Domain"
                externalReference="administration">
                    <Meta author="monica" date="2008-05-27"/>
                </MonolingualExternalRef>
                <MonolingualExternalRef externalSystem="Domain" externalReference="law">
                    <Meta author="monica" date="2008-05-27"/>
                </MonolingualExternalRef>
                <MonolingualExternalRef externalSystem="SuperSense"
externalReference="act">
                    <Meta author="monica" date="2008-05-27"/>
                </MonolingualExternalRef>
                <MonolingualExternalRef externalSystem="SUMO"
                externalReference="PoliticalProcess" relType="at">
                    <Meta author="monica" date="2008-05-27"/>
                </MonolingualExternalRef>
                <MonolingualExternalRef externalSystem="TCO"
externalReference="Agentive"/>
                <MonolingualExternalRef externalSystem="TCO"
externalReference="Purpose"/>
                <MonolingualExternalRef externalSystem="TCO" externalReference="Social"/>
                <MonolingualExternalRef externalSystem="TCO"
                externalReference="UnboundedEvent"/>
            </MonolingualExternalRefs>
        </Synset>
```

7.2.11 Definition and Statement

Definition allows to represent the gloss associated with each synset. It has an obligatory attribute 'gloss' and in turn contains an empty element *Statement* that allows to represent examples of use associated with the synset by means of the attribute 'example'.

Definition				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
gloss	<free text>	required	Statement	0..*

Statement				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
example	<free text>	required		

Example:

```
        <Definition gloss="bla bla">
            <Statement example="bla bla"/>
        </Definition>
```

7.2.12 SynsetRelations

This is a bracketing element for grouping together all *SynsetRelation* elements (see below). It must contain at least one of them.

SynsetRelations				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
			SynsetRelation	1..*

Example:

```
<SynsetRelations>
        <SynsetRelation targets="EU-16-06056130-n" relType="has_hyperonym">
                <Meta author="german" date="2008-05-12" status="yes" source="whatsoever"
                confidenceScore="99"/>
        </SynsetRelation>
        <SynsetRelation targets="EU-16-06060479-n" relType="has_mero_part">
                <Meta author="german" date="2008-05-12" status="true" source="whatsoever"
                confidenceScore="99"/>
        </SynsetRelation>
        <SynsetRelation targets="EU-16-00403152-n" relType="gloss">
                <Meta author="monica" date="2008-05-27" status="false" source="whatsoever"
                confidenceScore="0.3"/>
        </SynsetRelation>
</SynsetRelations>
```

7.2.13 SynsetRelation

Relations between synsets are codified by means of *SynsetRelation* elements, one per relation.
The required attribute 'target' contains the ID value of the synset that is target of the relation. The particular relation type (ex., hypernym, meronym, domain, etc.) is expressed as a value of the attribute relType. A list of possible values is enclosed in the Appendix.

SynsetRelation				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
Target	IDREF	required	Meta	0..1
relType	<free text>	required		

Example:

```
<SynsetRelation targets="EU-16-06056130-n" relType="has_hyperonym">
        <Meta author="german" date="2008-05-12" status="yes" source="whatsoever"
        confidenceScore="99"/>
</SynsetRelation>
<SynsetRelation targets="EU-16-06060479-n" relType="has_mero_part">
        <Meta author="german" date="2008-05-12" status="true" source="whatsoever"
        confidenceScore="99"/>
</SynsetRelation>
<SynsetRelation targets="EU-16-00403152-n" relType="gloss">
        <Meta author="monica" date="2008-05-27" status="false" source="whatsoever"
        confidenceScore="0.3"/>
</SynsetRelation>
```

7.2.14 SenseAxes

SenseAxes is a bracketing element that groups together elements (*SenseAxis*) used for interlingual correspondences. It has no attributes.

SenseAxes				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
			SenseAxis	1..*

Example:

```
<SenseAxes>
        <SenseAxis id="sa_en16-en30_001" relType="equal_synonym">
                <Meta author="monica" date="2008-05-27"/>
                <Target ID="EN-16-06060223-n"/>
                <Target ID="EN-30-08135342-n"/>
                <InterlingualExternalRefs>
                        <InterlingualExternalRef externalSystem="SUMO"
                        externalReference="PoliticalProcess" relType="at">
                                <Meta author="claudia" date="06-06-2008"/>
                         </InterlingualExternalRef>
                </InterlingualExternalRefs>
        </SenseAxis>
</SenseAxes>
```

7.2.15 SenseAxis

This element represents the relationships among different closely related senses in different languages. In WordNet terms, it encodes ILI correspondences. Any *SenseAxis* element groups together monolingual synsets that correspond one to another by means of a particular type of relation, specified by means of the 'relType' attribute. The set of inter-WordNet relations is given in the Appendix.

SenseAxis				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
Id	ID	required	Meta	0..1
relType	<free text>	required	Target	1..*
			InterlingualExternalRefs	0..1

Example :

```
<SenseAxis id="sa_en16-en30_001" relType="equal_synonym">
        <Meta author="monica" date="2008-05-27"/>
        <Target ID="EN-16-06060223-n"/>
        <Target ID="EN-30-08135342-n"/>
        <InterlingualExternalRefs>

                <InterlingualExternalRef externalSystem="SUMO"
                externalReference="PoliticalProcess" relType="at">
```

```
                    <Meta author="claudia" date="06-06-2008"/>
                </InterlingualExternalRef>
            </InterlingualExternalRefs>
        </SenseAxis>
```

Coding instructions:
The <SenseAxis> element is a means for grouping together synsets belonging to different monolingual wordnets and sharing the same equivalence relation to a pivot synset, which by convention is an English one.
This is a compact way of encoding correspondences among wordnets, avoiding to have several languageX-to English single correspondences.
For instance, suppose you have the following situation (Synset IDs are made up):
Italian synset ita-16-1251-n, Spanish synset spa-30-09686541-n and Chinese synset zho-30-05231501-n all map onto English WordNet eng-30-13480848-n by means of an eq_synonym relation.
We could represent this situation with several SenseAxis for each language pair:

```
<SenseAxis id="sa_ita16-eng30_001" relType="eq_synonym">
<Target ID="ita-16-1251-n" />
<Target ID="eng-30-13480848-n" />
</SenseAxis>

<SenseAxis id="sa_spa16-eng30_001" relType="eq_synonym">
<Target ID="spa-30-09686541-n" />
<Target ID="eng-30-13480848-n" />
</SenseAxis>

<SenseAxis id="sa_spa16-eng30_001" relType="eq_synonym">
<Target ID="zho-30-05231501-n" />
<Target ID="eng-30-13480848-n" />
</SenseAxis>
```

The representation we propose, instead, is the following one:

```
<SenseAxis id="sa_ita16-spa30-zho30-eng30_001" relType="eq_synonym">
<Target ID="ita-16-1251-n" />
<Target ID="spa-30-09686541-n" />
<Target ID="zho-30-05231501-n" />
<Target ID="eng-30-13480848-n" />
</SenseAxis>
```

As the <SenseAxis> element is used for expressing interlingual correspondences, it will not apply to representation of English WordNet. Mapping between different English WordNet versions are to be represented by means of the <MonolingualExternalRef> element (see above).

7.2.16 Target

The element Target encapsulates the monolingual synset ID that is referenced by each SenseAxis.

Target

Attributes	Values	Optionality	Child elements	Cardinality
ID	<free text>	required		

Example :

```
<SenseAxis id="sa_en16-en30_001" relType="equal_synonym">
        <Meta author="monica" date="2008-05-27"/>
        <Target ID="EN-16-06060223-n"/>
        <Target ID="EN-30-08135342-n"/>
        <InterlingualExternalRefs>
                <InterlingualExternalRef externalSystem="SUMO"
                externalReference="PoliticalProcess" relType="at">
                        <Meta author="claudia" date="06-06-2008"/>
                </InterlingualExternalRef>
        </InterlingualExternalRefs>
</SenseAxis>
```

7.2.17 InterlingualExternalRefs

This is a bracketing element for grouping together all *InterlingualExternalRefs* elements (see below). It must contain at least one of them.

InterlingualExternalRefs				
Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
			InterlingualExternalRef	1..*

Example :

```
<InterlingualExternalRefs>
        <InterlingualExternalRef externalSystem="SUMO" externalReference="PoliticalProcess"
        relType="at">
                <Meta author="claudia" date="06-06-2008"/>
        </InterlingualExternalRef>
</InterlingualExternalRefs>
```

7.2.18 InterlingualExternalRef

This element is used in KYOTO-LMF to express a linking between a SenseAxis instance and an external system such as an ontology, and will represent the means to anchor a group of synsets to an ontological node. In principle, however, the same element can hold a link to any system referenced by a homogeneous group of synsets.
Its intended use, thus, is to provide a representational device to link a group of synsets from different wordnets to the same ontological concept. In essence, it is an equivalent to the ILI.
It should not be used to link a monolingual synset to an ontology. To this end the element <MonolingualExternalRef> should be used instead.
The 'externalSystem' and 'externalReference' recommended attributes allow to encode, respectively, the name of the external system and the specific relevant nodes in the given external system.

InterlingualExternalRef

Attributes	*Values*	*Optionality*	*Child elements*	*Cardinality*
externalSystem	<free text>	required	Meta	0..1
externalReference	<free text>	required		
relType	at, plus, equal	optional		

Example :

```
<InterlingualExternalRef externalSystem="SUMO" externalReference="PoliticalProcess"
relType="at">
        <Meta author="claudia" date="06-06-2008"/>
</InterlingualExternalRef>
```

8 Ontologies

Ontologies are formal and explicit specifications of a shared conceptualization. They are mainly composed by: a set of concepts or classes that characterize the formalized knowledge, a set of rules, called also properties or relations between concepts and a set of instances or individuals belonging to the classes along with their specific properties. The individuals of a class may be characterized by a proper or not proper subset of all the relations of that class. In some way, a concept is the characterization of a set of individuals and a rule is a kind of relation that could hold between two individuals.

In general, ontologies are used to formally express knowledge about a defined domain of interest. When we want to formalize knowledge we must deal with three different levels of knowledge abstraction (see Figure 5); they must all be specified in order to provide an effective description of the information to be represented. The highest level of knowledge abstraction is the methodological knowledge: it is composed by all the knowledge representation languages or ontology languages like OWL or KIF that, based on a particular knowledge description formalism, provide expressive means to describe a set of classes along with their relations and constraints. All the specific sets of classes and relations constituting an ontology, defined referring to a particular ontology language and describing the general structure of a domain of interest belong to the level of conceptual knowledge (i.e., the Suggested Upper Merged Ontology). The set of individuals described as instances of the classes of a particular ontology, along with the relations holding between couples of them is referred to as factual knowledge.

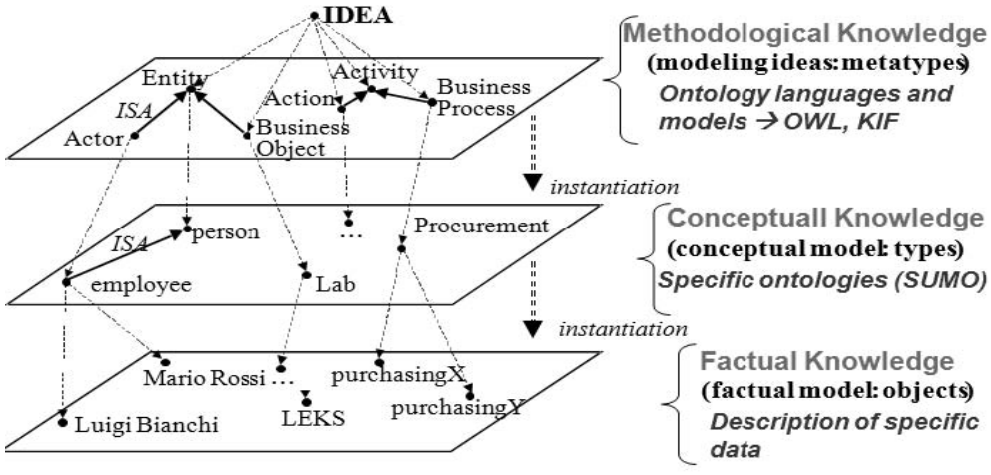

Figure 5 - The three levels of knowledge abstraction.

As said, in order to express knowledge and to make it computable, we need some sort of formalism that, supporting the specification of one or more ontology languages, allows for a standardized way to express, through ontologies, the information considered, making possible automated reasoning procedures. Ontologies can be

expressed adopting different formalisms, called also description languages. When we choose a formalism we have to determine the right trade-off between two main opposite needs: Expressive power and Complexity of reasoning.
Description logics are a family of knowledge representation formalisms; they are a decidable subset of the First Order Logic (FOL). The different description logics are distinguished by different sets of constructors of concepts (union, intersection, universal and existential quantifier, etc.) and rules (inverse rule, transitive rule, concepts subsumptions, etc.). Constructors are the distinct expressive means available to specify concepts (or classes) and rules (or properties). The set of all the descriptions of classes and relations defines the general structure of the domain of interest along with all its constraints. It constitutes a frame of reference exploited to characterize the concrete data, that are the individuals of the considered domain along with their relations. Two widespread knowledge description languages based on description logics are the Web Ontology Language (OWL) and the Knowledge Interchange Format (KIF).

In this section, first of all we give a brief and synthetic overview of the most important knowledge representation languages available, referring Web sites to search for further information. Then we focus our attention mainly on OWL and KIF. We describe them considering their purpose, their constructs as well as their usage and the tools adopted to edit and share ontologies.

8.1 Overview of Semantic Description Languages

In this section we present an exhaustive list of relevant formal languages used to express concepts terms and descriptions [22].

8.1.1 CycL

http://www.cyc.com/cycdoc/ref/cycl-syntax.html CycL was developed by Cycorp and it's it is a declarative language based on classical first-order logic. CycL is used to express common sense knowledge and to represent the knowledge stored in the Cyc Knowledge Base. It has six expression types: Constants, Formulas and Truth-function, Function-denotational, Variables and Quantifiers. CycL's is characterized by good expressiveness, precision, meaning and use-neutral representation. It is part of the Cyc project [26], aiming at assembling a comprehensive ontology and database of everyday common sense knowledge, with the goal of enabling AI applications to perform human-like reasoning. CycL is used to represent the knowledge stored in the Cyc Knowledge Base (the Cyc Knowledge Base), available from Cycorp. The source code written in CycL is licensed as open source, to increase its usefulness in supporting the semantic web.

8.1.2 F-Logic

http://www.cs.umbc.edu/771/papers/flogic.pdf - F-Logic was developed in 1995 at Karlsruhe University -Germany and it's a formalism to represent knowledge. F-logic stands in the same relationship to object-oriented programming as classical predicate calculus stands to relational database programming. Features include, among others, object identity, complex objects, inheritance, polymorphism, query methods, encapsulation [4]. F-Logic major strengths are extensibility and his capacities to

directly represent fundamental concepts that come from object oriented programming and frame based languages. F-Logic makes a number of central aspects of object oriented programming to become compatible with logic paradigm. F-Logic main weakness is related to mathematical and logical concepts needed to programme in this language. F-Logic does not possess cardinality restrictions.

8.1.3 LOOM

http://www.isi.edu/isd/LOOM/LOOM-HOME.html -Loom knowledge representation system has been developed by the University of Southern California's Information Sciences Institute (ISI) in 1986, under DARPA sponsorship. Loom is a language and environment for constructing intelligent applications. The heart of Loom is a knowledge representation system that is used to provide deductive support for the declarative portion of the Loom language. Declarative knowledge in Loom consists of definitions, rules, facts, and default rules. A deductive engine called a classifier utilizes forward-chaining,semantic unification and object-oriented truth maintenance technologies in order to compile the declarative knowledge into a network designed to efficiently support on-line deductive query processing. Loom implements a suite of KR functions whose use has been validated by the substantial Loom user community. Loom is a large and complex system.

8.1.4 KIF

http://logic.stanford.edu/kif/specification.html - It was originally created by Michael Genesereth and others participating in the DARPA Knowledge Sharing Project. There have been a number of versions of KIF, among which SUO-KIF [25] used by Adam Pease to define SUMO. Knowledge Interchange Format (KIF) is a language designed for use in the interchange of knowledge among disparate computer systems (created by different programmers, at different times, in different languages, and so forth).KIF was created to serve as a syntax for first-order logic that is easy for computers to process. It was intended as an interlingua, rather than a format for human authoring of knowledge, but it has since been more often used for that latter purpose. KIF features full semantic expressiveness. One inconvenience of this language is his computational complexity many times has been considered too high. Although the original KIF group intended to submit to a formal standards body, that did not occur. In order to read a more detailed description of KIF along with its constructs, see Paragraph 4.

8.1.5 Ontolingua

http://www.ksl.stanford.edu/software/ontolingua - Ontolingua, created in 1992 at Stanford University, is a language based in KIF (Knowledge Interchange Format). It provides a distributed collaborative environment to browse, create, edit, modify, and use ontologies. Combines frames paradigm and first order predicates. Beyond all the languages used to represent ontologies, Ontolingua language is the one with the biggest expressiveness. It can represent concepts, concepts taxonomies, n-ary relationships, axioms, instances and procedures. Also because of its expressiveness, Ontolingua doesn't permit reasoning.

8.1.6 RDF(S)

http://www.w3.org/TR/rdf-schema/ -RDF [13] stands for Resource Description Framework and is a W3C Recommendation. RDF is a graphical language used for representing information about resources on the web thus constituting a basic ontology language. Resources are described in terms of properties and property values using RDF statements. Statements are represented as triples, consisting of a subject, predicate and object (S, P, O). RDF is written in XML and uses URIs -Unique Resource Identifiers to identify resources. RDF Schema, along with RDF, provides basic capabilities for describing vocabularies that describe resources leaving however a lot of possibilities of extension through other important features. For a more detailed description of RDFS see Paragraph 3.

8.1.7 OWL

http://www.w3.org/TR/owl-features/ -Latest standard in ontology languages from the World Wide Web Consortium (W3C). OWL semantically extends RDF (S). It is based on its predecessor language DAML+OIL. OWL is an ontology language. Classes and relations are the basic building blocks of an OWL ontology. OWL has a rich set of modelling constructors. In order to allow usability by various users, OWL provides three increasingly expressive sublanguages: OWL-Lite, OWL-DL and OWL-Full. For a detailed description of OWL, its syntax and the tools that support the definition of OWL ontologies see Paragraph 3.

8.2 Web Ontology Language (OWL)

The Web Ontology Language (OWL) [9] is used to describe ontologies over the Web; it is intended to be a reference to specify, share and reuse processable knowledge in a distributed environment. It is built on the Resource Description Framework (RDF) [13] and RDF Schema (RDFS) [14] and provides additional vocabulary for describing properties and classes. RDF, as briefly mentioned in Paragraph 2, is a language representing information about resources over the Web. In particular in RDF each piece of information is represented as a triple composed by a property connecting two resources: the first one is referred to as the subject and the second one as the object (ie: subject:Claudia -property:isSisterOf -object:Miriam).
RDF Schema (methodological knowledge) provides basic constructs to define an ontology (conceptual knowledge) in order to specify RDF real data (factual knowledge); in particular it allows to define classes, properties and their subsumption hierarchies along with the domain and the range of each property. OWL was born from the need to extend RDFS to increase its expressivity, thus adding a consistent number of constructs useful to better formalize a domain.
OWL has been derived from DAML+OIL [2], an older semantic markup language for Web resources. The 1.0 version of OWL has been standardized at the beginning of 2004 as the outcome of the W3C Web Ontology Working Group. During the last few years has increased the need to extend OWL so as to add a useful set of features that have been requested by users, for which are now available effective reasoning algorithms, and that OWL tool developers are willing to support. After many proposal of extensions to OWL, since September 2007 the W3C OWL Working Group [21] has been constituted in order to formalize all these requests for extensions to produce a new

standard: OWL 2.0. Currently, OWL 1.0 is mainly used along with some particular extension supported by tools developers that is likely to be standardize in OWL 2.0.

OWL formalism is based on the SHOIN(D) [3] description logic family. In particular, three different OWL sublanguages has been defined, with a growing degree of expressive power (see Figure 6):

- OWL Lite: it provides only simple constructs to describe domains (cardinality restrictions, optional or required properties, etc.);
- OWL DL: it is based on the expressive power of the SHOIN(D) description logic; it is decidable, that is that exists an algorithm which compute from the stated knowledge, the entailed knowledge in a finite number of steps;
- OWL Full: it adds further expressive power to OWL DL but is no longer decidable.

Nowadays the great part of OWL ontologies over the Web is expressed using OWL DL; OWL Lite is not so less expressive than OWL DL, so people usually choose OWL DL. On the other side, OWL Full is not decidable and thus standard automatic reasoning techniques cant be applied.

In what follows we give a brief overview of the main constructs of OWL DL. We contextually refer to the corresponding elements of the OWL XML presentation syntax for those constructs added by OWL 1.0 to RDFS; we also list the XML elements corresponding to the native constructors of RDFS. Those elements are respectively collected in the following namespaces: http://www.w3.org/2002/07/owl which is referred by the abbreviation owl and 'http://ww.w3.org/2001/01/rdf-schema' which is referred by the abbreviation 'rdfs'.

Concept constructors:
- *union of concepts* (owl:UnionOf)
- *intersection of concepts* (owl:IntersectionOf)

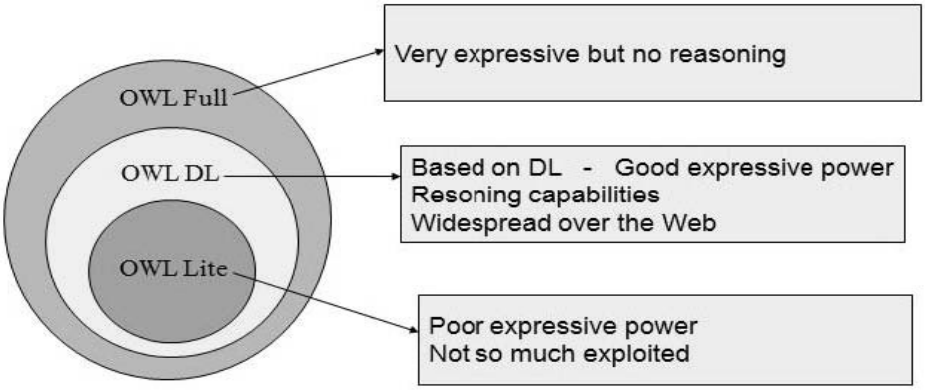

Figure 6 - The three OWL sublanguages.

- *negation of concepts* (owl:ComplementOf)

- choice of one among more concepts (owl:OneOf)
- universal quantifier (owl:AallValuesFrom)
- existential quantifier (owl:SomeValuesFrom)
- greater or equal cardinality constraint between one concept and another linked through a particular property (owl:MinCardinality)
- less or equal cardinality constraint between one concept and another linked through a particular property (owl:MaxCardinality)
- equal cardinality constraint between one concept and another linked through a particular property (owl:Cardinality)

Rules constructors and related axioms:

- concepts subsumption (rdfs:SubClassOf)
- properties subsumption (rdfs:SubPropertyOf)
- domain of a property (rdfs:Domain)
- range of a property (rdfs:Range)
- object property (owl:ObjectProperty)
- datatype property (owl:DatatypeProperty)
- concepts equivalence (owl:EquivalentClasses)
- properties equivalence (owl:EquivalentProperties)
- instances equivalence (owl:SameIndividual)
- disjunction (owl:DisjointClasses)
- instances difference (owl:DifferentFrom)
- inverse rule (owl:IinverseOf)
- symmetric rule (owl:SymmetricProperty)
- transitive rule (owl:TransitiveProperty)
- functional property (owl:FunctionalProperty)
- inverse functional property (owl:InverseFunctionalProperty)

The 'owl:oneOf' property allows the definition of enumerated classes. The element 'owl:ontology' allows expressing all the meta-information regarding the whole ontology: the URI reference for the ontology (rdf:about), a human-readable comment of the ontology (rdfs:comment), the references to previous versions (owl:priorVersion) and the references to other ontologies to include in the existing one (owl:imports).

We briefly expose some of the future extensions to OWL 1.0 that probably will be standardized by the W3C OWL Working Group during the next years.
All the proposed extensions to OWL 1.0 keep decidability and implementability; many of them are derived from the developments of description logic languages and reasoning techniques that have been achieved since the standardization of OWL 1.0, in 2004. First of all, some syntactic facilities needs to be introduced: the possibility not to define only pairwise disjoint classes but to specify a group of classes that are disjoint is one of them. It makes the description of domains more concise and optimizable by reasoners. OWL 1.0 users also need the possibility to define disjointness between properties (two properties cannot characterize the same entity at the same time) as

well as to specify irreflexive and antisymmetric properties. Moreover, it is a common requirement to express value ranges and relationships between values (a rectangle has the width different from height).
Also the possibility to include not semantically defined comments is a requirement for future versions of OWL. In the future directions of improvement of OWL there are also the need to better define an XML syntax for OWL in order to effectively exploit XPATH and XSLT processing patterns and to give users the possibility to extend OWL syntax thanks to macros.

In order to describe individuals, or better instances of the classes belonging to an OWL ontology of reference, along with their properties the RDF is usually adopted. In this way we can define the real knowledge to carry out automated reasoning tasks. Those tasks are performed by a reasoner on the basis of the contents of the ontology and on the factual information (factual knowledge) contained in RDF triples. Usually a reasoner, applying appropriate inferencing rules, can check if there are inconsistencies in the ontology, define properties of particular individuals or also expand the factual knowledge explicitly asserted through RDF, thus deriving the inferred data. In order to query for finding useful information inside RDF data collections is usually exploited SPARQL Query Language for RDF [16]. SPARQL has been standardized as a W3C Recommendation at the beginning of 2008. It gives users the possibility to query RDF graphs, defining specific information pattern to search for. In a certain sense SPARQL is important in RDF data collections like SQL is relevant to relational databases.

To better understand how all those pieces fit together we describe the simple example shown in Figure 7. On the top box is defined and graphically represented a simple OWL ontology of 'Naturally Occurring Water Sources'. It is composed by nine classes (NaturallyOccurringWaterSources, Steam, BodyOfWater and so on). They are linked in a subsumption hierarchy through the 'rdfs:subclass' property (one of the constructs available in OWL and derived from RDF(S) to define subsumption relations between classes). In the blue-backgrounded square there is the XML representation of some RDF knowledge. In particular we say that the individual Yangtze is a river (it is an instance of the class 'River', defined in th ontology previously described). Morover we specify two properties of this particular instance: its length (6300 kilometers) and the link to another instance of the class 'Sea' ('EastChinaSea'), through the relation 'emptiesInto'.

The user can query the RDF knowledge, using SPARQL for instance, or through some engine that translates natural language queries into SPARQL ones. In this way, thanks to the support of a reasoner that allows to make inferences over data relying upon the ontology, we can try to find, if it exists, a result set. As a result, the particular document, or better the particular RDF subset of data containing sensible information respect to the query is selected and show to the user, as represented in the lower yellow box of Figure 7.

OWL is one of the most diffused and supported languages used to describe and share ontologies over the Web; many ontologies or lexical resources are exposed exploiting OWL. As instance, the SUMO ontology has been translated into OWL [17], but also in 2006 the English 2.0 version of Wordnet lexical database has been represented by W3C in OWL and RDF [12]. In conclusion we have to mention Swoogle [18], developed by the UBMC eBiquity research group of the Department of Computer Science and Electrical Engineering of University of Meryland, baltimore Country; it represents an

interesting semantic search engine that analyzes a great amount of semantic data allowing, for instance, to search for specific classes over many indexed ontologies.

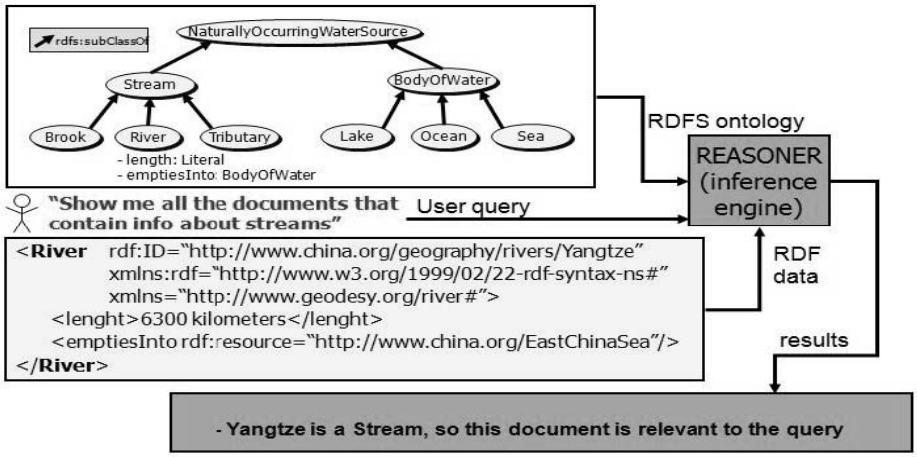

Figure 7 - Example of reasoning with an OWL ontology and RDF data.

It is a good resource to retrieve and explore many different OWL ontologies, in order to share, integrate and reuse conceptualizations of different domains. The great number of OWL ontologies available and the richness of their data explains the huge diffusion of OWL as a standard for ontology and knowledge description over the Web.

8.2.1 An examlpe of OWL ontology

In order to give a simple practical example of an OWL ontology we describe, relying on the OWL XML presentation syntax, an ontology including the classes Person, Man, Woman and Father and the property hasChild ; OWL rules constructors like classes subsumption, classes domain and range, classes disjointness, cardinality restrictions and inverse properties are applied. All these elements are extensively commented:

```
<?xml version=1.0?>
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns
xmlns:xsd=http://www.w3.org/2001/XMLSchema
xmlns:rdfs=http://www.w3.org/2000/01/rdf-schema
xmlns:owl=http://www.w3.org/2002/07/owl
xmlns=http://www.mylocation.it/myontology.owl
xml:base=http://www.mylocation.it/myontology.owl>

<!-- This OWL element specifies the metadata that characterize
the ontology; in this case the empty attribute rdf:about points
out that the URI of the whole ontology is those used to refer
the file that contains it over the Web -->
<owl:Ontology rdf:about= />
```

```
<!-- Definition of the class Person -->
<owl:Class rdf:ID=Person/>
<!-- Definition of the class Man which is a subclass of the class
Person and its set of instances is disjoint from those of the class
Woman -->
<owl:Class rdf:ID=Man>
<rdfs:subClassOf rdf:resource=#Person/>
<owl:disjointWith rdf:resource=#Woman/>
</owl:Class>


<!-- Definition of the class Woman which is a subclass of the
class Person and its set of instances is disjoint from those
of the class Man -->
<owl:Class rdf:ID=Woman>
<rdfs:subClassOf rdf:resource=#Person/>
<owl:disjointWith rdf:resource=#Man/>
</owl:Class>

<!-- Definition of the class Father as a subclass of the class Man,
stating that every instance of the class father must be the subject
of at least one RDF-triple characterized by the property hasChild -->
<owl:Class rdf:ID=Father>
<rdfs:subClassOf rdf:resource=Man/>
<owl:Restriction owl:minCardinality=1 />
<owl:onProperty rdf:resource=#hasChild/>
</owl:Restriction>
</owl:Class>

<!-- Definition of the property hasChild which must have as subject an
element/instance of the class Parent and as object an element/instance of
the class Person; its inverse property is hasParent -->
<owl:ObjectProperty rdf:ID=hasChild>
  <rdfs:domain rdf:resource=#Parent/>
<rdfs:range rdf:resource=#Person/>
<owl:inverseOf>
<owl:ObjectProperty rdf:about=#hasParent/>
</owl:inverseOf>
</owl:ObjectProperty>
</rdf:RDF>
```

In the last part of this section about OWL we will briefly describe the most important OWL editing and reasoning tools.

8.2.2 OWL Tools: editors and reasoners

As a consequence of OWL great diffusion, there is a huge amount of tools developed to create and edit OWL ontologies and also to reason with OWL ontologies and RDF data sets.

Among the most diffused OWL editing tools there are:

- **Protégé**: is an open source ontology editor developed by the Stanford Center for Biomedicl Informatic Research. It is a Java application, easily extensible thanks to a plugin mechanism. It has been adopted by a large community of users and is constantly updated and enriched with new functionalities. The Protégé-OWL extension fully supports the OWL 1.0 W3C Recommendation. Some of the common tasks that can be carried out thanks to Protégé are: load and save OWL and RDF ontologies, edit and visualize classes, properties, and SWRL rules [20], define logical class characteristics as OWL expressions, execute reasoners such as description logic classifiers, edit OWL individuals for Semantic Web markup. To download OWL or simply to obtain more information see [11].
- **Swoop**: is a tool for creating, editing, and debugging OWL ontologies. It was produced by the MIND lab at University of Maryland, College Park, but is now an open source project with contributors from all over; it is deployed as a Java application. It has many interesting facilities to edit ontologies even if it is no more constantly developed. To find some more information or to download SWOOP, see [19].
- **Ontotrack**: is an integrated browser/editor of ontologies accessible as abrowsing/editing system. It has many interesting interface features like sophisticate ontology layout and visualization possibilities, but it supports only OWL Lite; thus it is not possible to manage with Ontotrack the expressivity of OWL DL. To get more information about Ontotrack see [8].

Some of the most used reasoners supporting OWL are:

- **Pellet**: Pellet is an open source, OWL DL reasoner. It is distributed for free, but commercially supported. Pellet supports the full expressivity of OWL DL. As of version 1.4, Pellet supports many new features that has been proposed as extension for new versions of OWL, with the exception of n-ary datatypes. It is a java based web application. Pellet is widely used for reasoning tasks. To get more information about Pellet or to download it see [10].

- **FACT++**: is an OWL DL reasoner released under the GNU licence. It has been written in C++, thus maximizing performances. Beyond normal reasoning tasks, it provides some specific service like the HTML output of an OWLontologies. To download it or access to a more detailed description see [5].

8.3 Knowledge Interchange Format (KIF)

The Knowledge Interchange Format is a standard to describe knowledge among different computer systems so as to facilitate its exchange. KIF is intended not as an

internal memorization format within computer, but as a mean to enable data flows among distinct systems. Itsexpressivity is based on a version of first order predicate calculus, with extensions to support non monotonic reasoning and definitions [23].

KIF, as briefly mentioned in Paragraph 2, was originally created by Micheal Genesereth and others participating in the DARPA Knowledge Sharing Effort, a global group that wanted to develop techniques, methodologies and software tools for knowledge sharing and knowledge reuse, at design, implementation, orexecution time [6].
There have been a number of versions of KIF the original KIF group intended to submit to a formal standard body, that did not occur. A later version called Common Logichas since been developed for submission to ISO and has been approved and published. A variant called SUO-KIF is the language in which the Suggested Upper Merged Ontology is written [13].

We refer to the version of KIF the specifications which can be retrieved at [7].

KIF has declarative semantics; this means that it is possible to understand the meaning of expressions in the language without the intermediation of any interpreter. KIF is logically comprehensive, that means that it provides for theexpression of arbitrary sentences in the first order predicate calculus. Threefurther characteristics of KIF are: the translatability, or better the easiness of implementation of translation mechanisms to and from particular knowledge representation languages; the readability, in the sense that it should be easily readable by humans even if not explicitly intended for this purpose; the implementability, that is the possibility, if desired, to use KIF also as a representation language within a program.

We briefly describe KIF syntax. The basic building block of KIF syntax is thecharacter. Characters are divided into seven groups: upper case, lower case, digits, alpha characters (non alphabetical characters used in the same way letters are used), special characters, spacing characters and other ASCII characters. Through lexical analysis a flow of characters belonging to different groups is divided in lexemes, usually considering spacing characters as lexemes delimiters. In KIF syntax there are five types of lexemes, described in what follows. Special lexemes are composed by all the special characters (” - ’ - # - (-) - , - \). Words are another type of lexeme; they are sequences of characters. Words are case insensitive andin their text, special characters are escaped through ’\’. Another type of lexeme is the Character reference. It is composed by the characters ’\’ or ’#’ followed by any other character. They allow us to refer to characters as characters, differentiating them from one character symbols. Character strings are sequences of characters included in quotation marks (quotation marks are escaped by ’\’). Character blocks allow to write a sentence of an arbitrary number of bits without escaping; they are composed by ’#’ + decimal number of characters of the block + q/Q + sequence of characters. Variables are words in which the first character is ’ ?’ (individual variables) or ’@’ (sequence variables). Opera tors are words used to form expressions of various sort and are divided into term operators, function operators and definition operators. Constants are all words except variables and operators. There are object constants, used to denote individual objects, function constants, for functions on objects, relation constants, to denote relations and logical constants to express boolean conditions. Expressions in KIF are composed by one or more lexemes; according to particular rules of composition there are three types of expressions: terms, sentences and definitions. Terms are individual variables, character references, constants, character strings, character blocks, functional terms (function name + arguments), list term (finite list of elements), quoterm (quote

operator + arbitrary list of expressions) and logical terms (involving the if and the cond operators). Sentences are constants, equations (= operator), inequalities (\= operator), relational sentences (relation constant + arbitrary number of arguments), logical sentences (depending on the logical operator considered:conjunction, disjunction, implication, reverse implication, equivalence) and quantified sentences (existentially or universally quantified). There are three types of definitions: unrestricted, complete or partial. Within each type there are four classes of definitions: defobject, deffunction, defrelation, deflogical (defining respectively object, function, relation and logical constants). A form is a sentence or definition. A KIF knowledge base is a finite set of forms; the order of sentences is irrelevant. Speaking about KIF logics we must also say that:

- *functions are total* (there is a result for every combination of arguments; bottom is the undefined value);
- *in functions, list variables* (ie. @1 = 1 2 3) *are considered as multiple arguments of the same function*;
- *definitions are exploited to state sentences that are true by definition, in a way that distingushes them from properties that express contingent properties of the world*;
- *numbers are constant in base 10 representation and there is a huge set of functions useful to elaborate them*.

Considering KIF browsers and editors, we have to mention Sigma [16]. It has been created by Adam Pease; Sigma is an environment for creating, testing, modifying, and performing inference with ontologies. It is accessible by a browser with the support of Java libraries. As said in its presentation, Sigma shows a number of useful features for knowledge engineering work, including term and hierarchy browsing, the ability to load different files of logical theories, a full first order inference capability with structured proof results, a natural language paraphrase capability for logical axioms, support for displaying mappings to the WordNet lexicon and a numbe r of knowledge base diagnostics. In order to download the system or view the manual so as to deeply explore Sigma see [15].

8.3.1 An examlpe of KIF knowledge description

We comment a short example of a part of the Suggested Upper Merged Ontology (SUMO) expressed exploiting KIF. We refer to the class Beverage. In KIF sentences are expressed in the form: (operator/relation firstArgument secondArgument). Starting from this assumption, in line 1 we say that the class Beverage is a subclass of the class Food (a beverage is a particular type of food). We assume that in the previous part of SUMO there is the definition of the subclass relation. From line 2 to line 4 there is a natural language description of the class Beverage in English language, through the property documentation.
From line 5 to line 7 there is an expression, involving the implication operator (=¿). It says that if there is an instance ?BEV (individual variable) of the class Beverage (line 6), then this instance must have as characterizing attribute the fact that it is Liquid (line 7).

1. (subclass Beverage Food)

2. (documentation Beverage EnglishLanguage □ny &%Food that is ingested
3. by &%Drinking. Note that this class is disjoint with the other
4. subclasses of &%Food, i.e. &%Meat and &%FruitOrVegetable. □)
5. (=>
6. (instance ?BEV Beverage)
7. (attribute ?BEV Liquid))

8.4 Conclusions

KIF is based on a set of constructs and expressive possibilities greater than OWL; to give some example of these increased descriptive possibilities we can consider that in knowledge representation languages, the context permits to represent statements over statements, also said meta-statements, and hence, for example, situation duration and statement negation, modalities, creator and argumentation relations. As instance we could want to say that: 'Laura think that Mario likes her (now) in 2003, and that before he did not'. In KIF this kind of constructs and as a consequence this kind of expressivity is possible, while in OWL 1.0 it is not. To expose a further example of differences between the two languages considered, we can state that OWL 1.0, contrary to KIF, doesn't have the possibility to define n-ary relations.

Generalizing the expressivity of OWL is not so extensive as those of KIF, but on the other side OWL is the most widespread and supported language that allows, along with RDF(S), for ontology and knowledge description, constituting the de facto standard for ontology representation over the Web and not only. This is also underlined by the great number of browsing/editing and reasoning tools developed for OWL. The huge number of OWL ontologies diffused on the Web furtherly stresses the great diffusion of OWL. Moreover we must keep in mind that KIF is mainly intended as a common language to describe knowledge among different systems so as to support their exchange of data. As the last and general consideration we must say that in order to choose an ontology representation language besides all the factors just described, we must take into consideration the real need to exploit the complex descriptive capabilities of a particular language that is usually opposed to its easiness of use,reasoning and ontology editing; we must try to find the knowledge description language that better balances these two opposite needs.

9 Semantic search in Kyoto

9.1 Overall architecture and design

Requirements for semantic search are:

1. match concepts rather than words
2. match relations, properties and processes in which concepts are involved
3. represent results as events in time and place rather than as search results
4. point back to sources and search results
5. allow for search across the different languages
6. support structured views on the results
7. give the users a feel for completeness of the results
8. perform with reasonable speed

The semantic search consists of two layers:

1. basic retrieval layer that returns the best matching phrases in the best matching pages;
2. semantic retrieval layer that matches queries with concepts, properties and relations within a given range of time and locations and returns the events in a structured form;

The documents are represented in terms of pages, see Figure 8 below. For each page, we have:

- a linear KAF representation of the text in terms of wordforms, terms, chunks, dependencies and semantic relations and roles between terms;
- a generic KAF representation of semantic objects in terms locations, dates, processes and properties;

Whereas the linear KAF is language-specific, the generic KAF is language neutral but points back to the realizations in a particular language. In addition to KAF files at the page level, there are also KAF files for the complete document. In the case of the generic KAF, these are merges of the separate page files.

From the linear KAF we extract all the phrases from each page, as shown in Figure 8. A page to phrase index is built that holds all phrase identifiers and points to the pages in which each phrase occurs. Next a word index is built from the phrases, which points to the phrases and to the pages. For cross-lingual retrieval, the phrase structures in the KAF files are translated using the multilingual wordnet database. This means that for each KAF page in a language, we create an equivalent KAF page with the same phrase structures but for each indexable word the translations of that word in the target language. If no translation exists, the original word is maintained.

Instead of words, the index can also be based on concepts, provided that the KAF has wordnet synsets assigned to the words. There is no difference in architecture for word-based or concept-based retrieval. If we build a full concept based index, we do not need to translate the indexes to the target languages. It is also possible to have combined indexes of concepts and words. In that case, concepts identifiers are added as synonyms to the words in the indexable phrases. From the word index, a fuzzy

index can be built from the trigrams in each word. This is optional. For Chinese and Japanese, fuzzy indexing will not work.

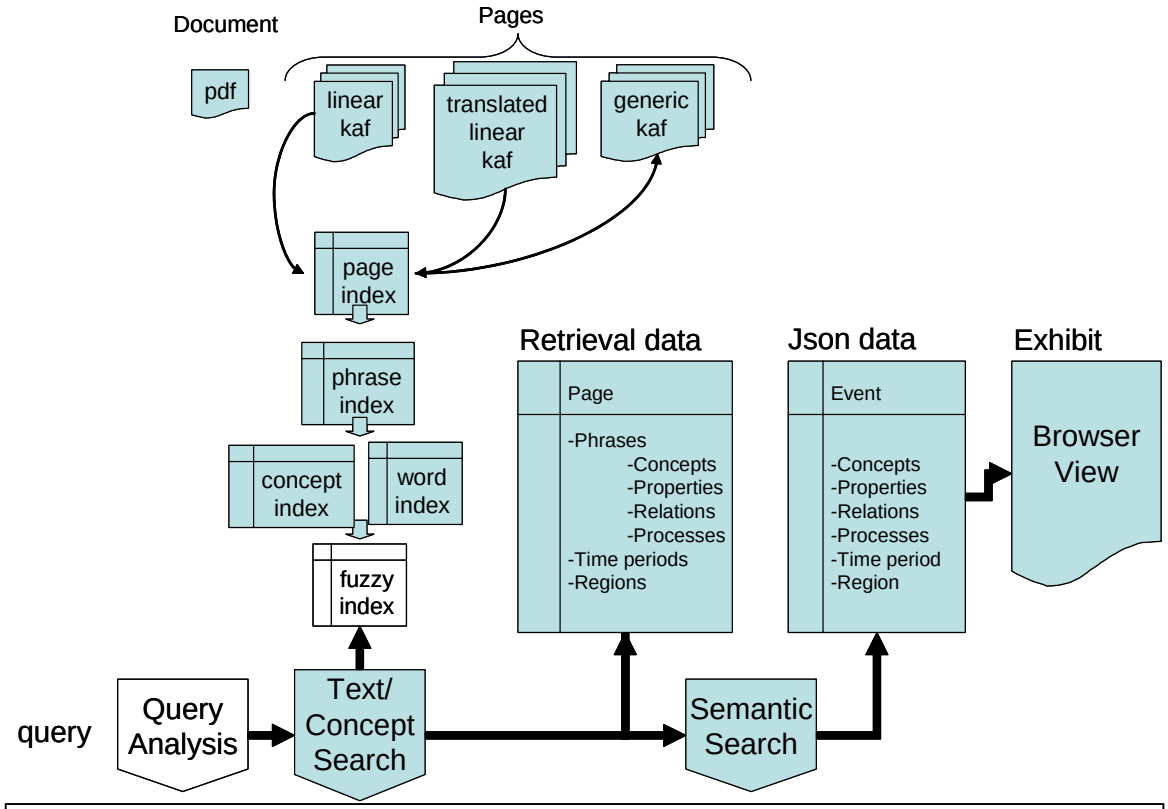

Figure 8: Architecture for semantic search

The overall retrieval is based on a selection of pages and phrases using the text retrieval engine, on top of which a semantic interpretation is built. Text retrieval works in 3 global steps:

1. Words in the query are matched with the fuzzy index to increase recall. It is possible to turn off the fuzzy matching (e.g. for Chinese and Japanese), in which case the query words are directly matched against the index words.
2. The fuzzy index returns index words which are passed to a vector-space engine to retrieve the most relevant pages. It is possible to skip this phase by returning all the pages from the index on which the query words occur.
3. Within the returned pages, the best matching phrases are returned. Phrases that include all the query words are preferred.

The final search result consists of a list of pages in documents with for each page a list of the matching phrases. The overall score of the page is based on the scores of the best phrases on that page.

In the case of a conceptual search, a query analysis is required that matches the query against a list of concepts, using the multilingual wordnet database. The query is then replaced by concept identifiers and fuzzy matching can be skipped. This is optional. The same steps are further applied to conceptual search: most relevant pages are returned and for each page the best matching phrases.

The matching of the phrases is based on the inclusion of concepts or words from the query in a phrase or adjacent phrases of the page. If all are present in the same phrase, a 100% match is returned. The score decreases proportionally to the number of included concepts or words.

For each result, the language of the source pages is given from which the phrase was extracted. In the case of cross-lingual retrieval, the system will thus point to pages in another language, based in the indexes in the query language, even when it accessed the index of the query language. Through the page identifier of a source result, we can access the original linear KAF file and any generic KAF files related to a page.

Once the scope of the relevant pages is defined through a text search, the semantic information is aggregated using the linear and generic KAF representation of each relevant page. From the search results, we build a data structure in Json format that represents so-called facts. A fact consists of:

1. A list of concepts
2. Quantities of each concept
3. Dynamic and static relations between the concepts
4. The region that applies to fact
5. A period that applies to a fact
6. Pointers to the search results on which the fact is based

The information for each fact can come from different phrases in each page and possible from the complete document. First each search result is represented as a separate pre-factual structure. Secondly, these pre-factual structures are combined in so far that they match.

To match the textual search with conceptual structures represented in KAF, the identifiers of the textual index units and the KAF units are shared. Below is an example for an indexable unit for the text search and the corresponding structures in the KAF annotation of the file. The indexable units are NPs with identifier attributes. The search returns the NP identifiers for all matching phrases. We also added attributes for the chunk, the head of the chunk and for earch word to the term and word identifiers in the KAF annotation.

Indexable unit for text search

```
<NP id="10" cid="c232" head="t1540" phrase="NP">
     <WRD POS="g" tid="t1539"><WF>iberian</WF><TWF
wid="w1613">Iberian</TWF></WRD>
     <WRD POS="n" tid="t1540"><WF>lynx</WF><TWF
wid="w1614">Lynx</TWF></WRD>
<PHR>Iberian Lynx </PHR></NP>
```

Corresponding structure in the KAF representation of the page:

```
<wf wid="w1613" sent="184" para="1" page="8"><![CDATA[Iberian]]></wf>
<wf wid="w1614" sent="184" para="1" page="8"><![CDATA[Lynx]]></wf>

<term tid="t1539" lemma="iberian" pos="g" type="open">
<span>
     <target wid="w1613"/>
</span>
```

```
….
<term tid="t1540" lemma="lynx" pos="n" type="open">
<span>
     <target wid="w1614"/>
</span>
</term>
….
<chunk cid="c232" head="t1540" phrase="NP">
<span>
     <target tid="t1539"/>
     <target tid="t1540"/>
</span>
</chunk>
….
<dep from="t1540" to="t1539" rfunc="mod"/>
```

These identifiers can be indexed so that we can quickly retrieve all derived KAF annotations for each result. Likewise, we can collect all textual, structural and semantic objects that are represented in any layer in KAF and that are directly or indirectly related to the units in the search index. We thus have the flexibility to add semantic results to any textual results without loosing the capacity of the textual search.

The system is also open to any type of semantic layer that is added to KAF in the Kyoto project.

9.2 Pre-factual search results and factual results

The matching of pre-factual structure is defined by:

1. The semantic match of the concepts
2. The semantic match of the dynamic or static relation
3. Overlap in regions
4. Overlap in time

The following example clarifies this approach. Assume that the query for "Decrease of lynx populations" yields the following search results:

Table 2: Distinct search results at the page level based on text search

Concepts	Quantity	Relation	Region	Period	Sources
The small population of lynx decreased since 1992, due to increase of agriculture activity, as reported by the local authorities in 1194.	small	decrease	Basque country, Madrid, Austria	1992-1994, 2008	doc1#5
The small population of lynx					
Large predators are threatened in the Pyrenees		threatened	Bilbao	1990-1993	doc1#8 doc12#4
lynx	250	lower	Rocky Mountains	2002-2004	doc3#1
lynx	25%	decrease	Spain	2002	doc1#15
cats		hunt	European cities	2001	doc18#2
cats		hunt	European cities	2001	doc18#3
feline species	some	eat	Zimbabwe, Thailand		doc45#3

In the search table, ear row represents a search result on the page level, where a concept is represented as a query word or a phrase that contains the query word. Using simple techniques, query words can be expanded to hyperonyms, hyponyms and synonyms, yielding some type of semantic search that increases the recall. For each search result, a number of columns are given in which we express data extracted for each result: quantities, relations, regions, periods. In principle, any property can be extracted from a search result and displayed as a row. The last column indicates the page result of the query. The page result is the actually unit of retrieval, therefore there is only a single page per row. Since we can extract more than one property from the search context on each page, we also see that multiple values can be given for regions and periods (or even relations and quantities). The first row also shows that there can be multiple phrases on the same page that match a query.

Now look at the next table, which represents a factual representation of the same results. In this table, we take an event as a unit. An event is defined by the combination of the concept and the relation within a region and a time-frame. If regions are too distinct, we need to split the event in two events. If periods are too distinct, we need to split the event as well and treat them as different events. On the other hand, any event (concept+relation) that is within the same time-frame and region can be grouped together even when they are from different pages, different documents or even different languages.

Table 3: Distinct facts based on matching search results

Concepts	Quantity	Relation	Region	Period	Sources
The small population of lynx decreased since 1992, due to increase of agriculture activity, as reported by the local authorities in 1994.	small	decrease	Basque country, Madrid	1990-1994	doc1#5,8 doc12#4
Large predators are threatened in the Pyrenees		threatened	Bilbao		
lynx	25%	decrease	Spain	2002	doc1#15
The small population of lynx	small	decrease	Austria	2008	doc1#5
lynx		lower	Rocky Mountains	2002-2004	doc3#1
cats		hunt	European cities	2001	doc18#2,3
feline species	some	eat	Zimbabwe		doc45#3
feline species	some	eat	Thailand		doc45#3

In this table, we see that the first row holds results from different pages and documents with the same region Spain (here region is defined at the country level) and within a limited time period. The results for the Basque country, Madrid and Bilbao are thus grouped together, spanning a period from 1990-1994. For the same reasons, we distinguished "The small population of lynx" in Austria as a distinct event, because it is a too distinct region at a too different date: 2008. The "25% decrease of lynx in Spain" is distinguished on the basis of a difference in time only: 2002. Finally, we merge *cats in European cities* (two similar hits on the same page) and we split *feline species* in Thailand and Zimbabwe.

Within this architecture, we have a lot of possibilities to vary the degree of matching for different pieces of information:

1. Conceptual matches for the involved concepts:
    a. synonymy
    b. hyponymy: cats, feline species
    c. roles: large predators
2. Conceptual matches for relations:
    a. synonymy: lower versus decrease
    b. hyponymy: change
    c. implications: extinction implies decrease
3. Matches of regions:
    a. meronymy relations
    b. distance in longitude and altitude coordinates
4. Matches of time periods:
    a. overlap of periods
    b. required distance

Possibly, these matches can be tuned by the user who can express the fine-grainedness of the search, i.e. the degree of conceptual matches or the discrimination of regions and periods.

The results are built on top of textual units for which KYOTO can provide conceptual layers in the associated KAF. This means that for any result, we can look for a conceptual representation of the textual match, which can be the basis for conceptual matches and for cross-lingual search.

9.3 Cross-lingual search

Cross-lingual functionality can be added at two different levels:

1. Using the standard Irion functionality, indexes in one language can be represented in any other target languages;
2. Using the semantic layers in KAF, we can present representations of search results in any language that provides appropriate labels for the concepts associated with the results;

The first option is based on the functionality of the Irion search engine to expand the representation of indexable units to other languages. Each WF element in the indexable unit is added to the Irion index of a language. By creating indexable units in another language, we can build indexes in other languages to that correspond to the index in the source language. This is done by translating the content words (part-of-speech is noun, verb or adjective) in the indexable units to content words in the target language and to maintain the structure of NPs and identifiers. As a result, we get parallel indexes for the source language and any other target language. The same technique is used to add synonyms and other variants within the same language. In the MEANING project (IST-2001-34460), we have shown that we can use the wordnets from other languages to do the expansion of the indexes [32]. We will do the same in the KYOTO project. Through the online wordnets that are connected through sense-axis relations, we can match any synset in any language to related synsets in all the other languages. It is

even possible to limit the translation and expansion by applying WSD to the NPs before we do the expansion. This was also demonstrated in the MEANING project. When the NPs are represented in the target languages, we create parallel index structures for each target language:

English source language
KAF document
-> KAF pages
-> NPs
-> NP-page index
-> word-NP index
-> word-page index
-> page-document index
-> word-tri-gram index

Target languages: Dutch, Spanish, Basque, Italian, Chinese, Japanese
-> NPs
-> NP-page index
-> word-NP index
-> word-page index
-> page-document index
-> word-tri-gram index

Cross-lingual retrieval is then facilitated as follows:

Query
-> select/detect query language
-> word-tri-gram index to match query words with index words through fuzzy matching
-> retrieve most relevant pages
-> retrieve most relevant NPs on each page
-> check the source language of the result page
-> get the KAF document of the source language: in the case of a cross-lingual result, these are the KAF representation for another language
-> retrieve the semantic objects of the source language related to the NPs in the search result: they can be displayed using the source language labels and/or using the query language words
-> build up the JSON result structure

In the basis query words are matched against indexes of the query language. We collect the resulting pages and NPs in the same way as in the case of mono-lingual retrieval. However, the system stores the source language in the meta data of each page. Likewise, it can see what pages are source language pages and what pages are artificially created representations in target languages.

If a result is a translated page, we know that the KAF representation of the result corresponds to the source language page and not with the translated page. Nevertheless, we know the correspondence between the NP words in each representation and likewise can proceed to build up the semantic Json structure on the basis of the data of the source language.

This strategy also allows us to combine results from different languages in one result table. For the retrieval system it makes no different to process results from mono-lingual or cross-lingual retrieval. The only difference that we expect is that cross-lingual retrieval results will suffer from translation errors.

The second approach for cross-lingual functionality is to make use of the semantic/conceptual representation of the text in the KAF representations. When we build up a table for the results based on a semantic interpretation, we can use the domain wordnets to display the results with labels in different languages as well. The baseline retrieval system of Irion Technologies that is now used already uses such a functionality to show the translations of query word matches in the NPs when you move the mouse over a search result.

In this case, it is easier than regular cross-lingual search because we do not need to translate a complete noun phrase or text fragments but only the isolated elements in the tables. Translation of periods and regions is straight-forward. The concepts and their relations and properties can be a bit more difficult but we expect them to be present in the domain wordnets.

9.4 Interfacing

The interface for the semantic search is built using the Exhibit API developed at MIT: http://static.simile.mit.edu/exhibit/. Exhibit [30] consists of Java-script packages that provide advanced functionality to display structured data. The structured data can be published by any server (e.g. as Google spreadsheets) and are loaded in the browser of the user together with the Java-script. The local database of the user is accessed to further present the data. The data model for Exhibit is as follows:

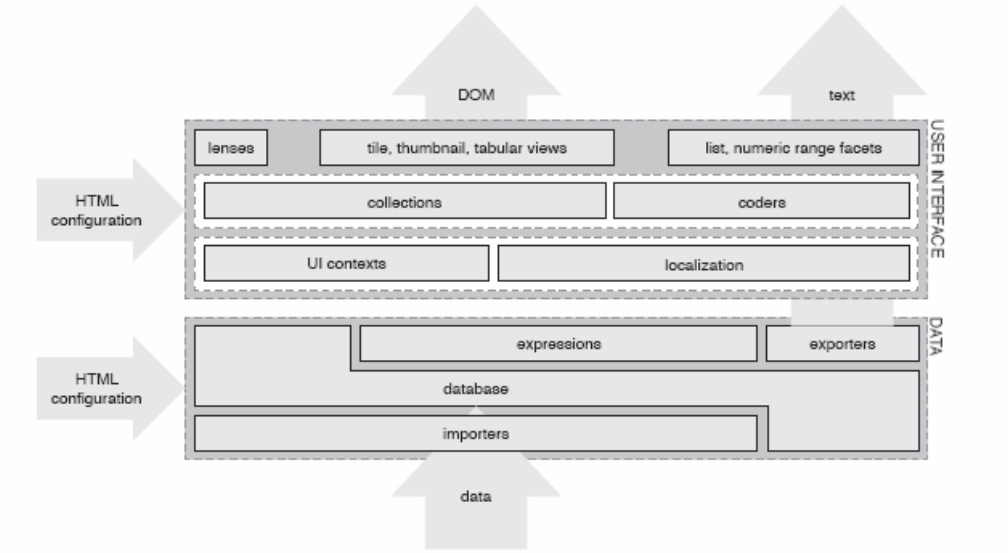

Figure 9: Taken from Huynh 2007, p. 69, figure 3.15. Exhibit's architecture

At the bottom is the data layer, consisting of the database, the expression language parser and evaluator, and importers and exporters. At the top is the user interface layer, which consists of three sub-layers:

- UI contexts and localization resources—storage of presentation settings for the rest of the user interface layer.
- collections and coders – components that do not render to the screen but determine what data widgets should render and how to render it.
- widgets which perform the actual rendering and support interactions.

Data in various formats is converted to a database in memory that is represented in the client interface of each user. Since the database is locally stored, it is easy and fast to manipulate the data in the client interface once it is loaded.

For the database, we generate a Json data file that can directly be loaded into the Exhibit script. Various display are given in Exhibit: tables, tiles, timelines and Google geomaps. New displays can be created as well.

The actual data structures and their display will be defined on the basis of the output of the Kybots in Kyoto and the user-requirements. Instead of the quantities and relations in the above example, we can relations between objects that are directly generated by the Kybots.

10 References

0. **Language tags in HTML and XML.** http://www.w3.org/International/articles/language-tags/

1. **Language Resource Management-Semantic Annotation Framework: time and events (SemAF-Time) rev-12** . http://lirics.loria.fr/doc_pub/SemAFCD24617-1Rev12.pdf, in: Proceedings of LREC 2008, Marrakech, Morocco, May 28-30, 2008

2. **Daml+oil (march 2001) reference description - w3c note.** http://www.w3.org/TR/daml+oil-reference.

3. **Description logics from wikipedia.** http://en.wikipedia.org/wiki/Description logic.

4. **F-logic description from wikipedia.** http://en.wikipedia.org/wiki/F-logic.

5. **Fact++ - web site.** http://owl.man.ac.uk/factplusplus/.

6. **Knowledge interchange format from wikipedia.** http://en.wikipedia.org/wiki/Knowledge Interchange Format.

7. **Knowledge interchange format specifications - draft proposed american national standard (dpans).** http://logic.stanford.edu/kif/dpans.html.

8. **Ontotrack - web site.** http://www.informatik.uni-ulm.de/ki/ontotrack/.

9. **Owl web ontology language overview - w3c recommendation.** http://www.w3.org/TR/owl-features/.

10. **Pellet - web site.** http://pellet.owldl.com/.

11. **Protégé-owl - web site.** http://protege.stanford.edu/overview/protege-owl.html.

12. **Representation of wordnet in rdf/owl - w3c working draft.** http://www.w3.org/TR/wordnet-rdf/.

13. **Resource description framework (rdf) - w3c recommendation.** http://www.w3.org/TR/REC-rdf-syntax/.

14. **Resource description framework schema (rdfs) - w3c recommendation.** http://www.w3.org/TR/rdf-schema/.

15. **Sigma knowledge engineering environment for kif knowledge - web site. http://sigmakee.sourceforge.net/.**

16. **Sparql query language for rdf - w3c recommendation.** http://www.w3.org/TR/rdf-sparql-query/.

17. **The suggested upper merged ontology expressed adopting owl.** http://www.ontologyportal.org/translations/SUMO.owl.txt.

18. **Swoogle: semantic web search - web site.** http://swoogle.umbc.edu/.

19. **Swoop - web site.** http://code.google.com/p/swoop/.

20. **Swrl: a semantic rule language combining owl and ruleml - w3c member submission.**
http://www.w3.org/Submission/SWRL/.

21. **The w3c owl working group 2007 - web site.**
http://www.w3.org/2007/OWL/wiki/OWL Working Group.

22. **Semantic domain system group of Universidade de Madeira. Semanic description languages.** http://seed.uma.pt/Projects/sds/index.php?page=lang.html.

23. **Labrou Finin and Mayfield. A brief introduction to the knowledge interchange format.** http://www.cs.umbc.edu/kse/kif/kif101.shtml.

24. **Martins. Knowledge representation/translation in rdf+owl, n3, kif, uml and the webkb-2 languages.** http://www.cit.gu.edu.au/phmartin/WebKB/doc/model/comparisons.html.

25. **Adam Pease. Suo-kif - standard upper ontology knowledge interchange format.**
http://sigmakee.cvs.sourceforge.net/*checkout*/sigmakee/sigma/suo-kif.pdf.

26. **The cyc project** - web site. http://www.cyc.com/.

27. Vossen 2008, **Linguistic knowledge for more precision, richer answers and flexible systems"**, in Special Issue of Revue Francaise de Linguistique Appliquee **(RFLA**) on "Extraction d'Information: l'apport de la linguistique", Vol XIII 2008-1, June 2008. Revue Française de Linguistique appliquée, Paris.

28. Vossen P., E. Agirre, N. Calzolari, C. Fellbaum, S. Hsieh, C. Huang, H. Isahara, K. Kanzaki, A. Marchetti, M. Monachini, F. Neri, R. Raffaelli, G. Rigau, M. Tescon (2008) **"KYOTO: A system for Mining, Structuring and Distributing Knowledge Across Languages and Cultures"** in: Proceedings of the Fourth International Global Word Net Conference - GWC 2008, Szeged, Hungary, January 22-25, 2008.

29. Huynh, David F., 2007, **User Interfaces Supporting Casual Data-Centric Interactions on the Web**. PhD at the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology

30. Huynh, David F., David R. Karger, Robert C. Miller, **Exhibit: Lightweight Structured Data Publishing**, International World Wide Web Conference Committee (IW3C2) WWW 2007, May 8–12, 2007, Banff, Alberta, Canada. ACM 978-1-59593-654-7/07/0005. MIT Computer Science and Artificial Intelligence Laboratory, The Stata Center, Building 32, 32 Vassar Street, Cambridge, MA 02139, USA

31 Dabernig, Josef (2008) **Creating interactive web pages using the Exhibit framewor**k, University of Applied Science Technikum Wien, Study course Computer Science.

32. Vossen, Piek, German Rigau, Iñaki Alegria, Eneko Agirre, David Farwell, Manuel Fuentes 2006 "**Meaningful results for Information Retrieval in the MEANING project**". In: Proceedings of the 3rd Global Wordnet Conference, Jeju Island, Korea, South Jeju, Januaru 22-26, 2006

33 **KAF: Kyoto Annotation Framework v0.5**, Kyoto consortium 2009.
https://kyoto.let.vu.nl/svn/kyoto/tags/KAF/v0.5/.

11 Appendix A - Terms annotation example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tmf SYSTEM "tmf.dtd">
<tmf>
       <struct type="TE" id="t001">
       <!-- all values of the "type" attribute in the example are taken as they are from
the original formats. In case of adoption
       of TMF as encoding format, they should be mapped against the ISO Data Category
Registry 12620 -->

              <!-- We deviate from TMF and represent features in a more compact way -->
              <!-- The motivation for this is that less standardization is necessary for
the term data since it depends too much on the application -->

              <!-- this is a language specific TE. I think that the terms should be kept
separate per language -->
              <!-- Use 3-letter language coding (e.g. ENG, NLD) -->
              <languageCoding>ISO 639-3</languageCoding>
              <languageLetterCoding>ENG</languageLetterCoding>


              <!-- Manually assigned domain -->
              <termDomain>environment</termDomain>

              <!-- domain classification based on all the forms in the term database -->
              <!-- microWorld stands for a thematic group of domains, e.g. all Wordnet
domains are grouped together in about 50 rough clusters that represent distinct
microworlds.The microWorlds are assigned automatically by the Irion classifier on the
basis of all the term forms listed in this file and their structural relations-->
              <treeProfile>
                            <microWorld score="0.88">Topography</microWorld>
                            <microWorld score="0.72">Finance</microWorld>
                            <microWorld score="0.7">Bio</microWorld>
              </treeProfile>

              <struct type="LS">
                     <!--English example based on data provided by Irion -->
                     <!-- In this case, ID value corresponds to the one from the Irion tool
-->
                     <struct type="TS" id="t66">
                            <!-- data related to the word forms of the terms -->

                            <normalizedTerm>climate change</normalizedTerm> <!-- normalized
form of the term, in the case of Irion these are not proper lemmas, e.g. diacritics are
removed -->

                            <partOfSpeech>noun</partOfSpeech> <!-- more silimar to LMF -->

                            <preferredForm>Climate Change</preferredForm> <!-- this is
needed only if normalized form are not nice lemmas and there are multiple form
occurrences. Currently, we take to shortest form as the preferredForm but we can also use
the form frequency. Note that case can be maintained-->

                            <!-- Introduced a structure to group all the actual forms that
belong to the same term class -->
                            <forms>
                                   <!-- Another layer to store occurrences for each form --
>
                                   <!-- the id is generic: a type that represents the
tokens in SEMAF -->
                                   <termFormData id="tf_1" frequency="3">
                                          <termForm>climate change</termForm>
```

```
                                        <!-- SEMAF uses span attributes and form
identifiers to refer to word tokens in the text -->
                                        <spans docId="1234">
                                                <span from="w2" to="w3"/>
                                                <span from="w23" to="w25"/>
                                                <span from="w27" to="w28"/>
                                        </spans>
                                </termFormData>
                                <!-- I also added a form identifier -->
                                <termFormData id="tf_2" frequency="6">
                                        <termForm>climate changes</termForm>
                                        <spans docId="124">
                                                <span from="w24" to="w43"/>
                                                <span from="w123" to="w125"/>
                                                <span from="w5627" to="w5628"/>
                                        </spans>
                                        <spans docId="7824">
                                                <span from="w24" to="w43"/>
                                                <span from="w123" to="w125"/>
                                                <span from="w5627" to="w5628"/>
                                        </spans>
                                </termFormData>
                                <!-- If we group synonyms as one term class (which we
should for creating dense term hierarchies) then there will also be real different forms.
I added made up synonyms here for illustration-->
                                <termFormData id="tf_3" frequency="4">
                                        <termForm>climatological changes</termForm>
                                        <spans docId="1824">
                                                <span from="w24" to="w43"/>
                                                <span from="w123" to="w125"/>
                                                <span from="w5627" to="w5628"/>
                                        </spans>
                                        <spans docId="88">
                                                <span from="w24" to="w43"/>
                                        </spans>
                                </termFormData>
                        </forms>

                        <parentData>
                                <!-- parentTerm is used to represent the term hierarchy
-->
                                <parentTerm target="t13">change</parentTerm>
                                <!-- Other parents can be derived from other sources -->
                                <wikiCategory source="http://en.wikipedia.org/wiki"
date="2008-06-20">Climate change feedbacks and causes</wikiCategory>
                                <wikiCategory source="http://en.wikipedia.org/wiki"
date="2008-06-20">Global warming </wikiCategory>
                                <wikiCategory source="http://en.wikipedia.org/wiki"
date="2008-06-20">History of climate</wikiCategory>
                                <wikiCategory source="http://en.wikipedia.org/wiki"
date="2008-06-20">Carbon finance</wikiCategory>
                                <wikiCategory source="http://en.wikipedia.org/wiki"
date="2008-06-20">Climate and weather statistics</wikiCategory>
                        </parentData>
                        <termStatistics>
                                <documentNumber>5</documentNumber> <!-- number of
documents in the domain index the terms occurs in, this is the sum of unique document ids
in the spans of the termForms-->
                                <termFrequency>13</termFrequency> <!-- frequency count
of the term in the domain corpus, this should be the sum of the termForm frequencies -->

                                <termSalience>0.04</termSalience> <!-- number between
zero and 1 based on some tf*idf calculation. The relative frequency of the term is divided
by the relative number of documents it occurs in some reference corpus.-->

                                <termConnectivity>13</termConnectivity> <!-- number of
connections of a term class in the tree, also including structural contextual relations.
In the case of Irion these are limited to siblings and most frequent modifiers -->
```

```
                                    <cumulativeFrequency>18</cumulativeFrequency> <!-- the
frequency of all descendants below this term as an indication of the salience of the term
as a concept in the hierarchy. This sum includes the termFrequency and thus should be
equal or higher-->

                                    <cumulativeDocumentNumber>5</cumulativeDocumentNumber>
<!-- the number of documents of all descendants below this term, as an indication of the
salience of the term as a concept in the hierarchy. This include documentNumber and should
be equal or higher-->

                                    <termSiblings>3</termSiblings> <!-- the number of
siblings of the term -->
                             </termStatistics>

                             <!-- domain classification based on all the forms in the term
hierachy branch starting from the top -->
                             <!-- the attribute profileMatch indicates the overlap with the
overall treeProfile -->
                             <termProfile profileMatch="0.69">
                                          <microWorld score="0.88">Geography</microWorld>
                                          <microWorld score="0.75">Finance</microWorld>
                                          <microWorld score="0.73">Metereology</microWorld>
                                          <microWorld score="0.7">Society</microWorld>
                             </termProfile>

                             <sources>
                             <!-- we could use this to indicate the status of the text from
which the term occurs. Here TOC stands for table of content. Other values are CAPTION,
HEADING, INTRODUCTION, CONCLUSION, BODY, FOOTNOTE -->
                             <!-- the score could be used to indicate the proportion of
occurrence in the type of sections. The total adds up to 1.0-->
                                    <termSource score="0.2">TOC</termSource>
                                    <termSource score="0.8">BODY</termSource>
                             </sources>

                             <semanticRelations>
                                    <!-- any kind of defining phrase, comment or description
-->
                                    <definitions>
                                          <termDefiniton
source="http://en.wikipedia.org/wiki/Climate_change" date="2008-06-20">Climate change is
any long-term significant change in the "average weather" that a given region experiences.
Average weather may include average temperature, precipitation and wind patterns. It
involves changes in the variability or average state of the atmosphere over durations
ranging from decades to millions of years. These changes can be caused by dynamic process
on Earth, external forces including variations in sunlight intensity, and more recently by
human activities.</termDefiniton>
                                          <termDefiniton source="googleSnippets" date="2008-
06-20">factors such as climate changes affecting our oceans</termDefiniton>
                                          <termDefiniton source="googleSnippets" date="2008-
06-20">environniental problems such as climate changes or acid rains</termDefiniton>
                                          <termDefiniton source="googleSnippets" date="2008-
06-20">global environmental issues such as climate changes</termDefiniton>

                                          <termDefiniton source="googleSnippets" date="2008-
06-20">environmental changes such as climate changes</termDefiniton>
                                          <termDefiniton source="googleSnippets" date="2008-
06-20">related activities such as climate changes and changes in land use pattern
explanatory events such as climate changes</termDefiniton>
                                          <termDefiniton source="googleSnippets" date="2008-
06-20">all kinds of other geographical data such as climate changes, plant growth,
radiation, rainfall, forest fires</termDefiniton>
                                    </definitions>
                                    <!-- Semantic layer that summarizes the best synset
mappings from all occurrences -->
                                    <!-- This notation is based on the current SEMAF
proposal. If we change SEMAF this needs to be adapted -->
```

```
                                <semanticMatch type="senseAlt" orig="urn:wordnet1.7">
                                        <sense source="EHU-WSD1" sensecode="ENG30-
00180570-n" weight="0.80"/>
                                        <sense source="EHU-WSD1" sensecode="ENG30-
00290564-n" weight="0.30"/>
                                </semanticMatch>

                                <!-- separate match to the ontology should be added
since it may resolve fine-grained ambiguities at the synset level -->
                                <semanticMatch type="ontologyAlt" orig="urn:sumo">
                                        <ontology source="EHU-WSD1" class="Process"
weight="0.65"/>
                                        <ontology source="EHU-WSD1" class="NaturalProcess"
weight="0.70"/>
                                </semanticMatch>
                            </semanticRelations>
                            <!-- Complete list of structural relations as they are now
presented in the term structure tables -->
                           <structuralRelations>
                                <!-- Next two examples are most simple left & right
constituents of the term -->
                                <!-- We only keep the context if it contains a head that
is also included as a term and therefore has a term id -->
                                <structuralRelation syntaxRole="leftnp" semanticRole="">
                                        <syntaxElement/>
                                        <termFormData id="tf_23" frequency="1">
                                                <termForm>sayan</termForm>
                                                <deps docId="1824">
                                                <!-- Here we use the dep instead of the
wordSpan -->
                                                <!-- dep is a higher level of the layered
SEMAF notation -->
                                                        <dep from="t3" to="t4"/>
                                                </deps>
                                                <termContext/>
                                        </termFormData>
                                </structuralRelation>

                                <structuralRelation syntaxRole="leftnp" semanticRole="">
                                        <syntaxElement/>
                                        <termFormData id="tf_89" frequency="1">
                                                <termForm>location</termForm>
                                                <deps docId="1824">
                                                        <dep from="t6" to="t7"/>
                                                </deps>
                                                <termContext/>
                                        </termFormData>
                                </structuralRelation>



                                <!-- Based on the term structure tables these can be
extended to provide richer data -->

                                <!-- PP occurring to the right of the term as NP -->
                                <structuralRelation syntaxRole="np_right_pp"
semanticRole="LOCATION">
                                        <syntaxElement>in</syntaxElement>
                                        <termFormData id="t65" frequency="2">
                                                <termForm>tropical area</termForm>
                                                <deps docId="124">
                                                        <dep from="t6" to="t8"/>
                                                </deps>
                                        </termFormData>
                                        <termFormData id="t68" frequency="1">
                                                <termForm>marine area</termForm>
                                                <deps docId="124">
                                                        <dep from="t10" to="t11"/>
```

```
                                        </deps>
                                        <termContext>climate change in marine
areas</termContext>
                                  </termFormData>
                            </structuralRelation>

                            <structuralRelation syntaxRole="np_right_pp"
semanticRole="TIME">
                                  <syntaxElement>from</syntaxElement>
                                  <termFormData id="t256" frequency="1">
                                        <termForm>1970</termForm>
                                        <deps docId="124">
                                              <dep from="t6" to="t8"/>
                                        </deps>
                                        <termContext>climate change from 1970 to
2005</termContext>
                                  </termFormData>
                            </structuralRelation>

                            <!-- NP to the left of the term as PP -->
                            <structuralRelation syntaxRole="pp_left_np"
semanticRole="CAUSE">
                                  <syntaxElement>of</syntaxElement>
                                  <termFormData id="t597" frequency="1">
                                        <termForm>impact</termForm>
                                        <deps docId="124">
                                              <dep from="t13" to="t15"/>
                                        </deps>
                                        <termContext>impact of climate
change</termContext>
                                  </termFormData>
                            </structuralRelation>

                            <!-- Term is the subject of the main verb in an ACTIVE
sentence -->
                            <structuralRelation syntaxRole="subj" active="true"
semanticRole="PATIENT">
                                  <syntaxElement/>
                                  <termFormData id="t33" frequency="1">
                                        <termForm>accellerate</termForm>
                                        <deps docId="124">
                                              <dep from="t20" to="t28"/>
                                        </deps>
                                        <termContext>climate change
accellerated</termContext>
                                  </termFormData>
                            </structuralRelation>


                            <!-- Term is the subject of the main verb in an ACTIVE
sentence -->
                            <structuralRelation syntaxRole="subj" active="true"
semanticRole="AGENT">
                                  <syntaxElement/>
                                  <termFormData id="t11" frequency="1">
                                        <termForm>cause</termForm>
                                        <deps docId="124">
                                              <dep from="t56" to="t63"/>
                                        </deps>
                                        <termContext>climate change causes a
decline of biodiversity</termContext>
                                  </termFormData>
                            </structuralRelation>


                            <!-- Term is the subject of the main verb in a PASSIVE
sentence -->
```

```
                                    <structuralRelation syntaxRole="subj" active="false"
semanticRole="PATIENT">
                                          <syntaxElement/>
                                          <termFormData id="t11" frequency="1">
                                                <termForm>cause</termForm>
                                                <deps docId="124">
                                                      <dep from="t356" to="t359"/>
                                                </deps>
                                                <termContext>climate change is caused by an
increase in industrial activity</termContext>
                                          </termFormData>
                                    </structuralRelation>

                                    <!-- Term is modified by an adjective or adverb-->
                                    <structuralRelation syntaxRole="mod"
semanticRole="ATTRIBUTE">
                                          <syntaxElement/>
                                          <termFormData id="t111" frequency="1">
                                                <termForm>rapid</termForm>
                                                <deps docId="124">
                                                      <dep from="t356" to="t359"/>
                                                </deps>
                                                <termContext>rapid climate
changes</termContext>
                                          </termFormData>
                                    </structuralRelation>

                              </structuralRelations>
                        </struct>
                  </struct>
</struct>
</tmf>
```

12 Appendix B – Kyoto-LMF wordnet: list of values of attribute ‘relType’ for SynsetRelation elements

antonym
antonym_comp
be_in_state
category
category_term
causes
co_agent_instrument
co_agent_patient
co_agent_result
co_instrument_agent
co_instrument_patient
co_instrument_result
co_patient_agent
co_patient_instrument
co_patient_result
co_result_agent
co_result_instrument
co_result_patient
co_role
for_purpose_of
fuzzynym
gloss
has_derived
has_holo_location
has_holo_madeof
has_holo_member
has_holo_part
has_holo_portion
has_holonym
has_hyperonym
has_hyponym
has_mero_location
has_mero_madeof
has_mero_member
has_mero_part
has_mero_portion
has_meronym
has_pertainym
has_subevent
has_xpos_hyperonym
has_xpos_hyponym
in_manner
instance
involved
involved_agent
involved_direction
involved_instrument
involved_location
involved_patient
involved_result
involved_source_direction
involved_target_direction
is_a_value_of
is_caused_by

is_derived_from
is_subevent_of
manner_of
near_antonym
near_synonym
nearest
pertains_to
region
region_term
related
related_to
results_in
rgloss
role
role_agent
role_direction
role_instrument
role_location
role_manner
role_patient
role_result
role_source_direction
role_target_direction
see_also_wn15
state_of
usage
usage_term
verb_group
xpos_fuzzynym
xpos_near_antonym
xpos_near_synonym

13 Appendix C – Kyoto-LMF wordnet: list of values of attribute ‘relType’ for SenseAxis elements

eq_synonym
eq_near_synonym
eq_has_hypernym
eq_has_hyponym
eq_involved
eq_role
eq_is_caused_by
eq_causes
eq_has_holonym
eq_has_meronym
eq_has_subevent
eq_is_subevent_of
eq_be_in_state
eq_is_state_of
eq_co_role
eq_generalization
eq_metonym
eq_diathesis
eq_in_manner
eq_has_instance
eq_belongs_to_class
eq_antonym

14 Appendix D – Kyoto-LMF wordnet: example representation of English synset “Department_of_Justice_1”

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LexicalResource SYSTEM "kyoto_wn.dtd">
<LexicalResource>
 <GlobalInformation label="Proposal for Kyoto-internal WordNet representation"/>
 <Lexicon languageCoding="ISO 639-3" label="English Wordnet 1.6, Meaning" language="eng"
owner="Princeton" version="1.6">
   <LexicalEntry id="Department_of_Justice">
                <Meta author="claudia" date="06-06-08"/>
        <Lemma writtenForm="Department_of_Justice" partOfSpeech="N"></Lemma>
        <Sense id="Department_of_Justice_1" synset="ENG-16-06060223-n">
        <MonolingualExternalRefs>
                <MonolingualExternalRef externalSystem="Wordnet3.0"
externalReference="department_of_justice%1:14:00::"/>
                </MonolingualExternalRefs>
                </Sense>
        </LexicalEntry>
   <Synset id="ENG-16-06060223-n" baseConcept="1">
   <Meta author="piek" date="2008-05-12"/>
        <Definition gloss="bla bla">
                <Statement example="bla bla"/>
        </Definition>
        <SynsetRelations>
                <SynsetRelation targets="EU-16-06056130-n" relType="has_hyperonym">
                <Meta author="german" date="2008-05-12" status="yes" source="whatsoever"
confidenceScore="99"/>
                        </SynsetRelation>
                <SynsetRelation targets="EU-16-06060479-n" relType="has_mero_part">
                <Meta author="german" date="2008-05-12" status="true" source="whatsoever"
confidenceScore="99"/>
                        </SynsetRelation>
                <SynsetRelation targets="EU-16-00403152-n" relType="gloss">
                <Meta author="monica" date="2008-05-27" status="false" source="whatsoever"
confidenceScore="0.3"/>
                        </SynsetRelation>
                </SynsetRelations>
        <MonolingualExternalRefs>
        <MonolingualExternalRef externalSystem="Domain" externalReference="administration">
                <Meta author="monica" date="2008-05-27"/>
  </MonolingualExternalRef>
        <MonolingualExternalRef externalSystem="Domain" externalReference="law">
                <Meta author="monica" date="2008-05-27"/>
  </MonolingualExternalRef>
        <MonolingualExternalRef externalSystem="SuperSense" externalReference="act">
                <Meta author="monica" date="2008-05-27"/>
  </MonolingualExternalRef>
        <MonolingualExternalRef externalSystem="SUMO" externalReference="PoliticalProcess"
relType="at">
                <Meta author="monica" date="2008-05-27"/>
  </MonolingualExternalRef>
        <MonolingualExternalRef externalSystem="TCO" externalReference="Agentive"/>
        <MonolingualExternalRef externalSystem="TCO" externalReference="Purpose"/>
```

```
        <MonolingualExternalRef externalSystem="TCO" externalReference="Social"/>
        <MonolingualExternalRef externalSystem="TCO" externalReference="UnboundedEvent"/>
        </MonolingualExternalRefs>
    </Synset>
  </Lexicon>
  <SenseAxes>
  <SenseAxis id="sa_en16-en30_001" relType="equal_synonym">
                <Meta author="monica" date="2008-05-27"/>
                  <Target ID="EN-16-06060223-n"/>
                  <Target ID="EN-30-08135342-n"/>
                <InterlingualExternalRefs>
       <InterlingualExternalRef externalSystem="SUMO" externalReference="PoliticalProcess"
relType="at">
                <Meta author="claudia" date="06-06-2008"/>
                </InterlingualExternalRef>
                </InterlingualExternalRefs>
  </SenseAxis>
  <SenseAxis id="sa_en16-en30_002" relType="equal_synonym">
                <Meta author="monica" date="2008-05-27"/>
                        <Target ID="EN-16-01661609-v"/>
                        <Target ID="EN-30-02439732-v"/>
                <InterlingualExternalRefs>
       <InterlingualExternalRef externalSystem="SUMO" externalReference="PoliticalProcess"
relType="at">
                <Meta author="claudia" date="06-06-2008"/>
                </InterlingualExternalRef>
                </InterlingualExternalRefs>
  </SenseAxis>
  <SenseAxis id="sa_en16-en30_003" relType="equal_synonym">
        <Meta author="monica" date="2008-05-27"/>
                        <Target ID="EN-16-00584005-a"/>
                        <Target ID="EN-30-00619433-a"/>
                <InterlingualExternalRefs>
       <InterlingualExternalRef externalSystem="SUMO" externalReference="PoliticalProcess"
relType="at">
                <Meta author="claudia" date="06-06-2008"/>
                </InterlingualExternalRef>
                </InterlingualExternalRefs>
  </SenseAxis>
  </SenseAxes>
</LexicalResource>
```

15 Appendix E - – Kyoto-LMF wordnet: DTD

```
<?xml version='1.0' encoding="UTF-8"?>
<!ELEMENT LexicalResource (GlobalInformation, Lexicon+, SenseAxes?)>
<!ELEMENT GlobalInformation EMPTY>
<!ATTLIST GlobalInformation
label CDATA #IMPLIED>
<!ELEMENT Lexicon (LexicalEntry+, Synset*)>
<!ATTLIST Lexicon
languageCoding CDATA #FIXED "ISO 639-3"
label CDATA #IMPLIED
language CDATA #REQUIRED
owner CDATA #REQUIRED
version CDATA #REQUIRED>
<!ELEMENT LexicalEntry (Meta?, Lemma, Sense*)>
<!ATTLIST LexicalEntry
id ID #IMPLIED>
<!ELEMENT Lemma EMPTY>
<!ATTLIST Lemma
writtenForm CDATA #REQUIRED
partOfSpeech CDATA #REQUIRED>
<!ELEMENT Sense (Meta?, MonolingualExternalRefs?)>
<!ATTLIST Sense
id ID #REQUIRED
synset IDREF #REQUIRED>
<!ELEMENT Meta EMPTY>
<!ATTLIST Meta
author CDATA #IMPLIED
date CDATA #IMPLIED
source CDATA #IMPLIED
status CDATA #IMPLIED
confidenceScore CDATA #IMPLIED>
<!ELEMENT Synset (Meta?, Definition?, SynsetRelations, MonolingualExternalRefs)>
<!ATTLIST Synset
id ID #REQUIRED
baseConcept (1|2|3) #REQUIRED>
<!ELEMENT Definition (Statement*)>
<!ATTLIST Definition
gloss CDATA #REQUIRED>
<!ELEMENT Statement EMPTY>
<!ATTLIST Statement
example CDATA #REQUIRED>
<!ELEMENT SynsetRelations (SynsetRelation+)>
<!ELEMENT SynsetRelation (Meta?)>
<!ATTLIST SynsetRelation
targets IDREFS #REQUIRED
relType CDATA #REQUIRED>
<!ELEMENT MonolingualExternalRefs (MonolingualExternalRef+)>
<!ELEMENT MonolingualExternalRef (Meta?)>
<!ATTLIST MonolingualExternalRef
externalSystem CDATA #REQUIRED
externalReference CDATA #REQUIRED
relType (at|plus|equal) #IMPLIED>
<!ELEMENT SenseAxes (SenseAxis+)>
```

```
<!ELEMENT SenseAxis (Meta?, Target+, InterlingualExternalRefs?)>
<!ATTLIST SenseAxis
id ID #REQUIRED
relType CDATA #REQUIRED>
<!ELEMENT Target EMPTY>
<!ATTLIST Target
ID CDATA #REQUIRED>
<!ELEMENT InterlingualExternalRefs (InterlingualExternalRef+)>
<!ELEMENT InterlingualExternalRef (Meta?)>
<!ATTLIST InterlingualExternalRef
externalSystem CDATA #REQUIRED
externalReference CDATA #REQUIRED
relType (at | plus | equal) #IMPLIED>
```