Common Language Resources and Technology Infrastructure

Interoperability and Standards

2010-12-23 Version: 1

Editors: Erhard Hinrichs, Iris Vogel

www.clarin.eu

The ultimate objective of CLARIN is to create a European federation of existing digital repositories that include language-based data, to provide uniform access to the data, wherever it is, and to provide existing language and speech technology tools as web services to retrieve, manipulate, enhance, explore and exploit the data. The primary target audience is researchers in the humanities and social sciences and the aim is to cover all languages relevant for the user community. The objective of the current CLARIN Preparatory Phase Project (2008-2010) is to lay the technical, linguistic and organizational foundations, to provide and validate specifications for all aspects of the infrastructure (including standards, usage, IPR) and to secure sustainable support from the funding bodies in the (now 23) participating countries for the subsequent construction and exploitation phases beyond 2010.

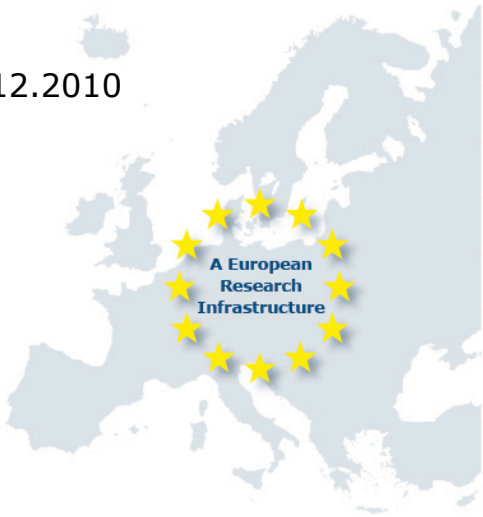
Interoperability and Standards

CLARIN-D5C-3

EC FP7 project no. 212230

Deliverable: D5.C-3 - Deadline: 31.12.2010

Responsible: Erhard Hinrichs



Contributing Partners: UTU, UHEL, IPPBAS, KTH, IMCS, IPIPAN, ILC-CNR, ILSP, MPI

Contributing Members: BBAW, U Hamburg, U Stuttgart, Vienna

With contributions from: Piotr Bański, Kathrin Beck, Gerhard Budin, Tommaso Caselli, Kerstin Eckart, Kjell Elenius, Gertrud Faaß, Maria Gavrilidou, Verena Henrich, Valeria Quochi, Lothar Lemnitzer, Wolfgang Maier, Monica Monachini, Jan Odijk, Maciej Ogrodniczuk, Petya Osenova, Petr Pajas, Maciej Piasecki, Adam Przepiórkowski, Dieter Van Uytvanck, Thomas Schmidt, Ineke Schuurman, Kiril Simov, Claudia Soria, Inguna Skadina, Jan Stepanek, Pavel Stranak, Paul Trilsbeek, Thorsten Trippel, Iris Vogel

Contents

Background

WP5 has the task of producing a deliverable entitled „Interoperability and Standards“. In the CLARIN Document of Work, this task is defined as follows:

Working Groups set up in this WP will specify the requirements for the registries of resources and tools for the representational standards for the various types of resources. Some of the work will be done in close collaboration with WP2 (technical standards). Based on these specifications, the WP will study generic frameworks such as LMF for lexical resources (ISO TC37/SC4) and their and give recommendations for converters which transforms existing language resources into the CLARIN representational standards and formats. Help will be given to users to transform the required resources into standard formats. Reports will describe the generic frameworks and the conversion tools required.

It is meant as an internal document to stimulate discussion on this important topic which is of immediate concern to at least WP2, WP3, WP5 and WP7.

Introduction

Standards play a central role in an integrative and interoperable domain of language resources and technology as envisaged by CLARIN. Wherever two or more parties interact with each other such agreements are a necessary prerequisite. The specification of encoding standards plays a role when for example the quality of services is evaluated or the interoperability with other services is determined.

This document gives an overview of the representative standards and best practices for language resources and their current state of development and usage. It also introduces methods and examples for converters in these standards. This is not an exhaustive list or a statement about the quality of the discussed standards and converters, but a snapshot of current activities, of the standards in common use and of best practices in the encoding and interchange of language resources (text corpora, lexica, multimedia, etc.).

CLARIN wants to develop a common, pan-European infrastructure for language resources, tools and services for use in the humanities. A key requirement of such a common infrastructure is the ability to exchange information across different resources and across different services that extract linguistic information or annotate language resources at different analysis levels. In short, the CLARIN infrastructure needs to be interoperable. This, in turn, presupposes that language resources and tools need to utilize, as much as possible, common data formats that are compliant with encoding standards and best practices.

CLARIN does rely as much as possible on existing standards, guidelines and best practices. Several CLARIN partners have been long-standing members of standards organizations so that there is easy information flow between these activities and the CLARIN membership. Apart from officially recognized standards, CLARIN will also track de-facto standards and best-practice encodings for those areas of language resources and tools for which no published standards are available yet. For standards that pertain to neighboring disciplines in the humanities, CLARIN will seek active collaboration with standards organizations and initiatives in these fields.

It is important that the specification of standards is easily accessible so that developers of language resources and tools can refer to them and, if appropriate, give feedback to the relevant standards committees. Ideally, existing standards are already followed by a large community of users. For these well-established standards, CLARIN will ensure that data conversions tools are available.

Standards serve different functions in the creation or curation of language resources and tools:

1. For existing resources and tools with often heterogeneous data formats, standards can be used for the specification of interchange (pivot) formats. This will facilitate the development of conversion tools that translate heterogeneous data formats into such pivot formats and back. The availability of such pivot formats will greatly simplify the specification and implementation of flexible processing chains for language resources.
2. For the development of new resources and tools, standards should be followed as much as possible. Creators of new corpora and lexical resources, for example, should consult the ISO standards pertaining to language as well the TEI guidelines on text encoding. Adherence to such established or emerging standards will minimize the need for data conversion in the first place.
3. Other standards may guide the interaction and exchange of resources and tools that have been created by other initiatives world-wide. Through the process of such international collaborations, best practices will continue to emerge that are empirically validated and that are broadly accepted by the scientific community.

International standardization committees, such as the ISO/TC 37/SC 4 Language Resource Management, the W3C Consortium, and other standardization bodies play a central role in developing guidelines for the processing, storage and modeling of language resources to facilitate reusability, merging and comparison of linguistic information independent of the language used.

A standard developed by the International Standardization Organization (ISO) goes through six stages from the first proposal to the final publication of the International Standard. Initial approval of a new work item leads to the formation of a technical committee (TC) and subcommittees (SC) for specific work items.

These expert groups prepare working drafts (WD) which -if approved by the relevant TC/SC members- will lead to the registration of an approved work item (AWI, Preparatory stage). The ISO registration process then starts with the first Committee Draft (CD), which will be distributed in form of a draft International Standard (DIS) to all ISO member bodies for commenting and voting (Committee stage). Once approved for submission as a Finalized Draft International Standard (FDIS, Enquiry stage), it has to pass a final vote by all ISO members (Approval stage) to become an ISO Standard. The new International Standard will be published by the ISO Central Secretariat (Publication stage).

Resource	Standard	Description	Responsibility	Version
Text	Unicode	Universal Character Set consisting of a repertoire of ca. 100.000 characters	ISO 10646 Working Group (SC 2/WG 2), Unicode Consortium	Unicode 5.1.0
Text	Language Codes	Codes for the representation of names of	TC 37/SC 2/WG 1	ISO 639-x, etc.

		languages		
Text	Character Encoding	Coded Character Set	JTC 1/SC 2	ISO/IEC 8859-X, etc.
Text	XML	Extensible Markup Language SGML (ISO 8879) extended by TC2 ISO/IEC JTC 1/SC 34 N 029:1998-12-06	W3C XML Working Groups	XML 1.0 XML 1.1
Text	TermLR	Terminology for Language Resource Management	ISO/TC 37/SC 4 WG1	ISO/WD 21829
Text Lexicon	LMF	Lexical Markup Framework	ISO/TC 37/SC 4/WG 4	ISO 24613:2008 60.60
Text Corpus	LAF	Linguistic Annotation Framework	ISO/TC 37/SC 4/WG 1	ISO/DIS 24612 40.60
Text Corpus	MAF	Morpho-Syntactic Annotation Framework	ISO/TC 37/SC 4 WG2	ISO/DIS 24611 40.60
Text Corpus	FSR	Feature Structure Representation	ISO/TC 37/SC 4 WG1	ISO/24610-1:2006 90.92
Text Corpus	FSD	Feature Structure Description	ISO/TC 37/SC 4 WG1	ISO/DIS 24610-2 40.99

Text Corpus	WordSeg1	Word segmentation of written texts for mono-lingual and multi-lingual information processing — Part 1: Basic concepts and general principles	ISO/TC 37/SC 4 WG2	ISO/DIS 24614-1 40.99
Text Corpus	WordSeg2	Word segmentation: Part 2 Chinese, Japanese, Korean	ISO/TC 37/SC 4 WG 2	ISO/WD 24614-2 30.60
Text Corpus	SemAF	Semantic Annotation Framework	ISO/TC 37/SC 4 WG 2	Part 1: ISO/CD 24617-1 40.99 Part 2: ISO/DIS 24617-2 30.60
Text Corpus	SynAF	Syntactic Annotation Framework	ISO/TC 37/SC 4/WG 2	ISO/DIS 24615:2010 60.60
Text Corpus Lexicon?	MLIF	Multilingual Information Framework	ISO/TC 37/SC 4/ WG3	ISO/WD 24616 40.20
Text Terminology	DCR	Data Category Registry	ISO/TC 37/SC 4 WG1	ISO 12620:1999 ISO/DIS 12620.2

1. General Standards

1.1. Unicode

The Unicode Standard aims at providing a consistent method of storing letters and characters for most of the world's writing systems independent of the operating system or program used. The idea behind it is to assign a unique number to each character, symbol or control sequence used to display the writing systems used around the globe and thereby allow data of different languages to be passed

through different platforms without the risk of corruption. It is developed and maintained by the Unicode Consortium, a non-profit organization open to individuals and organizations and financed solely by membership dues. The Unicode Standard has been widely adopted in the commercial and academic world and has become the official way to implement ISO/IEC 10646.

The current Version Unicode 5.2 comprises 107,296 characters, ranging from the ASCII character set over the contemporary European alphabetic scripts, Middle Eastern right-to-left scripts, and scripts of Asia and Africa. An increasing number of historic scripts are included as well (The Unicode Standard / the Unicode Consortium. Version 5.2, 2009). For the implementation of the Unicode Character Set (UCS) the Unicode transformation formats UTF-8 and UTF-16 and – to a lesser extent – UTF-32 are the most commonly used encodings. They cover the same character set, but are using different approaches in mapping the character code points to character sequences. Therefore lossless conversion between the different Unicode transformation formats is guaranteed. Conversion of character data in local character encodings into the standard Unicode encodings (UTF-8, UTF-16 and UTF-32) is supported by common text processing software and programming languages.

The choice of encoding largely depends on the requirements of the application, the kind of characters being represented, respectively available storage space, and also the environment (UTF-8 is most popular encoding on the WWW, partly because of its ASCII compatibility, UTF-16 is mainly used by Java and Windows, UTF-32 by some Unix-based systems). Unicode is about to replace a myriad of different encodings and thereby create an environment for true multilingual cross-platform processing.

Unicode is recommended by CLARIN as the key standard for character encoding in data processing and interchange. Most tools and services provided by CLARIN use UTF-8 or UTF-16 as import and export format as well as for internal encoding. Extensive documentation, code charts and character databases are available at the organizations website (http://www.unicode.org).

1.2. XML

XML stands for Extensible Markup language and describes a class of data objects called XML-documents. XML is a restricted form of SGML (Standardized Generalized Markup Language, ISO 8879) and was developed in the 1990s as a meta-language formalism to be deployed on the World Wide Web. The current Specification XML 1.0 is an open standard available at the W3C website (http://www.w3.org/TR/REC-xml).

The encoding of information in XML documents relies exclusively on characters for content (character data) and markup (logical or structural information), which makes them human readable and – in combination with support via Unicode – paves the way for the processing of such documents independent of a particular application or hardware. There are numerous programming interfaces and applications available and a variety of schema languages (e.g., DTD, XML-Schema, RelaxNG, Schematron) support the definition of a markup language customized for a particular purpose. In addition many pre-defined XML-based markup languages for specific needs (e.g., RSS for newsfeed, XHTML for web design, SOAP for web services) are also publicly available, which makes XML a key player in the field of interoperability and data exchange.

CLARIN supports and recommends the use of XML as a text format in combination with Unicode for information encoding and interchange. Even though internal data encoding might differ due to the limitations of XML, all of the promoted text encoding standards and even some of the media

encoding standards are based on XML. XML-based markup languages specified for linguistic purposes include TEI, LMF, VoiceXML, MaF, GraF, SynAF and many more. Therefore examples in XML will be an integral part of the following chapters.

1.3. TEI

The Text Encoding Initiative (TEI) Consortium is a non-profit organization supported by the academic community. It develops and maintains guidelines for the representation of digital text resources in form of XML encoding schemes as published along with documentation on the TEI web site (http://www.tei-c.org). The aim of the initiative is to provide an open format for academic community that allows for full interoperability.

Even though the TEI is an ongoing project and work is not complete in all areas, it has become a de-facto standard for text markup in the humanities. The current version P5 of the TEI guidelines defines general modules, such as the TEI header for metadata about the resource, generic elements common to all kinds of texts and close to 500 elements arranged in modules according to the kind of information or the resource type they distinctively model.

TEI is designed to be applicable to a wide variety of textual resources in various disciplines. The inbuilt flexibility leads to multiple possibilities to encode the same phenomenon, which basically makes it necessary to choose between several options and thereby constraining or simplifying the TEI specifications in order to ensure interoperability (Ide, 1998). Since the specifications aim to capture all aspects of textual resources, they contain not only markup for the logical structuring of textual data but also for typographic information needed for purposes of publication.

Due to the versatile nature of TEI, most of the following chapters include details on encoding digital text by following the P5 guidelines and conversion methods. However information on the TEI header is only mentioned for the sake of completeness and will not be essential part of this deliverable, since the topic of metadata development is being covered in detail by Work Package 2.

2. Lexica and Terminology Standards

2.1. Terminology, Definitions

Author: Thorsten Trippel

In comparison to other standardization initiatives, the standardization in the areas of terminology is much more advanced, looking back at a tradition of several decades, the International Organization of Standardization (ISO) Technical Committee 37 (TC 37) was organized 1947 resulting from the affinity to the technical field and the distance to traditional linguistics, combined with the requirements of standardization for technical writers and translators. This technical committee works on the theory and practice of terminology, lexicons and dictionaries as published by the publishing industry, and electronic representation of such resources. Associated with this Technical Committee is also the subcommittee on language resources (ISO TC 37 SC4), established 2001, when similar issues and requirements for standardization were seen and earlier projects on standardization of language resources (such as EAGLES and ISLE) were integrated into the formal ISO work.

In the context of CLARIN, the following non-SC 4 standards are relevant or may seem relevant:

1. ISO 639 (Parts 1-6): Codes for the representation of names of languages
2. ISO 1951:2007 Presentation/representation of entries in dictionaries — Requirements, recommendations and information
3. ISO 1987 (Part 1 and 2):2000 Terminology work — Vocabulary
4. ISO 12200:1999 Computer applications in terminology — Machine-readable terminology interchange format (MARTIF) – Negotiated interchange
5. ISO 12620:2009 Terminology and other language and content resources — Specification of data categories and management of a Data Category Registry for language resources
6. ISO 16642:2003 Computer applications in terminology – Terminological markup framework
7. ISO/CD 22274 Systems to manage terminology, knowledge and content — Internationalization and concept-related aspects of classification systems
8. ISO 30042:2008 Systems to manage terminology, knowledge and content — TermBase eXchange (TBX)

ISO 639 is widely adopted by various fields in the knowledge industry as it defines the 2 letter codes (ISO 639-1) for languages and two separate 3 letter codes — ISO-639-2 resulting originally from a Library of Congress notation adapted by librarians and the more extensive ISO 639-3, which is more linguistically motivated. In the CLARIN context, i.e., in the context of language resources, the use of ISO 639-3 is recommended. The registrar — also for adding languages or change requests — is the SIL (see http://www.sil.org/iso639-3/). ISO 639-4, “Language families”, describes general principles of language codings, and groups are assigned codes in ISO 639-5. Ways of encoding language variants are finally provided with 4 letter codes in ISO 639:6.

ISO 1951 can be seen as relevant for the representation of lexical resources, but this standard is driven by publishing houses and targeted at commercial dictionaries and similar resources. For research oriented lexical resources, this standard can be discarded.

ISO 1987 provides the terminology of terminology that is principles and practices of terminology work as used in the translation processes and terminological practices. Though this is rather

fundamental, the practical consequences for language resources can be seen in the need for explicit and structured definitions, precise terminology, etc.

The family of standards dealing with the exchange of terminological databases is ISO 12200, ISO 16642 and ISO 30042. ISO 12200 (MARTIF) defines a data format for the interchange of terminological databases. This interchange requires the participants in the process to negotiate the data structures. The background is that different applications use data categories and substructures within terminological databases in different ways, but usually the content of the data categories in a source structure can be relocated to other areas of the target structures.

ISO 12200 is partially superseded by the introduction of a two level model of terminological databases in ISO 16642 (TMF) and ISO 30042 (TBX). TMF defines the model of terminological databases, which is also applicable to all forms of onomasiological lexical resources. It also provides a serialization in XML, i.e., a generic representation of onomasiological lexicons that is not sufficient for interchange, as data categories and data category structures are not defined. TBX is a standard that uses the TMF model to describe two levels of termbases — the organization of data categories with the relation between them, and the syntax of interchange files. With both descriptions, a blind interchange becomes possible, i.e. the interchange between terminological databases without defining a mapping of data categories.

The most central standard for language resources is ISO 12620, defining the procedure of registering and using data categories. Originally this standard was created listing the terminological data categories as needed by MARTIF. However, when a similar standard was developed for other language resources, it became apparent that the overlaps were large and the principles similar. In the standardization discussion it was also made clear that a closed list of data categories was deemed to fail. The result was the development of the Data Category Registry ISOcat (see http://www.isocat.org), which is an online database for data categories, in which every interested party can insert new data categories. If such a data category is intended for standardization and reuse, a group of subject matter experts nominated by the standardization organization can accept a category as standard or send it back to the originator for revision. Though ISOcat does not contain a formal semantic description of the data categories, it contains a standard reference point for the use and reuse of data categories, names and concepts. The data categories are referenced by Persistent Identifiers (PID) hence schemas and other document descriptions can point there for defining reference.

One of the newest standards is ISO 22274, currently (December 2010) close to the pre-final stage of being published as an international standard. Though the motivation of creating classification systems is again industry driven, it is apparent for example in the discussion of component models for metadata that classification systems are also important for the organization of language resources. Classification systems, in contrast to concept systems, require a unique classification of a concept or device, without requiring theoretical soundness. Ideally a concept system and a classification system would be closely related but due to theoretical arguments, ambiguity and different underlying theories, this cannot be assumed for concept systems. Classification systems, however, accept those shortcomings and define procedures for transparently defining a unique system of objects, discarding theoretical problems of underspecified concepts for a sound and unique concept system.

For language resources the most important standards from the fields of terminology therefore seem to be the ISO 639-3 (3 letter codes for languages), ISO 12620 (the data categories), ISO 16642 (the LMF twin standard for onomasiological resources), and ISO 22274 (classification systems). ISO 1087 (terminology of terminology), ISO 1959 (print dictionary representation), ISO 12200 (MARTIF), and ISO 30042 (TBX) may be relevant for restricted contexts as well.

2.2. Lexical Markup Framework (LMF)

Author: Valeria Quochi

This paragraph is intended to be a general introduction to the Lexical Markup Framework (LMF). For a detailed description of the formalism the reader should refer to the ISO-24613 specification document (http://www.tc37sc4.org/new_doc/LMF_rev14_For_DIS_Ballot%5B1%5D.pdf).

Some detail can be derived from the examples reported at the end of the paragraph.

LMF (ISO 24613:2008) is an abstract meta-model for the representation of computational lexicons. LMF allows the encoding of linguistic information in a way that should enable reusability of the data in different applications and for different tasks. LMF provides a common, shared representation of lexical objects that allows for the representation of rich linguistic information including morphological, syntactic, and semantic aspects. LMF does not deal with the general grammar of a language or with world knowledge representation.

The goals of LMF are to provide a common model for the construction of NLP lexicons of different magnitudes, to manage the exchange of data between and among these resources, and to enable the merging of a large number of individual electronic resources to form extensive global electronic repositories. Therefore, LMF presents a model that may in the future allow the development of XML serializations that will enable content interoperability across electronic lexical resources.

LMF is organized in several different packages: the core package describes the basic hierarchy of information included in a lexical entry that has to be taken as mandatory by lexicon developers. The core package then is enriched by resources that are part of the definition of LMF. These resources include:

- Specific data categories that are to be used to adorn both the core model and by the other resource types associated with LMF;
- Standard procedures for defining and expressing data categories and for attributing them to the various lexical objects, both of the core model and of extensions;
- Extensions of the core package are: Machine Readable Dictionaries, Natural Language Processing electronic lexical resources (the Morphology, Syntax, Semantics extensions).

Extensions are anchored to the model as a subset of the core package classes. An extension cannot be used to represent lexical data independently of the core package: core classes are mandatory; however, according to the types of linguistic data involved, an extension can depend on another extension, or a core class may be bypassed.

Individual instantiations of LMF can include, among many others, monolingual, bilingual or multilingual specialized lexicons. LMF provides general structures and mechanisms for designing new electronic lexical resources; therefore, the same specifications in principle can be used for both small and large lexicons (Francopoulo et al., 2006).

In the following subsections, we will briefly introduce the LMF meta-model with its NLP extensions. For more detailed explanations please refer to the documentation to be found at http://www.lexicalmarkupframework.org/ (LMF-revision-16).

LMF Core Model

The LMF core model consists of a minimal set of objects fundamental to encode a lexical resource. GlobalInformation and LexicalResource should be used to encode administrative and general information on the lexicon or set of lexica represented (i.e., character encoding etc.). The Lexicon

object represents the actual lexical resource and contains at minimum all lexical entries, plus patterns (like subcategorization frames), and other shared linguistic objects (such as semantic predicates). Lexical Entry is the main entry which then has 2 main objects as its aggregates that constitute the three major representation layers in linguistic description: Form (i.e., morphosyntax), and Sense (semantics). Different orthographies of the same entry should be captured in the RepresentationForm.

LMF NLP Morphology Extension

The purpose of the Morphology Extension is to provide the mechanisms to support the development of lexicons that need to have either an extensional or an intensional description of lexical entries. With "extensional" we mean that all inflected forms are explicitly described in the Lexicon; whereas with “intensional” we mean that not all forms are represented in the lexicon, but a Paradigm Class may be provided that specifies the mechanisms for the generation of all inflected forms. The morphology extension consists of two types of Form subclasses: one set or subclasses that represent sets of grammatical variants that make up the abstract lexeme, and another set of subclasses that represent words, morphemes, and MWEs that can be related to a form in another Lexical Entry.

LMF NLP Syntax Extension

The Syntax extension has the goal of describing the properties of a word when it combines with other words in a sentence. When recorded in a lexicon, the syntactic properties make up the syntactic description of a LexicalEntry instance. What is represented in a Lexicon, however, is not the general grammar of a language, but only the description of specific syntactic properties of words. The syntax extension is aggregated to the Lexical Entry and to the Sense. In this triangle, formed by the 3 key objects (LexicalEntry, SyntacticBehavior, the Sense), only the Lexical Entry is mandatory, the others are optional.

SyntacticBehaviour is the class that represents one of the possible syntagmatic behaviors of a word. A detailed description of the syntactic behavior of a lexical entry is then defined by the SubcategorizationFrame, which represents one syntactic construction. A SubcategorizationFrame may be shared by more LexicalEntry instances, i.e, by words that show the same syntagmatic behavior in one language. Because a SubcategorizationFrame can also inherit properties from another more generic SubcategorizationFrame, it is possible to implement a hierarchical structure of syntactic constructions.

LMF NLP Semantics Extension

The Semantics extension is to be used to describe the senses of lexical items and their relationships with other senses. This extension also provides the linking mechanisms and structures between syntax and semantics. The semantics extension is plugged-in the general model through the Sense class; i.e. all objects describing semantic aspects of an entry are be related to Sense, and hence indirectly to LexicalEntry. Sense, however, may not be shared by different Lexical Entries. Among the representational devices of this extension, the most important ones are PredicativeRepresentation, SemanticPredicate and SynSemArgMap.

The PredicativeRepresentation describes the link between Sense and Semantic Predicate; SemanticPredicate, instead, represents an abstract meaning together with its arguments, and may be used to represent the common meaning between different senses. SynSemArgMap is the class for the representation of the links between semantic arguments (of semantic predicates) and syntactic arguments (of syntactic behaviors).

LMF MRD Extension

LMF defines also an extension for the representation of monolingual and bilingual machine-readable dictionaries (MRD), which is meant to be used both by human translators and NLP systems. The MRD Package, which is based on the NLP Morphological Extension, requires at least one Sense class. Associated with the Sense class is the Equivalent class that is used in bilingual dictionary to represent the translation of the word form. Other classes that may optionally be associated with the Sense class are a Context class, representing the word form in context, and a Subject Field representing information about domain or status.

LMF NLP multilingual notations extension

The Multilingual notation extension is used to represent the sense or syntactic equivalences between two or more languages in a lexical (bilingual or multilingual) resource. The basic notion is that of Axes, that is used to link Sense and Syntactic Behavior instances belonging to different languages. The model consists of three main classes: the Sense Axis, the Transfer Axis and the Example Axis. The Sense Axis and Sense Axis Relation classes (implementing the interlingual pivot approach to automatic translation) are used to describe how lexemes are translated from own language to another, while the Transfer Axis and Transfer Axis Relation classes (implementing the transfer approach) are used to describe the multilingual mappings of syntactic behaviors. The Example Axis includes previously translated examples to be linked to the Lexical Entry.

Summing up, LMF can

- represent words in languages (or sublanguages) where multiple orthographies are possible,
- represent multiword expressions,
- represent specific syntactic behaviors,
- represent event types,
- allow for complex argument mappings between syntax and semantic descriptions,
- allow for a semantic organization based on SynSets (like in WordNet) or on semantic predicates (like in FrameNet), and
- represent bi or multi-lingual lexica

2.2.1. Content Descriptors/Tags: the Data Categories

In line with the principles of the ISO/TC 37/SC 4 committee, a complete lexical resource is thus organized in two levels: a high level constituted by a structural model, and a low level made of content descriptors.

LMF can be used to represent or formalize the structural organization of the lexicon (the model) and for this it provides objects and relations between them. Content descriptors in the ISO family of standards for language resources are the Data Categories (as defined in ISO 12620:2009); each lexical resource would define its appropriate Data Category Selection, possibly drawing data categories from the Data Category Registry (Ide and Romary, 2004; Wright, 2004) or mapping through them. CLARIN requires users to define their own DCS by using ISOCat (http://www.isocat.org).

An instantiated lexicon model is therefore represented by a number of lexical objects (or lexical classes), by relations among such classes, and by a set of data categories, i.e. attribute-value pairs, used to describe the individual instances of objects.

2.2.2. Sample Entries from different types of lexica

2.2.2.1. Samples from the Morphalou project

The following two entries represent articles from the Morphalou project. Figure 1 is using the LMF example serialization from the informative part of the standard, Figure 2 shows a TEI compliant serialization of the same article (taken from (Romary, 2010)). They are two versions of "the same" article, i.e., both entries contain the same information. For details, see (Romary et al., 2004). For Morphalou, see http://www.cnrtl.fr/lexiques/morphalou/.

```
<lexicalEntry xml:id="championne_1">
<feminineVariantOf target="#champion_1">champion</feminineVariantOf>
<formSet>
<lemmatizedForm>
<orthography>championne</orthography>
<grammaticalCategory>commonNoun</grammaticalCategory>
<grammaticalGender>feminine</grammaticalGender>
</lemmatizedForm>
<inflectedForm>
<orthography>championne</orthography>
<grammaticalNumber>singular</grammaticalNumber>
</inflectedForm>
<inflectedForm>
<orthography>championnes</orthography>
<grammaticalNumber>plural</grammaticalNumber>
</inflectedForm>
</formSet>
</lexicalEntry>
```

Figure 1: LMF compliant example

```
<entry>
<form type=”lemma”>
<orth>championne</orth>
<gramGrp>
<pos>commonNoun</pos>
<gen>feminine</gen>
<gramGrp>
</form>
<form type=”inflected”>
<orth>championne</orth>
<num>singular</num>
</form>
<form type=”inflected”>
<form type=”inflected”>
<orth>championnes</orth>
<num>plural</num>
```

```
</form>
</inflectedForm>
</entry>
```

Figure 2: TEI compliant example

2.2.2.2. Samples from the BioLexicon:

The BioLexicon is a comprehensive lexicon for biology and gene regulation created within an EU funded project (BOOTStrep). It includes terms from various domain terminological resources and morphological, syntactic and semantic information automatically derived from texts through text mining methods. For details on the content and format please refer to (Quochi et al., 2008; Quochi et al., 2009). Figure 3 shows a sample entry.

```
<LexicalResource>
<GlobalInformation >
<feat att="Authors" val="PISA-GROUP"></feat>
</GlobalInformation>
<Lexicon>
<feat att="name" val="Bio-lexicon"/>
<feat att="language" val="eng"/>

<!--Lexical entry starts: it carries PoS info and SouceId PoS is implemented as a special subType of
feat; SourceId belongs to the set of feat-->

<LexicalEntryID="LE_interleukin-2">
<feat Att="POS" Val="Noun"/>
<feat SourceAtt="Source" SourceVal="Q4U313a"></SOURCEDC>

<!--The morphological component. Lemma starts: it carries an attribute for preferred basename. The
object can be adorned with other info taken from the DatCat Repository -->

<Lemma ID="LM_interleukin-2" basename="interleukin-2" >
<feat att="GRAMMATICALNUMBER" val="Singular"/>
<feat att="GRAMMATICALGENDER" val="Feminine"/>

<!--FormRepresentation carries info concerning othographical variants -->
<FormRepresentation ID="RF_interleukin-2" writtenform="interleukin-2">
<feat Att="VariantType" Val="FullForm"/>
</FormRepresentation>
<FormRepresentation ID="RF_IL-2" writtenform="IL-2">
<featAtt="VariantType" Val="Orthographic"/>
</FormRepresentation>
</Lemma>

<SyntacticBehaviour id="SB_interleukin-2" subcategorizationFrames="N0">
</SyntacticBehaviour>
```

```
<Sense id="S_interleukin-2">
<SenseRelation targets="S_T-Cell_growth_factor">
<feat att="semanticrelation" val="synonym"/>
</SenseRelation>
</Sense>
</LexicalEntry>
<LexicalEntryID="LE_T-Cell_growth_factor">
<feat Att="POS" Val="Noun"/>
<feat Att="Source" Val="Q4U313b"/>
<!--#The morphological component. Lemma starts: it carries an attribute for preferred basename. The
object can be adorned with other info taken from the DatCat Repository -->

<Lemma ID="LM_T-Cell_growth_factor" basename="T-Cell_growth_factor" >
<feat att="GRAMMATICALNUMBER" val="Singular"/>
<feat att="GRAMMATICALGENDER" val="Feminine"/>

<!--FormRepresentation carries info concerning othographical variants -->
<FormRepresentation ID="RF_T-Cell_growth_factor" writtenform="T-Cell_growth_factor">
<feat att="RFDC1" val="RFDCVAL1"/>
<feat Att="VariantType" Val="FullForm"/>
</FormRepresentation>
<FormRepresentation ID="RF_TCGF" writtenform="TCGF">
<feat att="RFDC1" val="RFDCVAL1"/>
<feat Att="VariantType" Val="Achronym"/>
</FormRepresentation>
</Lemma>

<SyntacticBehaviour id="SB_T-Cell_growth_factor" subcategorizationFrames="N0">
</SyntacticBehaviour>
<Sense id="S_T-Cell_growth_factor">
</Sense>
</LexicalEntry>
<SubcategorizationFrame id="N0">
</SubcategorizationFrame>
</Lexicon>
</LexicalResource>
```

Figure 3: Sample entry from the BioLexicon

2.2.2.3. Example from the PAROLE/SIMPLE lexicon

PAROLE/SIMPLE was a project for the creation of syntactic and semantic lexica in 12 EU languages. SIMPLE was the lexical semantic extension on the PAROLE lexica and links all 12 lexicons via a base concepts ontology (drawn from the EuroWordnet Top Ontology).

These lexica were designed well before the birth of LMF, and partially contributed to its genesis.

The example reported in Figure 4 is taken from the Italian SIMPLE lexicon which has been extended and maintained after the end of the EU project and has been recently mapped and made exportable also according to the serialization in the informative appendix of the LMF standard.

```
<LexicalResource>
<GlobalInformation>
<feat att="label" val="SimpleLMFExport"/>
</GlobalInformation>
<Lexicon>
<LexicalEntry id="LE_leggere_1" morphologicalPatterns="GINP403">
<Lemma>
<feat att="pos" val="V"/>
<feat att="phoneticform" val="'lEgga"/>
<FormRepresentation>
<feat att="writtenform" val="leggere"/>
</FormRepresentation>
</Lemma>
<WordForm>
<feat att="morphofeat" val="S1CP"/>
<FormRepresentation>
<feat att="writtenform" val="legga"/>
<feat att="phoneticform" val="'lEgga"/>
</FormRepresentation>
</WordForm>
<WordForm>
<feat att="morphofeat" val="S2CP"/>
<FormRepresentation>
<feat att="writtenform" val="legga"/>
<feat att="phoneticform" val="'lEgga"/>
</FormRepresentation>
</WordForm>
…
<!—list of all forms of the lemma -->
<Sense id="USem68921leggere">
<PredicativeRepresentation correspondences="ISObivalent " predicate="PREDleggere#2">
<feat att="link" val="Master"/>
</PredicativeRepresentation>
<SenseRelation targets="USem64096decifrare">
<feat att="relation_type" val="Isa"/>
</SenseRelation>
</Sense>
<Sense id="USem68924leggere">
<PredicativeRepresentation correspondences="ISOtrivalent " predicate="PREDleggere#3">
<feat att="link" val="Master"/>
```

```
</PredicativeRepresentation>
<SenseRelation targets="USemD5576dire">
<feat att="relation_type" val="Isa"/>
</SenseRelation>
</Sense>
<Sense id="USem72261leggere">
<PredicativeRepresentation correspondences="ISObivalent " predicate="PREDleggere#4">
<feat att="link" val="Master"/>
</PredicativeRepresentation>
<SenseRelation targets="USemD5371agire">
<feat att="relation_type" val="Isa"/>
</SenseRelation>
</Sense>
<Sense id="USem73482leggere">
<PredicativeRepresentation correspondences="ISObivalent " predicate="PREDleggere#5">
<feat att="link" val="Master"/>
</PredicativeRepresentation>
<SenseRelation targets="USem72488interpretare">
<feat att="relation_type" val="Isa"/>
</SenseRelation>
</Sense>
<Sense id="USemD5495leggere">
<PredicativeRepresentation correspondences="ISOtrivalent " predicate="PREDleggere#1">
<feat att="link" val="Master"/>
</PredicativeRepresentation>
<SenseRelation targets="USem74769acquisire">
<feat att="relation_type" val="Isa"/>
</SenseRelation>
</Sense>
</LexicalEntry>

<MorphologicalPattern id="GINP403">
<TransformSet>
<Process>
<feat att="operator" val="remove"/>
<feat att="string" val="3"/>
</Process>
<Process>
<feat att="operator" val="add"/>
<feat att="string" val="ERE"/>
</Process>
<GrammaticalFeatures>
<feat att="morphofeat" val="F"/>
```

```
</GrammaticalFeatures>
</TransformSet>
<TransformSet>
<Process>
<feat att="operator" val="remove"/>
<feat att="string" val="3"/>
</Process>
<Process>
<feat att="operator" val="add"/>
<feat att="string" val="O"/>
</Process>
<GrammaticalFeatures>
<feat att="morphofeat" val="S1IP"/>
</GrammaticalFeatures>
</TransformSet>
<!—other transformation sets -->
</MorphologicalPattern>
<SemanticPredicate id="PREDleggere#2">
<SemanticArgument id="ARG0leggere#2">
<feat att="label" val="ARG0"/>
<feat att="semanticRole" val="Role_ProtoAgent"/>
<feat att="restriction" val="ArgAgentive"/>
</SemanticArgument>
<SemanticArgument id="ARG1leggere#2">
<feat att="label" val="ARG1"/>
<feat att="semanticRole" val="Role_ProtoPatient"/>
<feat att="restriction" val="Information"/>
</SemanticArgument>
</SemanticPredicate>
<SemanticPredicate id="PREDleggere#3">
<SemanticArgument id="ARG0leggere#3">
<feat att="label" val="ARG0"/>
<feat att="semanticRole" val="Role_ProtoAgent"/>
<feat att="restriction" val="ArgHuman"/>
</SemanticArgument>
<SemanticArgument id="ARG1leggere#3">
<feat att="label" val="ARG1"/>
<feat att="semanticRole" val="Role_ProtoPatient"/>
<feat att="restriction" val="Information"/>
</SemanticArgument>
<SemanticArgument id="ARG2leggere#3">
<feat att="label" val="ARG2"/>
<feat att="semanticRole" val="Role_2Participant"/>
```

```
<feat att="restriction" val="ArgHumanHumanGroup"/>
</SemanticArgument>
</SemanticPredicate>
<SemanticPredicate id="PREDleggere#4">
<SemanticArgument id="ARG0leggere#4">
<feat att="label" val="ARG0"/>
<feat att="semanticRole" val="Role_ProtoAgent"/>
<feat att="restriction" val="ArgHuman"/>
</SemanticArgument>
<SemanticArgument id="ARG1leggere#4">
<feat att="label" val="ARG1"/>
<feat att="semanticRole" val="Role_ProtoPatient"/>
<feat att="restriction" val="Semiotic_artifact"/>
</SemanticArgument>
</SemanticPredicate>
<SemanticPredicate id="PREDleggere#5">
<SemanticArgument id="ARG0leggere#5">
<feat att="label" val="ARG0"/>
<feat att="semanticRole" val="Role_ProtoAgent"/>
<feat att="restriction" val="ArgHuman"/>
</SemanticArgument>
<SemanticArgument id="ARG1leggere#5">
<feat att="label" val="ARG1"/>
<feat att="semanticRole" val="Role_ProtoPatient"/>
<feat att="restriction" val="Entity"/>
</SemanticArgument>
</SemanticPredicate>
<SemanticPredicate id="PREDleggere#1">
<SemanticArgument id="ARG0leggere#1">
<feat att="label" val="ARG0"/>
<feat att="semanticRole" val="Role_ProtoAgent"/>
<feat att="restriction" val="ArgHuman"/>
</SemanticArgument>
<SemanticArgument id="ARG1leggere#1">
<feat att="label" val="ARG1"/>
<feat att="semanticRole" val="Role_ProtoPatient"/>
<feat att="restriction" val="Information"/>
</SemanticArgument>
<SemanticArgument id="ARG2leggere#1">
<feat att="label" val="ARG2"/>
<feat att="semanticRole" val="Role_Underspecified"/>
<feat att="restriction" val="Semiotic_artifact"/>
</SemanticArgument>
```

```
</SemanticPredicate>
</Lexicon>
</LexicalResource>
```

Figure 4: Example from the PAROLE/SIMPLE Lexicon

2.2.2.4. Example from a MRD Entry

Figure 5 shows a representation according to the examples from the LMF informative appendix of the example entry "Ski" that makes use of the MRD extension (taken from (Lemnitzer et al., 2010)).

```
<LexicalEntry>
<Lemma id="l1">
<FormRepresentation>
<feat orthographyName="GermanVariantD"/>
<feat writtenForm="Ski"/>
</FormRepresentation>
<FormRepresentation>
<feat orthographyName="GermanVariantB"/>
<feat writtenForm="Schi"/>
</FormRepresentation>
</Lemma>
<Equivalent>
<feat lang="German"/>
<feat writtenForm="Schneeschuh"/>
</Equivalent>
<etymology>
<etymon id="l2">
<form>
<orth xml:lang="norwegian">ski</orth>
<pos>commonNoun</pos>
</form>
<sense>
<gloss>device for sliding on snow</gloss>
<note>aus anord. sk.. ‚Scheit, Schneeschuh'; s. das im Dt. etymologisch entsprechende Scheit.</note>
</sense>
</etymon>
<etymologicalLink source="l2" target="l1">
<etymologicalClass>loan word </etymologicalClass>
</etymologicalLink>
</etymology>
</LexicalEntry>
```

Figure 5: Use of the MRD extension

2.3. TEI

The Text Encoding Initiative (TEI; www.tei-c.org) is an international endeavor in the humanities to provide a set of reference guidelines for the representation of digital textual information, including (printed) dictionaries. Initiated in 1987 as a forum of major text archives worldwide, it has been an early adopter of SGML, and in turn XML, and has issued five editions of its guidelines so far, providing more than 500 elements for representing prose, poetry, drama, manuscript and of course, dictionary content. The TEI has evolved to become a real infrastructure for specifying and customizing textual formats, while allowing a quick entry into its technology (Romary, 2009).

It is based on a specification language (ODD – One Document Does it all), from which is generated, on the one hand, the full textual documentation, and on the other hand, the actual formal specification in one of several available schema languages (DTD, Relax NG, W3C).

One important mechanism available in the TEI infrastructure is a class management system allowing elements to be grouped together when they bear either a joint semantics or when they occur in similar structural contexts. Classes are essential for providing an abstract entry point (e.g. all elements providing grammatical information in a dictionary entry) for the specification of a given construct, onto which one can easily customize a specialist profile (e.g. for my own dictionary, I just need part of speech and gender).

The "print dictionary" chapter of the TEI guidelines has been designed, right from the beginning (Ide and Véronis, 1995) as a generic model that could, in turn, be customized to deal with the variety of form and structure that print dictionaries or born digital machine readable dictionaries may take. It has also been a compromise between providing a highly structured format for controlling dictionary content, and accounting for the many and varied permutations and combinations that surface forms can take, especially in older dictionaries. This is why all elements belonging to the TEI dictionary chapter may occur within two main constructs: An `<entryFree>` element, where each component is tagged independently from one another in the order they appear in the printed text and an `<entry>` element, which provides a highly structured organization of content, somehow closer to a database-like organization.

In the following, we will present the major characteristics of the TEI dictionary principles in more detail. First, the TEI is conformant to the semasiological view of lexical structures and actually maps rather precisely onto the LMF model, which has been introduced earlier in this report.

A typical entry in a semasiological dictionary starts with an account of the formal features of the lexical sign, which are shared by all its senses. In the case that the formal and/or etymological features of a linguistic sign vary even if the headword is the same, the TEI provides the `<hom>`-element to split an entry into two or more homographs. This view is reflected in the TEI model through the provision of two main constructs:

- A `<form>` element, grouping together all descriptive elements (phonetic, orthographic, grammatical) of a surface realization of the dictionary entry;
- A `<sense>` element, which organizes, both sequentially and hierarchically, the various meanings associated to the entry

This high-level distinction between information about the formal aspects of the described linguistic sign and the content or meaning aspects reflects usual practice when producing lexical descriptions in the form of a dictionary entry. The distinction is, however, not a strict one. In a sense description, a part of the form description can still be further specified or constrained. For example, a lexical unit

which can in general be used in the plural form might not be used in a particular sense as such and this information should be assigned to this individual sense, cf. the following example, taken from the "Wörterbuch der deutschen Gegenwartssprache" (Klappenbach and Steinitz, 1962-1977) (pieces which are not relevant are again left out):

```
<entry xml:id="E_s_957">
    <form type="lemma">
        <orth extent="full">Sand</orth>
        <gramGrp>
            <pos value="N"></pos>
            <gram rend="sep:semicolon" type="determiner">der</gram>
            <gen value="masculine"></gen>
            <gram rend="sep:comma" type="genitive">-(e)s</gram>
            <gram rend="sep:comma" type="plural">-e</gram>
            <usg type="plev">auch</usg>
            <gram type="plural">Sände</gram>
        </gramGrp>
    </form>
    <sense level="0" xml:id="S_s_528_1">
        <sense n="1." xml:id="S_s_528">
            <form>
                <gramGrp>
                    <gram type="singular-only">ohne Pl.</gram>
                </gramGrp>
            </form>
            <def>lockere und feinkörnige, meist vorwiegend aus Quarz bestehende Substanz des
Erdbodens</def>
             [...]
        </sense>
        <sense n="2." xml:id="S_s_529">
            <form>
                <gramGrp>
                    <gram rend="ldelim:slash rdelim:slash" type="plural-only">nur im Pl.</gram>
                </gramGrp>
            </form>
            <colloc>
                <quote>Sande</quote>
            </colloc>
            <def>Sandarten</def>
            <cit type="example">
                <quote>alluviale Sande</quote>
            </cit>
              [....]
        </sense>
        <sense n="3." xml:id="S_s_530">
```

```
            <form>
                <gramGrp>
                    <note rend="ldelim:slash">Pl.</note>
                    <gram rend="sep:comma" type="plural">Sande</gram>
                    <usg type="plev">auch</usg>
                    <gram rend="rdelim:slash" type="plural">Sände</gram>
                </gramGrp>
            </form>
            <def>Sandbank</def>
             [...]
            <cit type="quotation">
                <quote>Zwanzig Schritte weiter zerbrach das Wasser mit eifrigem Plätschern auf einem
Sand</quote>
                <ref cRef="LUSMARDEB" type="bibl">
                    <bibl>
                        <author>Luserke</author>
                        <title>Erzwungener Bruder</title>
                        <biblScope>16</biblScope>
                    </bibl>
                </ref>
            </cit>
              [...]
        </sense>
    </sense>
</entry>
```

Figure 6: Example from Wörterbuch der Deutschen Gegenwartssprache

Within each of the senses, the use is either constrained (to singular of plural only) and / or one of the alternative plural forms ('Sande', 'Sände') is selected.

The <form>-element can be used to group the following types of information: spelling of a word, including alternative spellings or other information related to the use of the word in written discourse, e.g., syllabification; pronunciation(s) of a word (<pron>-element) and other information related to the use of the word in spoken discourse, e.g., word stress; grammatical information, e.g., part of speech, gender, inflection, to mention only a few prominent examples. Each piece of information can be further qualified with a "usage"-marker (the <usg>-element). The scope of a description can thus be constrained to a certain regional variety or a certain diachronic stage of the described language. Grammatical features of the described linguistic sign, e.g. part of speech, inflections, subcategorization, can be grouped into a grammatical-group structure (the <gramGrp>-element). Iteration as well as nesting of the <gramGrp>-elements allows the dictionary writer to group features that are functionally related so that the change of one feature implies the change of the other feature. For example, take a look at the gender and inflection of the German lexical unit "Gischt" (engl. "spin drift"). The gender varies without a corresponding change in the meaning of the word. The inflection of the word depends on the gender, and it changes accordingly. For both inflectional paradigms there is a restriction on the use of the plural. This can be modeled, using the TEI specification, as follows:

```
<form type="headword">
<orth extent="full">Gischt</orth>
<gramGrp>
<pos value="N"/>
<gramGrp>
<gram type="determiner">der</gram>
<gram type="genitive">-es</gram>
<gram type="plural">-e</gram>
</gramGrp>
<usg type="plev">auch</usg>
<gramGrp>
<gram type="determiner">die</gram>
<gram type="genitive">-</gram>
<gram type="plural">-e</gram>
</gramGrp>
<gram type="singular-preferred">Pl. ungebräuchl.</gram>
</gramGrp>
</form>
```

Figure 7: Lexical unit "Gischt" (spin drift)

The <sense>-element assumes a similar function, i.e., grouping of elements that are related and together describe a particular sense (or reading) of the lexical unit. Prominent information types are definitions (<def>-element), citations (<cit>-elements) and usage examples, often taken from corpora (<example>-element).

Sample Entry from German dictionary

In the following we present an example from a German multi-volume general language dictionary. The source has been compiled between 1962 and 1977. The printed books have been digitized 2003. Afterwards, the merely typographic structure was transformed into a logical dictionary entry structure, using the TEI guidelines for the encoding of the information.

First, the entry for "Bahnhof" (railway station) copied from the printed version of the dictionary (with some minor parts left out):

> *Bahn- ...- hof, der, Halle, Gebäude am Halteplatz von Eisenbahnzügen: am B. sein; jmdn. Am B. erwarten, vom B. abholen, zum B. bringen; auf welchem B. kommt er an?; wie weit ist es bis zum B.?; der Zug rollte aus dem B.; im Gedränge des Bahnhofes; Neupräg. salopp ich verstehe immer nur B. (ich verstehe gar nichts); Neupräg. großer B. festlicher Empfang: der berühmte Gast wurde mit großem B. empfangen; von einem großen B. absehen*

(entry from the digitized version of (1962-1977), online at: www.dwds.de.)

This is the TEI conformant representation of the article:

```
<entry xml:id="E_b_437">
```

```
<form type="headword">
<orth extent="suffix">-hof</orth>
<gramGrp>
<pos value="N"/>
<gram type="determiner">der</gram>
</gramGrp>
</form>
<sense xml:id="S_b_234" level="0">
<def xml:id="N_b_140">Halle, Gebäude am Halteplatz von Eisenbahnzügen</def>
<cit type="example">
<quote>am B. sein</quote>
</cit>
<cit type="example">
<quote>jmdn. am B. erwarten, vom B. abholen, zum B. bringen</quote>
</cit>
<cit type="example">
<quote>auf welchem B. kommt er an?</quote>
</cit>
<cit type="example">
<quote>wie weit ist es bis zum B.?</quote>
</cit>
<cit type="example">
<quote>der Zug rollte aus dem B.</quote>
</cit>
<cit type="example">
<quote>im Gedränge des Bahnhofes</quote>
</cit>
<cit type="example">
<usg type="reg">salopp</usg>
<quote>ich verstehe immer nur B.</quote>
<quote type="paraphrase">ich verstehe gar nichts</quote>
</cit>
</sense>
</entry>
```

Figure 8: TEI conformant representation of lexical entry for "Bahnhof"

2.4. Terminological Markup Framework (TMF)

Author: Gerhard Budin

TMF, the Terminological Markup Framework, has been published as an international standard in 2003 as ISO 16642. It specifies a framework for representing data recorded in terminological data collections. This framework includes a meta-model and methods for describing specific terminological markup languages (TMLs) expressed in XML. ISO 16642 is designed to support the

development and use of computer applications for terminological data and the exchange of such data between different applications.

The core of TMF is a terminological meta-model. It is based on established methods and principles of terminology management. Terminology collections consist of terminological entries. One of the most important characteristics of a terminological entry - compared to a lexicographical entry - is its concept orientation. A terminological entry treats one concept in a given language and, in the case of multilingual terminological entries, one or more totally or partially equivalent concepts in (an)other language(s), whereas a lexicographical entry contains one lemma (the base form of a single lexical unit) and one or more definitions (representing different meanings) in one or more languages. The TMF meta-model gives guidelines for designing terminological data collections and their terminological entries.

A terminological data collection comprises global information about the collection and a number of entries.

Each entry performs three functions, it describes one concept, or two or more totally or partially equivalent concepts, in one or more languages, it lists the terms that designate the concept(s), and it describes the terms themselves.

Each entry can have multiple language sections, and each language section can have multiple terminological units. Each data element in an entry can be associated with various kinds of descriptive and administrative information. In addition, there are various other resources that are not part of any one entry, but that can be linked to one or more entries. Such resources include bibliographic references, descriptions of ontologies, and binary data such as images that illustrate concepts.

By instantiating the generic architecture, the terminological meta-model is described through seven instances from the structural node class, with the following core structure:

- TDC (terminological data collection): this is a top level container for all information contained in a terminological data collection.
- GI (global information): Information that applies to all elements represented in a file, as opposed to information that may pertain to some but not to all components of the file. GI usually contains, the title of the (XML) file, the institution or individual originating the file, address information, copyright information, update information, etc.
- TE (terminological entry): Information that pertains to a single concept. TE usually contains descriptive information pertinent to a concept, and administrative information concerning the concept. It can contain one or more language sections depending on whether the termbase is monolingual, bilingual, or multilingual.
- CI (complementary information): CI usually contains textual bibliographical or administrative information residing in or external to the file, static or dynamic graphic images, video, audio, etc. CI can also include references to other terminological resources or contextual links to related text corpora. These items are often designated as shared resources because they are available to all points in a termbase and are not repeated for different entries.
- LS (language section): The language section contains all the term sections for a terminological entry that are used in a given language. It usually contains definitions, contexts, etc. associated with that language or the terms in that language.

- TS (term section): The term section contains information about terms and contains a single term used to designate the concept that is the subject of the terminological entry, as well as any other information (e.g. definitions, contexts, etc.), associated with that term.
- TCS (term component section): The term component section contains information about morphemic elements, words, or contiguous strings from which a poly-morphemic (or multiword) term is formed.

Figure 9 visualizes the structural form of the TMF meta-model:

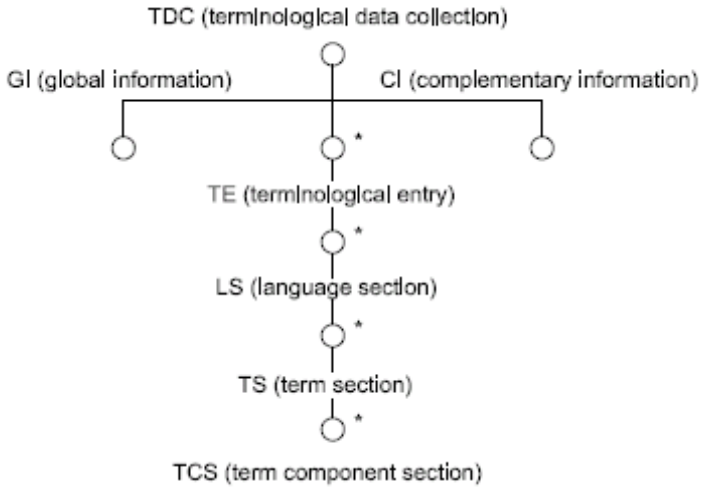


Figure 9: Structure of TMF meta-model

- A TE can contain any number of LSs (0..*).
- An LS can contain any number of TSs (0..*).
- A TS can contain any number of TCSs (0..*).
- A TDC must contain exactly one GI (1..1), at most one CI-Level (0..1) and any number of TEs (0..*).

Hierarchical organization is ensured by the 1..1 limitations expressed for the dual cardinalities for each relation.
On the basis of the meta-model, representations of terminological data are generated according to a specific terminological markup language (TML) that conforms to the framework, i.e. TMF. ISO 16642 specifies the conformance conditions that a TML must satisfy.

A terminological markup language (TML) uses the meta-model as the structural skeleton, data category specifications (DCS) for defining the semantics of the meta-data of the terminology collection and other details governing the data. In order to use ISO 16642, another international standard is necessary: ISO 12620 that specifies data categories registry (DCR). On the basis of the DCR, a specific selection of data categories is prepared as a DCS to be used in a terminology markup language (TML). The dialectal specification (Dialect) includes the various elements needed to describe a given TML as an XML document. These elements comprise expansion trees and data category instantiation styles, together with their corresponding vocabularies.

The combination of the meta-model and a given DCS is enough to define conditions of interoperability, encompassing the full informational properties of the TML from a terminological point of view. Any information structure that corresponds to such conditions has a canonical

expression as an XML document using the GMT representation. The interoperability between two different TMLs depends solely on their compatibility at that level, as Figure 10 shows.

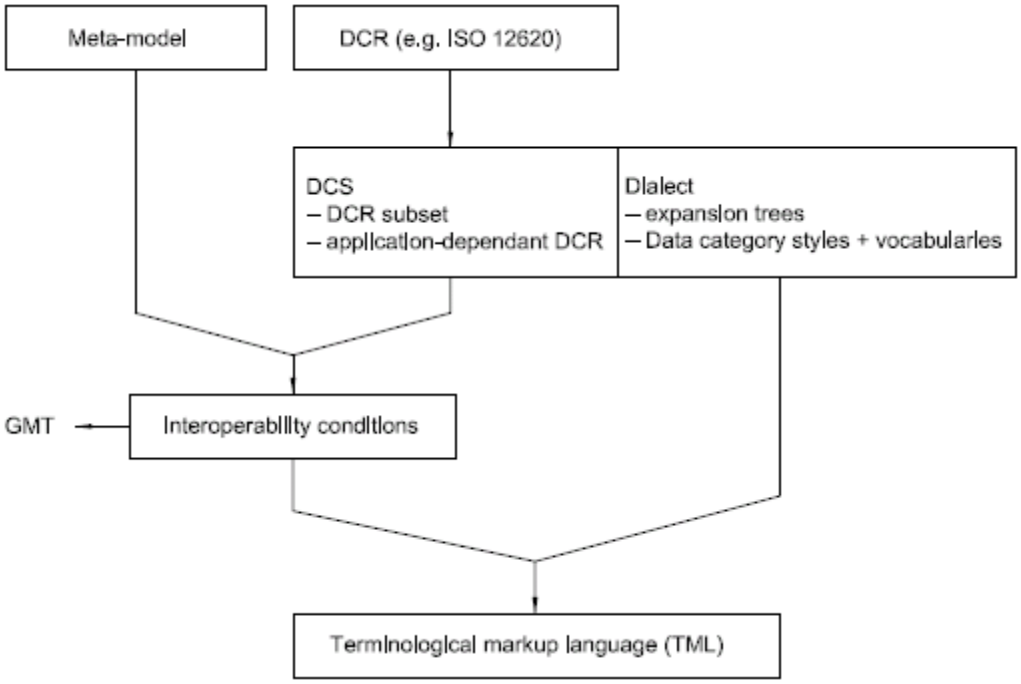


Figure 10: Interoperability

ISO 16642 contains several types of representation forms of TMF, including a UML model and XML specifications. The SALT project, where TMF was developed, also generated some utilities that facilitate the application of the standard in real-world scenarios. TBX (a new ISO standard as well as a LISA open standard) and Geneter are most widely known TMLs according to ISO 16642.

2.5. Widespread Lexicon Formats

2.5.1. Wordnet Structure and Formats

Authors: Verena Henrich, Maciej Piasecki

A wordnet means different things to different people, there is no consensus on its nature but a wordnet is a network involving words and groups of words constructed along the lines defined the WordNet (Fellbaum, 1998), called here the Princeton WordNet (henceforth PWN). PWN began as a psychological experiment that aimed to explain how lexical meaning is stored in the mind. Recently, it is treated as a language resource, and is perceived as a lexicalized ontology, (lexical) semantic network, or thesaurus – to name the most popular perspectives. A wordnet describes lexical meanings and contrary to a traditional lexicon is organized around lexical meanings themselves not words. The basic structure of a wordnet consists of *synsets* (short for: synonymy set) and binary relations defined over synsets. A synset is a set of (near) synonyms and according to PWN (Fellbaum, 1998), a synset represents a *lexicalized concept*. Synsets are linked by instances of semantic relations like: hypernymy/hyponymy, meronymy / holonymy and different forms of entailment relation in the case of verbs. Relations linking synsets are called *conceptual relations*

(Fellbaum, 1998), however, they owe their names due to lexico-semantic relations well established in lexicography.

Besides synset relations, we can also find relations defined over synset members – words or more formally, in some wordnets, *lexical units*, e.g. (Piasecki et al., 2009). Relations linking lexical units are called *lexical relations* and are a subset of lexico-semantic relations identified in lexicography and lexical semantics, e.g. antonymy and different kinds of derivational relations.

The significant differences between wordnets and semantic lexicons resulted in specific formats assumed for the latter.

2.5.1.1. Native Format of the Princeton WordNet

The original format of Princeton WordNet was influenced by perspective of WordNet as a kind of thesaurus (Beckwith et al., 1993) in which sets of synonyms, i.e., synsets, are the basic building blocks. Two tasks were identified as basic for the technical environment supporting wordnet development:

- writing the source files, called *lexicographer files*, including the description of the wordnet structure, and
- generation of the display format for the user.

In relation to the WordNet format the first task was primary, and the second is derivative. Source files are written in a formal language defined for this task. As this language and the file format have been utilized in many projects and had influence on many applications due to free distribution of WordNet, we will describe this format more closely in the rest of this section.

The basic building blocks of the source files are "word forms" and "word meanings". The former are represented in the orthographic form, the latter are represented by synsets. Synsets are arranged into lexicographers' source files: one for adverbs and several for adjectives, nouns and verbs. The division into source files is not a part of the main WordNet semantic structure, but, nevertheless, is semantically motivated. This kind of the two level division of synsets, i.e. into Part of Speech sub-databases and semantic domains (in WordNet corresponding to source files) can be observed in several wordnets, e.g. in plWordNet (Piasecki et al., 2009).

A source description of a synset includes:

- a set of synonymous word forms,
- *relational* pointers, and
- other information (e.g. textual gloss).

Relational pointer is a formal representation of an instance of: relation between synsets or a relation between word forms, which are called lexical relations.

Word form is described by: orthographic form, syntactic category, semantic field and the sense number. Orthographic form together with the sense number define a unique key which identifies the given word form in wordnet. Thus word form understood in this way corresponds to a lexical unit of a dictionary.

Lexical relations were defined in WordNet 1.x exclusively between words of different syntactic categories.

A wordnet relation can be reflexive, but this information is not directly encoded in the source code. Instead the appropriate reflexive pointers are automatically generated for the needs of searching by users and visualising WordNet.

Relation pointers to particular word forms can be entered in square brackets, i.e., […], which follow a given word form.

Verb synsets contain a list of *verb frames* each. A verb frame can be restricted to a particular word of a synset by describing it in the in square brackets following the form.

As a summary, the description of the syntax of a synsets as is defined in (Beckwith et al., 1993) is given below:

[1] Each synset begins with a left curly bracket ({).

[2] Each synset is terminated with a right curly bracket (}).

[3] Each synset contains a list of one or more word forms, each followed by a comma.

[4] To code semantic relations, the list of word forms is followed by a list of relational pointers using the following syntax: a word form (optionally preceded by *"filename***:***"* to indicate a word form in a different lexicographer file) followed by a comma, followed by a relational pointer symbol.

[5] For verb synsets, *"***frames:***"* is followed by a comma-separated list of applicable verb frames. The verb frames follow all relational pointers.

[6] To code lexical relations, a word form is followed by a list of elements from [4] and/or [5] inside square brackets (**[...]**).

[7] To code adjective clusters, each part of a cluster (a head synset, optionally followed by satellite synsets) is separated from other parts of a cluster by a line containing only hyphens. Each entire cluster is enclosed in square brackets.

2.5.1.2. Czech WordNet and DEB VisDic Format

VisDic (Horák and Smrž, 2004) and its direct descendant DEB VisDic (Horák et al., 2006) are wordnet editing and management tools that have been developed primarily for Czech WordNet, but next were applied in several wordnet and wordnet-related projects, including BalkaNet (Tufiş et al., 2004) and KYOTO project (Vossen et al., 2008).

DEB VisDic stores wordnet structure in a set of database files but also in XML export/import file. An example of the synset definition in the VisDic format is given in Figure 11.

```
<SYNSET><ID>ENG20-02853224-n</ID><POS>n</POS>
<SYNONYM>
<LITERAL sense="1">car</LITERAL><WORD>car</WORD>
<LITERAL sense="1">auto</LITERAL><WORD>auto</WORD>
<LITERAL sense="1">automobile</LITERAL><WORD>automobile</WORD>
<LITERAL sense="4">machine</LITERAL><WORD>machine</WORD>
<LITERAL sense="1">motorcar</LITERAL><WORD>motorcar</WORD>
</SYNONYM>
<ILR type="hypernym">ENG20-03649150-n</ILR><ILR type="eng_derivative">ENG20-01874890-v</ILR>
<DEF>4-wheeled motor vehicle; usually propelled by an internal combustion engine</DEF>
<USAGE>he needs a car to get to work</USAGE>
<BCS>1</BCS><DOMAIN>tourism</DOMAIN>
<SUMO type="+">TransportationDevice</SUMO>
</SYNSET>
```

Figure 11: Example of a synset represented in the VisDic XML format.

The synset representation in the VisDic XML format is organized into 4 levels, where the 0 level is the top level of synsets. The following levels are:

- 1 ID – identifier
- 1 POS – Part of Speech
- 1 SYNONYM – list of synonyms:
  - 2 LITERAL – word form ()
    - 3 SENSE – sens number
    - 3 LNOTE – sense description
- 1 ILR – lexical relation
  - 2 TYPE relation type
- 1 RILR – synset relation
- 1 BCS – pointer to basic koncept
- 1 DEF – definition
- 1 USAGE – note on the Osage
- 1 SNOTE
- 1 STAMP

A similar organization of elements of the representation in the EuroWordNet format (Vossen, 2002) is also used for encoding the fact that the levels with the higher numbers are optional.

2.5.1.3. plWordNet 1.0

plWordNet (Polish name *Słowosieć* is a wordnet of Polish whose first version 1.0 was built in the years 2005-2009 (Piasecki et al., 2009). Presently, the much larger version 2.0 is under construction and will be released by the end of 2012. From the very beginning, it was assumed that plWordNet would be constructed in a semi-automatic way with the intensive support of the tools suggesting new lexical units and instances of lexico-semantic relations. Moreover, the plWordNet linguistic team was distributed across different localizations and a network system supporting wordnet editing was necessary. Thus a database system was developed, plWordNet was stored in the central database and all editing was performed via specialized editors – database system clients.

A XML-based format for plWordNet was developed mainly for the needs of archiving. Moreover the core part of the format was defined during the very first phase of the plWordNet 1.0 construction during which some linguists were working without a permanent access to Internet. Parts of the wordnet structure were created locally, stored in the first version of the XML format and next merged into one central version. Thus the developed format has a rather technical character and reflects the structure of the database tables. However, all data are kept in one file in order to facilitate easy transmission and file management by linguists.

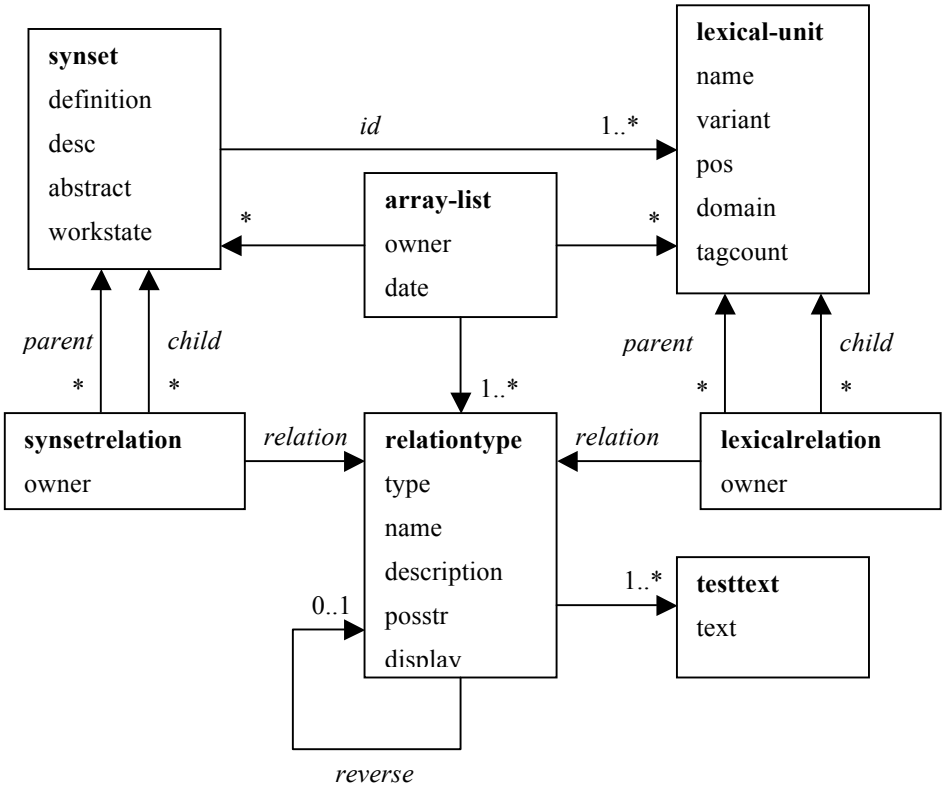


Figure 12: Scheme of the XML-based format of plWordNet 1.0

The scheme of the plWordNet XML file structure is presented in Figure 12. All elements of the plWordNet structure are grouped together under a technical grouping XML element called `array-list`. The `array-list` attributes describe the owner of the archive, date of creation and the version of the application with which the given archive was created. As the format was evolving, the application version determines the version of the format used to store the data.

The format includes also description of lexico-semantic relations used in the given version of a wordnet. Each relation is described by the `relationtype` XML element. In `relationtype` the attributes express:

- `type` – type of the relation, theoretically the number of different types is unlimited, however, in plWordNet
  - only two types are in use: synset relation and lexical unit relation1,
- `name`, `display` and `shortcut` – different forms of the relation name,
- `description` – a descriptive definition of the relation written in the natural language and presented on demand to the operators of the wordnet editing application,
- `posstr` – Parts of Speech for which the given relation is defined,
- `autoreverse` – is set to true if for each instance of the given relation an instance
  - of the *reverse* relation is automatically created – the reverse relation is defined by the value
  - of the *reverse* XML link (technically, an attribute with the identifier as a value.)

Each `relationtype` XML element has its unique identifier used in the optional `reverse` XML link pointing to a relation which is an opposite relation to the given one. Each relation is associated with at least one substitution test supporting the identification of relation instances, i.e. pairs of lexical units associated by the given relation. Substitution tests are stored as natural language expressions (`text` attribute) including variables marking places of substitution. Tests are instantiated with the appropriate words by the application.

Relations and substitution test are defined around lexical units as lexical units are the basic building elements of the plWordNet structure, see (Derwojedowa et al., 2008) and (Piasecki et al., 2009). All synset relations originate from the linguistic lexico-semantic relations and a synset is defined as a kind of abbreviation for a group of lexical units sharing the same lexico-semantic relations.

Lexical units are described by the `lexical-unit` XML elements in which:

- `name` – stores lemma for the given lexical unit,
- `variant` – the consecutive number of a sense represented by the given lemma – name plus variant form the unique name of the given lexical unit,
- `pos` – Part of Speech,
- `domain` – the name of a semantic domain to which the given lexical unit belong – the division into semantic domains has a technical character in plWordNet and follows the division into lexicographer files introduced in Princeton WordNet, only minor modifications were introduced, cf. (Piasecki et al., 2009),
- `tagcount` – frequency of the lemma in the corpora from which data were extracted during the process of plWordNet semi-automated construction – this information can be treated as a part of lexical unit meta-data,
- `desc` – description of the meaning or purpose of the given lexical unit, also a place for comments, it was intended as a part of the meta-data,
- `source` – meta-data describing the source of origin of the given lexical unit, e.g. a list defined by coordinators on the basis of large corpora or a lexical unit added by linguists during wordnet editing.

A synset is represented by an XML element grouping lexical units linked by their ids. The synset description includes also:

- `definition` – a place for a gloss, very rarely used in the present version of plWordNet,
- `desc` – stores examples plus comments, more for the future than present use in plWordNet,
- `abstract` – marks abstract synsets representing meanings non-lexicalized but represented in the network for the needs of the readability of the network structure, see (Piasecki et al., 2009),
- `workstate` – stores a code describing the state of editing process in relation to the given synset, e.g. `added but not verified yet' – a part of synset meta-data,
- `owner` – the name of a linguist who introduced the last modification to the given synset – a part of meta-data.

As plWordNet is being constructed by a relatively large team of linguists over longer period of time, the need for various kinds of meta-data is continuously increasing. In order to make an archive self-

dependent – a complete snapshot of plWordNet – all this meta-information must be present in the XML format.

Relations are technically divided into relations linking synsets (`synsetrelation`) and lexical units (`lexicalrelation`), however, there is no difference between them on the conceptual level. Both kinds of relation originate from particular linguistic relations. If a synset relation links two synsets it means that the given lexico-semantic relation can be attributed to all pairs of lexical units created from lexical units belonging to the respective synsets. From the definition, each relation is a set of ordered pairs, where the order of elements is represented by the links: `parent` and `child`[1].

Each relation instance is described by meta-data representing the linguist who created the given link.

Summing up, there are three characteristic elements of the plWordNet XML-based format:

- tendency to include all information characterizing the wordnet structure into one file, including the meta-description of the semantic relations,
- a central role played by lexical units and relations between lexical units, and
- and rich meta-data describing different aspects of the wordnet creation stored in the file in order to form a complete description of a wordnet as a evolving object.

2.5.1.4. GermaNet Structure and GermaNet XML Format

GermaNet is a German lexical semantic network that is modeled after the Princeton WordNet for English. As mentioned before, it partitions the lexical space into a set of concepts that are interlinked by semantic relations. A semantic concept is modeled by a synset. A synset is a set of words (called lexical units) where all the words are taken to have (almost) the same meaning. Thus a synset is a set-representation of the semantic relation of synonymy, which means that it consists of a list of lexical units and a paraphrase (represented as a string). The lexical units in turn have frames (which specify the syntactic valence of the lexical unit) and examples. The list of lexical units for a synset is never empty, but any of the other properties may be.

There are two types of semantic relations in GermaNet: conceptual and lexical relations. Conceptual relations hold between two semantic concepts, i.e. synsets. They include relations such as hyperonymy, part-whole relations, entailment, or causation. Lexical relations hold between two individual lexical units. Antonymy, a pair of opposites, is an example of a lexical relation.

GermaNet covers the three word categories of adjectives, nouns, and verbs, each of which is hierarchically structured in terms of the hyperonymy relation of synsets.

GermaNet development started in 1997, and is still ongoing. The current version 5.3 of GermaNet was released in April 2010.

Traditionally, lexicographic work for extending the coverage of GermaNet utilized the Princeton WordNet development environment of lexicographer files (see section 2.5.1.1). These lexicographer files are specified in plain text and contain synsets for a particular semantic domain and part of speech. To get an impression of what these GermaNet lexicographer files looked like see section 3 in (Henrich and Hinrichs, 2010).

The development with these lexicographer files was very time consuming and error-prone. For these reasons, the GermaNet data was converted to a relational database format in 2009. The working

[1] Due to the database construction process the names are taken from the domain of programming.

development copy of all GermaNet data is now stored in a relational database and no longer in the lexicographer files. The database model follows the internal structure of GermaNet. This means that there are tables to store synsets, lexical units, conceptual and lexical relations, etc. The complete database structure for GermaNet is described in detail in (Henrich and Hinrichs, 2010).

The GermaNet Editing Tool GernEdiT (Henrich and Hinrichs, 2010) supports various export functionalities. For example, it is possible to export all GermaNet contents into XML files, which are used as an exchange format of GermaNet.

The structure of the XML files closely follows the internal structure of GermaNet, which means that the file structure mirrors the underlying relational organization of the data. There are two DTDs that jointly describe the XML-encoded GermaNet. One DTD represents all synsets with their lexical units and their attributes. The other DTD represents all relations, both conceptual and lexical relations.

The GermaNet XML format was initially developed by (Kunze and Lemnitzer, 2002; Kunze and Lemnitzer, 2002; Lemnitzer and Kunze, 2002), but modifications of the GermaNet data itself led to an adopted XML format, which is presented here.

XML Synset Files

The XML files that represent all synsets and lexical units of GermaNet are organized around the three word categories currently included in GermaNet: nouns, adjectives, and verbs (altogether 54 synset files since the semantic space for each word category is divided into a number of semantic subfields).

The structure of each of these files is illustrated in Figure 13[2]. Each synset represents a set of lexical units (*lexUnits*) which all express the same meaning. This grouping represents the semantic relation of synonymy. Further properties of a synset (e.g., the word category or a describing paraphrase) and a lexical unit (e.g., a sense number or the orthographical form (*orthForm*)) are encoded appropriately.

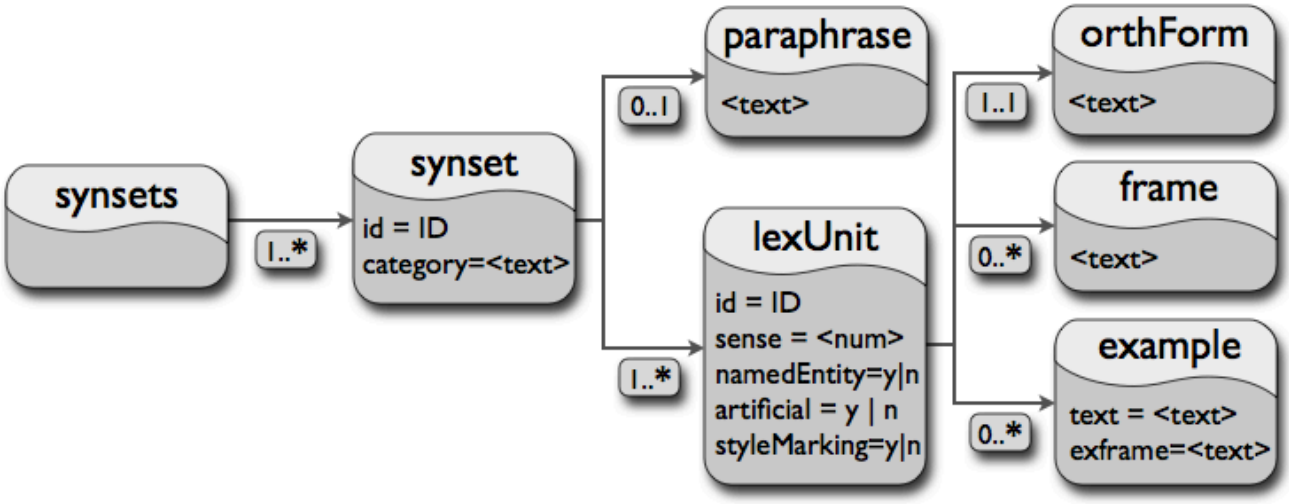


Figure 13: Structure of the XML synset files

[2] This figure is not complete for the reason of simplicity.

Figure 13 describes the underlying XML structure. Each box in the figure stands for an element in the XML files, and the properties in each box (listed underneath the wavy line) represent the attributes of an XML element. This means, for example, that a *synset* element has the attributes of an *id* and a *category*.[3]

Figure 14 shows an example of a *synset* with two lexical units (*lexUnit* elements) and a *paraphrase*. The *lexUnit* elements in turn contain several attributes and an orthographical form (the *orthForm* element), e.g., *leuchten* (German verb for: *to shine*). The first of the two lexical units even has a *frame* and an *example*.

```
<synset id="s58377" category="verben">
  <lexUnit id="l82207" sense="1" namedEntity="no" artificial="no" styleMarking="no">
    <orthForm>leuchten</orthForm>
    <frame>NN</frame>
    <example>
      <text>Der Mond leuchtete in der Nacht. </text>
      <exframe>NN</exframe>
    </example>
  </lexUnit>
  <lexUnit id="l82208" sense="2" namedEntity="no" artificial="no" styleMarking="no">
    <orthForm>strahlen</orthForm>
  </lexUnit>
  <paraphrase>Lichtstrahlen aussenden, große Helligkeit verbreiten</paraphrase>
</synset>
```

Figure 14: Synset file example

XML Relation File

This type of XML file represents both kinds of relations: conceptual and lexical relations. All relations are encoded within one XML file, whose structure is illustrated in Figure 16. The boxes in Figure 15 again represent XML elements, which means that there is one *relations* element that contains all lexical relations (*lex_rel* elements) and conceptual relations (*con_rel* elements). Both relation types contain several attributes.

[3] Note that in this section, XML element or attribute names appear *italic* if they are referenced in the text.

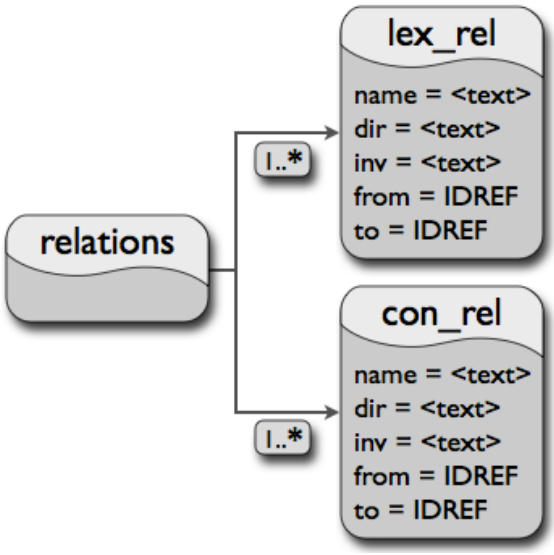


Figure 15: Structure of the XML relation file

Figure 16 illustrates an example for each of the two relation types. The type of the conceptual relation is hyperonymy (indicated by the *name* attribute), and it holds between the synset with ID s58377 (*from* attribute) and the synset with ID s58376 (*to* attribute). The lexical relation is of type antonymy (again indicated by the name attribute), and holds between the lexical units with the IDs l2471 (*from* attribute) and l12470 (*to* attribute).

```
<con_rel name="hyperonymy" from="s58377" to="s58376" dir="revert" inv="hyponymy" />
<lex_rel name="antonymy" from="l2471" to="l2470" dir="both" />
```

Figure 16: Example from relation file.

2.5.2. Existing Wordnet-LMF Formats

The Lexical Markup Framework (ISO 24613:2008), is an ISO standard for encoding natural language processing lexicons and machine readable dictionaries (Francopoulo et al., 2006). The intention of LMF is to provide a common model for the creation and use of lexical resources, to manage the exchange of data between and among these resources, and to enable the merging of a large number of individual electronic resources to form extensive global electronic resources.

The core structure of LMF is based on the prototypical structuring of a lexicon in terms of lexical entries, each of which enumerates the different senses of the lexical item in question. This word-driven perspective contrasts the synset-driven relational structure of wordnets – the grouping of word senses (i.e., lexical units) that express the same meaning into synsets. Exactly these two radically different organizing principles (relation-based in the case of wordnets versus lexical-entry-based in the case of LMF) constitute the challenge of encoding wordnets in LMF. We take up this challenge in this document and show how synset-based wordnets, e.g. plWordNet or GermaNet, can be represented in a word-driven format like LMF.

The conversion of plWordNet and GermaNet to LMF build on Wordnet-LMF (Lee et al., 2009; Soria et al., 2009), an existing Lexical Markup Framework dialect described in the following subsection.

2.5.2.1. KYOTO LMF Wordnet Format

Wordnet-LMF has been developed in the context of the EU KYOTO project[4] and is especially tailored to encode wordnets in the LMF standard. Wordnet-LMF is specified by a Document Type Definition (see Appendix E in (Soria and Monachini, 2008)) and fully complies with standard LMF.

The Wordnet-LMF XML structure is shown in Figure 17[5]. There is a *Lexical Resource* which contains at least one *Lexicon* (in this case a wordnet lexicon).[6] A *Lexical* Entry represents a word entry in a *Lexicon*, where the word itself is represented by the *writtenForm* attribute of the *Lemma* element. *Lexical Entries* group different *Senses* of a particular word. The *Senses* have a *synset* attribute that relates them to a *Synset* element by the corresponding ID. If two *Senses* have the same *synset* attribute, they belong to the same *Synset* and are thus synonyms. A *Synset* can have several relations to other *Synsets*. These relations are encoded in *SynsetRelation* elements.

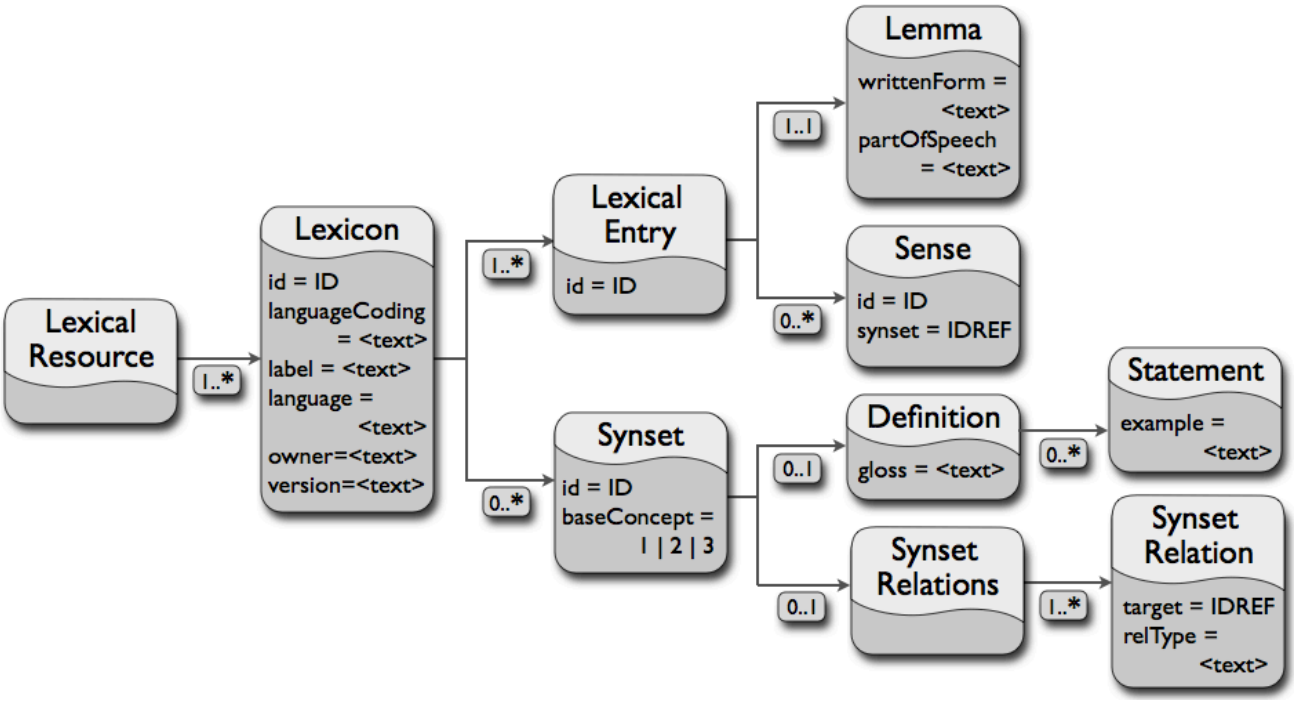


Figure 17: The Wordnet-LMF structure (Henrich and Hinrichs, 2010)

2.5.2.2. plWordNet LMF Representation

The primary format of plWordNet is the XML-based format described in section 2.5.1.3. However, this specific native format was considered to be a possible obstacle for the interoperability in the future CLARIN network. Thus, the plWordNet team decided to use a more general format as the basic format for the constructed plWordNet Web Service.

---

[4] See http://www.kyoto-project.eu

[5] Note that this figure does not show the whole Wordnet-LMF model. Only the monolingual part that is relevant for this paper is represented. The representation of multi-lingual resources (i.e., the optional *SenseAxis* element with its children) is not considered in this paper. For a complete picture, see (Soria and Monachini, 2008).

[6] Here, XML element or attribute names again appear *italic* if they are referenced in the text.

Due the LMF status as an ISO standard the use of an LMF compliant format was assumed as the basic rule. In order to avoid construction of a yet another variant of the LMF implementation the already existing Wordnet-LMF was chosen as a basis. As Wordnet-LMF was proposed in the KYOTO project applying wordnets for several languages, the initial intention of the plWordNet team was to use the KYOTO proposal in its exact shape. However, it very quickly became clear that KYOTO Wordnet-LMF is too selective in relation to the information represented in plWordNet. Thus, the information returned by the plWordNet WS includes only those parts of plWordNet structure that can be encoded in the KYOTO Wordnet-LMF. The missing elements include:

- all information related to the relations linking lexical units,
- meta-description of relations,
- and meta-information associated with lexical units, synsets and relation instances.

Yet another problem are elements required in the KYOTO Wordnet-LMF, but not present in the plWordNet structure, i.e. monolingual external links of the two subtypes. plWordNet does not yet include such mapping in the present version.

```
<?xml version="1.0"?>
<!DOCTYPE LexicalResource SYSTEM
 "http://www2.let.vu.nl/twiki/pub/Kyoto/LexicalResourceRepresentation/kyoto_wn.dtd">
<LexicalResource>
  <GlobalInformation label="Search result for a written form `biały`, plWordNet 1.0"  />
  <Lexicon languageCoding="ISO 639-3"
           label="Polish Wordnet 1.0" language="pol"
           owner="Wrocław University of Technology"  version="1.0">
    <LexicalEntry id="biały">
      <Lemma writtenForm="biały" partOfSpeech="a"/>
      <Sense id="biały_1" synset="pol-10-771-a"/>
      <Sense id="biały_2" synset="pol-10-772-a"/>
    </LexicalEntry>
    <Synset id="pol-10-771-a" baseConcept="1">
      <Meta author="Tomasz.Stępień"/>
      <SynsetRelations>
        <SynsetRelation target="artificial" relType="artificial"/>
      </SynsetRelations>
      <MonolingualExternalRefs>
        <MonolingualExternalRef externalSystem="artificial"
                                externalReference="artificial"/>
      </MonolingualExternalRefs>
    </Synset>
    <Synset id="pol-10-772-a" baseConcept="1">
      <Meta author="Magdalena.Zawisławska"/>
      <SynsetRelations>
        <SynsetRelation target="artificial" relType="artificial"/>
      </SynsetRelations>
```

```
      <MonolingualExternalRefs>
        <MonolingualExternalRef externalSystem="artificial"
                                externalReference="artificial"/>
      </MonolingualExternalRefs>
    </Synset>
  </Lexicon>
</LexicalResource>
```

Figure 18: Example of the LMF-based description returned by plWordNet web service for *biały* (Polish for: *white*)

2.5.2.3. Representing GermaNet in LMF

The differences between the synset-driven structure of GermaNet (see Figure 13 and Figure 15) and the word-driven format of Wordnet-LMF (see Figure 17) are obvious. But there is also a strong commonality: Both formats have synset elements that cluster synonymous words. In GermaNet, the words are represented by lexical units that are child elements of a synset. In Wordnet-LMF, senses, which correspond to the lexical units in GermaNet, are linked to a synset (by an attribute containing a synset ID) (Henrich and Hinrichs, 2010).

The conversion of GermaNet to Wordnet-LMF proceeds as follows: Each lexical unit of GermaNet is turned into a *Sense* element in Wordnet-LMF (see Figure 17). The *synset* attribute (containing a synset ID) of the *Sense* element links this *Sense* with the *Synset* that it is a member of. The different *Sense* elements are grouped by their orthographical form (the *Lemma* in Wordnet-LMF) into *Lexical Entries*.

An example of a GermaNet *LexicalEntry* in Wordnet-LMF is shown in Figure 19. This *LexicalEntry* represents the word *leuchten* (German verb for: *to shine*), as the written-Form attribute of the *Lemma* element indicates. This *LexicalEntry* has two *Senses*, which belong to different *Synsets* (see the different *synset* attributes of the *Sense* elements).

Each *Sense* has a *MonolingualExternalRefs* element with at least one *MonolingualExternalRef* representing a reference to an external system. In this case, each *Sense* is linked to the corresponding entry in the GermaNet database; the *externalReference* attribute of a *MonolingualExternalRef* specifies the database table name with a database ID.

```
<LexicalEntry id="deu-52-l4601-v">
  <Lemma writtenForm="leuchten" partOfSpeech="v" />
  <Sense id="deu-52-l4601-v_1" synset="deu-52-s58377-v">
    <MonolingualExternalRefs>
      <MonolingualExternalRef externalSystem="GermaNet-Database"
        externalReference="lex_uni_table#id=82207" />
    </MonolingualExternalRefs>
  </Sense>
  <Sense id="deu-52-l4601-v_2" synset="deu-52-s58718-v">
    <MonolingualExternalRefs>
      <MonolingualExternalRef externalSystem="GermaNet-Database"
        externalReference="lex_uni_table#id=82677" />
    </MonolingualExternalRefs>
  </Sense>
```

```
</LexicalEntry>
```

Figure 19: Example of a *LexicalEntry*

In the next conversion step, all synsets of GermaNet are listed with their relations to other synsets. The corresponding *Synset* (with the ID *deu-52-s58377-v*) of the first Sense in Figure 19 is illustrated in Figure Figure 20. It has, inter alia, a describing gloss and two example sentences.

The element *SynsetRelations* encodes relations to other *Synset* instances. The relations are simply encoded with a *target* attribute that contains the ID of the referencing *Synset*. The *Synsets* in Wordnet-LMF are logically the "same" as the synsets in GermaNet XML, i.e. the concept that a synset expresses is exactly the same in both formats.

Each *Synset* has a reference to the GermaNet database. Therefore, the *MonolingualExternalRef* element links to the corresponding entry in the GermaNet database; the *externalReference* attribute specifies the database table name with the synsets database ID.

```
<Synset id="deu-52-s58377-v" baseConcept="1">
  <Definition gloss="Lichtstrahlen aussenden, große Helligkeit verbreiten">
    <Statement example="Der Mond leuchtete in der Nacht."/>
    <Statement example="Die Lichter der Stadt strahlen in die Nacht."/>
  </Definition>
  <SynsetRelations>
    <SynsetRelation target="deu-52-s58376-v" relType="has_hyperonym"/>
  </SynsetRelations>
  <MonolingualExternalRefs>
    <MonolingualExternalRef externalSystem="GermaNet-Database"
        externalReference="synset_table#id=58377"/>
  </MonolingualExternalRefs>
</Synset>
```

Figure 20: Example of a *Synset*

These two figures – Figure 19 and Figure 20 – represent the same example in Wordnet-LMF that was already shown in the GermaNet XML format in Figure 13.

2.5.3. Suggested Wordnet-LMF Format

As the previous discussion has shown, Wordnet-LMF provides a very useful basis for converting plWordNet and GermaNet into LMF. However, a number of modifications to Wordnet-LMF are needed if this conversion is to preserve all information present in the original resource. The present section will discuss a number of modifications to Wordnet-LMF that are needed for conversion of wordnets in general. In addition, we will also discuss a set of extensions to Wordnet-LMF that are needed for conversion of plWordNet and GermaNet in particular.

The most glaring omission in Wordnet-LMF concerns the modeling of lexical relations which hold between lexical units (i.e., *Senses* in the terminology of Wordnet-LMF). In the current Wordnet-LMF DTD only conceptual relations (i.e., *SynsetRelations* in the terminology of Wordnet-LMF), which hold between synsets, are modeled. Thus antonymy, which is a typical example of a lexical relation (see (Fellbaum, 1998)) for further details), can currently not be modeled without violating the Wordnet-LMF DTD.

Among the synset relations specified in Wordnet-LMF, the entailment relation is missing, which plays a crucial role in the modeling of verbs in the Princeton WordNet and in GermaNet alike. The list of values of attribute *relType* for *SynsetRelation* elements (see Appendix A in (Soria and Monachini, 2008)) therefore has to be amended accordingly.

Wordnet relations are not fixed across wordnets, e.g. among Slavic wordnets different semantic relations motivated derivationally have been proposed recently, e.g. (Pala and Hlaváčková, 2007; Koeva, 2008). To some extent the problem of evolving set of wordnet relations is related to the synset relations, too. Any fixed list of relation types can be a limitation for the LMF-based representation. Inclusion of the specification of relations inside the LMF-based wordnet representation, e.g. in a similar way as it is done in the native format of the plWordNet 1.0 could be a possible solution to this problem

A third omission in the current Wordnet-LMF DTD concerns syntactic frames used in the Princeton WordNet to indicate the syntactic valence of a given word sense. Syntactic frames are also used in GermaNet, albeit using a different encoding[7]. Syntactic frames together with example sentences, which illustrate the meaning and prototypical usage of a particular word, help to distinguish among word senses.

In WordNet both syntactic frames and examples are linked to synsets. However, at least in the case of syntactic frames the linkage to synsets seems problematic since different members of the same synset may well have different valence frames. For example, the German verbs *beeindrucken* and *imponieren* both mean *to impress* and thus belong to the same synset. Both are transitive verbs, but their object NPs have different cases: accusative case for *beeindrucken* and dative case for *imponieren*. As this example shows, syntactic frames need to be associated with lexical units rather than synsets. This is exactly the design choice made in GermaNet, as shown in Figure 15.

A related question concerns the anchoring of example sentences which illustrate the meanings and prototypical usage of a particular word sense. In both the Princeton WordNet and GermaNet such examples are associated with synsets[8]. GermaNet correlates examples additionally with particular syntactic frames and treats both examples and syntactic frames as properties of lexical units, i.e. Senses in the terminology of Wordnet-LMF.

The above issues lead to a modified version of the Wordnet-LMF DTD as shown in Figure 21. Compared to Figure 17, the *Sense* element is enriched by three optional subelements: *SenseRelations*, *Examples*, and *Frames*. The *SenseRelation* elements represent relations between different *Senses* (the lexical units in GermaNet).

The *Examples* and *Frames* elements in Figure 21 both group several *Example* or *Frame* instances. A *Frame* element represents the syntactic valence of a word sense. An *Example* shows the prototypical usage of a word sense as an example sentence. The syntactic valence for a concrete example sentence can be specified with the optional frame attribute of an *Example*.

[7] In WordNet, frames are encoded in a controlled language using paraphrases such as Somebody ----s something for a transitive verb with an animate subject and an inanimate object. The frames in GermaNet use complementation codes provided with the German version of the CELEX Lexical Database (Baayen et al., 2005) such as NN.AN for transitive verbs with accusative objects.

[8] In WordNet, the examples are placed at the synset level, but referencing to a word sense at the same time.

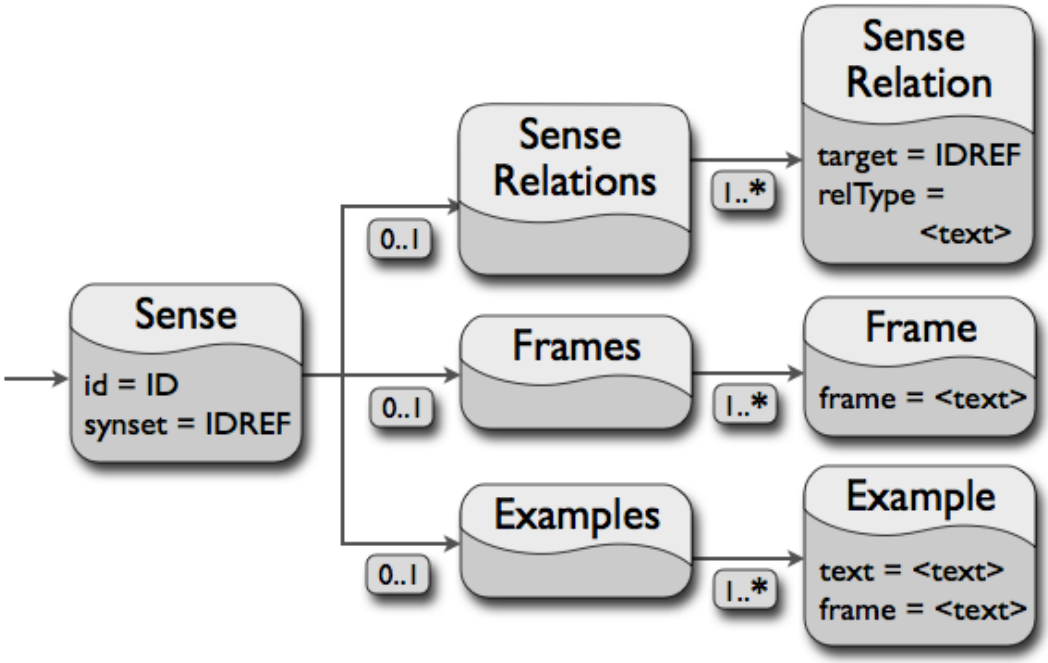


Figure 21: Revised Wordnet-LMF structure (Henrich and Hinrichs, 2010).

Wordnets are mostly dynamic objects that are developed during years. It is necessary to maintain mapping of the newer versions to the older, as well, as to keep meta-data of different types facilitating the process of wordnet development and maintenance. A situation in which one data format serves all needs is preferable to the use of several formats and mappings among them. If we agree to the idea of one-basic format, we should consider enrichment of LMF-based wordnet format with the optional meta-data attached to different elements of the wordnet structure.

3. Ontologies

Authors: Kiril Simov and Petya Osenova

An *ontology* defines the conceptual knowledge for modeling of a domain of discourse. The conceptual knowledge is represented as a set of concept terms organized in a hierarchy related by relations and constrained by a set of axioms. The term ontology has its origin in the field of philosophy where it denotes the science of existence. In the last decades, it became popular also in the field of information science. The ontology was established in the area of knowledge representation as a level between the logical level underlining the knowledge representation languages and the actual vocabulary of predicates for modeling of the world or a part of it – the domain of interest. At the ontology level the knowledge primitives are interpreted as satisfying the restrictions of a formal ontology which distinguishes: (1) among the entities of the world (physical objects, events, processes...); and (2) among the meta-level categories used to model the world (concepts, properties, states, roles, attributes, various kinds of part-of relations...) (Guarino, 1994). Definition of what an ontology is in the area of the information science is given by many authors. The most popular definition (Gruber, 1993; Gruber, 1995) is:

> "An ontology is an explicit specification of a formal conceptualization."

where a conceptualization consists of Universe of Discourse (a set of objects to be modeled) and Relational Structure over it. An ontology in the sense of this definition specifies the necessary concepts, functions, relations to express the knowledge about the universe of discourse (or in other words – the domain). Although there are critics of this definition, it is widely accepted. The main goals of an ontology are (Noy and McGuiness, 2001):

- To share common understanding of the structure of information among people or software agents;
- To enable reuse of domain knowledge;
- To make domain assumptions explicit;
- To separate domain knowledge from the operational knowledge;
- To analyze domain knowledge.

The ontologies became a key ingredient of Semantic Web initiative where it is supposed they to represent a shared knowledge in machine readable form. It is expected that the real Semantic Web will be populated by a large number of cooperatively created ontologies. This expectation reflects the fact that the world can be modeled in many different ways and that there is no a single authority able to specify all the knowledge. On the basis of this many different ontologies already existing and there are many ways in which these ontologies are represented.

Usually the ontologies are classified on the basis of their generality with respect to the knowledge represented in them. On this level there are two types of ontologies: *Upper Ontology* and *Domain Ontology*. Upper (or Top-level) ontologies define general notions like space, time, objects, qualities. Thus upper ontologies contain concepts common for all domains. Domain ontologies represent specific concepts and properties for a given area of interest. Such a domain can be characterized by specific concepts and properties, but also and specific restriction of the interaction of these concepts and properties. Each domain ontology explicitly or implicitly is connected to an upper ontology. The upper ontology constraint the knowledge represented in the domain ontology by inheritance. The

upper ontology provides also a way to connect the domain ontology to other domain ontologies related to the same upper one.

Another classification of ontologies distinguishes between *Object Ontology*, *Task Ontology* and *Application Ontology*. The object ontology is an ontology of things existing in the world (domain). The task ontology is an ontology of tasks to be performed within the domain. The application ontology is an ontology necessary to support the functionality of an application.

Each ontology is represented in a formal language comprising:

- *Vocabulary* – reflecting the relations in the conceptualization of the world;
- *Formal theory* – constraining the interpretation of the vocabulary with respect to the conceptualization;
- *Inference* – rules for explication of implicit information.

Natural language represents the human conceptualization of the world. Thus, often it is accepted that lexical knowledge can be considered as ontology. The obvious missing part of such assumption is the formal language with its inference mechanism. Putting lexicons and ontologies together (Guarino, 2000) represents the following classification of ontologies on the basis of their precision and complexity:

- *Lexicon* (Machine Readable Dictionaries) – Vocabulary with NL definitions
- *Simple Taxonomy* (Classifications) – Thesaurus, e.g., WordNet
- *Taxonomy plus related-terms* – Lexical Relations
- *Relational Model – Light-weight ontologies*
- *Unconstrained use of arbitrary relations* – Fully Axiomatized Theory / *Heavy-weight ontologies*

Usually, the first three items of this classification are called *linguistic ontologies*. The lightweight and heavyweight ontologies are the ontologies within Semantic Web world. They are represented formally and provide basis for inference. Some linguistic ontologies (WordNet) are formalized in an ontology language and used as lightweight ontologies. It is assumed that linguistic ontologies are language dependent as much as the different natural languages reflect different conceptualization of their speakers. The non-linguistic ontologies are assumed to be language independent. From application point of view it is necessary to have both kinds of ontologies interacting. The formal ontologies are more machine oriented and the linguistic ontologies are human oriented. The ontologies of the first type are used for extracting of new knowledge via inference mechanisms. The ontologies of the second type are used to present the knowledge to the users in an appropriate form. Defining an appropriate interaction of the language and the world knowledge is the basis for successful applications.

We can see two main roles of ontologies in the world of language resources:

- Concept repository for formal language semantics;
- Modelling language for linguistic annotation.

In the first case ontologies are considered as a repository of conceptual part of the meaning of the natural language (usually, the lexical semantic). The ontology represents formal concepts and properties that are necessary to define the meaning of a given lexical unit or to compose the meaning of a language expression. A recent overview of the relation between ontologies and lexicons is presented in (Hirst, 2004). The paper discusses the structure of lexical entries, the knowledge

recorded in them and mechanisms for interrelation of the lexicon elements. Special attention is given to the definition of 'word sense', its conceptual structure, relations between senses and problematic cases. The main topics under discussion are near-synonyms, gaps in the lexicon, and linguistic categorizations that are not ontological.

In the second case the area of linguistic annotation is considered as domain on its own and thus it can be modeled within an appropriate ontology. The modeling of the annotation as RDF graphs provides a natural way to present different relations between the language material (text, speech, etc) and the annotations or among the annotations themselves. The standard languages for representation of ontologies provide mechanisms for querying the annotated corpora, to infer new knowledge or to find contradictions within the annotations. An example of this type of usage of ontologies is (Burchardt et al., 2008).

Further information on ontologies can be found on the following websites:

- Semantic Web Page (http://semanticweb.org/wiki/Ontology);
- Laboratory for Applied Ontology (http://www.loa-cnr.it/index.html);
- Buffalo Ontology Site (http://ontology.buffalo.edu/);
- John Bateman's ontology portal (http://www.fb10.uni-bremen.de/anglistik/langpro/webspace/jb/info-pages/ontology/ontology-root.htm).

In this section we first present some basic terminology in the area of ontology modeling, then some concrete ontologies and ontological languages.

3.1. Terminology/Definitions

This section contains short definition of the basic notions used in the area of ontology in information science. The sources of these definitions includes (Gruber, 1993; Guarino, 1998; Suárez-Figueroa et al., 2007) and others.

Concept

Class

> A concept (or a class) in an ontology is an expression – name or term – that models a distinct set of objects in a domain. Each concept in an ontology is defined by its relations to other concepts in the ontology and the set of axioms. The concept can be a *defined* or *primitive* depending on the fact whether its definition specifies sufficient and necessary conditions one object to be an instance of the concept or just sufficient ones. Class is used in some ontological languages to denote concepts in a domain.

Inference

> A formal method for explication of new knowledge (not explicitly defined) from an ontology (or knowledge base).

Instance data

> A set of instance description with respect to an ontology. For example, descriptions of all models of Volkswagen cars with respect to an ontology of automobiles. In the terms of KL-ONE-like languages a set of instance data is called *A-Box*.

Instance

Individual

An instance in an ontology is an expression that models a given object in the intended domain of the ontology. Individual is used as a synonym of instance in some ontology languages.

Knowledge Base

An ontology a set of individuals (instances) are called knowledge base.

Linked Open Data (LOD)

LOD is a W3C SWEO community project, which aims to facilitate the emergence of a web of linked data, by means of publishing and interlinking open data on the Web in RDF.

OntoClean

A methodology for ontology-driven conceptual analysis[9]. Defines an inventory of (meta-)properties for distinguishing of different types of concepts.

Ontology

An explicit specification of a formal conceptualization.

A set of concepts, relations and axioms defining a theory about a domain.

Ontology Commitment

Partial semantic account of the intended domain for an ontology (intended domain can be expressed as a formal conceptualization or it can be implicit).

Ontology Alignment

Ontology Mapping

The activity of finding the correspondences between two or more ontologies and storing/exploiting them. The term refers sometimes to the result of the activity.

Ontology Semantics

A semantic theory based on ontology and constructing knowledge content of utterances of a natural language. (Nirenburg and Raskin., 2004)

[9] http://www.ontoclean.org/

Ontology Learning

A knowledge acquisition activity that relies on (semi-) automatic methods to transform unstructured (e.g. corpora), semi-structured (e.g. folksonomies, html pages, etc.) and structured data sources (e.g. data bases) into conceptual structures.

Ontology Population

A knowledge acquisition activity that relies on (semi-) automatic methods to transform unstructured (e.g. corpora), semi-structured (e.g. folksonomies, html pages, etc.) and structured data sources (e.g. data bases) into instance data (e.g. A-Box).

Ontology Matching

An activity of finding or discovering relationships or correspondences between entities of different ontologies or ontology modules.

Relation

Property

A relation in an ontology is an expression that models relationship between instances of concepts. Usually in ontology languages binary relations are used. Property is a synonym of binary relation within some ontology languages.

3.2. Ontology

3.2.1. Upper Ontologies

3.2.1.1. SIMPLE Core Ontology

SIMPLE Core Ontology (Lenci et al., 2000) is created within SIMPLE Project for creation of semantic dictionaries for 12 European languages. The role of the SIMPLE Core ontology is to cluster the meaning of the words in the dictionaries. The meaning is represented via means of lexical templates, where each template is connected to a concept in the ontology and the fields of the template corresponds to the restrictions over the concepts expressed in the ontology. The lexical model of SIMPLE project extends the Generative Lexicon (GL) model. The SIMPLE project developed further the GL model by constructing an upper ontology that reflects the four qualia roles of GL model – Formal, Constitutive, Telic, and Agentive. The SIMPLE Core Ontology can be seen as a set of four interconnected ontologies: One ontology consisting of a hierarchy of concepts, additionally connected to other concepts via relations and functions. This sub-ontology corresponds to Formal qualia role. The other three ontologies consist of hierarchies of relations. Each of the relational ontologies corresponds to one of the qualia roles. Additionally, the concept ontology is extended with complex types which consist of two (or more) concepts which represent colocalized entities (see also DOLCE). The role of the complex types is to represent the regular polysemy in the lexicon. The ontology is represented as taxonomy of concepts and flat lists of relations. The SIMPLE Ontology contains 193 concepts and about 60 relations.

3.2.1.2. DOLCE Ontology[10]

DOLCE (a Descriptive Ontology for Linguistic and Cognitive Engineering) is the first module of the WonderWeb Foundational Ontologies Library. DOLCE is an axiomatic ontology (thus, it is a heavy-weight ontology, in contrast to light-weight ontologies, which mainly define taxonomic structures). DOLCE is an ontology of particulars. A basic choice made in DOLCE is the so-called multiplicative approach: different entities can be co-located in the same space-time. The ontology is developed with respect to the OntoClean methodology (Guarino and Welty, 2002). OntoClean is a methodology for validating the ontological adequacy of taxonomic relationships. DOLCE is axiomatized in a very expressive modal logic, but from it several simplified sub-ontologies are isolated. Such sub-ontologies are: DOLCE-LITE which includes 80 classes, 80 properties, 24 axioms represented in OWL; DOLCE-Lite-Plus which contains additional extensions in direction to Plans, Information Objects, Semiotics, Temporal relations, Social notions, etc. DOLCE UltraLite is a very light version of DOLCE, which uses friendly names and comments for classes and properties, has simple restrictions for classes. All these ontologies can be seen as lightweight versions of the fully axiomitized ontology.

3.2.1.3. OntoWordNet[11]

OntoWordNet is the result of an application of the OntoClean methodology to lexical knowledge bases and usage of DOLCE. The project aims at:

- aligning the upper-level of WordNet to DOLCE, in order to obtain an "ontologically sweetened" lexical resource, meant to be conceptually more rigorous, cognitively transparent, and efficiently exploitable in several applications;
- reengineering WordNet lexicon as a formal ontology;
- checking the consistency of the overall result, and correcting the cases for inconsistency;
- modularizing the resulting ontology according to an initial proposal of 'domains';
- learning and revising formal domain relations (either from glosses or from corpora).

An update of the WordNet 1.6 noun synsets linking to DOLCE-Lite-Plus is available. The materials are available at the same site as DOLCE ontology.

3.2.1.4. SUMO Ontology[12]

The Suggested Upper Merged Ontology (SUMO) is a free, formal upper ontology with many associated tools, translations, explanatory documents and domain ontologies. SUMO and all domain ontologies consist of 20,000 terms and 60,000 axioms. These consist of SUMO itself, the MId-Level Ontology (MILO), and ontologies of Communications, Countries and Regions, Distributed computing, Economy, Finance, Engineering components, Geography, Government, Military, North American Industrial Classification System, People, Physical elements, Transnational Issues, Transportation, Viruses, World Airports A-K, World Airports L-Z. All terms are formally defined. Meanings are not dependent on a particular inference implementation. An inference and ontology

[10] http://www.loa-cnr.it/DOLCE.html

[11] http://wiki.loa-cnr.it/index.php/LoaWiki:Ontologies

[12] http://www.ontologyportal.org/

management system however is provided. An additional system that supports visual editing, and does a better job of displaying the ontologies, especially in non-Western languages is the KSMSA system. SUMO is the only formal ontology (according to the website) that has been mapped to the entire WordNet lexicon. SUMO is written in the SUO-KIF language.

3.2.1.5. OpenCyc Ontology[13]

OpenCyc is the open source version of the Cyc technology. Release 1.0 of OpenCyc will include:

- 6,000 concepts: an upper ontology for all of human consensus reality.
- 60,000 assertions about the 6,000 concepts, interrelating them, constraining them, in e®ect (partially) defining them.
- A compiled version of the Cyc Inference Engine and the Cyc Knowledge Base Browser.
- A suite of tools for rapidly extracting knowledge from a domain expert, such as a physician or an oil-drilling specialist.
- Documentation and self-paced learning materials to help users achieve a basic- to intermediate-level understanding of the issues of knowledge representation and application development using Cyc.
- A specification of CycL, the language in which Cyc (and hence OpenCyc) is written. There are CycL-to-Lisp, CycL-to-C, etc. translators.
- A specification of the Cyc API, by calling which a programmer can build an OpenCyc application with very little familiarity with CycL or with the OpenCyc KB.
- The ability to import and export CycML files. A few sample programs that demonstrate use of the Cyc API for application development.

3.2.1.6. Basic Formal Ontology[14]

Basic Formal Ontology (BFO) is the third module of the WonderWeb Ontology Library. BFO is a highest-common-denominator upper ontology designed to support interoperability between 5 domain ontologies supporting shared use of scientific research data across disciplinary boundaries. Basic Formal Ontology consists in a series of sub-ontologies (most properly conceived as a series of perspectives on reality). It is claimed by the authors that the ontology is very small.

3.2.1.7. PROTON Ontology[15]

PROTON (PROTo ONtology) is a basic upper ontology. It contains about 300 classes and 100 properties, providing coverage of the general concepts necessary for a wide range of tasks, including semantic annotation, indexing, and retrieval of documents. The design principles can be summarized as follows (i) domain-independence; (ii) light-weight logical definitions; (iii) alignment with popular standards; (iv) good coverage of named entities and concrete domains (i.e. people, organizations, locations, numbers, dates, and addresses). The ontology is originally encoded in a fragment of OWL Lite and split into four modules: System, Top, Upper, and KM (Knowledge Management).

---

[13] http://www.cyc.com/cyc/opencyc/overview

[14] http://ontology.buffalo.edu/bfo/BFO.htm

[15] http://proton.semanticweb.org/

3.2.1.8. UMBEL (Upper Mapping and Binding Exchange Layer)[16]

UMBEL is based on OpenCyc represented in RDF format using OWL and SKOS. UMBEL provides two valuable functions: (1) it provides a vocabulary for the construction of concept-based domain ontologies, designed to act as references for the linking and mapping of external content, and (2) it is its own broad, general reference structure of 21,000 concepts, which provides a scaffolding to orient other datasets and domain vocabularies.

3.2.2. Domain Ontologies

There are thousands of domain ontologies with different size and different level of formalization. Together with some of the upper ontologies listed above there are also domain ontologies which can be used as models for creation of new domain ontologies. Here we give some repository of domain ontologies that can be searched for a particular ontology.

3.2.2.1. DAML Ontology Library[17]

DAML stands for DARPA Agent Markup Language Program. It officially began in August 2000. The goal of the DAML effort is to develop a language and tools to facilitate the concept of the Semantic Web. Within the DAML program an ontology library was created. It contains 282 ontologies. Most of them are toy ontologies containing up to 20 concepts. There are several ontologies with several thousand concepts.

3.2.2.2. SWOOGLE: Semantic Web Search[18]

A search engine for ontologies. Currently the search is performed over 10 000 ontologies.

3.2.2.3. WATSON: Exploring the Semantic Web[19]

Another search engine for ontologies based on key words.

3.2.2.4. Protégé-OWL Ontologies[20]

A small number of OWL ontologies (including also some upper-level ontologies) which are used to demonstrate some of the features of the Protégé ontology development framework.

3.2.2.5. Omega Ontology[21]

Omega Ontology: Omega is a 120,000-node terminological ontology constructed at USC ISI as the reorganization and synthesis of WordNet (versions 2.0 and 2.1), which is a lexically oriented network constructed on general cognitive principles; and Mikrokosmos, a conceptual resource originally conceived to support translation, into a new upper model, created expressly in order to

[16] http://umbel.org/

[17] http://www.daml.org/ontologies/

[18] http://swoogle.umbc.edu/

[19] http://kmi-web05.open.ac.uk/WatsonWUI/

[20] http://protege.stanford.edu/plugins/owl/owl-library/

[21] http://omega.isi.edu/

facilitate the merging of lower models into a functional whole. Omega, like its close predecessor SENSUS, can be characterized as a "shallow", lexically oriented, term taxonomy. By far the majority of its concepts can be stated in English by a single word. Omega contains no formal concept definitions and only relatively few interconnections (semantic relations) between concepts. By making few commitments to either specific theories of semantics or particular representations, Omega enjoys a malleability that has allowed it to be used in a wide variety of applications, from translation to question answering to information integration. Omega ontology is an example of domain conceptual knowledge aligned to an upper ontology.

3.2.2.6. GOLD Ontology (General Ontology for Linguistic Description)[22]

GOLD is an ontology for descriptive linguistics. It was developed initially within E-MELD project[23]. GOLD ontology gives a formalized account of the most basic categories and relations (the "atoms") used in the scientific description of human language. GOLD is intended to capture the knowledge of a well-trained linguist, and can thus be viewed as an attempt to codify the general knowledge of the field. It will facilitate automated reasoning over linguistic data and help establish the basic concepts through which intelligent search can be carried out. Furthermore, GOLD is meant to be compatible with the general goals of the Semantic Web.

3.3. Ontology Languages

The ontology management and usage systems have to support the following services:

- *Navigation* – these services have to support traversing of explicit information represented in the ontology;
- *Consistency* – these services have to support the check for contradiction within a repository. At least the consistency of the whole ontology and the consistency of a class description with respect to an ontology have to be supported;
- *Classification* – this service has to support for a class **C** the finding of the minimal classes in the ontology that are more general than **C** and the maximal classes in the ontology that are more specific than **C**.
- *Realization* – this service has to support for an instance **I** the finding of the minimal classes in the ontology that describe **I**.
- *Instance Checking* – this service has to support for an instance **I** and a class **C** the checking whether **I** is an instance of **C**.
- *Retrieval* – this service has to support for a class **C** the finding of all of its instances in the ontology.
- *Rules* – this service has to support the creation of user-defined rules. User-defined rules are a very powerful mechanism for changing the ontology. There will be a mechanism for defining the scope of such rules to one or few repositories.
- *Mapping* – this service provides mechanisms to align two or more ontologies.

[22] http://linguistics-ontology.org/

[23] http://emeld.org/index.cfm

For the implementation of these services within different ontology management and usage system a large set of ontology related languages were defined and standardized. These languages cover different aspects of ontology services and can be classified in the following way: (1) definitions of ontologies, (2) rules over ontological knowledge, (3) quering ontology repositories of instance data; and (4) mapping between ontologies. Here we represent a short list of languages of these categories.

3.3.1. Languages for definition of ontologies

3.3.1.1. RDF (Resource Description Framework)[24]

RDF is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata model but which has come to be used as a general method of modeling information, describing resources on the web, through a variety of syntax formats.

The RDF metadata model is based upon the idea of making statements about resources in the form of subject-predicate-object expressions, called triples in RDF terminology. The subject denotes the resource, and the predicate expresses relation between the subject and the object, both the predicate and the object represent resources also. There are named and anonymous resources. The first ones are identified by a URI (Uniform Resource Identifier) and can be accessed directly using it and the second kind of resources, called blank nodes, carry system names and can't be directly accessed.

3.3.1.2. RDF(S) – (RDF Schema/ RDF Vocabulary Description Language)[25]

RDF Schema is a semantic extension of RDF. RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute-value pairs. RDF properties also represent relationships between resources.

RDF however, provides no mechanisms for describing these properties, nor does it provide any mechanisms for describing the relationships between these properties and other resources. That is the role of the RDF vocabulary description language, RDF Schema. RDF Schema supplies formal syntax for classes and properties definition that may be used to describe specification of different types (groups) of resources (classes, properties, individuals of classes), with semantics for generalization hierarchies of such properties and classes.

3.3.1.3. FLORA-2[26]

FLORA-2 is an advanced object-oriented knowledge base language and application development environment. The language of FLORA-2 is a dialect of F-logic (Frame logic) with numerous extensions, including meta-programming in the style of HiLog and logical updates in the style of

---

[24] http://www.w3.org/RDF/

http://en.wikipedia.org/wiki/Resource_Description_Framework

http://www.w3schools.com/rdf/default.asp

[25] http://www.w3.org/RDF/

http://en.wikipedia.org/wiki/Resource_Description_Framework

http://www.w3schools.com/rdf/default.asp

[26] http://flora.sourceforge.net/florahome.php

Transaction Logic. FLORA-2 was designed with extensibility and flexibility in mind, and it provides strong support for modular software design through its unique feature of dynamic modules.

Applications of FLORA-2 include intelligent agents, Semantic Web, ontology management, integration of information, and others.

Future plans for FLORA-2 include standardization of the syntax, which will bring greater compatibility with other F-logic based languages, such as OntoBroker, WSML (Web Service Modeling Language), and SWSL-Rules.

3.3.1.4. DAML+OIL[27]

On the Semantic Web, the DARPA agent markup language (DAML) aims to enable the next generation of the web — a web that moves from simply displaying content to one that actually understands the meaning of the content. The DAML program has generated the DAML+OIL markup language. The submission of the DAML+OIL language to the World Wide Web consortium captures the work done by DAML contractors and the EU/U.S. Joint Committee on Markup Languages. This submission was the starting point for the language to be developed by W3C's web ontology working group, WebOnt. DAML+OIL is a syntax, layered on RDF and XML, that can be used to describe sets of facts making up an ontology. DAML+OIL and its friend OIL (ontology integration language) use RDF namespaces to organize and assist with integration of arbitrarily many different and incompatible ontologies. Current research into DAML is leading toward the expression of ontologies and rules for reasoning and action. Much of the work in DAML has now been incorporated into OWL.

3.3.1.5. OWL (Web Ontology Language)[28]

OWL adds more vocabulary for describing properties and classes, possible to express by using RDF and RDF(S) syntax like complex classes and additional relations between classes – disjointness of classes, equivalence, difference, union, intersection, anonymous (complex restrictions) classes, cardinality restrictions, equality, richer typing of properties, characteristics of properties (symmetry, transitivity, functional and inverse properties), and enumerated classes.

OWL provides three increasingly expressive sublanguages designed for use by specific communities of implementers and users.

OWL-Lite

> OWL Lite supports those users primarily needing a classification hierarchy and simple constraint features. For example, while OWL Lite supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and provide a quick migration path for thesauri and other taxonomies.

OWL-DL

> OWL DL supports those users who want the maximum expressiveness without losing computational completeness (all entailments are guaranteed to be computed) and decidability (all computations will finish in finite time) of reasoning systems. OWL

[27] http://www.daml.org/

[28] http://www.w3.org/TR/owl-features/

DL includes all OWL language constructs with restrictions such as type separation (a class can not also be an individual or property; a property can not also be an individual or class). OWL DL is so named due to its correspondence with description logics, a field of research that has studied a particular decidable fragment of first order logic. OWL DL was designed to support the existing Description Logic business segment and has desirable computational properties for reasoning systems.

OWL-Full

OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. In OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. Another significant difference from OWL DL is that a owl:DatatypeProperty can be marked as an owl:InverseFunctionalProperty. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support every feature of OWL Full.

3.3.2. Rules definition languages

3.3.2.1. RuleML[29]

RuleML is a markup language developed to express both forward (bottom-up) and backward (top-down) rules in XML for deduction, rewriting, and further inferential-transformational tasks. It is defined by the Rule Markup Initiative, an open network of individuals and groups from both industry and academia that was formed to develop a canonical Web language for rules using XML markup and transformations from and to other rule standards/systems. One or more rule engines will be needed for executing RuleML rulebases. On 2000-11-15, the RuleML Initiative thus joined forces with the Java Specification Request. This cooperation will enable a direct cross-fertilization between the specifications of the open XML-based Rule Markup Language and of the Java runtime API for rule engines.

3.3.2.2. SWRL (Semantic Web Rule Language)[30]

SWRL is based on a combination of the OWL DL and OWL Lite sublanguages of OWL with the Unary/Binary Datalog RuleML sublanguages of RuleML. SWRL extends the set of OWL axioms to include Horn-like rules. An extension of the OWL model-theoretic semantics is also given to provide a formal meaning for OWL ontologies including rules written in an abstract syntax. The rules are of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold.

[29] http://www.ruleml.org/

[30] http://www.w3.org/Submission/SWRL/

3.3.2.3. R2ML (REWERSE Rule Markup Language)[31]

R2ML is developed by the REWERSE Working Group for the purpose of rules interchange between different systems and tools, enriching of ontologies with rules, connecting of custom rule system with R2ML-based tools for vizualization, verbalization, verification and validation. R2ML is comprehensive in the sense that it integrates the Object Constraint Language (OCL) - a standard used in information systems engineering and software engineering, the Semantic Web Rule Language (SWRL)- a proposal to extend the Semantic Web ontology language OWL by adding implication axioms, the Rule Markup Language (RuleML) - a proposal based on Datalog/Prolog, and it includes four rule categories: derivation rules, production rules, integrity rules and ECA/reaction rules. R2ML provides a rich syntax for expressing rules supporting conceptual distinctions between different types of terms and different types of atoms, which are not present in standard predicate logic.

3.3.3. Queries definition languages

3.3.3.1. RDQL (Query Language for RDF)[32]

RDQL has been implemented in a number of RDF systems for extracting information from RDF graphs. An RDQL consists of a graph pattern, expressed as a list of triple patterns. Each triple pattern is comprised of named variables and RDF values (URIs and literals). An RDQL query can additionally have a set of constraints on the values of those variables, and a list of the variables required in the answer set. An RDQL query treats an RDF graph purely as data. If the implementation of that graph provides inferencing to appear as "virtual triples", then an RDQL will include those triples as possible matches in triple patterns. RDQL makes no distinction between inferred triples and ground triples. RDQL was first released in Jena 1.2.0. The following systems are known to provide RDQL: Jena, RDFStore, Sesame, PHP XML Classes, 3Store, RAP - RDF API for PHP. In addition, RDQL is one language used for remote query by the Joseki RDF Server.

3.3.3.2. SPARQL[33]

SPARQL is a query language for RDF. The SPARQL language includes IRIs, a subset of RDF URI References that omits spaces. All IRIs in SPARQL queries are absolute; they may or may not include a fragment identifier. IRIs include URIs and URLs. The abbreviated forms (relative IRIs and prefixed names) in the SPARQL syntax are resolved to produce absolute IRIs. The SPARQL query language is based on matching graph patterns. Graph patterns contain triple patterns. Triple patterns are like RDF triples, but with the option of query variables in place of RDF terms in the subject, predicate or object positions. Combining triple patterns gives a basic graph pattern, where an exact match to a graph is needed. SPARQL has several query forms. The SELECT query form returns tabular information. The CONSTRUCT query form returns an RDF graph. The graph is built based on a template which is used to generate RDF triples based on the results of matching the graph pattern of the query. Basic graph patterns are sets of triple patterns. SPARQL graph pattern matching

---

[31] http://oxygen.informatik.tu-cottbus.de/rewerse-i1/?q=R2ML

http://en.wikipedia.org/wiki/R2ML

[32] http://www.w3.org/Submission/RDQL/

[33] http://www.w3.org/TR/rdf-sparql-query/

is defined in terms of combining the results from matching basic graph patterns. A sequence of triple patterns interrupted by a filter comprises a single basic graph pattern. Any graph pattern terminates a basic graph pattern.

3.3.3.3. RQL (RDF Query Language)[34]

RQL is a typed language following a functional approach and supports generalized path expressions featuring variables on both labels for nodes (classes) and edges (properties). RQL relies on a formal graph model (as opposed to other triple-based RDF QLs) that captures the RDF modeling primitives and permits the interpretation of superimposed resource descriptions by means of one or more schemas.

RQL is able to combine schema and data querying while exploiting the taxonomies of labels and multiple classification of resources, using pattern-matching facilities. The RQL Interpreter consists of four modules a parser, analyzing the syntax of queries; a graph constructor, capturing the semantics of queries in terms of typing and interdependencies of involved expressions; a SQL Translator, which rewrites RQL to efficient SQL queries; and a evaluation engine, accessing the underlying database via SQL queries.

3.3.3.4. SeRQL (Sesame RDF Query Language)[35]

SeRQL is a RDF/RDFS query language that is currently being developed by Aduna as part of Sesame. It combines the features of other (query) languages (RQL, RDQL, N-Triples, N3) and adds some of its own. Some of SeRQL's most important features are:

graph transformation, RDF Schema support, XML Schema datatype support, expressive path expression syntax, optional path matching. The SeRQL query language supports two querying concepts. The first one can be characterized as returning a table of values, or a set of variable-value bindings. The second one returns a true RDF graph, which can be a subgraph of the graph being queried, or a graph containing information that is derived from it. The first type of queries are called "select queries", the second type of queries are called "construct queries".

3.3.3.5. nRQL (new Racer Query Language)[36]

nRQL can be seen as a straightforward extension and combination of the individuals repository querying mechanisms. nRQL allows the use of variables within queries, as well as much more complex queries. The variables in the queries are to be bound against the individuals in the repository that satisfy the specified query. Queries make use of concept and role terms. It is possible to use individuals in query expressions as well. A nRQL query is composed of a query head and a query body. The query body consists of the query expression whereas the query head corresponds to the variables mentioned in the body that will be bound to the result. Visually, the query head is symbolized by the selection of the concepts whose instances are to be part of the result.

---

[34] http://139.91.183.30:9090/RDF/RQL/

http://www.openrdf.org/doc/rql-tutorial.html

[35] http://www.openrdf.org/doc/sesame/users/ch06.html

http://www.openrdf.org/sesame/serql/serql-examples.html

[36] http://users.encs.concordia.ca/~haarslev/racer/racer-queries.pdf

3.3.3.6. TRIPLE (A Query Language for the Semantic Web)[37]

TRIPLE is an RDF query, inference, and transformation language for the Semantic Web. TRIPLE formulates queries in an ontology to correspond to a user-specific view. The approach is based on maintenance of multiple views expressed in ontologies simpler than the domain ontology. This allows users to query heterogeneous data repositories. Ontology developers can define view ontologies and corresponding mapping rules. The use of the language TRIPLE is useful when not only representation of RDF data is necessary but also a definition of views and mappings at the same time. TRIPLE provided an infrastructure to evaluate queries issued against the user-specific ontology and in this way to realize the view. The view techniques have central importance when creating interoperability. TRIPLE has the following overview features: native support for resources and namespaces, abbreviations; models (sets of RDF statements); reification; rules with bodies; transformations; syntactical extension of Horn Logic; syntactically similar to F-Logic; has both ASCII and RDF syntax.

3.3.4. Mapping definition languages

3.3.4.1. Ontology Mapping Language[38]

The ontology mapping language allows specifying correspondences between two ontologies. What makes it specific is the possibility to represent correspondences independently from the language the ontologies are modeled in. The mapping language gives a way to represent different kinds of schema mappings. Lisp style syntax is also maintained. The Mapping Language allows describing mapping documents that each describe an alignment. The mapping document consists of a header (annotations, ontologies – the data resources that have to be mapped to each other) and a number of cells (mapping features- represent the concrete correspondences between units of the data resources).

3.3.4.2. SKOS[39]

The Simple Knowledge Organization System is a data-sharing standard, bridging several different fields of knowledge, technology and practice. The Simple Knowledge Organization System is a common data model for knowledge organization systems such as thesauri, classification schemes, subject heading systems and taxonomies. Using SKOS, a knowledge organization system can be expressed as machine-readable data. It can then be exchanged between computer applications and published in a machine-readable format in the Web. The ontology is very expressive – OWL Full sublanguage.

---

[37] http://triple.semanticweb.org/

[38] http://www.omwg.org/TR/d7/rdf-xml-syntax/

http://www.deri.at/fileadmin/documents/DERI-TR-2004-06-30.pdf

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

http://www.w3.org/TR/skos-reference/

4. Written Corpora

The need for text encoding standards for language resources (LRs) is widely acknowledged: within the ISO/TC 37/SC 4, work in this area has been going on since the early 2000s, and working groups devoted to this issue have been set up in two current pan-European projects, CLARIN (http://www.clarin.eu/) and FLaReNet (http://www.flarenet.eu/). It is obvious that standards are necessary for the interoperability of tools and for the facilitation of data exchange between projects, but they are also needed within projects, especially where multiple partners and multiple levels of linguistic data are involved.

One such project is the international project KYOTO (Knowledge Yielding Ontologies for Transition based Organization; http://www.kyoto-project.org/), involving 11 institutions from Europe and Asia. Another is the much smaller National Corpus of Polish project (Pol. Narodowy Korpus Języka Polskiego; NKJP; http://nkjp.pl/; (Przepiórkowski et al., 2008) and (Przepiórkowski et al., 2010)) involving 4 Polish institutions. What these two very different projects have in common is the strong emphasis on the conformance with current XML standards in LR encoding. It is interesting that this common objective gives rise to very different practices in these projects.

In this chapter, two encoding standards for LR are presented, namely TEI P5 and CES/XCES. TEI P5 will be presented on the basis of an example, the TEI encoding of the National Corpus of Polish, in section 4.2. CES/XCES will be presented in section 4.3.

4.1. Scope, Terminology Definitions

A *written corpus* is a collection of written texts (*primary data)*. Data sources can vary greatly. They include newspapers, legal documents, literary texts, and others. While some corpora exclusively rely on a single source (e.g., the English Penn Treebank or the German Tübingen Treebank of Written German only consist of newspaper text), other treebanks, such as the Bulgarian BulTreebank or the National Corpus of Polish (see 4.2) mix sources of different kinds.

Corpora become exploitable through *annotation*. We distinguish three categories of information with which a corpus can be augmented (cf. 4.3).

- *documentation*: global information about the text, its content, and its encoding. Bibliographic description of the document, documentation of character sets and entities, description of encoding conventions, etc.
- *primary data*: two types of information may be encoded:
    - *gross structure*: structural units of text (volume, chapter, paragraph, but also footnotes, titles, headings, tables, figures, etc.), features of typography and layout (to cover legacy data from printed material), non-textual information (graphics, etc.).
    - *sub-paragraph structures*: elements appearing at the sub-paragraph level (orthographic sentences, orthographic words, abbreviations, names, dates, etc.)
- *linguistic annotation*: morphological annotation, syntactic annotation, alignment of parallel texts, prosody markup etc.

This chapter is concerned with standards for the encoding of all three levels. Two standards are in active use for the encoding of such information. These are TEI P5 and CES/XCES. TEI P5 will be

presented on the basis of an example, namely the TEI encoding of the National Corpus of Polish, in section 4.2. CES/XCES will be presented in section 4.3. Schemes for linguistic annotation of different type are presented in the following chapters.

4.2. The TEI encoding of the National Corpus of Polish

Authors: Adam Przepiórkowski, Piotr Bański

4.2.1. Introduction

National Corpus of Polish (Pol. Narodowy Korpus Języka Polskiego; NKJP; http://nkjp.pl/) is a project carried out in 2008–2010, involving 4 Polish institutions: Institute of Computer Science of the Polish Academy of Sciences (coordinator), Institute of Polish Language of the Polish Academy of Sciences, University of Łódź and Polish Scientific Publishers PWN. Each of these institutions contributes texts from their own corpora, and each — apart from the coordinator — acquires new texts for the National Corpus of Polish (NKJP, henceforth): books, newspapers and magazines, blogs, transcripts of spoken data, etc. All these texts are imported into two very different search engines available in NKJP (cf. the "Demo" link at http://nkjp.pl/).

Obviously, before NKJP texts can be indexed or automatically processed by any other tools they must be converted to a common interchange format. Such interchange format should allow for the representation of various types of texts mentioned above, and also for the encoding of various kinds of metadata and structural information. The only text encoding standard sufficiently versatile to meet these requirements is TEI P5, presented in the Guidelines of the Text Encoding Initiative (TEI; (Burnard and Bauman, 2008); http://www.tei-c.org/). It is not an official ISO standard, but a mature and very specific XML-based *de facto* standard for text encoding in the humanities, with a rich user base and supporting tools.

The reason for continuing beyond the previous paragraph is that TEI is a large treasure trove of solutions, rather than a lean and highly focused formalism, and a particular text encoding schema must still be designed by choosing the most appropriate mechanisms from the TEI toolbox and — in rare specific cases — by introducing new XML elements or attributes. The aim of this section is to present and document one such particular schema, developed within NKJP. As there are few well-documented TEI corpora around, and hardly any corpora following the current P5 version of the TEI Guidelines (substantially differing from the previous TEI versions), we hope that this presentation will facilitate the development of other TEI P5 corpora.

4.2.2. Corpus Header

Following the TEI Guidelines, the NKJP corpus header consists of 4 sections contained in the `<teiHeader xml:lang="en" type="corpus">` element: `<fileDesc>`, `<profileDesc>`, `<encodingDesc>` and `<revisionDesc>`.

Two of these have very simple structure. First, `<profileDesc>` identifies the main languages used in the TEI encoding of texts and metadata, and it is cited in its entirety below:

```
<profileDesc>
 <langUsage>
  <language ident="pl">Polish</language>
  <language ident="en">English</language>
```

```
  </langUsage>
 </profileDesc>
```

The values of `@ident` attributes may be used for any element to specify the language of the content of that element. In fact, the `xml:lang="en"` specification in the `<teiHeader>` element is inherited by other elements in the header, unless explicitly overridden by `xml:lang="pl"`, thus making English the default language of the NKJP header.

Another simple and homogeneous section is `<revisionDesc>`: it contains a sequence of `<change>` statements like the following:

```
<change who="#adamp" when="2009-08-01">
  Added <gi>profileDesc</gi>.
</change>
```

The `<fileDesc>` section contains 4 subsections. The first, `<titleStmt>`, specifies the name of the corpus and describes the responsibility of various institutions and persons involved in its creation. One such responsibility statement is referenced by `who="#adamp"` in the example above, another may look as follows:

```
<respStmt>
 <persName xml:id="bansp">Piotr Baski</persName>
 <resp>initial design of various XML schemata</resp>
</respStmt>
```

The other three subsections of `<fileDesc>` are: `<editionStmt>` — a brief statement concerning the stability of the current version of NKJP, `<publicationStmt>` — defining availability and distribution of NKJP, and `<sourceDesc>` — specifying the origin of texts in general terms (specific source descriptions are contained in the headers of particular texts).

Finally, `<encodingDesc>` characterizes NKJP in various ways, e.g., `<projectDesc>` repeats the description of the project given at http://nkjp.pl/, `<samplingDecl>` says that Whole texts are included, whenever possible and provides some information on text structure, as discussed in 4.2.4, and `<editorialDecl>` briefly discusses anonymization of spoken data and other editorial interventions in NKJP texts.

While these subsections contain free-text statements, many other `<encodingDesc>` subsections are more structured. Perhaps the most important are `<classDecl>` subsections, which specify text classifications referenced in particular text headers. For example, one of the ways in which NKJP texts are classified is according to the Universal Decimal Classification, so the following declaration is present in the corpus header:

```
<classDecl>
 <taxonomy xml:id="ukd">
  <bibl>
   <title xml:lang="pl">Uniwersalna Klasyfikacja
    Dziesitna</title>
   <title xml:lang="en">Universal Decimal
    Classification</title>
   <edition>UDC-P058</edition>
```

```
  </bibl>
 </taxonomy>
</classDecl>
```

Within a text header (cf. 4.2.3), a reference to this classification may be made as follows: `<classCode scheme="#ukd">821.162.1-3</classCode>`. Similarly, in order to control the good balance of the corpus with respect to genres, a taxonomy of text types is defined; its fragment is presented below:

```
<classDecl>
 <taxonomy xml:id="taxonomy-NKJP-type">
 <!-- ... --->
   <category xml:id="typ_lit_proza">
    <desc xml:lang="pl">proza</desc>
    <desc xml:lang="en">prose</desc>
   </category>
   <category xml:id="typ_lit_poezja">
    <desc xml:lang="pl">poezja</desc>
    <desc xml:lang="en">poetry</desc>
   </category>
   <category xml:id="typ_lit_dramat">
    <desc xml:lang="pl">dramat</desc>
    <desc xml:lang="en">drama</desc>
   </category>
 <!-- ... --->
 </taxonomy>
</classDecl>
```

Again, the type of a particular text may be defined by referencing one of the categories defined in such a classification.

The final `<encodingDesc>` subsection in the NKJP header to be mentioned here is `<nkjp:fsLib>`. As the namespace prefix nkjp suggests, this element is not defined by TEI but introduced within NKJP for a specific purpose.

TEI specifications contain the ISO standard on the XML representation of feature structures (ISO 24610-1:2006) and, within NKJP, feature structures are used for representing various types of linguistic annotation (Przepiórkowski and Bański, 2009). The standard makes it possible to define feature structure libraries containing, e.g., feature structures representing complex morphosyntactic information. Such feature structures may subsequently be referenced from an appropriate linguistic layer by their identifier, thus simultaneously increasing readability and compactness of linguistic representations. Curiously, according to TEI specifications it is not possible to define such feature structure libraries in a header, which seems to be the most natural place for such libraries: once they are in the corpus header, they may be referenced in a way analogous to how particular categories defined within `<classDecl>` are referenced for classification. Hence the need for the project-specific element `<nkjp:fsLib>`.

The presence of such `nkjp:…` elements and attributes is what makes the NKJP schema TEI conformant in a weaker sense: it is a TEI Extension rather than TEI Conformant (with a capital 'C'), as defined in (Burnard and Bauman, 2008). Nevertheless, there are only a few conservative and well-justified modifications of this kind in the NKJP schema presented here, so it may be regarded as a "nearly" TEI Conformant TEI Extension.

4.2.3. Text Header

Each NKJP text is represented as a number of XML files, two of which are relevant here: `header.xml` and `text_structure.xml`. The structure of the latter will be presented in 4.2.4. The structure of text headers, `header.xml`, is similar to that of the corpus header: the `<teiHeader>` element, implicitly marked as `type="text"` here, contains three sections: `<fileDesc>`, `<profileDesc>` and `<revisionDesc>`. The last section, `<revisionDesc>`, is fully analogous to that of the corpus header and contains a sequence of `<change>` elements describing modifications to any of the files representing the text and its annotation.

On the other hand, `<profileDesc>` differs from that of the corpus header and it comprises one element, `<textClass>`, which contains classifiers of the text, referencing appropriate taxonomies posited in the corpus header. For example, the content of `<profileDesc>` for Manuela Gretkowska's novel *Namiętnik* may look as follows:

```
<profileDesc>
 <textClass>
  <classCode scheme="#ukd">821.162.1-3</classCode>
  <keywords scheme="#bn">
   <list>
    <item>Opowiadanie polskie -- 20 w.</item>
   </list>
  </keywords>
  <catRef scheme="#taxonomy-NKJP-type"
   target="#typ_lit_proza"/>
  <catRef scheme="#taxonomy-NKJP-channel"
   target="#kanal_ksiazka"/>
 </textClass>
</profileDesc>
```

References to 4 classification schemes are made here: two external to the project (the Universal Decimal Classification mentioned above and the classification of the Polish National Library, cf. `#bn`), and two internal (text type: `#typ_lit_proza`, i.e., prose, and text channel: `#kanal_ksiazka`, i.e., book).

Finally, `<fileDesc>` contains a variety of information about the text: its title in NKJP (e.g., "`TEI P5 encoded version of "Namiętnik"`"), bibliographic information about the source of the text (title, author, publisher, etc.), a note about the origin of the text in NKJP (e.g., "`<note type="text_origin">IPI PAN Corpus</note>`"), the original header, if available, of the text as it was defined in the corpus from which the text is inherited, as well as a `<publicationStmt>`, exemplified below:

```
<publicationStmt nkjp:subcorpus="balanced">
 <availability status="restricted">
  <p>For all NKJP purposes.</p>
 </availability>
</publicationStmt>
```

The `@nkjp:subcorpus` attribute shown above is another example of a non-Conformant modification of TEI, needed here in order to represent the information about the target NKJP subcorpus for which the text was acquired.

For transcripts of spoken data, the text header may also contain information about the person responsible for transcription (encoded as `<respStmt>` within `<fileDesc>`), various kinds of information about the source of the text (different from written texts, because here the source is a recording rather than a publication), as well as another element from the `nkjp` namespace, `<nkjp:topic>`, describing the topic of the conversation. Moreover, apart from `<textClass>`, `<profileDesc>` also contains a `<langUsage>` element specifying the level of formality of the conversation, `<particDesc>`, containing background information about participants in the conversation, as well as `<settingDesc>`, mentioning when and where the conversation took place; some of these elements specific to transcripts of spoken data are exemplified below:

```
<langUsage>
 <language ident="pl-x-formal"/>
</langUsage>
<nkjp:topic xml:lang="pl">Rozmowa o immunitecie Zbigniewa
 Ziobro, sytuacji w Gruzji i reakcji unii europejskiej na
 ni.</nkjp:topic>
<particDesc>
 <!-- ... --->
 <person xml:id="sp2" role="speaker">
  <persName>Zbigniew Ziobro</persName>
  <sex value="1">male</sex>
  <education xml:lang="pl">wysze</education>
  <age>40</age>
  <residence>unknown</residence>
 </person>
 <!-- ... --->
</particDesc>
<settingDesc>
 <setting>
  <name type="place">TVP Info</name>
  <name type="voivodship" xml:lang="pl">mazowieckie</name>
  <date type="recorded" when="2008-09-02"/>
 </setting>
```

```
</settingDesc>
```

4.2.4. Text Structure

For any corpus document, `text_structure.xml` contains the actual text, as well as the structural markup of the document.

It is often considered best practice to have a read-only pure text file referenced by a stand-off file containing structural information. The main justification for this requirement is the need for an immutable text level. While stand-off annotation is used for all other NKJP layers, it has turned out to be impractical to separate primary data and structure. The reason for that is that corpus data are virtually never acquired as pure text, but almost always come with some markup already present: XML, HTML, or even implicit markup in Microsoft Word and OpenOffice files. Separating this markup from text for the sake of stand-off annotation (rather then converting it *in situ* into the appropriate TEI markup), and then logically combining them again for processing at later stages, would only generate unnecessary work. Hence, for the purposes of NKJP, it is `text_structure.xml` that is considered immutable.

The outline of `text_structure.xml`, containing a single text and any structural annotation, is as follows, with the `<front>` and `<back>` matter elements optional (always absent in transcripts of spoken data):

```
<teiCorpus xmlns:xi="http://www.w3.org/2001/XInclude" xmlns="http://www.tei-c.org/ns/1.0">
<xi:include href="NKJP_header.xml"/>
<TEI>
<xi:include href="header.xml"/>
<text xml:id="struct_text">
<front><! - front matter -></front>
<body><! - main text body -></body>
<back><! - back matter -></back>
</text>
</TEI>
</teiCorpus>
```

It should be noted that each text is a `<teiCorpus>` and logically includes not only the text header (`<xi:include href="header.xml"/>`), but also the entire corpus header (`<xi:include href="NKJP_header.xml"/>`).

For `<front>` and `<back>`, any structural elements defined in TEI P5 are allowed. Typically, within front matter there will be a title statement, possibly distinguishing between the main title and the subtitle, as in the following example:

```
<docTitle>
 <titlePart type="main">Pieni ndzy i zagady</titlePart>
 <titlePart type="sub">Twrczo Mordechaja Gebirtiga
  w Salonie Poezji</titlePart>
</docTitle>
```

On the other hand, in NKJP, the content of <body> is constrained with respect to the range of possibilities offered by TEI P5. For spoken texts, only a sequence of <u>tterances (and perhaps <incident>s between them) may occur within <body>, as in the following fragment:

```
<body xml:id="txt_body">
 <u who="#sp3" xml:id="u1">ale zostaw to w ogle dajcie
  buziaka przepraszam was laski</u>
 <u trans="overlap" who="#sp1" xml:id="u2">no daymy
  sobie buziaka no</u>
 <!-- ... -->
</body>
```

For written texts, the main blocks are <p> (paragraph), <ab> (anonymous block, i.e., a paragraph-sized chunk of text exceptionally used for texts without division into paragraphs) and <head> (for headings starting a textual division at any level, e.g., for chapter and section titles). These blocks can be grouped into chapters, sections, subsections, etc., using the — possibly nested — elements, e.g.:

```
<body>
 <!-- ... -->
 <div type="chapter" n="1">
  <head>Rozdzia 1 Skd si bior paradygmaty?</head>
  <p><!-- ... --></p>
  <p><!-- ... --></p>
  <p><!-- ... --></p>
 </div>
 <div type="chapter" n="2">
  <head>Rozdzia 2 wiat wedug Pszczki Mai</head>
  <p><!-- ... --></p>
  <p><!-- ... --></p>
  <p><!-- ... --></p>
 </div>
 <!-- ... -->
</body>
```

The content of the elements <u>, <p>, <ab> and <head> is (almost) pure text; the only XML elements which may appear there are: <gap> (to mark places where tables, pictures, etc., were removed from text), <hi>, with the obligatory attribute @rend specifying how the highlighted text is rendered (in written texts only), <lb> (always empty, to mark line breaks in poetry or in a motto; only in written texts) and two elements used in spoken texts for marking non-verbal events, <vocal> and <incident>.

These restrictions on <u>, etc., are caused by the fact that the content of these elements will be further marked linguistically and indexed by search engines, so any additional markup which would complicate the operation of these tools must be well justified. More specifically, the following XPath expressions define the elements to be further processed:

- `//body//(p—ab—u—incident—head)`
- `//front//titlePart`

In other words, the content of any `<p>`, `<ab>`, `<u>`, `<incident>` and `<head>` elements anywhere within `<body>`, as well as the content of `<titlePart>` anywhere within `<front>` will be marked linguistically and indexed by corpus search engines.

4.3. Written corpora in CES/XCES

Author: Maria Gavrilidou

4.3.1. Introduction

XCES (Corpus Encoding Standard for XML) instantiates the EAGLES Corpus Encoding Standard (CES) DTDs for linguistic corpora, developed by the Department of Computer Science, Vassar College, and Equipe Langue et Dialogue, LORIA/CNRS.

To understand XCES, the documentation on CES is very informative. CES is based on the *EAGLES Guidelines* developed by the Expert Advisory Group on Language Engineering Standards (EAGLES). It was designed in the 90's to be used in the field of HLT for corpus encoding at various levels. The CES is an application of SGML ISO 8879:1986.

The CES specifies a minimal encoding level for corpora. It also provides encoding specifications for linguistic annotation, as well as a data architecture for linguistic corpora. The CES covers the encoding of objects in primary data (i.e. "un-annotated" data, most often originally created for non-linguistic purposes such as publishing, broadcasting, etc.); the objects include paragraphs, chapters, etc. together with titles, footnotes, etc., but also sub-paragraph-level elements of interest for linguistic analysis, such as sentences, names, dates, abbreviations, terms, etc. Besides primary data objects, the CES covers linguistic annotation of text and speech, such as morphosyntactic tagging, parallel text alignment, prosody, phonetic transcription, etc.

The CES provides recommendations and tagsets for the documentation and encoding of primary data and linguistic annotation.

According to the CES, the encoding of linguistic annotation should be maintained separately from the primary data, to which it is linked. Three broad categories of information are distinguished:

- *documentation*: global information about the text, its content, and its encoding. This type of markup corresponds roughly to the TEI header and includes bibliographic description of the document, documentation of character sets and entities, description of encoding conventions, etc.
- *primary data*: two types of information may be encoded:
    - *gross structure*: structural units of text (volume, chapter, paragraph, but also footnotes, titles, headings, tables, figures, etc.), features of typography and layout (to cover legacy data from printed material), non-textual information (graphics, etc.).
    - *sub-paragraph structures*: elements appearing at the sub-paragraph level (orthographic sentences, orthographic words, abbreviations, names, dates, etc.)
- *linguistic annotation*: morphological annotation, syntactic annotation, alignment of parallel texts, prosody markup etc.

4.3.2. The CES Header

Central to the CES is the notion of the header, as adopted from the TEI and customized to the needs of HLT; headers are foreseen for the entire corpus, as well as for each individual text within a corpus. The header consists of four main constituents:

- the file description `<fileDesc>` element documents the whole corpus (in the case of a corpus header) or the individual text to which the header applies (in the case of a text header).
- the encoding description `<encodingDesc>` element contains information about the relationship between an encoded text and its original source (such as the description of the sampling method used for the creation of the corpus, information about the tagging applied, the project for and by which the text or corpus was created etc.)
- the profile description `<profileDesc>` element contains descriptive information related to the creation of the corpus or text, the languages, sublanguages, registers, dialects etc. represented therein, the topic of a text, it links the text to its translations (if applicable) or to annotation files associated with the text
- the revision description `<revisionDesc>` element records changes to the corpus (such as versions etc.).

4.3.3. Encoding of primary data

For the encoding of primary data the CES identifies three levels of encoding, Level 1 being the minimum encoding level (level of paragraph), Level 2 demanding encoding of information at the sub-paragraph level and Level 3 being the most detailed level of encoding (tokenization of abbreviations, numbers, dates, names, foreign words and phrases, etc.)

4.3.4. Linguistic annotation

The CES adopts a strategy whereby annotation information is not merged with the original, but rather retained in separate SGML documents (with different DTDs) and linked to the original or other annotation documents. The hyper-document comprising each text in the corpus and its annotations will consist of several documents. The base or "hub" document is the unannotated document containing only primary data markup. The hub document is not modified in the annotation process. Each annotation document is a proper SGML document with a DTD, containing annotation information linked to its appropriate location in the hub document or another annotation document.

5. Annotation

5.1. General annotation frameworks (TEI, LAF)

As mentioned above the TEI P5 Guidelines are de facto, constantly maintained XML standard for encoding and documenting primary data, with an active community, detailed guidelines (Burnard and Bauman, 2008) and supporting tools. Its recommendations for the encoding of linguistic information are limited, but it includes the ISO FSR standard for representing feature structures, which can be used to encode various kinds of information.

5.1.1. TEI annotation of the National Corpus of Polish

Authors: Adam Przepiórkowski and Piotr Bański

For reasons discussed in section 4 of (Przepiórkowski and Bański, 2009) TEI P5 has been adopted as the main standard in NKJP. However, TEI is a rich toolbox, providing a variety of tools to address particular problems. Whenever there is a choice, an attempt has been made to select a solution isomorphic with other proposed, official and de facto standards.

5.1.1.1. Metadata, primary data and structure

The CLARIN short guide on metadata (CLARIN:CM, 2009) makes the following recommendation: We recommend using. . . (1) IMDI and its special profiles including TEI elements or (2) OLAC, and later adds: Also components and profiles will be offered that contain IMDI, TEI and OLAC specifications to take care of the already existing metadata records. Hence, the use of TEI headers is in line with current best practices, and natural for LRs otherwise represented according to the TEI Guidelines. Apart from a TEI header for each text (`header.xml`), there is a general TEI corpus header, describing NKJP as a whole (`NKJP_header.xml`).

There is also no viable alternative to TEI for the representation of primary data and text structure. Texts are acquired for NKJP from a variety of sources, including previous participating corpora, publishers, Internet, media, original recordings of spontaneous conversations. They come with different kinds of structural information and different front and back matters. Some are divided into paragraphs or paragraph-like blocks, others into conversation turns. TEI Guidelines provide well-defined elements for all these situations. TEI P5 encoding of metadata, primary data and structural information, as employed in the National Corpus of Polish, is presented in detail in (Przepiórkowski and Bański, 2009). The outline of `text_structure.xml`, containing a single text and any structural annotation, is as follows, with `<front>` and `<back>` (matter) elements optional:

```
<teiCorpus
xmlns:xi="http://www.w3.org/2001/XInclude"
xmlns="http://www.tei-c.org/ns/1.0">
<xi:include href="NKJP_header.xml"/>
<TEI>
<xi:include href="header.xml"/>
<text xml:id="struct_text">
```

```
<front><!-- front matter --></front>
<body><!-- text to annotate --></body>
<back><!-- back matter --></back>
</text>
</TEI>
</teiCorpus>
```

In the case of written texts, the element <body> contains possibly nested elements, expressing the overall structure of the text and containing <p>aragraphs (or paragraph-like anonymous blocks, <ab>). For spoken data, <body> consists of <u>tterances.

5.1.1.2. Segmentation

Within any ann_segmentation.xml file, the <body> element contains a sequence of <p>, <ab> or <u> elements mirroring those found in the <body> of the corresponding text_structure.xml. The parallelism is expressed via TEI @corresp attributes on these elements; their values refer to the corresponding elements in text_structure.xml. Any other structural markup is not carried over to this or other linguistic levels. Each paragraph or utterance is further divided into <s>entences and even further into <seg>ments which define the span of each segment, by providing offsets to an appropriate element in text_structure.xml.[40] Each such <seg> element bears the implicit attribute @type="token".

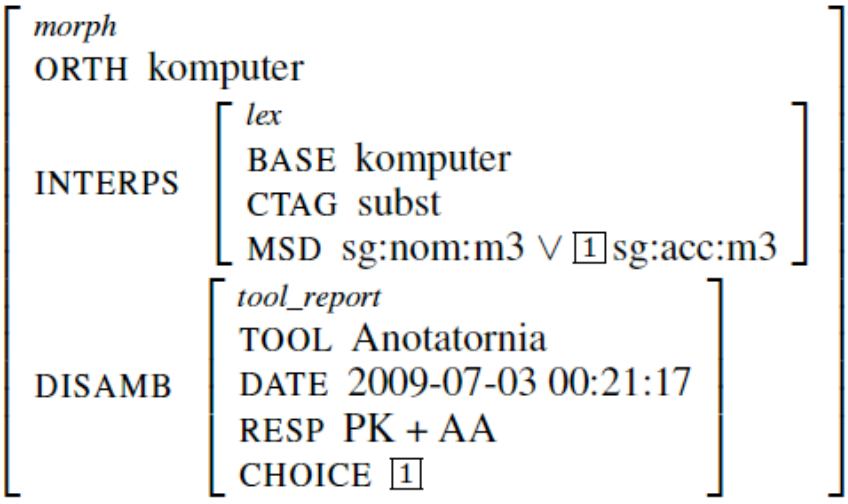
```
<seg xml:id="segm_1.1-seg"
corresp="text_structure.xml#C string-range(txt_1.1-p,0,6)"/>
```

5.1.1.3. Morphosyntax

The overall structure of ann_morphosyntax.xml, down to the level of <seg> (also implicitly marked as @type="token"), is identical to that of ann_segmentation.xml, with each <seg> referring — via the value of @corresp — to the corresponding segment at the segmentation level. Within <seg>, however, a feature structure — encoded in conformance with the FSR ISO standard — represents information about all morphosyntactic interpretations of a given segment, as well as about the tool used to disambiguate between them and the result of the disambiguation. For example, the logical structure of the content of a <seg> representing the noun komputer (singular, inanimate masculine, nominative or accusative) may be represented as follows:[41]

[40] Two complexities concerning alternative segmentations and information about boundedness of segments are discussed—and solutions are proposed — in (Banski and Przepiórkowski, 2009).

[41] In this case manual disambiguation was performed by two annotators, anonymised here as PK and AA, with the help of a tool called Anotatornia.



Note that the names of features BASE, CTAG and MSD are taken from XCES. The value of INTERPS may actually be a list of feature structures, representing interpretations differing in base form (BASE) or grammatical class (CTAG). In cases where interpretations differ only in morphosyntactic description (MSD), they are listed locally, as alternative values of MSD. Hence, it is the value of MSD that is used for the disambiguation information within DISAMB|CHOICE.

5.1.1.4. Syntactic words

Word segmentation in the sense of the previous two levels, as produced by a morphological analyzer used in NKJP, is very fine-grained: segments never contain spaces, and sometimes orthographic ("space-to-space") words are broken into smaller segments. For this reason an additional level is needed that will contain multi-token words, e.g., analytical tense forms of verbs. It is this level that corresponds most closely to MAF, a morphological annotation framework in development by ISO (ISO/CD 24611). However, while MAF assumes that `<token>`s and `<wordForm>`s reside in the same file (with `<token>` perhaps referring to primary data in a different file), we need a stand-off encoding referring to `ann_morphosyntax.xml`. Down to the `<s>`entence level, `ann_words.xml` follows the same design as other levels, and links its `<s>` elements to those in `ann_morphosyntax.xml`, again via `@corresp`. Each sentence at this level is a list of `<seg>`ments of `@type="word"` covering the whole original sentence. In the default case, a `<seg>`ment at this level will be co-extensive with a `<seg>` at the lower level, but it may also correspond to a possibly discontinuous list of such token-level `<seg>`ments. Two different syntactic words may also overlap, as in BAŁ SIĘ ZAŚMIAĆ '(He) feared (to) laugh', where for two inherently reflexive verbs, BAĆ SIĘ 'fear' and ZAŚ MIAĆ SIĘ 'laugh', one occurrence of the reflexive marker się suffices. One way to represent such syntactic words in TEI is given schematically below. The feature structure `<fs>` contains information about the lemma and the morphosyntactic interpretation of the word, similarly to the information at the morphosyntactic levels, but without ambiguities. Segments in `ann_morphosyntax.xml` (and possibly syntactic words in `ann_words.xml`) within the given word are referenced via the `<ptr>` element.

```
<seg xml:id="word13">
<fs>...</fs>
<ptr target="ann_morphosyntax.xml#seg15"/>
<ptr target="ann_morphosyntax.xml#seg16"/>
```

```
<ptr target="ann_morphosyntax.xml#seg18"/>
</seg>
```

5.1.1.5. Named entities and syntactic groups

Files representing the following two levels, `ann_named.xml` for named entities (NEs) and `ann_groups.xml` for syntactic groups, also have the same overall structure down to the `<s>` level, but within each sentence only the information pertinent to the current level is represented, so, in particular, some `<s>` elements within `ann_named.xml` may be empty, if the relevant sentences do not contain any named entities. Both levels refer independently to the level of syntactic words.

Within `ann_groups.xml`, each sentence is a sequence of `<seg>`ments of `@type="group"` structured in a way analogous to the word-level `<seg>` elements described above: they consist of a feature structure describing the syntactic group, as in the following simplified example. Note that the `@type` attribute of `<ptr>` defines the kind of relation between the node and its immediate constituent; note also that `<ptr>` elements have `@xml:id` values and, hence, may be referenced from within the `<fs>` description of the group.

```
<seg xml:id="group4">
<fs>...</fs>
<ptr xml:id="id1" type="head" target="ann_words.xml#word10"/>
<ptr xml:id="id2" type="nonhead" target="ann_words.xml#word12"/>
<ptr xml:id="id3" type="nonhead" target="#group3"/>
</seg>
```

The representation of NEs is analogous, with the following differences: 1) the implicit value of `@type` is `"named"` instead of `"group"`, 2) different information is represented within the `<fs>` description; this includes the type of the named entity, as well as the base form of the NE, which, obviously, does not need to be a simple concatenation of base forms of words within the NE, 3) there seems to be no need for the `@type` attribute within `<ptr>`.

5.1.1.6. Word senses

Within NKJP, a limited number of semantically ambiguous lexemes will be disambiguated.[42] In a manner analogous to the morphosyntactic level, each `<s>` contains a sequence of token-level `<seg>`ments, with `@corresp` references to `<seg>`ments in ann_segmentation.xml.[43] Each `<seg>` contains a feature structure with a reference to the appropriate sense in an external word sense inventory, e.g.:

```
<seg xml:id="seg2" corresp="ann_segmentation.xml#seg17">
<fs type="sense">
<f name="sense" fVal="NKJP_WSI.xml#sam.2"/>
</fs>
```

[42] See also (Młodzki and Przepiórkowski, 2009).

[43] This is a technical decision; in the future, the word sense level may be changed to reference syntactic words rather than segments.

```
</seg>
```

In a way analogous to the two levels described in the preceding subsection, only those segments are represented here which were semantically disambiguated, so some <s> elements will be empty.

5.1.1.7. Conclusion

For each specific TEI P5 solution presented above there are other ways of representing the same information in a way conformant with the TEI P5 Guidelines. For example, instead of recycling the <seg> element with different @type values, TEI elements such as <w> (for words), <phr> and <cl> (for syntactic groups), and even <persName>, <geogName>, <orgName> and <date> (for various kinds of named entities) could be used at different levels. Instead of using <ptr> links, nested structures could be represented straightforwardly via the nesting of XML elements, or—much less straightforwardly — as feature structures (Witt et al., 2009) etc. The encoding proposed here was designed with the view of maximizing compatibility with other standards, whether sanctioned by ISO or de facto in use. It is directly mappable to specific encodings such as TIGER-XML and PAULA, and it is an instantiation of sometimes rather abstract models developed within ISO TC 37 / SC 4. We conjecture that—given the stability, specificity and extensibility of TEI P5 and the relative instability and generality of some of the other proposed standards — this approach is currently the optimal way of following corpus encoding standards.

5.1.2. Linguistic Annotation Framework (LAF)

Author: Kerstin Eckart

LAF, the Linguistic Annotation Framework is an upcoming ISO standard. It is developed within ISO, technical committee 37 (ISO/TC 37/SC 4), under the title: Language resource management – Linguistic annotation framework (LAF).

LAF is related to other (upcoming) standards developed within ISO/TC 37 such as the Specification of data categories and management of a Data Category Registry for language resources (ISO 12620:2009) and Feature structures – Part 1: Feature structure representation (ISO 24610-1:2006).

5.1.2.1. Status

LAF is a standard under development in enquiry stage and is considered as a draft international standard ISO/DIS 24612 by the latest version of this deliverable.

5.1.2.2. Aspects

LAF provides a meta model to represent primary data and different kinds of linguistic annotations (without commitment to one particular linguistic theory). LAF provides stand-off annotations, i.e. each annotation layer is kept in a separate document and can contain references to elements in other annotation layers.

In LAF, primary data documents must not contain any markup to be interpreted. If markup-strings are included in a primary data document, they are treated like primary data itself. Segmentation annotations are kept in a different document. Therefore it is possible to have different segmentation annotations for the same primary data. Primary data is referenced by defining virtual nodes in-between each character (or byte sequence denoting the base unit of representation) of the primary data document (see example below). The default encoding for textual data is UTF-8.

As various types of annotation layers can be represented, each document has a document header either included into the annotation document itself or kept in a standalone file; the latter is obligatory in case of a primary data document. The header is based on the XCES (http://www.xces.org/) and TEI (http://www.tei-c.org) header.

To provide high interoperability LAF stipulates an XML-serialization as a pivot format, i.e. as an exchange format. Users can transform their data format into the LAF pivot format, and from the pivot format into any other format for which an exporter exists. The pivot format may also serve as a basis for comparing annotations from different formats.

Although the current XML-Serialization of LAF focuses on textual data, LAF is designed to handle other types of media as well.

5.1.2.3. Serialization

The XML-serialization of the LAF pivot format is called GrAF (Graph Annotation Format). It is based on graph data structures and includes nodes, edges and annotations as well as regions for referencing primary data. Annotations state the annotation set which is used and are realized as feature structures (cf. ISO 24610-1). Therefore annotations at different levels of complexity can be represented (Ide and Suderman, 2007).

Excerpt of an example of the current format:

The following examples are extracted from MASCI Corpus (cf. http://www.anc.org/MASC/Home.html).

In a segmentation document, regions of primary data are denoted by anchors:

```
<region xml:id="seg-r194" anchors="560 563"/>
<region xml:id="seg-r196" anchors="564 567"/>
<region xml:id="seg-r198" anchors="568 575"/>
```

Figure 22: Anchors

The virtual anchors are situated between the characters:

```
|h|i|s| |o|w|n| |w|e|b|s|i|t|e|
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
```

Figure 23: Virtual Anchors

An annotation document node can reference a region with a link; annotations reference the element they belong to. Figure 24 reproduces a part-of-speech annotation from the Penn Treebank project of the region denoting ‘website‘:

```
<node xml:id="ptb-n00198">
  <link targets="seg-r198"/>
</node>
<a label="tok" ref="ptb-n00198" as="PTB">
  <fs>
    <f name="msd" value="NN"/>
```

```
    </fs>
  </a>
```

Figure 24: Part-of-Speech Annotation

Edges in annotation documents denote their source node and target node with 'from' and 'to' attributes. The node referenced by the edge attributes can also be defined in another annotation document. In this case the annotation document containing the edge depends on the annotation document containing the referenced node.

In Figure 25, 'his own website' constitutes a noun phrase. The syntactic annotation (Penn Treebank project) in Figure 25 depends on the part of speech annotation in Figure 24: The node "ptb-n00198" is annotated **"NN"** in Figure 24, and referenced as target of the first edge ("ptb-e00192").

```
<node xml:id="ptb-n00195"/>
<a label="NP" ref="ptb-n00195" as="PTB">
  <fs>
    <f name="cat" value="NP"/>
  </fs>
</a>
<edge xml:id="ptb-e00192" from="ptb-n00195" to="ptb-n00198"/>
<edge xml:id="ptb-e00191" from="ptb-n00195" to="ptb-n00197"/>
<edge xml:id="ptb-e00190" from="ptb-n00195" to="ptb-n00196"/>
```

Figure 25: Syntactic Annotation

Annotation documents can also include their own segmentation regions. Figure 26 reproduces an annotation for events produced by researchers at Carnegie-Mellon University using GATE:

```
<region xml:id="ev-r4" anchors="894 900"/>
<node xml:id="ev-n4">
  <link targets="ev-r4"/>
</node>
<a label="Setting Up" ref="ev-n4" as="xces">
  <fs>
    <f name="arg1" value="companies"/>
    <f name="arg2" value="offices"/>
  </fs>
</a>
```

Figure 26: Event Annotation

MASC structure and annotation details taken from http://www.anc.org/MASC/MASC_Structure.html.

5.1.2.4. Converters

There are plugins to GATE (http://gate.ac.uk/) for input and output of GrAF-encoded data and converters to use the data in UIMA. It will also soon be possible to use the data in NLTK (www.nltk.org).

This converters are realized in conjunction with MASC and can be found here: http://www.anc.org/MASC/Download.html

There is also a GrAF API (http://www.anc.org/graf-api/) and a subset of MASC I is also available in a CONLL (http://ifarm.nl/signll/conll/) format.

5.1.2.5. Documentation and Use

(Ide and Suderman, 2007)

Projects: MASC (http://www.anc.org/MASC/Home.html)

5.2. Standards for morphological annotation

Authors: Gertrud Faaß and Kerstin Eckart

5.2.1. Morpho-Syntactic Annotation Framework (MAF)

MAF, the morpho-syntactic annotation framework, is an upcoming ISO standard. It is developed within ISO, technical committee 37 (ISO/TC 37/SC 4), under the title Language resource management – Morpho-syntactic annotation framework.

MAF is related to other (upcoming) standards developed within ISO/TC37, such as the TC47/SC3 lSO 12 620 Computer applications in terminology – Data categories – Data category registry, and the TC37/SC3 ISO 16642, Computer applications in terminology – TMF (Terminological Markup Framework). Regarding the SC4 group, ISO/DIS 24610-1 Language Resource Management – Feature Structures – Part 1: Feature Structure Representation is considered. The recommendations of the OLAC metadata set are to be considered when annotating metadata.

5.2.1.1. Status

MAF is a standard under development. The last version for voting was published in July, 2008 (ISO/DIS 24611). Voting ended in December 2008 and members of the committee handed in a number of comments. A newer version is currently being developed, therefore our report should be considered to be preliminary.

5.2.1.2. Aspects

MAF provides a reference format for the representation of morpho-syntactic annotations, however, currently it focuses on the encoding of parts of speech and other grammatically relevant data on the level of "word". The draft contains descriptions for the encoding of `<token>`: an element resulting from a word segmentation process and an - optional - post-processing step of abstraction (e.g. transcriptions and transliterations). The second layer of description foreseen is that of `<wordForm>`, which is built of tokens. This element may contain additional morphological information; it may also refer to a lexicon. Structural ambiguities in compounding can be represented as DAGs ("lattices"). The third major element in MAF, `<tagset>`, describes sets of morpho-syntactic labels. Such a set can be linked directly with the isocat Data category registry. Figure 1 shows a (simplified) view of the MAF model (Clément and Villemonte de la Clergerie, 2005). Note that MAF allows for standoff and inline annotations.

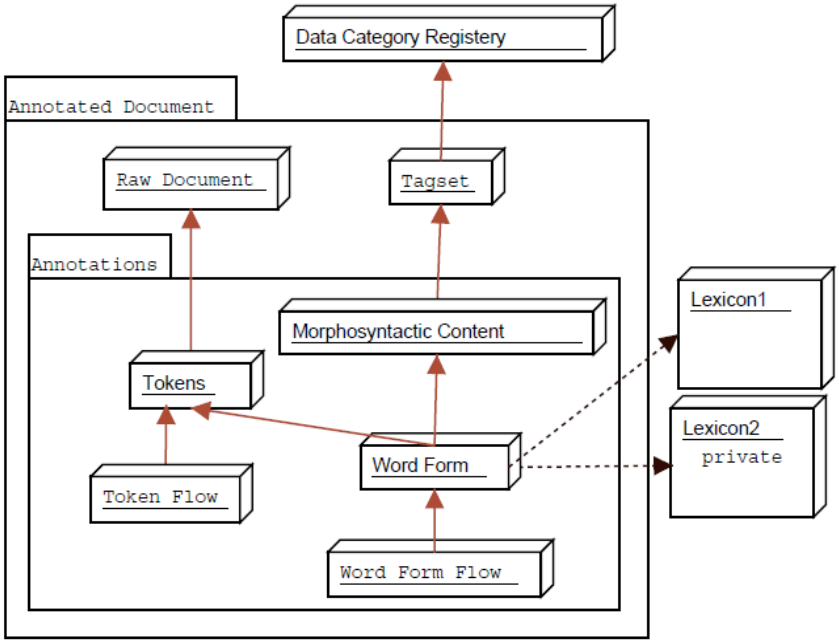


Figure 27: MAF

5.2.1.3. Serialization

The draft includes examples and formal descriptions of a XML-serialization. In the version of 2008, three major elements, `<token>`, `<wordform>` and `<tagset>`, are `forseen`. Tokens may be described by either explicitly defining the begin-end position in the text (as foreseen by LAF (ISO/DIS 24612)), or by the "join" information (left, right, both, see example below). Tokens are identified by an id.

The element wordForm is obviously to describe linguistic words, including compounds and multiword units. One wordForm element can entail one or several tokens (several wordForms may also be described by one token, e.g. compounds; the element is thus described to be recursive). The formal definition allows for theoretical elements that cannot be assigned to specific tokens (to be represented with an empty string as token id).

The objective in describing the `<tagset>` element is to allow the users to refer to entries of the Data category registry, recommended by the ISO 12620 proposal. By making use of the `<tagset>` element, a subset of these data categories (dcs) is assigned for annotating a certain document. However, as not all definitions may be useable as they are, MAF allows for defining a relation between the local and the registered tagset (by utilizing the attribute @rel in the *dcs* description): "eq" describing equality, "subs" describes a subsuming relation, "gen" a generalization. Other relations can be described, too. If it is not possible to assign a category to a registered one, a textual description can be given. The elements fvLib and fLib are foreseen so that libraries making use of feature elements can be utilized.

5.2.1.4. Examples

A floating text may contain the sequence "the prime minister". It can be represented in MAF as follows (the element "entry" refers to an external lexicon):

```
<token form=”the” id=”t0”>the</token>
<token form=”prime” id=”t1”>
<token form=”minister” id=”t2>
<wordForm lemma=”the” tokens=”t0”/>
<wordForm lemma=”prime_minister” entry=”urn:lexicon:en:prime_minister”
tokens=”t1 t2”/>
```

Figure 28: MAF annotation example 1

Using embedding notation, it is possible to specify how tokens are joined (text sequence: "L'on dit"):

join left:

```
<token id=”t1”>L’</token>
<token id=”t2”
join=”left”>on</token>
<token id=>”t3”>dit</token>
```

alternative: describe separator as token:

```
<token id=”t1”>L</token>
<token id=”t2”
join=”both”>’</token>
<token id=”t3”>on</token>
<token id=>”t4”>dit</token>
```

Figure 29: MAF annotation example 2

When encoding links to a *data category registry*, a <tagset> entry can e.g. contain the following entries (here: encoding of the local morpho-syntactic category “masc” as referring to the *dcs* category “masculine”, PID 246):

```
<tagset
…
<dcs local=”masc”
  registered=”http://www.isocat.org/datcat/DC-246”
rel=”eq”/>
</tagset>
```

Figure 30: MAF annotation example 3

5.2.1.5. Summary and Comments

In summary, the 2008-Version of MAF can represent metadata that is usually provided by Part-of-Speech taggers. In terms of definitions, it does not really fit in what has been described before, for example, a `<wordForm>` which is used for lemmatizable units, is defined very generally as a “morpho-syntactic unit” though such a definition would describe morphemic (sub-word) elements as well. The MAF draft contains descriptions for morphological terms (e.g. morpheme, morphological pattern, morphology), but does not (yet) make use of them. The element <token> on the other hand is described as being “identified by a morpho-phonological analysis” (which would be expected to be a morpheme), though it only represents surface elements of the text.

MAF does not take ISO 24614-1:2010 into account, though this ISO standard on Word segmentation explicitly refers to MAF.

We are in the process of suggesting the adaption of the MAF standard to what was foreseen by the ISO 24614-1:2010 standard on word segmentation, i.e. a more narrow definition of the elements `<token>` and `<wordForm>` which is harmonized with the foreseen use. Secondly, we are in the process of preparing a suggestion towards an extension of the standard so that it will contain an additional element `<morph>` describing morphemic units building linguistic words, so that MAF will be capable of representing the output of morpho-phonological analysing processes, too.

Regarding the XML-serialization, <wordForm> is not yet identifiable by an id, it is thus problematic to get access to a specific <wordForm> element from ”outside” (e.g. from a SynAF representation).

5.2.1.6. Documentation and Use

(Clément and Villemonte de la Clergerie, 2005)
Projects: A project group from D-Spin will provide the German fairy-tale "Das Rotkäppchen" with TEI, MAF and SynAF documents, as well as annotation of Propp's narrative functions.

5.3. Standards for syntactic annotation

5.3.1. Syntactic Annotation Framework (SynAF)

Author: Kerstin Eckart

SynAF, the Syntactic Annotation Framework, is an ISO standard. It is developed within ISO, technical committee 37 (ISO/TC 37/SC 4), under the title: Language resource management - Syntactic annotation framework (SynAF).

SynAF is related to other (upcoming) standards developed within ISO/TC 37 such as the Linguistic annotation framework LAF (ISO/DIS 24612), the Morphosyntactic Annotation Framework MAF (ISO/DIS 24611), the Lexical markup framework LMF (ISO 24613:2008) and the Specification of data categories and management of a Data Category Registry for language resources (ISO 12620:2009).

5.3.1.1. Status

SynAF, the Syntactic Annotation Framework is described by an ISO standard document (ISO 24615:2010).

5.3.1.2. Aspects

SynAF provides a metamodel to represent annotations on the syntactic layer. The data structures are graph-based, and therefore include terminal nodes, inner nodes (also known as non-terminal nodes), and edges.

Like LAF, SynAF aims at interoperability with as many existing annotation formats as possible, but as SynAF focuses on syntactic annotations, it is more specific than LAF. Nevertheless it is also designed to be flexible, and different kinds of syntactic annotations can be represented (generic examples: dependency annotations as well as constituency annotations).

Figure 31 shows the SynAF metamodel and its relationship to the layer of morphosyntactic annotations.

Throughout the LAF framework a separation of annotation layers is recommended, which means to keep each annotation layer in a separate document, using references to refer to other layers. For SynAF it is recommended to have references from the SynAF terminal nodes to morphosyntactic wordform elements. It is also possible to use MAF for the morphosyntactic annotation layer. Nevertheless, if a user is uncomfortable with MAF or with stand-off annotations, it is possible to include e.g. morphosyntactic annotations as well as the wordforms themselves in the terminal nodes or to refer to other documents like e.g. a TEI document as segmentation layer.

In SynAF, nodes as well as edges can be annotated, and the annotations can be related to with the data categories in the syntax profile of the ISOcat inventory, as defined in ISO 12620:2009 (Terminology and other language and content resources – Specification of data categories and management of a Data Category Registry for language resources). The SynAF document also provides a list of relevant data categories for syntactic annotation.

5.3.1.3. Serialization

The current XML-serialization of SynAF is <tiger2/> and is based on TIGER-XML (cf. (König et al., 2003), Chapter V). TIGER-XML was chosen as the starting point for the SynAF XML-serialization as it has been used over several years and in several countries. From this starting point two main development steps are prepared to extend TIGER-XML such that mappings to and from as many different syntactic annotation formats as possible can be easily achieved. The first step has already taken place.

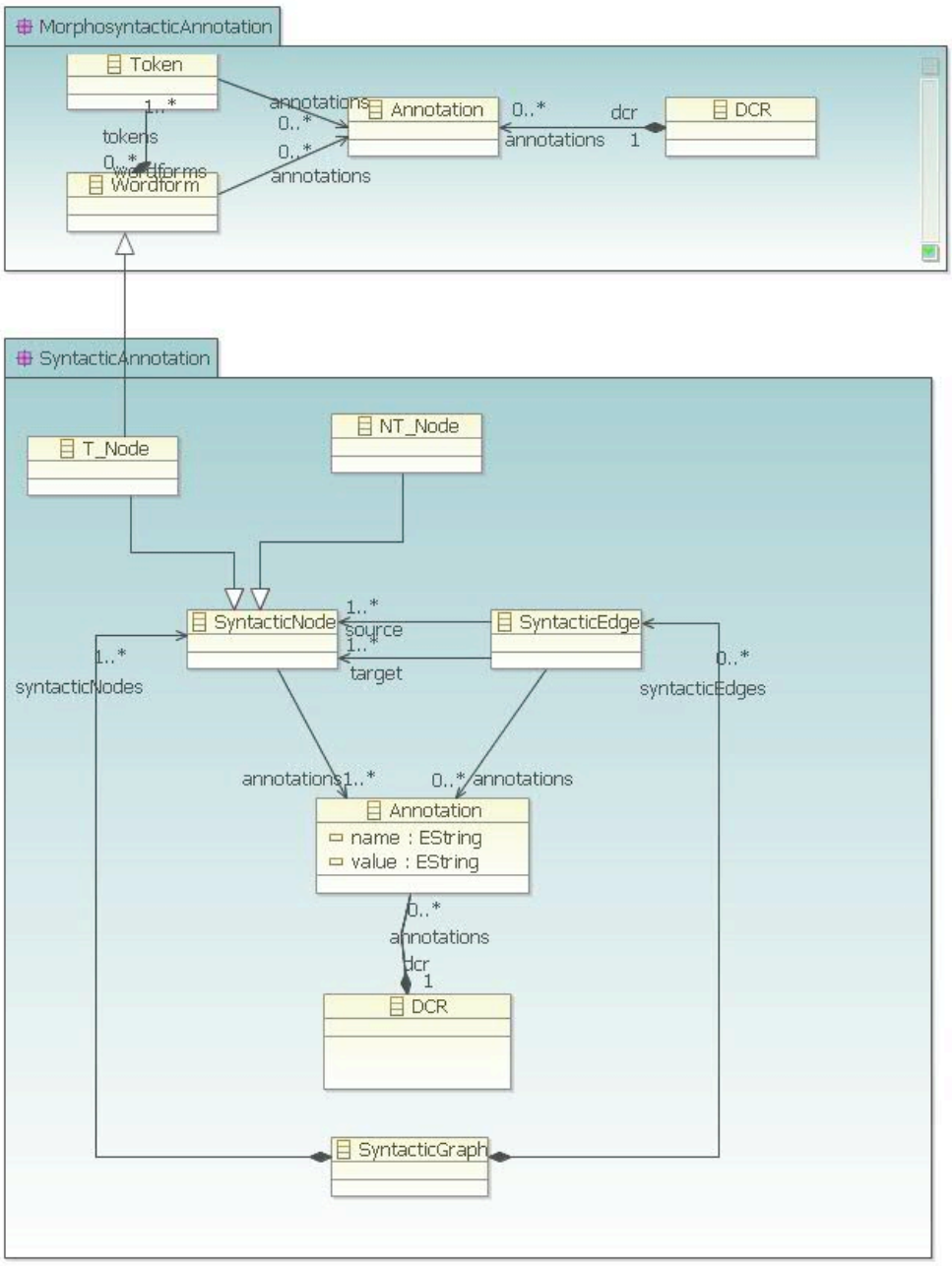


Figure 31: SynAF Metamodel

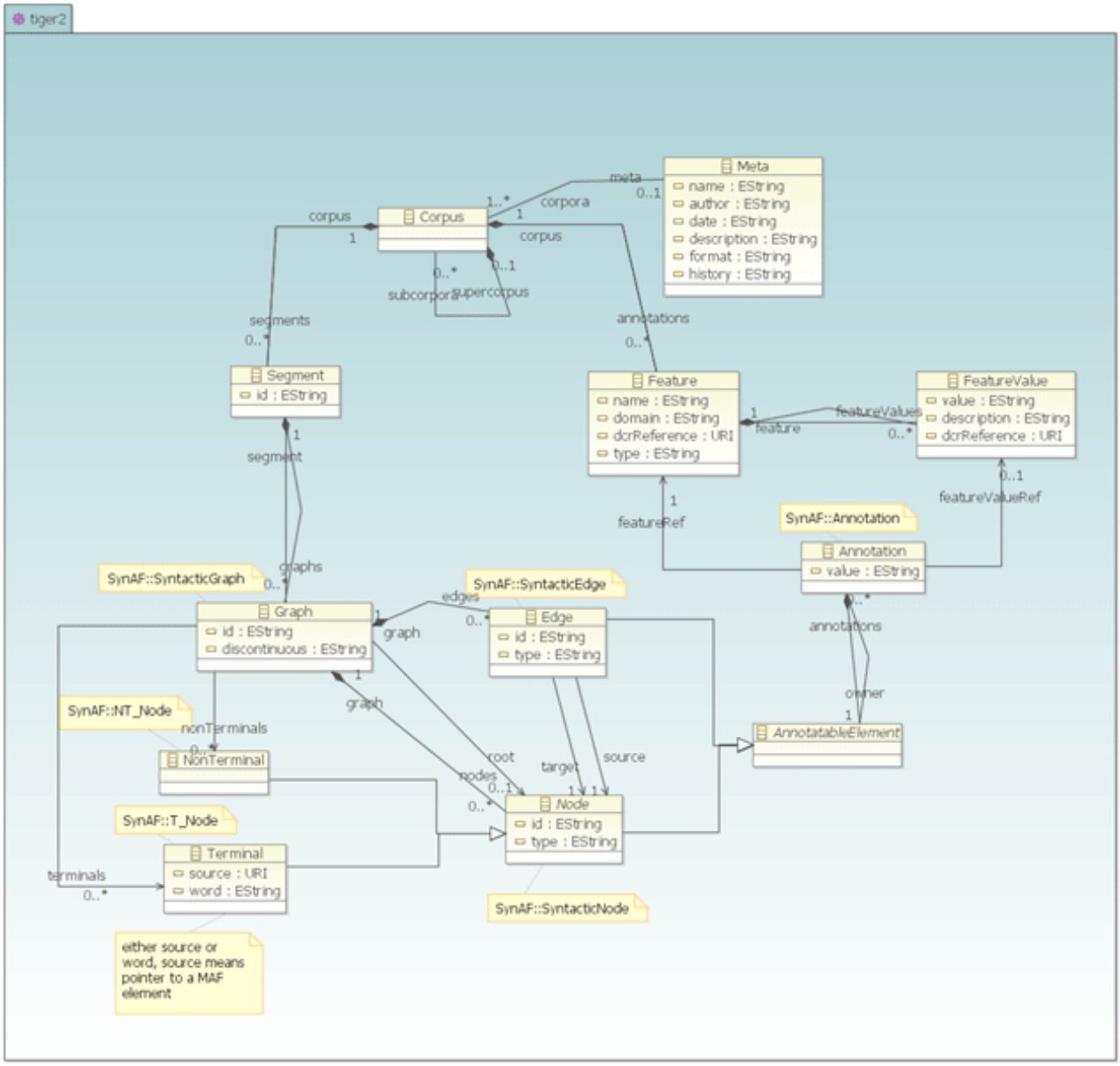


Figure 32: <tiger2/> Metamodel

The second step is in process now. In both steps care was and is taken to extend TIGER-XML as little as possible and as much as necessary.

In the first development step, small changes have been made, such as including a referencing mechanism to refer to ISOcat data categories. The user can also define different node and edge types, which allow for flexible annotations (e.g. dependency edges, complex nodes, etc.). The format has been presented i.a. to the D-SPIN partners, who added comments and changes regarding the interoperability with their own resources. <tiger2/> was released 31-08-2010.

In the second step, the experiences users will make with <tiger2/> shall be included in the development of an extended format. Some greater changes might be made on the verge of the step to this format, such as e.g. including the FSR (ISO 24610-1:2006 Feature structures – Part 1: Feature structure representation) functionality for annotation.

Although <tiger2/> is not the final format, at the end of the second step a conversion of <tiger2/> data into the extended format will also be provided, so that the second step will mean no extra work for the user. The user working with <tiger2/> can participate in the development, especially to test and ensure interoperability with his or her own resources, in sending comments to: tiger2@lists.hu-berlin.de. Such feedback will be accounted for in the final format.

To find out about the current stage of the development, visit http://korpling.german.hu-berlin.de/tiger2/.

The definition of the XML-Serialization is not part of the ISO document yet, but might become an appendix.

5.3.1.4. Excerpt of an example of the current format

See also http://korpling.german.hu-berlin.de/tiger2/ for examples.

In a MAF document – denoting the morphosyntactic layer (i.e. one layer beneath SynAF) – word forms are annotated with morphosyntactic information:

```
    <wordForm xml:id="wordForm_9" lemma="Unternehmer" tokens="t10">
      <fs>
        <f name="pos"><symbol value="NN"/></f>
        <f name="morph"><symbol value="Acc.Sg.Masc"/></f>
      </fs>
    </wordForm>
```

Figure 33: MAF morphosyntactic layer

In a <tiger2/> document, the terminal nodes of a syntactic graph can refer to the MAF word forms:

(Here, Tiger_V2_const.subtoken.maf.xml is the filename of the respective MAF document.)

```
    <terminals>
      <t xml:id="s9_8" source="Tiger_V2_const.subtoken.maf.xml#wordForm_8" /> <!-- einen -
->
      <t xml:id="s9_9" source="Tiger_V2_const.subtoken.maf.xml#wordForm_9" /> <!--
Unternehmer -->
    </terminals>
```

Figure 34: SynAF linking to MAF

There are also non-terminal nodes of the graph in this example: “einen Unternehmer“ constitutes a noun phrase:

```
    <nonterminals>
      <nt xml:id="s9_501" cat="NP">
        <edge type="prim" func="NK" target="#s9_8" />
        <edge type="prim" func="NK" target="#s9_9" />
      </nt>
    </nonterminals>
```

Figure 35: Non-terminal nodes

Edges are embedded beneath their source node. The target node is given by a reference (cf. Figure 35)

Annotation sets are given in a separate annotation section. There, references to ISOcat data categories can be included:

```
<annotations>
  <feature xml:id="f2" name="pos" domain="t" dcr:datcat="http://www.isocat.org/datcat/DC-396">
    <value xml:id="f2_1" name="PP" dcr:datcat="http://www.isocat.org/datcat/DC-1463"/>
  </feature>
</annotations>
...
<terminals>
  <t id="s1_t1" word="I" pos="PP"/>
</terminals>
```

Figure 36: Reference to ISOcat

5.3.1.5. Converters

A conversion from TIGER-XML (cf. (König et al., 2003)) into <tiger2/> will be realized by a Java API.

Other converters into <tiger2/> will hopefully be created by the testing users during the second development step. There will also be a converter from <tiger2/> into the extended format of the next development step, such that it means no extra work for the user testing the interoperability of <tiger2/> with respect to his or her own resources.

5.3.1.6. Documentation and Use

Website: http://korpling.german.hu-berlin.de/tiger2/

Projects: A project group from D-Spin will provide the German fairy-tale "Das Rotkäppchen" with TEI, MAF and SynAF documents, as well as annotation of Propp's narrative functions.

5.3.2. Penn Treebank (Phrase Structure Treebank)

Authors: Kathrin Beck, Wolfgang Maier

Developers:	Computer and Information Science Department, University of Pennsylvania
URL:	http://www.cis.upenn.edu/~treebank/
Documentation:	ftp://ftp.cis.upenn.edu/pub/treebank/doc/old-bktguide.ps.gz, ftp://ftp.cis.upenn.edu/pub/treebank/doc/arpa94.ps.gz

Penn Treebank format is an annotation format for part-of-speech tagged and syntactically parsed corpora. It was developed at the University of Pennsylvania for the Penn Treebank.

Syntactic dependencies of tree structures are realized by hierarchical “bracketing” of words and phrases. Due to that ordering principle, analysis is essentially context-free, and non-contiguous structures and dependencies are not possible.

```
( (S
    (NP-SBJ
      (NP (NNP Pierre) (NNP Vinken) )
      (, ,)
      (ADJP
        (NP (CD 61) (NNS years) )
        (JJ old) )
      (, ,) )
    (VP (MD will)
      (VP (VB join)
        (NP (DT the) (NN board) )
        (PP-CLR (IN as)
          (NP (DT a) (JJ nonexecutive) (NN director) ))
        (NP-TMP (NNP Nov.) (CD 29) )))
    (. .) ))
```

Figure 37: Penn Treebank format; Treebank I bracketing format; first sentence of the Penn Treebank

5.3.2.1. Annotation format

Penn Treebank corpus files do not contain any metadata but only consist of a series of annotated sentences. There is no formal specification how tag sets etc. should be published.

Bracketed structures can be arbitrarily complex, and one bracketed structure can nest within another.

Brackets are labeled with their syntactical category; every bracket has exactly one label (except the bracket which surrounds the entire sentence, it has no label). Phrases can contain an unlimited number of elements; its head element is not marked explicitly. Penn Treebank uses symbols for different kinds of null elements. A format example of the Penn Treebank is presented in Figure 37.

5.3.2.2. Bracketing formats of Treebank I and II

Treebank I bracketing was used until 1992. In 1994, guidelines for a refined bracketing format were published. Wishes were to explicitly provide some form of predicate-argument structure, to provide richer annotation forms and to mark non-contiguous structures (Marcus et al. 1994, http://www.ldc.upenn.edu/Catalog/docs/LDC95T7/arpa94.html).

Changes to the new format are:

- Unified analysis of predicate-argument structure (copular “be” is treated as a main verb with predicate complements)
- Distinction between arguments and adjuncts is abandoned. Instead, a small set of clearly distinguishable roles is labeled. Each constituent can have at least one label but as many as four tags, including numerical indices.
- Null elements are co-indexed with the lexical material they stand for by suffixing integers to their syntactic label.

- In the first bracketing format, discontinuous constituents are either trapped in the deeper or in the higher phrase. Treebank II format solves the "trapping" problem with null elements and index numbers.
- Gapping is solved with a template notation.

If only the Penn Treebank file format is used and own annotation standards are created, the only difference between formats I and II lies in the number of constituent labels and the number of tags allowed.

5.3.2.3. Software

tgrep, tgrep2, tregex/tsurgeon

Developers: Richard Pito (tgrep), Dough Rohde (tgrep2), Roger Levy and Galen Andrew (tregex), Roger Levy (tsurgeon), Anna Rafferty (tsurgeon/tregex GUI)

URL: http://tedlab.mit.edu/~dr/Tgrep2/

http://nlp.stanford.edu/software/tregex.shtml

http://nlp.stanford.edu/software/tsurgeon.shtml

The tgrep/tregp2/tregex ("grep for trees") tools offer a syntax for querying collections of trees available in Penn Treebank annotation format. The oldest of the tools is tgrep (before 1995), which was included in the Penn Treebank II release. It might not run on current systems. tgrep2 is an extension of grep which should still run on current Linux systems. tregex is a further extension, written in Java, which is actively supported. In its current release (1.4) it includes the tsurgeon utility for tree transformation.

TIGERSearch

Developers: Esther König, Holger Voormann, Wolfgang Lezius, University of Stuttgart

URL: http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/oldindex.shtml

TIGERSearch is a software for exploring annotated corpora. Corpus queries can be submitted via a graphical interface, where nodes can be linked and enriched with linguistic properties. Aditionally, there is an XML-based interface with with its own query language. It allows variables and is adapted at first order predicate logic, but without all-quantification.

Graphs can be viewed and exported as a picture and as XML-file.

5.3.2.4. Converters

Corpora encoded in the Penn Treebank annotation format or in a simpler bracketing format can be fed into TIGERSearch and exported from there into TIGER-XML format (http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus/, refer to SynAF (section 5.3.1) which is an adaptation of TIGER-XML).

5.3.3. NeGra Format (Phrase Structure Treebank)

Author: Kathrin Beck

Developers: Thorsten Brants, Roland Hendriks, Sabine Kramp, Brigitte Krenn, Cordula Preis, Wojciech Skut, and Hans Uszkoreit, Saarland University

URL: http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/negra-corpus.html

Documentation: http://www.coli.uni-sb.de/~thorsten/publications/Brants-CLAUS98.ps.gz

The NeGra Export annotation schema for annotated corpora was developed during the construction of the NeGra corpus (http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/negra-corpus.htmlhttp://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/negra-corpus.html) in the research project "Nebenläufige grammatische Verarbeitung" at Saarland University. The final version was published in 1997.

NeGra Export is a line-oriented and ASCII-based format. Nodes are separated by line breaks; properties of nodes and their edges are separated by tabs. Annotation is organized in-line.

This kind of format has several advantages:

- The format is both easy to read by humans and very efficient to process.
- NEGRA format is easy to parse and to convert into other formats.
- NEGRA format is supported by the annotation tool @nnotate for editing, visualizing and rudimentary search.
- NEGRA format is supported by the corpus viewer TIGERSearch for visualization, both easy and sophisticated search and for export of sentence trees.
- By importing a NeGra-format file into those tools, the corpus format can be checked on correctness (code table problems, missing tags, etc).

Three of the larger syntactically annotated corpora of German are available in NeGra Export format: NeGra corpus (http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/negra-corpus.html), TIGERCorpus (http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus/), and TüBa-D/Z (http://www.sfs.uni-tuebingen.de/en/tuebadz.shtml, with some more corpora available in NeGra Export: http://www.sfs.uni-tuebingen.de/en/corpora.shtml).

5.3.3.1. Format

There exist two versions of the NeGra Export format, format 3 and format 4, with "3" and "4" referring to the number of features in the terminal layer.

In the creation of a corpus file, this leads to differences at three points:

- The format version is specified in the first uncommented line of the corpus file (see Figure 38).
- Between the initial tables of and the sentence level, the used annotation layers are specified (see Figure 39 and Figure 40).
- The terminal nodes, i.e. words, contain three or four layers of information.

The corpus file starts with a "header" containing the format specification, a list of the sentence origins, a list of the corpus editors and tables of the used word tags, morphological tags, node tags, edge tags, and secondary edge tags. Elements in tables are enumerated; tags as well as their definitions can be defined according to individual purposes (see Figure 38).

```
%% database tuebaaktuell (corpus TUEBADZaktuell)
%%
#FORMAT 4
#BOT ORIGIN
0     T990507.2
1     T990507.3
...
#EOT ORIGIN
#BOT EDITOR
-1    __-1        <Automatisch>
0     __0         <Nicht_zugeordnet>
1     kuebler           Sandra
...
#EOT EDITOR
#BOT WORDTAG
-1    UNKNOWN     N     <unbekanntes POS-Tag>
0     --    N     <nicht zugeordnet>
1     $,    N     Komma
2     $.    N     satzbeendende Interpunktion
...
#EOT WORDTAG
#BOT MORPHTAG
-1    UNKNOWN     <unbekannte morphologische Analyse>
0     --    ohne Morphologie
1     *     ambig
2     a     akk.
...
#EOT MORPHTAG
#BOT NODETAG
-1    UNKNOWN     <unbekanntes Knotenlabel>
0     --    <nicht gebunden>
1     ADJX  Adjektivphrase
2     ADVX  Adverbialphrase
...
#EOT NODETAG
#BOT EDGETAG
-1    UNKNOWN     <unbekanntes Kantenlabel>
0     --    <nicht gebunden>
1     -     Nicht-Kopf (non-head)
2     APP   Apposition
…
#EOT EDGETAG
#BOT SECEDGETAG
-1    UNKNOWN     <unbekanntes sekundäres Kantenlabel>
0     --    <nicht gebunden>
1     EN    phraseninterne Relation zwischen zwei Teilen eines Eigenname
2     refcontr    Dependenzrelation zwischen Kontrollverb und Komplement
...
#EOT SECEDGETAG
```

Figure 38: Table definitions of the TüBa-D/Z. Entries are replaced by "…"

Between the corpus tables and the sentence level, a text line defines the order of the used annotated properties.

The stored corpus is divided into **sentences**. Metadata on the individual sentences are stored in the beginning-of-sentence line BOS.

The first sentence of the TüBa-D/Z corpus in Figure 39 contains the following information, coded in a list of numbers:

`#BOS 1`: beginning of sentence no. 1

`3`: the sentence was last annotated by editor no. 3 (confer to the list of editor at the head of the corpus file)

`1202391857`: this encodes the time of the last modification: 07/02/08, 14:44:17

`0`: the origin of the sentence is referenced to in the BOT table no. 0: `T990507.2`, referring to the newspaper article no. 2 of the edition of 07/05/1999 of the newspaper *tageszeitung* (taz).

`%%`: separates manual comments from the automated metadata

`HEADLINE`: manual mark of the headlines of newspaper articles

The corpus annotation is stored sentence-wise between the BOS and EOS (end-of-sentence) tags.

Words are implicitly numbered starting with 1, nodes are numbered starting with 500.

The word “Veruntreute” in Figure 39 contains the following information:

`word`: Veruntreute (the word layer of the corpus)

`tag`: VVFIN (finite full verb)

`morph`: 3sit (3rd person singular indicative past tense)

`edge`: HD (head relation)

`parent`: 500 (head of the word edge is the first non-terminal node = 500)

`secedge`: no secondary edge

`comment`: no comment; this layer is used in TüBa-D/Z for misspelled words

The list of non-terminals following the list of words is defined the same way referring from one non-terminal to the next up to the root non-terminal no. 0.

```
%% word           tag   morph       edge  parent      secedge     comment
#BOS 1 3 1202391857 0 %% HEADLINE
Veruntreute       VVFIN 3sit        HD    500
die               ART   nsf         -     504
AWO               NN    nsf         -     501
Spendengeld       NN    asn         HD    502
?                 $.    --          --    0
#500              VXFIN --          HD    503
#501              EN-ADD      --    HD    504
#502              NX    --          OA    505
#503              LK    --          -     506
#504              NX    --          ON    505
#505              MF    --          -     506
#506              SIMPX --          --    0
#EOS 1
#BOS 2
…
#EOS …
```

Figure 39: NeGra Export format 3; first sentence of the TüBa-D/Z (English: Did the AWO misappropriate donations?)

Figure 40 shows an example for the NeGra Export format 4 with the lemma layer as fourth terminal node layer.

```
%% word           lemma              tag    morph        edge  parent
      secedge comment
#BOS 1 0 1098266456 1 %% @SB2AV@
``                --                 $(     --           --    0
Ross              Ross               NE     Nom.Sg.Masc PNC    500
Perot             Perot              NE     Nom.Sg.Masc PNC    500
wäre              sein               VAFIN 3.Sg.Past.Subj     HD    502
vielleicht        vielleicht         ADV    --           MO    502
ein               ein                ART    Nom.Sg.Masc NK     501
prächtiger        prächtig                  ADJA  Pos.Nom.Sg.Masc   NK    501
Diktator                Diktator                    NN    Nom.Sg.Masc NK    501
''                --                 $(     --           --    0
#500              --                 PN     --           SB    502
#501              --                 NP     --           PD    502
#502              --                 S      --           --    0
#EOS 1
#BOS 2
…
#EOS …
```

Figure 40: NeGra Export format 4; first sentence of the TIGERCorpus (English: "Perhaps Ross Perot would be a magnificent dictator")

5.3.3.2. Software

There are two tools that are designed explicitly for the NeGra Export format, @nnotate and TIGERSearch.

@nnotate

Developers: Oliver Plaehn, Tania Avgustinova, Saarlandy University

URL: http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/annotate.html, http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/negra-corpus.html

@nnotate is a tool for the semi-automatic annotation of corpus data. It offers a graphical user interface for efficient sentences-wise annotation. Corpora are organized via a mySQL database.

For semi-automatic annotation, it has the underlying Part-of-Speech tagger "TnT" for parts-of-speech, grammatical functions and phrasal categories, a statistical parser based on Cascaded Markov Models and the NP-chunker "Chunkie" for noun phrases and prepositional phrases.

@nnotate software is not supported anymore. With some effort of providing obsolete libraries it still runs on Linux.

TIGERSearch

Developers: Esther König, Holger Voormann, Wolfgang Lezius, University of Stuttgart

URL: http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/oldindex.shtml

TIGERSearch supports the NeGra format. For details on the software, see Section 5.3.2.3

5.3.3.3. Converters

Corpora encoded in Negra Export format can be fed into TIGERSearch and exported from there into TIGER-XML format (http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus/, refer to SynAF (section 5.3.1) which is an adaptation of TIGER-XML). The only detriment is that the "comments" layer of NeGra Export format is neither displayed in TIGERSearch, nor is it exported into TIGER-XML.

The SaltNPepper framework (http://korpling.german.hu-berlin.de/trac/saltnpepper) provides converters from TIGER-XML into EXMARaLDA, PAULA, Treetagger, relANNIS, and GrAF.

5.3.4. Prague Markup Language (Dependency Structure Treebank)

Authors: Petr Pajas, Jan Stepanek, Pavel Stranak

5.3.4.1. Introduction

Prague Markup Language (PML) is a format designed to capture all current types of linguistic annotation and represent them in a standardized and unified form. Specifically, it can capture various kinds of morphological and both dependency and phrase structure syntactic annotation. Moreover, many types of additional information can be represented: coreference, valency, discourse relations, etc. An interesting distinction of PML is that it has been successfully implemented on an extensive set of treebanks (see page 98). PML was originally created as the main data format for the Prague Dependency Treebank 2.0 (PDT).

PML is an XML language aimed to provide a better and unifying alternative to various legacy formats used in various areas of corpus linguistics and specifically in the field of structured annotation.

PML conforms to the following important requirements for modern data formats:

- **Stand-off annotation principles:** Each layer of the linguistic data is cleanly separated from the other annotation layers as well as from the original data. This allows for making changes only to a particular layer without affecting the other parts of the annotation and data.
- **Cross-referencing and linking:** Both links to external document and data resources and links within a document are represented coherently. Diverse flexible types of external links are required by the stand-off approach. Supposed that most data resources (data, tag-sets, and dictionaries) use the same principles, they can be more tightly interconnected.
- **Linearity and structure:** The data format is able to capture both linear and structure types of annotation. Linear type includes e.g. word and sentence order (in case of written text) or temporal information (in case of speech data). As for the structural annotation, our primary concern was to allow capturing tree-like structures in a way that mirrors their logical nesting.
- **Structured attributes:** The representation allows for associating the annotated units with complex and descriptive data structures, similar to feature-structures.
- **Alternatives:** The vague nature of the language often leads to more than one linguistic interpretation and hence to alternative annotations. This phenomenon occurs on many levels, from atomic values to compound parts of the annotation, and is treated in a unified manner.
- **Human-readability:** The format is relatively human-readable. This is very useful not only in the first phases of the annotation process, when the tools are not yet mature enough to

reflect all evolving aspects of the annotation, but also later, especially for emergency situations when e.g. an unexpected data corruption occurs that breaks the tools and can only be repaired manually. It also helps the programmers while creating and debugging new tools.

- **Extensibility:** The format is extensible to allow new data types, link types, and similar properties to be added. The same applies to all specific annotation formats derived from the general one, so that one can incrementally extend the vocabulary with markup for additional information.
- **XML based:** Employing a commonly used generic markup language as the underlying data representation made achieving the above mentioned goals much easier. XML is widely deployed and offers many tools and libraries for various programming languages already. This also means, that existing validation tools and schema languages for XML can be applied on the PML format, too.

5.3.4.2. PML Schemata, Data Types and Roles

In PML, individual layers of annotation can be stacked one over another in a stand-off fashion and linked from in a consistent way. Each layer of annotation is described in a *PML Schema* file, which can be imagined as a formalization of an abstract annotation scheme for the particular layer of annotation. In brief, the PML Schema describes which elements occur on the layer, how they are nested and structured, of which types the values occurring in them are, and what role they play in the annotation scheme. This *role* information can also be used by applications such as TrEd (see below) to determine an adequate way to present a PML instance to the user. Based on a PML Schema, it is possible to generate various validation schemata, such as RelaxNG (http://www.relaxng.org), hence formal consistency of instances of the PML schema can be verified using conventional XML-oriented tools.

PML format offers unified representations for the most common annotation constructs, such as

- **Attribute-value structures**, i.e. structures consisting of attribute-value pairs.
- **Lists**, allowing several values of the same type to be aggregated in either an ordered sequence or an unordered list.
- **Sequences**, representing sequences of values of different types and also providing rudimentary support for XML-like mixed content.
- **Alternatives**, used for aggregating alternative annotations, ambiguity, etc.
- **References**, providing a unified method for cross-referencing within a PML instance and linking both among various PML instances (which includes links between layers of annotation) and to other external resources (in the present revision, these resources have to be XML-based).

As already briefly mentioned, PML introduces a concept of so called *PML-roles*, which is orthogonal to the concept of data typing. The information provided by PML roles identifies a construct as a bearer of additional higher-level property of the annotation, such as being a node of a dependency tree, being a unique identifier, etc.

It is common practice to derive an annotation schema for a new corpus (or treebank) from an existing one. PML allows to derive a PML Schema for a new corpus from an existing PML Schema in a similar fashion. For instance, the format of the Chinese Treebank 7.0 is based on the Penn Treebank

format; thus after conversion to PML, its PML Schema is also derived from the PML Schema of the Penn Treebank. In this case, the resulting schema is about half the size.

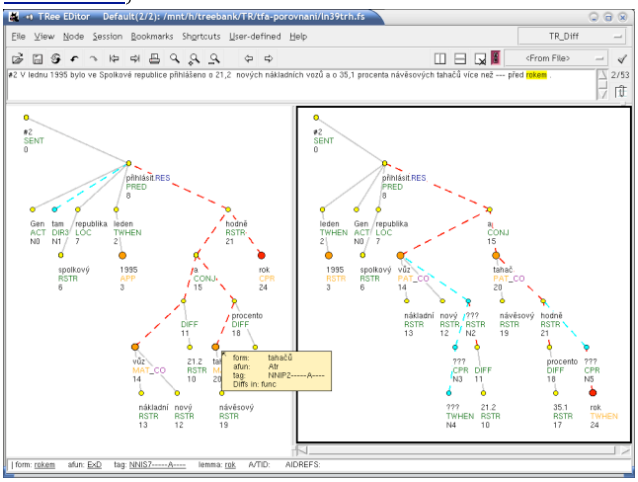
Both the PML data and the PML Schemata can be easily validated using Relax NG. There is a single Relax NG schema that can validate any PML Schema. For data, there is a tool to convert the corresponding PML Schema to Relax NG schema (via XSLT) and validate the data with it.

There is an inherent support for versioning of the PML standard itself and optional versioning of PML Schemata. Therefore, the PML Schema can require a particular version of the PML standard and/or a revision of PML Schemata it imports.

5.3.4.3. Applications (TrEd + extensions, MEd, PML-TQ)

TrEd

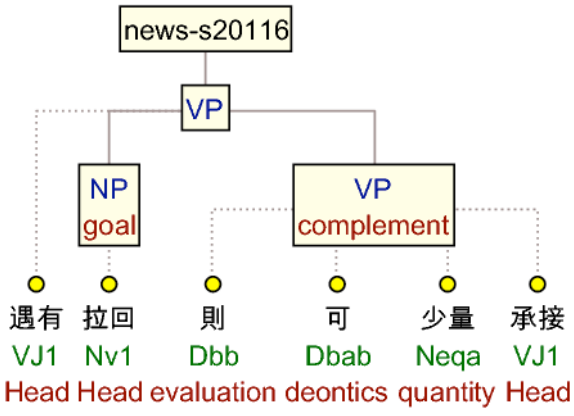
TrEd is a fully customizable and programmable graphical editor and viewer of tree-like structures such as dependency or phrase structure trees. Among other projects, it was used as the main annotation tool for syntactical and tectogrammatical annotations of The Prague Dependency Treebank, as well as for decision-tree based morphological annotation of The Prague Arabic Dependency Treebank.



TrEd's default behavior can be customized (extended or narrowed) using so called annotation modes (previously called binding contexts) that can provide different interfaces for different types of data or annotation. An annotation mode usually consists of a set of macros (subroutines programmed in Perl) that can be bound to keyboard shortcuts, menus, toolbars, or certain automatically triggered events. TrEd comes with an extensive library of pre-defined macros, which greatly simplifies writing new annotation modes.

TrEd extensions

TrEd extension packages are used to package related annotation modes (macros and Perl modules they use), stylesheets, resources (e.g. PML schemas), XSLT transformations, etc. to TrEd users. Extension pacakges are deployed using package repositories on the web. The users can install extensions from TrEd by adding the URL to the repository to a list of repositories using a Repository Manager accessible from the Extension Manager and selecting the extensions of their choice. Extensions can depend on other extensions and the TrEd installer attempts to resolve the dependencies automatically. There are currently 36 extensions in the default public



Sinica Treebank

repository at http://ufal.mff.cuni.cz/~pajas/tred/extensions/.

Among other types, there are extensions providing support for data formats, other annotation modes, parallel treebank alignment, parser, etc.

PML-TQ

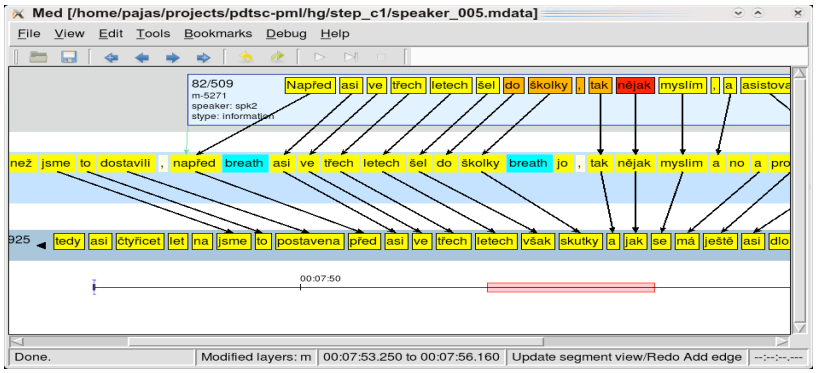
PML Tree Query (PML-TQ) is a query language and search engine targeted for querying multi-layer annotated treebanks stored in the PML data format. It can be used to query all kinds of treebanks: dependency, constituency, multi-layered, parallel treebanks, as well as other kinds of richly structured types of annotation.

The query language is declarative and offers both textual and graphical representation of queries. Currently, there are two implementations of the query engine, one based on a relational database (Oracle or PostgreSQL >= 8.4), the other based on Perl and the TrEd toolkit. Three user interfaces are available: a web-based interface for the database-based query engine displaying query results as SVG, a full-featured graphical user interface for both engines available as a plug-in to the tree editor TrEd, and a text-only command-line interface.

A PML-TQ server with many converted treebanks is run at the Institute of Formal and Applied Linguistics at Charles University in Prague.

MEd

MEd is an annotation tool in which linearly-structured annotations of text or audio data in can be created and edited. The tool supports multiple stacked layers of annotations that can be interconnected by links. MEd can also be used for other purposes, such as word-to-word alignment of parallel corpora.



MEd is the main annotation tool for the project Prague Dependency Treebank of Spoken Czech (PDTSC).

5.3.4.4. Supported Formats, Convertors

To convert data from any format to the PML format (or vice versa) is relatively easy, given the format is well documented. Several existing formats are already supported, see the list below.

TrEd supports PML data directly, but can also work with data in any format, provided there is a conversion script that converts the data on the fly. The conversion program can be written in Perl, shell, XSLT, or other languages.

Currently Supported Formats

The currently supported formats are PDT 2.0, Valency Lexicons, Hyderabad (see example below), Prague Arabic Dependency Treebank, Slovene Dependency, PCEDT, CoNLL 2009 ST (2006, 2007), Penn (Chinese), Tiger, Penn Arabic, Sinica Treebank (see picture above).

```
<Sentence id="8">
...
3       ((      NP      <af=maMxi,n,,pl,,d,0,0/drel=nmod:2>
3.1     eVMwo   INTF    <af=eVMwo,avy,,,,,0,0_avy>
3.2     maMxi   CL      <af=maMxi,n,,pl,,d,0,0>
        ))
...
</Sentence>
```

The original format of the Telugu Treebank:

```
<chunk id="tec22">
   <drel>nmod</drel>
   <feats>
      <lemma>ముంద</lemma>
      <wxlemma>maMxi</wxlemma>
      <pos>n</pos>
      <n>pl</n>
      <c>d</c>
      <v>0</v>
      <t>0</t>
   </feats>
```

```
   <ord>5</ord>
   <phrase>NP</phrase>
   <children>
      <word id="tew42">
         <feats>
            <lemma>ఎంత</lemma>
            <wxlemma>eVMwo</wxlemma>
            <pos>avy</pos>
            <v>0</v>
            <t>0_avy</t>
         </feats>
         <form>ఎంత</form>
         <ord>6</ord>
         <phrase>INTF</phrase>
         <wxform>eVMwo</wxform>
      </word>
      <word id="tew43">
         <feats>
            <lemma>మంద</lemma>
            <wxlemma>maMxi</wxlemma>
            <pos>n</pos>
            <n>pl</n>
            <c>d</c>
            <v>0</v>
            <t>0</t>
         </feats>
         <form>మంద</form>
         <ord>7</ord>
         <phrase>CL</phrase>
         <wxform>maMxi</wxform>
      </word>
   </children>
</chunk>
```

Figure 41: Telugu treebank example

The Telugu Treebank converted from the original format into PML. The forms and lemmas were generated by decoding the ‘wxlemma’ and ‘wxform’.

5.3.5. Kyoto Annotation Format (KAF)

Author: Monica Monachini

KAF is a layered annotation format, based on XML. The annotation is stand-off, meaning that the original source document remains unchanged and is kept read-only. KAF provides annotation layers for basic natural language processing and is open to extensions with other annotation layers needed by specific applications, which may be standardized later on. If a process adds information which cannot be held by existing layers, a layer of annotation is added. Layers may be linked by means of references from one layer to items in another (lower level) layer. Any previous layers remain intact and can still be used by other processes.

KAF is compatible with LAF (ISO Linguistic Annotation Framework) but imposes a more specific standardization of the annotation format itself.

KAF can be seen as a multi-layer format for text annotation: the first two layers, explicitly dedicated to representing morphosyntactic and syntactic information, are inspired by MAF and SynAF and are implemented “over” the semantic layer. For semantic annotation, the ISO community provides

SemAF which is especially dedicated to the representation of events and time. We decided to boost semantic annotation and devised a dialect of the ISO standards, where semantic notation is tailored to the specific purposes of the project. KAF layers are to be seen as dialects of the ISO standards, yet maintaining (different degrees of) mappability to them. The KYOTO dialects do not corrupt the compliance with ISO standards and their underlying philosophy; instead, they are in line with the strategy in ISO which provides high-level models (meta-models) able to be adapted, tailored and implemented according to specific needs.

KAF provides the following layers:

Administrative (external metadata) layer: KAF documents may have a header for describing information about the document, such as its original name, URI or a list of the linguistic processors which generated the KAF document. Optionally, the following information can be represented: the title of the document, the author of the document, the original file name, the original format, number of pages of the original document (optional). The header also stores the information about which linguistic processors produced the KAF document. The KAF header is represented within the `<kafHeader>` element.

```
<kafHeader>
  <fileDesc filename="11611.pdf" filesize="1468626" filetype="pdf"
metakey="HumberEsturary" pages="8"/>
  <public dmsid="11611" project="estuaries_english"
uri="\www.humberems.co.uk\downloads\ebbflow\EBB &amp;amp; FLOW NO.5.pdf"/>
  <captureDesc dateString="2009.10.19 AD at 09:30:40 AM CEST"/>
  <linguisticProcessors layer="1">
      <lp name="MultiwordTagger" timestamp="2010-04-07T22:54:12Z" version="0.1"/>
  </linguisticProcessors>
</kafHeader>
```

The text layer contains the tokens of the document. Optionally, sentence, paragraph and page boundaries are indicated. Optional offsets are used to link tokens with portions of the source document. This layer – the text element in KAF – is the result of sentence splitting and tokenization. All word forms are annotated within the `<text>` element, and each form is marked by a `<wf>` element.

```
<text>
        <wf page="1" sent="10" wid="w162">The</wf>
        <wf page="1" sent="10" wid="w163">current</wf>
        <wf page="1" sent="10" wid="w164">proactive</wf>
        <wf page="1" sent="10" wid="w165">role</wf>
        <wf page="1" sent="10" wid="w166">played</wf>
        <wf page="1" sent="10" wid="w167">by</wf>
        <wf page="1" sent="10" wid="w168">the</wf>
        <wf page="1" sent="10" wid="w169">Humber</wf>
        <wf page="1" sent="10" wid="w170">Advisory</wf>
        <wf page="1" sent="10" wid="w171">Group</wf>
</text>
```

The term layer is used to annotate words and multi-words. This layer refers to previous word forms (and to groups of multi-word forms) and attaches information such as lemma, part of speech, synset and name entity information. It also includes meta-information, references to other resources such as wordnet senses, whether or not it is a named entity, compound elements (in case of a compound). Since (multi-)words consist of tokens, they refer to tokens in the text layer. The part-of-speech tagset is the following:

[N] common noun

[R] proper noun

[G] adjective

[V] verb

[P] preposition

[A] adverb

[C] conjunction

[D] determiner

[O] other

Terms are marked by the `<term>` element.

```
<term lemma="the" pos="D" tid="t146" type="close">
 <span><target id="w162"/></span>
</term>
<term lemma="current" pos="G" tid="t147" type="open">
 <span><target id="w163"/></span>
<externalReferences>
<externalRef resource="wn30g" reference="eng-30-00666058-a" confidence="1"/>
</externalReferences>
</term>
<term lemma="proactive" pos="O" tid="t148" type="open">
 <span><target id="w164"/></span>
</term>
<term lemma="role" pos="N" tid="t149" type="open">
<span><target id="w165"/></span>
<externalReferences>
<externalRef resource="wn30g" reference="eng-30-05929008-n" confidence="0.267746"/>
<externalRef resource="wn30g" reference="eng-30-00720565-n" confidence="0.255975"/>
<externalRef resource="wn30g" reference="eng-30-05149325-n" confidence="0.245081"/>
<externalRef resource="wn30g" reference="eng-30-00722061-n" confidence="0.231198"/>
</externalReferences>
</term>
```

Compound terms can be represented in KAF by including <component> elements within <term> elements.

```
<term lemma="wading bird" pos="N" tid="t645mw" type="open">
 <span><target id="w718"/><target id="w719"/></span>
 <component id="t644" lemma="wade" pos="V">
<externalReferences>
<externalRef resource="wn30g" reference="eng-30-01916214-v" confidence="1"/>
</externalReferences></component>
<component id="t645" lemma="bird" pos="N">
<externalReferences><externalRef resource="wn30g" reference="eng-30-01503061-n"
confidence="0.299209"/>
</externalReferences>
</term>
```

The term layer can be enriched with further information. For each synset in KAF, the corresponding Base Concept, the correct ontology type and the appropriate relation to the ontology can be included. Each information is represented by means of nested XML elements <externalRef>:

```
<term lemma="role" pos="N" tid="t149" type="open">
 <span><target id="w165"/></span>
<externalReferences>
<externalRef confidence="0.267363" reference="eng-30-05929008-n" resource="wneng_domain">
  <externalRef confidence="1.0" reference="eng-30-00407535-n"    reftype="baseConcept"
resource="wn30g"/>
  <externalRef confidence="1.0" reference="Kyoto#activity-eng-3.0-00407535-n"
reftype="sc_subClassOf" resource="ontology">
  </externalRef>
</externalRef>
</externalReferences>
</term>
```

The chunks layer contains chunks of words (marked by a <chunks> element), such as noun phrases, prepositional phrases, etc. Since chunks consist of words, they refer to words in the terms layer. Each chunk has a head, which is also an item in the terms layer. They are marked with the <chunk> element.

```
<chunks>
<chunk cid="c121" head="t146" phrase="D">
            <span><target tid="t146"/></span>
</chunk>
<chunk cid="c122" head="t147" phrase="G">
            <span><target tid="t147"/></span>
</chunk>
<chunk cid="c124" head="t149" phrase="NP">
            <span><target tid="t149"/></span>
```

```
</chunk>
</chunks>
```

The dependency layer contains dependency relations between words (marked by a <deps> element). Since words participate in dependency relations, they refer to words in the terms layer. Each dependency is represented by a <dep> element.

```
<deps>
<dep from="t145" rfunc="mod" to="t144"/>
<dep from="t149" rfunc="subj" to="t163"/>
<dep from="t149" rfunc="mod" to="t147"/>
<dep from="t149" rfunc="mod" to="t148"/>
</deps>
```

The KAF semantic layer allows for tagging text elements such as expression of time, events, quantities, locations, …

Event information, including roles, spanning chunks is encoded in the <events> element. Event are compliant with ISO-TimeML annotation. An <event> element can be encoded as follows:

Given the following sentence:

```
<!-- 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>
<!-- every Monday -->
<chunk cid="c7" head="t7" phrase="NP">
<span>
<target id="t6"/>
<target id="t7"/>
</span>
</chunk>
<!-- in New York -->
<chunk cid="c9" head="t9" phrase="PP">
<span>
<target id="t8"/>
<target id="t9"/>
</span>
</chunk>
</chunks>
```

Event KAF encoding is as follows:

```
<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>
```

Quantifiers are annotated within `<quantifiers>` element. An example of quantifier annotation is:

```
<!-- every -->
<quantifiers>
<quantifier qid="q1" span="c6"/>
</quantifiers>
```

Time expressions are annotated within <timexs> element. Time expressions are compliant with ISO-TimeML. An example of time expression information is:

```
<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>
```

5.4. Standards for semantic annotation

Author: Tommaso Caselli

5.4.1. Dialogue Acts (DiAML)

The ISO standard for Dialogue Acts (ISO-DiAML) is an abstract meta-model for the annotation of dialogue corpora, following up on the EU-supported project LIRICS (Linguistic Infrastructure for the Interoperable Resources and Systems) developed in collaboration with TC 37/SC 4 ad-hoc Thematic group 3, Semantic content. DiAML is still under development and has been accepted for Draft International Standard balloting. The standard has been designed in accordance with the ISO Linguistic Annotation Framework (LAF, ISO 24612, 2009).

DiAML offers a number of domain independent concepts for dialogue acts annotation, a formal language for their expressions, strategies and guidelines for the extension of the core concepts and for selecting coherent subsets of the core concepts. The standard is intended for use by human annotators and automatic annotation systems.

The DiAML framework, being compliant with the LAF framework, draws a distinction between the concepts of *representation* and *annotation*. Annotation "refers to the linguistic information that is added to segments of language data, independent of the format in which the information is represented" (ISO/DIS 24617-2: 24), while representation "refers to the format in which an annotation is rendered, independent of its content" (ISO/DIS 24617-2: ibid). In DiAML this distinction is preserved through the concrete and abstract syntax, respectively. The abstract syntax defines-theoretical structures, called annotation structures, which contain all and exactly those elements that constitute the annotation of a functional segment with dialogue act information according to the metamodel. On the other hand, the concrete syntax defines a particular rendering of the annotation structures.

Applying the abstract syntax of DiAML different types of annotation schemes compliant with the DiAML standard can be specified. The abstract syntax can be understood as a toolkit for developing concrete representations, as it includes formal descriptions of of the conceptual inventory, annotation construction rules and link structures.

5.4.1.1. The backbone concepts of the metamodel

The core concepts of the DiAML metamodel as reported in the ISO/DIS 24617-2 document are the following:

- sender: a dialogue participant who produces a dialogue act;
- addressee: a dialogue participant oriented to by the speaker in a manner to suggest that his utterances are particularly intended for him, and that some response is therefore anticipated from him/her, more so than from the other participants;
- participants in other roles;
- functional segment: the minimal stretch of communicative behavior that has one or more communicative functions;
- dialogue act: the communicative activity of a participant in a dialogue interpreted as having a certain communicative function and semantic content, and possibly also having certain functional dependence relation, rhetorical relations and feedback dependence relations;
- communicative function: property of a dialogue act , specifying how the act's semantic content changes the addressee's information state upon successful performance of the dialogue act;
- communicative function qualifier;
- semantic content category: the semantic content type (kind of information, situation, action, event, or objects) that form the semantic content of a dialogue act;
- functional dependence relation: a relation between a dialogue act which depends semantically on a previous dialogue act and the previous act that it depends on (e.g. the relation between an answer and corresponding question);
- rhetorical relation: a relation between two dialogue acts, indicating a pragmatic connection between the two;
- feedback dependence relation: a relation between a feedback act and the stretch of communicative behavior whose processing the act provides or elicits information about function qualifier.

"A dialogue consists of two or more functional segments [...]. Each functional segment is related to one or more dialogue acts, reflecting the possible multifunctionality of functional segments. Each dialogue act has exactly one sender, one or more addressees, and possibly other participants [...]. It has a semantic content of a certain type, and one communicative function, which may have any number of function qualifiers; and is possibly related to other dialogue acts through functional dependence and rhetorical relations, and to functional segments through feedback dependence relations." (ISO/DIS 24617-2: 9).

Multidimensionality in DiAML is structured around the notion of *dimension*, based on the "observation that participation in a dialogue involves a range of communicative activities beyond those strictly related to performing the task that underlies the dialogue" (ISO/DIS 24617-2: 12).

The identification of core dimensions and communicative functions (*core dialogue acts*) id based on two different sets of parameters as reported in Figure 42:

Criteria for dimension identification	**Criteria for communicative function identification**
Theoretically justified – it must form a well established and well studied aspect of communication	Empirically observed
Empirically justified – it must be observed I the functions of dialogue utterances	Theoretically validated
Independent from other dimensions	Relevant – it must be used to obtain a good coverage of the phenomena in the dimension in analysis
Recognizable by human annotators and automatic systems	Recognizable by human annotators and automatic systems
Present in a number of existing dialogue schemes	Present in a number of existing dialogue schemes
	Be a member of a semantic connected set of functions

Figure 42: Criteria for the identification of core dimensions and communicative functions.

Communicative functions should be informed by the property of semantic connectedness, i.e. orthogonally organized and mutually exclusive. Applying these criteria DiAML proposes a set of 9 core dimensions, 26 general purpose and 31 dimension specific communicative functions

The use of functions qualifiers captures phenomena as modality, conditionality, partiality and accompanying emotions and avoids the introduction of very detailed functions.

5.4.1.2. XML-based Syntax Realization

This section describes a concrete XML-based syntax realization of the abstract syntax. The XML representation is composed by:

- `<dialogueAct>`: this tag is used to mark up the functional segments.

  Attributes:

  - sender: idref
  - addressee: idref
  - target: idref
  - otherParticipant: idref
  - id
  - communicative function: list of (at least) all core communicative function values;
  - dimension: list (of at least) all core dimensions;
  - conditionality: binary value expressing conditionality wrt. the communicative function;

- certainty: binary value expressing certainty wrt. the communicative function;
- partiality: binary value expressing partiality wrt. the communicative function;
- sentiment: open class of values expressing accompanying emotion to the communicative function;

Notice that the first four attributes are assumed to be identified in the metadata of the annotated data, and the functional segments as spans in the original data (compliance with stand-off annotation).

- `<functionalDependence>`: functional dependence link.

  Attributes:

  - dact: idref; referring to the currently annotated dialogue act
  - functAntecedent: idref; the functional antecedent of currently annotated dialogue act

- `<feedbackDependence>`: feedback dependence link.

  Attributes:

  - dact: idref; referring to the currently annotated dialogue act
  - fbSegment: idref; the segment whose processing the current dialogue acts elicits or provides information about;

- `<rhetoricalLink>`: rhetorical link.

  Attributes:

  - dact: idref; referring to the currently annotated dialogue act
  - rhetoAntecedent: idref; the dialogue act which is connected to the current one by means of a rhetorical relation
  - rhetoRel: (open class); list of rhetorical relations expressing the connection between two dialogue acts

5.4.1.3. DIAML Example

The following example is taken for the DiAML document.

```
P1: Do you know what time the next train to Utrecht leaves?
TA Fs1: Do you know what time the next train to Utrecht leaves?
P2: The next train to Utrecht leaves I think at 8:32.
AuFB fs2.1: The next train to Utrecht
TA fs2.2: The next train to Utrecht leaves I think at 8:32.


<diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#fs1"
 sender="#p1" addressee="#p2"
 communicativeFunction="setQuestion" dimension="task"
 conditionality="conditional"/>
<dialogueAct xml:id="da2" target="#fs2"
 sender="#p2" addressee="#p1"
 communicativeFunction="autoPositive" dimension="autoFeedback"/>
<feedbackDependence dact="#da2" fbSegment="#fs1"/>
<dialogueAct xml:id="da3" target="#fs2"
  sender="#p2" addressee="#p1"
```

```
  communicativeFunction="answer" dimension="task"/>
<functionalDependence dact="#da2" functAntecedent="#da1"/>
</diaml>
```

Figure 43: DiAML

No converter has been developed yet for this standard.

5.4.2. Events and Time Expression (TimeML-ISO)

Authors: Ineke Schuurman, Tommaso Caselli

An ISO-standard for temporal annotation is under development. It is called ISO-TimeML (ISO/DIS 24617-1, part I: Time and Events), and based on TimeML. It contains a set of guidelines, considered to be normative.

ISO-TimeML offers a format for the annotation of temporal entities, namely: temporal expressions, eventualities (both events and states), signals, such as temporal prepositions and conjuncts, and, finally, a set of relations between these entities, namely temporal relations, aspectual or phasal relations and subordinating relations which should facilitate the development of reasoning algorithms. TimeML is designed to address four problems in event and temporal expression markup: (i) time stamping of events (identifying an event and anchoring it in time); (ii) ordering events with respect to one another (lexical vs. discourse ordering); (iii) reasoning with contextually underspecified temporal expressions (temporal expressions such as 'last week' and 'two weeks before'); (iv) reasoning about events.

5.4.2.1. ISO-TimeML tags for markables

ISO-TimeML identifies three main markables elements and uses three different tags for their annotation:

`<event>`: this tag is used for annotating every event instance. The annotation philosophy of the `<event>` tag is inspired by the notion of the minimal chunk: one token per event. Generic events are not marked up. Special rules apply for events which span over more than one token.

Attributes:

- id: unique ID for each markabe (REQUIRED)
- anchor: HREF; used to anchor segments in stand-off annotation (REQUIRED)
- pred: CDATA; event predicate
- class: event classes. 7 possible values (OCCURRENCE; I_ACTION; I_STATE; ASPECTUAL; PERCEPTION; REPORTING; STATE); (REQUIRED)
- pos: event part of speech; (REQUIRED)
- tense: superficial verbal tense form (REQUIRED)
- aspect: standard distinction in grammatical category of aspect (REQUIRED)
- vform: grammatical category of non-tensed verbal forms; values are language specific; (REQUIRED)
- polarity: polarity of the event in question (REQUIRED)

- mood: mood of the event; values are language specific (REQUIRED)
- modality: modality nature of the event; values are language specific (OPTIONAL)
- comment:(OPTIONAL)

<timex3>: this tag is used for the annotation of temporal expressions. The tag span is determined on the basis of grammatical and relational criteria, with the latter beig more relevant than the former. It elaborates and extends the TIDES <timex2> tag.

Attributes:

- id: unique ID for each markabe (REQUIRED)
- anchor: HREF; used to anchor segments in stand-off annotation (REQUIRED)
- type: type of the temporal expressions; (values: DATE; DURATION; TIME; SET) (REQUIRED)
- functionInDocument: it indicates the function of the temporal expression in the document (OPTIONAL)
- beginPoint: HREF; used for anchored durations. It expresses the time expression ID indicating the begin point of the markable; (OPTIONAL)
- endPoint: HREF; used for anchored durations. It expresses the time expression ID indicating the end point of the markable; (OPTIONAL)
- quant: CDATA; it is used in conjunction of temporal expressions of type SET. It expresses the quantification over the temporal expression; (OPTIONAL)
- freq: CDATA; it is used in conjunction of temporal expressions of type SET. It expresses the frequency with which the temporal expression occurs; (OPTIONAL)
- value: it expresses the value of the temporal expressions in a standard format; the format is dependent on the type of the temporal expression; (REQUIRED)
- mod: expresses vague reference; (OPTIONAL)
- anchorTimeID: HREF; it expresses the ID of the temporal expression to which the markable is temporally anchored; (OPTIONAL)
- comment: CDATA (OPTIONAL)

<signal>: this tag is used to annotate every items which signals a relation between two events, or two timexes or an event and a temporal expression.

Attributes:

- id: unique ID for each markabe (REQUIRED)
- anchor: HREF; used to anchor segments in stand-off annotation (REQUIRED)

5.4.2.2. ISO-TimeML tags for links

Three types of link tags are proposed to create relations between the markables:

- <tlink>: this link is used to annotate any temporal relations between two elements

  Attributes:

  - id: unique ID for each link (REQUIRED)

- eventID: HREF; it expresses the ID of event markable involved in a temporal relation; (REQUIRED)
- timeID: HREF; it expresses the ID of the timex markable involved in a temporal relation; (REQUIRED)
- signalID: HREF; it expresses the ID of the signal markable involved in a temporal relation; (OPTIONAL)
- relatedToEvent:HREF; it expresses the ID of the event markable being related in a temporal relation; (REQUIRED)
- relatedtoTime: HREF; it expresses the ID of the timex markable being related in a temporal relation; (REQUIRED)
- relType: values of the temporal relations; (REQUIRED)

- `<slink>`: it is used to put in relation events on the basis of the syntactic structure. It does not offer strict temporal information but it allows to develop reasoning algorithms on event factivity.

  Attributes:

  - id: unique ID for each link (REQUIRED)
  - eventID: HREF; it expresses the ID of event markable involved in a subordination relation; (REQUIRED)
  - subordinatedEvent: HREF; it expresses the ID of event markable being involved in a subordination relation; (REQUIRED)
  - signalID: HREF; it expresses the ID of the signal markable involved in a temporal relation; (OPTIONAL)
  - relType: values of the subordination relations; (REQUIRED)

- `<alink>`: it is used to annotate relations between (lexical) aspectual events and their arguments

  Attributes:

  - id: unique ID for each link (REQUIRED)
  - eventID: HREF; it expresses the ID of aspectual event markable; (REQUIRED)
  - relatedToEvent: HREF; it expresses the ID of event markable related to the aspectual event; (REQUIRED)
  - signalID: HREF; it expresses the ID of the signal markable involved in the aspectual relation; (OPTIONAL)
  - relType: values of the aspectual relations; (REQUIRED)

5.4.2.3. Standard

As ISO-TimeML is a very detailed annotation format for temporal expressions, in many cases people only need to implement a subset (for example for temporal anchoring), and maybe even less detailed as in ISO-TimeML.

On the other hand, in some respects (ISO-)TimeML is not specific enough. An expression like '*Summer*', for example, is not associated with dates. People may want to be more specific, for example when comparing events described in several documents.

Another somewhat problematic issue is that for temporal annotation based on (formal-)semantic concepts, the definition of event and state as used in ANNEX A (normative temporal annotation guidelines) of ISO/DIS 24617-1, part I is not fully acceptable, states in (ISO-)TimeML being considered a subtype of events.

A last point is that (ISO-)TimeML contains a number of specifications that are in fact not related to temporal annotation *in se,* but do belong to a more general component. This holds i.e. for expressing whether polarity, or evidential, resp. neg-evendential expressions are involved. Although this is very valuable information, it does not belong to the temporal component as such.

We therefore suggest using ISO-TimeML as a recommended format for temporal annotation, (parts of) which may also serve as a pivot when converting other formats into each other.

5.4.2.4. Converters

We are not aware of existing converters.

For creation of converters it must be taken into account the fact that ISO-TimeML is a very detailed annotation scheme, and poorer formats may not have all the corresponding information or use other values. Moreover, the annotation philosophy of the various ISO-TimeML tags – attention to the superficial form, minimal chunk for events, grammatical and relational criteria for temporal expressions – must be taken carefully into account.

Within the Dutch/Flemish CLARIN pilot project TTNWW, the combined spatiotemporal STEx annotation format is to be mapped onto (ISO-) TimeML (as well as onto (ISO-)SpatialML). This will only be done towards the end of this project, early 2012.

5.4.2.5. Annotated Examples

All examples are taken from the ISO-TimeML document; ISO/DIS 24617-1.

John taught from September to December last year.

```
<EVENT id="e1" pred="TEACH" anchor="token1"
class="OCCURRENCE" pos="VERB"
tense="PAST" aspect="NONE" polarity="POS"/>

<SIGNAL id="s1" pred="FROM" anchor="token2"/>

<TIMEX3 id="t1" pred="SEPTEMBER" anchor="token3"
type="DATE" value="xxxx-09"/>

<SIGNAL id="s2" pred="TO" anchor="token4"/>

<TIMEX3 id="t2" pred="DECEMBER" anchor="token5"
type="DATE" value="xxxx-12"/>
```

```
<TIMEX3 id="t5" anchor="" type="DURATION" value="P4M"
beginPoint="t1" endPoint="t2" temporalFunction="true"/>

<TIMEX3 id="t3" pred="LAST_YEAR" anchor="token6 token7"
type=DATE" value="1995" temporalFunction="true"
anchorTimeID="t4"/>

<TIMEX3 id="t4" anchor="" type="DATE"
value="1996-03-27" functionInDocument="CREATION_TIME"/>

<TLINK timeID="t1" signalID="s1"
relatedToTime="t5" relType="BEGINS"/>

<TLINK timeID="t2" signalID="s2" relatedToTime="t5"
relType="ENDS"/>
```

Bill denied that John taught on Monday.

```
<EVENT id="e1" pred="DENY" anchor="token1"
class="I_ACTION" pos="VERB"
tense="PAST" aspect="NONE" polarity="POS"/>

<EVENT id="e2" pred="TEACH" anchor="token4"
class="OCCURRENCE" pos="VERB"
tense="PAST" aspect="NONE" polarity="POS"/>

<SIGNAL id="s1" anchor="token5"/>

<TIMEX3 id="t1" pred="MONDAY" anchor="token6"
type="DATE" value="XXXX-WXX-1"/>

<TLINK eventID="e2" signalID="s1"
relatedToTime="t1" relType="IS_INCLUDED"/>

<SLINK eventID="e1" subordinatedEvent="e2"
relType="NEG_EVIDENTIAL"/>
```

John is believed to have lived in Rome.

```
<EVENT id="e1" pred="BELIEVE" anchor="token3"
class="I_STATE" pos="VERB"
tense="PAST" aspect="PERFECTIVE" polarity="POS"/>


<EVENT id="e2" pred="LIVE" anchor="token5"
class="OCCURRENCE" pos="VERB"
tense="NONE" aspect="PERFECTIVE" vform="INFINITIVE"
```

```
polarity="POS"/>

<SLINK eventID="e1" subordinatedEvent="e2"
relType="INTENSIONAL"/>
```

5.4.3. Coreference (MATE)

Authors: Claudia Soria, Tommaso Caselli

The MATE web site is no longer maintained.

The MATE coreference annotation scheme is concerned with annotation of anaphoric elements in text. In MATE the term 'coreference' is used in a broad definition so as to include all anaphoric relations between two items (strict coreference - items which co-specify and co-refer- and large coreference – items which co-specify but do not co-refer, i.e. associative anaphora). The MATE proposal is based on the review of five previous annotation schemes, namely: the MUCSS scheme developed for MUC-7 (Hirschman and Chinchor, 1997), the DRAMA scheme (Passonneau, 1997), the Lancaster University UCREL scheme (Fligelstone, 1992) the scheme developed by (Bruneseaux and Romary, 1997) and the MapTask annotation of landmarks.

A first difference with respect to previous scheme is that the MATE proposals are explicitly based on the discourse model assumption (Webber, 1979; Heim, 1982; Gundel et al., 1993; Kamp and Reyle, 1993), i.e. that the interpretation of a discourse involves building a shared discourse model containing discourse entities which may *refer* to specific objects in the world, as well as the relations between these entities. The annotation for which the MATE scheme was developed is meant as a partial representation of the discourse model.

Since virtually any word in a text can be used to establish an anaphoric link, restriction of the amount of anaphoric information to annotate is obligatory for any annotation scheme.

The MATE coreference annotation scheme is a meta-scheme, i.e. an XML based markup language for marking up anaphoric relations. It was first develop for anaphoric relations in dialogue but it can be adapted easily to anaphoric relations in discourse as well.The meta-scheme includes:

- a set of tags for identifying the elements of text that may enter into anaphoric relations
- tags for specifying these relations.

The meta-scheme is further articulated into a) a “core scheme”, corresponding more or less to the MUC scheme, of information that can be annotated reliably and b) possible extensions to the scheme.

The MATE workbench support stand-off annotation and HREF mechanism to link the separate levels via a base file.

The core-scheme

- `<coref:de>`: this tag is used to mark every text span that may enter in an anaphoric relation.

  Attributes:

  - ID
  - HREF
- `<coref:link>`: this tag is used to indicate anaphoric relations.

Attributes:

- HREF (obligatory)
- TYPE (obligatory; with values as specified under the schemes)
- WHO-BELIEVES (optional; default value SHARED; other values to be set to the participants in the dialogue).

Embedded elements:

- <coref:anchor>: this tag is used to indicate the antecedent in an anaphoric relation.

Attributes:

- HREF

- <coref:universe>: this element is used to introduce the objects in the visual situation of discourse.

  Attributes:

  - ID (obligatory)
  - modifies (optional, only permitted value is COMMON, used when the universe extends the common universe

- <coref:ue>: there should be one such element for each object in the universe.

  Attributes:

  - ID

- <coref:seg>: this tag is used to mark elements that participate in anaphoric relations but are not expressed by NPs, such as incorporated clitics in Romance languages and the antecedents of discourse deixis.

  Attributes:

  - ID

Description of elements

This section provides a short description of the element tags.

- <coref:de>: it is used to annotate the text spans that introduce a discourse entity - that is, that can be subsequently referred to by means of anaphoric expressions. These are commonly noun phrases or other elements which can be further specified. Provided the stand-off annotation methodology supported by MATE, the annotation for <coref:de> should be included in a file with pointers (the HREF attribute) to a base file which has already been XML tagged with information about the structure of the document (dialogue transcript or a written document), ideally using TEI coding.
- <coref:link>: it is used to mark anaphoric relations between discourse entities, the most basic of which is the identity relation. When items in the document, annotated with the tag <coref:de>, co-specify, a <coref:link> element is added. The HREF attribute of this element points to the anaphoric expression, and contains at least one <coref:anchor> element specifying the antecedent (by means of a second HREF pointer). The type of relation that holds between the two discourse entities is specified by the TYPE attribute of the <coref:link> element, whose values depend on the exact scheme implemented. More than

<coref:anchor> element may be embedded in a <coref:link> element to account for ambiguity.

- <coref:universe>: it is used in dialogues, to mark up reference to items in the visual situation. This element may also be used to annotate references to items in the non-visible 'universe' of shared knowledge, the so-called `larger-situation' elements. The items in the visual situation are listed as universe entities and annotated with the tag <coref:ue>, embedded within a <coref:universe> tag. Each <coref:ue> element has an ID, like <coref:de>. This means that an anaphoric relation between a linguistic item and an object in the visual situation can be encoded by a link between a <coref:de> and a <coref:ue>.
- <coref:seg>: this element, like the TEI <seg> element, can be used to mark up arbitrary pieces of text, such as empty constituents, clitics, verbal ellipsis, discourse deixis (i.e. portions of text denoting a topic) and so on and so forth. <coref:seg> elements are given an id which can then be pointed at by a <coref:link> element just like for other anaphoric relations.

Annotated Examples

The following examples are taken from http://www.ims.uni-stuttgart.de/projekte/mate/mdag/cr/cr_4.html, (Poesio et al., 1999) and (Poesio, 2000).

For clarity's sake we will not report complete stand-off annotation.

```
we're gonna take <coref:de ID="de_07">the engine E3</coref:de> and shove
<coref:de ID="de_08">it</coref:de> over to Corning, hook
<coref:de ID="de_09">it</coref:de> up to the tanker car...


<coref:link href="coref.xml#id(de_08)" type="ident"><coref:anchor
href="coref.xml#id(de_07)"/></coref:link>
<coref:link href="coref.xml#id(de_09)" type="ident">
  <coref:anchor href="coref.xml#id(de_08)"/>
</coref:link>
```

```
<coref:universe ID="u1">
  <coref:ue ID="ue1">Diamond mine</coref:ue>
  <coref:ue ID="ue2">Graveyard</coref:ue>
  <coref:ue ID="ue3">Fast running creek</coref:ue>
  <coref:ue ID="ue4">Fast flowing river</coref:ue>
  <coref:ue ID="ue5">Canoes</coref:ue>
</coref:universe>


FOLLOWER: Uh-huh. Curve round. To your right.
GIVER:    Uh-huh.
FOLLOWER: Right.... Right underneath <coref:de ID="de_50">the diamond mine.</coref:de>
          Where do I stop.
GIVER:    Well..... Do. Have you got <coref:de ID="de_51">a graveyard?</coref:de>
          Sort of in the middle of the page? ... On on a level to
          <coref:de ID="de _52">the c-- ... er diamond mine.</coref:de>
FOLLOWER: No. I've got <coref:de ID="de_53">a fast running creek.</coref:
GIVER:    <coref:de ID="de _54">A fast flowing river</coref:de>,... eh.
FOLLOWER: No. Where's <coref:de ID="de_55"> that </coref:de>. Mmhmm,... eh.
          <coref:de ID="de _56">Canoes</coref:de>
```

```
<coref:link href="coref.xml#id(de_50)" type="ident">
  <coref:anchor href="coref.xml#id(ue1)"/>
</coref:link>
<coref:link href="coref.xml#id(de_51)" type="ident">
  <coref:anchor href="coref.xml#id(ue2)"/>
</coref:link>
<coref:link href="coref.xml#id(de_52)" type="ident">
  <coref:anchor href="coref.xml#id(ue1)"/>
</coref:link>
<coref:link href="coref.xml#id(de_53)" type="ident">
  <coref:anchor href="coref.xml#id(ue3)"/>
</coref:link>
<coref:link href="coref.xml#id(de_54)" type="ident">
  <coref:anchor href="coref.xml#id(ue4)"/>
</coref:link>
<coref:link href="coref.xml#id(de_55)" type="ident">
  <coref:anchor href="coref.xml#id(de_54)"/>
</coref:link>
<coref:link href="coref.xml#id(de_56)" type="ident">
  <coref:anchor href="coref.xml#id(ue5)"/>
</coref:link>
```

```
A: Dov'e` <coref:de ID="de_157">Gianni?</coref:de>
   [Where is Gianni?]
B: <coref:seg type="pred" ID="seg_158>
     e` andato a mangiare
   </coref:seg>
   [_ went to have lunch]


<coref:link href="coref.xml#id(seg_158)" type="ident">
  <coref:anchor href="coref.xml#id(de_157)"/>
</coref:link>
```

```
<coref:seg type="event" ID="seg_130">
The 23-year-old had hit his head against another player
</coref:seg> during a game of Aussie-rules football.
McGlinn remembered nothing of
<coref:de ID="de_131">
the collision
</coref:de>, but developed a headache and had several seizures.

 <coref:link href="coref.xml#id(de_131)" type="ident">
  <coref:anchor href="coref.xml#id(seg_130)"/>
</coref:link>
```

Converters

No converters have been created for the MATE annotation format. On the other hand, MATE, being a meta-scheme, has been used to develop MATE-compliant annotation schemes for anaphora resolution. One of these is the GNOME scheme ((Poesio et al., 1999) and (Poesio, 2000), and http://cswww.essex.ac.uk/Research/nle/ corpora/GNOME/anno_manual_4.htm).

The GNOME scheme follows very closely the MATE proposals for markables, even if:

- the <coref:de> element is renamed <ne>, since only NPs are annotated as discourse markables;
- the <coref:link> element is renamed <ante>
- stand-off annotation was not used, due to absence of tools for "knitting" back the various layers, though it is highly recommended and supported;
- new elements have been introduced for investigating the notion of utterance in Centering
- new attributes for the <ne> element have been introduced;
- In case converters to MATE meta-scheme were created, the richness of many annotation schemes for anaphora annotation in terms of attributes would be lost preserving only the core information.

MATE Core Scheme DTD

```
<!-- DTD for MATE coreference Core Scheme   -->
<!ELEMENT coref:de   ANY>
<!ATTLIST coref:de
                id      ID      #REQUIRED
                type    CDATA   #IMPLIED>
<!ELEMENT coref:link    (coref:anchor)+>
<!ATTLIST coref:link
                hrefAttr;
                type    (ident | user_values)#REQUIRED
                who-believes CDATA #IMPLIED>
<!ELEMENT coref:anchor  EMPTY>
<!ATTLIST coref:anchor
                hrefAttr;>
<!-- UNIVERSES -->
<!ELEMENT coref:universe  (coref:ue)*>
<!ATTLIST coref:universe
                id      ID      #IMPLIED
                modifies (common)       "common">
<!ELEMENT coref:ue  (#PCDATA)>
<!ATTLIST coref:ue
                id  ID #REQUIRED>
<!ELEMENT coref:seg  (#PCDATA |user_values)*>
<!ATTLIST coref:seg
                id      ID      #IMPLIED
                type    CDATA   #IMPLIED>
```

5.4.4. Named Entity Recognition/Classification (NER)

Named Entity Recognition (NER) is the task of recognising Named Entities and classifying them as belonging to a previously defined category. NER is frequently used in syntactic and semantic text analysis.

In contrast to Proper Names, there are no solid criteria for the definition of Named Entities (NE); their definition and classification is always dependent on the current purpose. NEs can be e.g. references to persons, locations, etc., but also to time and dates and whatever is relevant for the annotator. The annotation of NE is not restricted to Proper Nouns but also comprises common nouns and pronouns referring to the same entity. The NE-annotated biomedical corpus "GENIA" e.g. contains NE-classes of DNA, RNA, proteins, cell-lines, etc.

The two best-elaborated schemas for classification have been developed for the English language, but there are schemas for a number of other languages, and several schemas have been adapted to other languages.

5.4.4.1. MUC Schema

The MUC schema was developed in 1995 (Message Understanding Conferences) (MUC-6: Sixth Message Understanding Conference, 1995) for English. In MUC, NEs are defined as „proper names, acronyms, and perhaps miscellaneous other unique identifiers". Their heuristic of including all words with capital letters is suitable for most languages in Latin alphabet, except for German. Possible categories of NEs are (refer to MUC-Appendix 1995: 322, http://www.aclweb.org/anthology-new/M/M95/M95-1024.pdf):

- ORGANIZATION: named corporate, governmental, or other organizational entity;
- PERSON: named person or family
- LOCATION: name of politically or geographically defined location (cities, provinces, countries, international regions, bodies of water, mountains, etc.)

In the MUC schema, NEs are annotated as spans:

```
<LOC>North</LOC> and <LOC>South America</LOC>
<ORG>Mips</ORG> Vice President <PER>John Hime</PER >
<LOC>the Big Apple</LOC>
```

5.4.4.2. ACE Schema

The ACE (Automatic Context Extraction, http://www.itl.nist.gov/iad/mig//tests/ace/) Program started in 2004. It distinguishes more categories than MUC, namely Person, Organization, Location, Facility and Geo-Political Entity. Each type is further divided into subtypes (for instance, subtypes of Person include Individual, Group and Indefinite).

In ACE Pilot Format, one entity is annotated the following way:

```
<entity ID="DOCID-37639-3">
      <entity_type>ORGANIZATION</entity_type>
        <entity_mention TYPE="NAME">
          <extent>
            <charspan>
```

```
            <!-- string = "Congress" -->
              <start>567</start><end>575</end></charspan>
          </extent>
          <head>
            <charspan>
            <!-- string = "Congress" -->
              <start>567</start><end>575</end></charspan>
          </head>
        </entity_mention>
        <entity_name>
          <extent>
            <charspan>
            <!-- MAX = "Congress" -->
            <start>567</start><end>575</end></charspan>
          </extent>
        </entity_name>
      </entity>
```

6. Multimedia Encoding and Annotation

6.1. General distinctions / terminology

Authors: Thomas Schmidt, Kjell Elenius

By a **Multimedia Corpus** we understand a systematic collection of language resources involving data in more than one medium. Typically, a multimedia corpus consists of a set of digital audio and/or video data and a corresponding set of textual data (the transcriptions and/or annotations of the audio or video data). Before we start discussing tools, models, formats and standards used for multimedia corpora in the following sections, we will devote a few paragraphs to clarifying our terminology.

6.1.1. Different types of multimedia corpora

Multimedia corpora are used in a variety of very different domains including, among others, speech technology, several subfields of linguistics (e.g. phonetics, sociolinguistics, conversation analysis), media sciences and sociology. Consequently, there is no consistent terminology, let alone a unique taxonomy to classify and characterize different types of multimedia corpora. Instead of attempting to define such a terminology here, we will characterize different types of corpora by describing some of their prototypical exponents. The categories and their characterising features are meant neither to be mutually exclusive (in fact, many existing corpora are a mixture of the types), nor to necessarily cover the whole spectrum of multimedia corpora.

A **Spoken Language (or: Spontaneous Speech) Corpus** is a corpus constructed with the principal aim of investigating language as used in spontaneous spoken everyday interaction. A typical spoken language corpus contains recordings of authentic (as opposed to experimentally controlled or read) dialogues or multilogues (as opposed to monologues). It is transcribed orthographically, typically in a modified orthography which takes into account characteristic features of spontaneous speech (like elisions, assimilations or dialectal features), and it takes care to note carefully certain semi-lexical (or paraverbal) phenomena like filled pauses (hesitation markers) or laughing. A spoken language corpus may contain information about behaviour in other modalities (like mimics and gesture) in addition to the transcribed speech, but its main concern lies with linguistic behaviour. In that sense, corpora used in conversation or discourse analysis (e.g. the Santa Barbara Corpus of Spoken American English or the FOLK corpus) are prototypes of spoken language corpora. Meeting corpora like the ISL Meeting Corpus or the AMI Meeting Corpus can be regarded as another group of typical spoken language corpora. Less prototypical, but equally relevant members of this class are child language corpora (e.g. corpora in the CHILDES database), corpora of narrative (or: 'semi-structured') interviews (e.g. the SLX Corpus of Classic Sociolinguistic Interviews by William Labov), or corpora of task-oriented communication (like Map Tasks, e.g. the HCRC Map Task Corpus). Often, spoken language corpora are restricted to a specific interaction type (e.g. doctor-patient communication, classroom discourse or ad-hoc interpreting) or speaker type (e.g. adolescents or a certain social group). A truly multimodal database is the Swedish dialogue Spontal Corpus, in which the participants are captured on four channels of high-quality audio, two channels of high-resolution video and on a motion capture system tracking torso, arm, hand and head movements.

A **Speech (Technology) Corpus** is a corpus constructed with the principal aim of building, training or evaluating speech technology applications like speech recognizers, dialogue systems or text-to-speech systems. A speech corpus typically contains recordings of read or prompted speech – by one speaker in a studio for text-to-speech systems, or by more than hundreds of persons for speech recognition systems. The latter are often recorded in an office setting or recorded over mobile and/or fixed telephones. They are usually balanced over age and gender. Speech corpora are usually transcribed in standard orthography (possibly with labels for various acoustic noises and disturbances as well as truncated utterances), thus abstracting over idiolectal, dialectal or otherwise motivated variation in speech. Other modalities (like gestures or mimics) usually play no role in speech corpora (which are therefore usually audio corpora). In that sense, the SpeechDat project with up to 5000 speakers per language recorded over fixed telephone networks is a prototypical speech recognition database, as well as the corpora from the Speecon project, in which the recordings were made on location with high quality and also simple hands-free microphones. Also recordings of broadcast news are frequently used for speech recognition research and are relatively inexpensive to record. Besides containing spontaneous speech they include further challenges such as speaker change and non-speech events like music and jingles. The 1996 English Broadcast News Speech corpus is an example of these.

A **Phonetic (Speech) Corpus** is a corpus with the principal aim of carrying out research into the phonetics, phonology and prosody of language. A typical phonetic corpus contains recordings of elicited, i.e. read or prompted, speech, typically monologues. The utterances are often syllables, words or sentences that are phonetically transcribed, sometimes augmented with prosodic labels. Typically, a phonetic corpus contains no information about other modalities. TIMIT, an Acoustic-Phonetic Continuous Speech Corpus of American English, is a prototypical phonetic corpus. Also monologues and dialogues may be used for phonetic research. In the Swedish SweDia 2000 project more than 100 persons were recorded for dialect research. The informants were encouraged to speak monologues but also read word lists. Also recordings of endangered languages or languages that have no writing system may be seen as phonetic corpora. The UCLA Phonetics Lab Language Archive contains more than 200 languages.

An **Expressive Speech Corpus** is recorded in order to capture how emotions affect the acoustic speech signal. There are in principle three different ways of recording these:

- Let a person, usually an actor/actress, record utterances with different emotions. An exponent is the Berlin Database of Emotional Speech, Emo-DB.
- Induced emotions, e.g. a procedure in which the subjects watch a film intended to evoke specific emotions or the Velten technique (Velten, 1968) where the subjects read emotional scenarios or emotionally loaded sentences to "get into" the wanted mood.
- “Real” emotions from recorded spontaneous speech. Typical corpora with spontaneous emotions are recorded at call centers. An example is the French CEMO corpus that contains real agent-client recordings from a medical emergency call center.

A **Multimodal Corpus** is a corpus with the principal aim of making the multimodal aspect of interaction accessible for analysis. In contrast to the other types, it does not prioritize any single modality, but rather treats speech, gestures, mimics, body posture etc. as equally important aspects. In contrast to a typical spoken language corpus, a multimodal corpus will use systematic coding schemes, rather than free descriptions, to describe non-verbal behaviour. In that sense, the SmartKom corpus is a prototypical multimodal corpus.

A **Sign Language Corpus** is a corpus in which the focus is not on the acoustic/articulatory modality of a spoken language, but on the visual/gestural modality of a signed language. Transcription of sign language corpora is typically done with the help of specialized transcription systems for sign languages. The Corpus NGT is a prototypical example of a sign language corpus.

The following table summarizes distinguishing features of the corpus types.

Corpus type	Recordings	Modalities	Research interest	Transcription	Prototype(s)
Spoken language	Multilogues, Audio or Video	Verbal more important than non-verbal	Talk in interaction	Modified Orthography	SBCSAE corpus FOLK corpus
Speech	Monologues, Audio	Verbal	Speech technology	Standard Orthography	SpeechDat corpora
Phonetic	Monologues, Audio	Verbal	Phonetic/Phonology	Orthographic	CEMO, Emo-DB
Emotional	Audio	Verbal and Non-verbal equally important	Emotions	Phonetic	TIMIT
Multimodal	Video	Verbal and Non-verbal equally important	Multimodal behaviour	Modified or standard orthography	SmartKom corpus
Multimodal	Video, audio, movements	Verbal and Non-verbal equally im-portant, gestures	Multimodal behaviour	Modified or standard orthography	Spontal corpus
Sign Language	Video	Signs	Sign language	Sign language transcription system	NGT corpus

It is important to note that many of the tools described in section 6.3 are suitable to be used with several or all of these corpus types, although they may have a more or less outspoken preference towards a single one of them (e.g. FOLKER for spoken language, Praat for phonetic, ANVIL for multimodal corpora). Still, standardization efforts may have to take into account that different corpus types and the corresponding research communities have diverging needs and priorities so that evolving standards may have to be differentiated according to this (or a similar) typology.

6.1.2. Media encoding vs. Media annotation

By definition, a multimedia corpus contains at least two types of content: (audio or video) recordings and (text) annotations. Insofar as annotations are derived from and refer to the recordings, recordings are the primary and annotations the secondary data in multimedia corpora. Standardization is a relevant issue for both types of content. However, whereas the annotations are a type of content specific to the scientific community, audio or video recordings are used in a much wider range of contexts. Consequently, the processes for the definition and development of standards, as well as their actual state and their intended audience differ greatly for the two content types. We will treat standards and formats for the representation of audio and video data under the heading of "Media formats" in section 6.2, and tools and formats for annotating audio and video data under the heading of "Media annotation" in section 6.3.

6.1.3. Data models/file formats vs. Transcription systems/conventions

Annotating an audio or video file means systematically reducing the continuous information contained in it to discrete units suitable for analysis. In order for this to work, there have to be rules which tell an annotator which of the observed phenomena to describe (and which to ignore) and how to describe them. Rather than providing such concrete rules, however, most data models and file formats for multimedia corpora remain on a more abstract level. They only furnish a general structure in which annotations can be organized (e.g. as labels with start and end points, organized into tiers which are assigned to a speaker) without specifying or requiring a specific semantics for these annotations. These specific semantics are therefore typically defined not in a file format or data model specification, but in a transcription convention or transcription system. Taking the annotation graph framework as an example, one could say that the data model specifies that annotations are typed edges of a directed acyclic graph, and a transcription convention specifies the possible types and the rules for labelling the edges. Typically, file formats and transcription systems thus complement each other. Obviously, both types of specification are needed for multimedia corpora, and both can profit from standardization. We will treat data models and file formats in sections 6.3.1 to 6.3.4 and transcription conventions/systems in section 6.3.5. Section 6.3.5.4 concerns itself with some widely used combinations of formats and conventions.

6.1.4. Transcription vs. Annotation/Coding vs. Metadata

So far, we have used the term **annotation** in its broadest sense, as, for example, defined by (Bird and Liberman, 2001), pp. 25:

> [We think] of 'annotation' as the provision of any symbolic description of particular portions of a pre-existing linguistic object

In that sense, any type of textual description of an aspect of an audio or video file can be called an annotation. However, there are also good reasons to distinguish at least two separate types of processes in the creation of multimedia corpora. (MacWhinney, 2000), p. 13, refers to them as **transcription** and **coding**.

> It is important to recognize the difference between transcription and coding. Transcription focuses on the production of a written record that can lead us to understand, albeit only vaguely, the flow of the original interaction. Transcription must be done directly off an audiotape or, preferably, a videotape. Coding, on the other hand, is the process of recognizing, analyzing, and taking note of phenomena in transcribed speech. Coding can often be done by referring only to a written transcript.

Clear as this distinction may seem in theory, it can be hard to draw in practice. Still, we think that it is important to be aware of the fact that media annotations (in the broad sense of the word) are often a result of two qualitatively different processes – transcription on the one hand, and annotation (in the narrower sense) or coding on the other hand. Since the latter process is less specific to multimedia corpora (for instance, the lemmatization of an orthographic spoken language transcription can be done more or less with the same methods and formats as a lemmatization of a written language corpus), we will focus on standards for the former process in section 6.3 of this chapter.

For similar reasons, we will not go into detail about **metadata** for multimedia corpora. Some of the formats covered here (e.g. EXMARaLDA) contain a section for metadata about interactions, speakers and recordings, while others (e.g. Praat) do not. Where it exists, this kind of information is clearly separated from the actual annotation data (i.e. data which refers directly to the event recorded rather than to the circumstances in which it occurred and was documented) so that we think it is safe to simply refer the reader to the relevant CLARIN documents on metadata standards.

6.2. Media formats

Author: Paul Trilsbeek

When speaking about media formats, one has to distinguish between container formats and codecs (coder-decoder). Container formats are file formats that can contain one or more media streams that may have been encoded with a certain codec. Most container formats can contain media that have been encoded with a variety of codecs, e.g. an AVI container can contain audio and video in almost any encoding that the Windows Media Framework can play. Codecs are used to reduce the storage space or bandwidth required for the media streams by applying data compression algorithms. These compression algorithms can be either (mathematically) lossless or lossy. In the latter case, information that is deemed irrelevant or less relevant for human perception of the signal is removed. This reduction of information is irreversible and while it might not influence the human perception of the signal, it could lead to differences in results of computational analyses. Before applying lossy compression, one should make sure that this does not have a significant influence on the analyses one is going to perform.

A technique that is often used to reduce the bandwidth of a video signal is so-called Chroma Subsampling. This technique makes use of the fact that the human eye is less sensitive for color (chroma) information than for brightness (luma) information. Instead of storing the color information for every pixel, it is only stored for every group of 2 or 4 pixels. This reduces the required bandwidth by at least one third. Despite the fact that a reduction of information has taken place, chroma subsampled signals without any further compression applied to them are generally still referred to as "uncompressed".

Storing media for the long term involves both keeping the bit-stream intact as well as keeping the file interpretable. It is likely that certain container formats and codecs will become obsolete in the future. Converting from one lossy compressed format to another will result in a degradation of the signal; therefore it is better to store media in a lossless compressed or uncompressed form whenever feasible. For audio this is generally no issue since storage costs for uncompressed audio are manageable for most individuals and institutions these days. For video this is not yet the case. Only some of the larger specialized archives are at this point in the position to store large amounts of lossless compressed video.

The usage of proprietary formats should in principle be avoided as much as possible when long-term preservation is required. For practical reasons however this is not always feasible, especially for video formats, in which case it is at least recommended to use widely used ISO standardized formats such as MPEG2 and MPEG4.

6.3. Media annotation

Author: Thomas Schmidt

6.3.1. Tools and tool formats

6.3.1.1. ANVIL (Annotation of Video and Language Data)

Developer: Michael Kipp, DFKI Saarbrücken, Germany

URL: http://www.anvil-software.de/

File format documentation: Example files on the website, file formats illustrated and explained in the user manual of the software

ANVIL was originally developed for multimodal corpora, but is now also used for other types of multimedia corpora. ANVIL defines two file formats, one for specification files and one for annotation files. A complete ANVIL data set therefore consists of two files (typically, one and the same specification file will be used with several annotation files in a corpus, though).

The specification file is an XML file telling the application about the annotation scheme, i.e. it defines tracks, attributes and values to be used for annotation. In a way, the specification file is thus a formal definition of the transcription system in the sense defined above. The annotation file is an XML file storing the actual annotation. The annotation data consists of a number of annotation elements which point either into the media file via a start and an end offset or to other annotation elements and which contain one ore several feature value pairs with the actual annotation(s). Individual annotation elements are organized into a number of tracks. Tracks are assigned a name and one of a set of predefined types (primary, singleton, span).

ANVIL's data model can be viewed as a special type of an annotation graph. It is largely similar to the data models underlying ELAN, EXMARaLDA, FOLKER, Praat and TASX.

```
<annotation>
 <head>
  <specification src="lq-demo-spec.xml"/>
  <video src="lq1-2-reich.avi"/>
  <!-- [...] -->
 </head>
 <body>
  <track name="trl" type="primary">
   <el index="0" start="1.514341115" end="1.747961878">
    <attribute name="token">wir</attribute>
   </el>
   <el index="1" start="1.747961878" end="2.130250453">
    <attribute name="token">reden</attribute>
    <attribute name="emphasis">moderate</attribute>
   </el>
   <!-- [...] -->
  </track>
  <!-- (ctd.) -->
  <track name="rst" type="span" ref="trl">
   <el index="1" start="4" end="11">
    <attribute name="relation1">elaboration</attribute>
    <attribute name="direction1">backward</attribute>
   </el>
   <el index="2" start="12" end="13">
    <attribute name="relation2">evidence</attribute>
    <attribute name="relation1">attribution</attribute>
    <attribute name="direction2">backward</attribute>
    <attribute name="direction1">forward</attribute>
   </el>
   <!-- [...] -->
  </track>
 </body>
</annotation>
```

Figure 44: Excerpt for an ANVIL annotation file

6.3.1.2. CLAN (Computerized Language Analysis)/CHAT (Codes for the Human Analysis of Transcripts) / Talkbank XML

Developers: Brian MacWhinney, Leonid Spektor, Franklin Chen, Carnegie Mellon University, Pittsburgh

URL: http://childes.psy.cmu.edu/clan/

File format documentation: CHAT file format documented in the user manual of the software, Talkbank XML format documented at http://talkbank.org/software/, XML Schema for Talbank available from http://talkbank.org/software/.

The tool CLAN and the CHAT format which it reads and writes were originally developed for transcribing and analyzing child language. CHAT files are plain text files (various encodings can be used, UTF-8 among them) in which special conventions (use of tabulators, colons, percentage signs, control codes, etc.) are used to mark up structural elements such as speakers, tier types, etc. Besides defining formal properties of files, CHAT also comprises instructions and conventions for

transcription and coding – it is thus a file format as well as a transcription convention in the sense defined above.

The CLAN tool has functionality for checking the correctness of files with respect to the CHAT specification. This functionality is comparable to checking the well-formedness of an XML file and validating it against a DTD or schema. However, in contrast to XML technology, the functionality resides in software code alone, i.e. there is no explicit formal definition for correctness of and no explicit data model (comparable to a DOM for XML files) for CHAT files.

CHAT files which pass the correctness check can be transformed to the Talbank XML format using a piece of software called chat2xml (available from http://talkbank.org/software/chat2xml.html). There is a variant of CHAT which is optimized for conversation analysis style transcripts (rather than child language transcripts). The CLAN tool has a special mode for operating on this variant.

```
@UTF8
@Begin
@Languages:	en
@Participants:	CHI Ross Target_Child, MAR Mark Brother,
MOT Mary Mother,
		FAT Brian Father
*CHI:	I decided to wear my Superman shirt .
%snd:"boys49a1"_0_6130
%mor:	pro|I part|decide-PERF prep|to n|wear
pro:poss:det|my n:prop|Superman
		n|shirt .
%xsyn:	1|2|SUBJ 2|0|ROOT 3|2|JCT 4|3|POBJ 5|7|MOD
6|7|MOD 7|4|OBJ 8|2|PUNCT
%com:	first use of "I decided ." In morning while dressing .
@New Episode
@Tape Location:	185
@Date:	20-MAR-1982
@Situation:	Marky asked where the Darth_Vader was .
```

```
*FAT:	you mean the Darth_Vader head ?
%snd:"boys49a1"_6130_14572
%mor:	pro|you v|mean det|the n:prop|Darth_Vader n|head ?
%xsyn:	1|2|SUBJ 2|0|ROOT 3|5|DET 4|5|MOD 5|2|OBJ
6|2|PUNCT
*CHI:	but you really call it the Darth_Vader collection caser .
%snd:"boys49a1"_14572_19030
%mor:	conj:coo|but pro|you adv:adj|real-LY v|call pro|it
det|the n:prop|Darth_Vader
		n|collection n:v|case-AGT .
%xsyn:	1|0|ROOT 2|4|SUBJ 3|4|JCT 4|1|COORD 5|4|OBJ
6|9|DET 7|9|MOD 8|9|ENUM
		9|4|JCT 10|1|PUNCT
```

Figure 45: Excerpt of a CHAT text file

6.3.1.3. ELAN (EUDICO Linguistic Annotator)

Developer:	Han Sloetjes, MPI for Psycholinguistics, Nijmegen
URL:	http://www.lat-mpi.eu/tools/elan/
File format documentation:	Example set on the tool's website, XML schema inside the source distribution (available from the tool's website), explanation of format and data model explained to be published.

ELAN is a versatile annotation tool and one of the major components of the LAT (Language Archiving Technology) suite of software tools from the MPI in Nijmegen. ELAN has been extensively used for the documentation of endangered languages, for sign language transcription and for the study of multimodality, but its area of application probably goes beyond these three corpus types.

ELAN reads and writes the EAF format, an XML format based on an annotation graph inspired data model, which has many similarities with the data models underlying ANVIL, EXMARaLDA, FOLKER, Praat and TASX. Annotations are organized into (possibly interdependent) tiers of different types. Controlled vocabularies can be defined and also stored inside an EAF file. The tool and its format provide mechanisms for making use of categories inside the ISO-CAT registry and for relating annotations to IMDI metadata.

```
<ANNOTATION_DOCUMENT AUTHOR="" DATE="2006-06-13T15:09:43+01:00" FORMAT="2.3" VERSION="2.3">
  <HEADER MEDIA_FILE="" TIME_UNITS="milliseconds">
    <MEDIA_DESCRIPTOR MEDIA_URL="file:///D:/Data/elan/elan-example1.mpg" MIME_TYPE="video/mpeg"/>
    <MEDIA_DESCRIPTOR EXTRACTED_FROM="elan-example1.mpg" MEDIA_URL="file:///D:/Data/elan/elan-example1.wav"
        MIME_TYPE="audio/x-wav"/>
  </HEADER>
  <TIME_ORDER>
    <TIME_SLOT TIME_SLOT_ID="ts1" TIME_VALUE="0"/>
    <TIME_SLOT TIME_SLOT_ID="ts2" TIME_VALUE="280"/>
    <TIME_SLOT TIME_SLOT_ID="ts3" TIME_VALUE="440"/>
    <!-- [...] -->
  </TIME_ORDER>
  <TIER DEFAULT_LOCALE="en" LINGUISTIC_TYPE_REF="utterance" PARTICIPANT="" TIER_ID="K-Spch">
    <ANNOTATION>
      <ALIGNABLE_ANNOTATION ANNOTATION_ID="a1" TIME_SLOT_REF1="ts2" TIME_SLOT_REF2="ts5">
        <ANNOTATION_VALUE>so from here.</ANNOTATION_VALUE>
      </ALIGNABLE_ANNOTATION>
    </ANNOTATION>
    <ANNOTATION>
      <ALIGNABLE_ANNOTATION ANNOTATION_ID="a2" TIME_SLOT_REF1="ts22" TIME_SLOT_REF2="ts24">
        <ANNOTATION_VALUE>yeah</ANNOTATION_VALUE>
      </ALIGNABLE_ANNOTATION>
    </ANNOTATION>
    <!-- [...] -->
  </TIER>
  <TIER DEFAULT_LOCALE="en" LINGUISTIC_TYPE_REF="part of speech" PARENT_REF="W-Words"
    PARTICIPANT="" TIER_ID="W-POS">
    <ANNOTATION>
      <REF_ANNOTATION ANNOTATION_ID="a120" ANNOTATION_REF="a23">
        <ANNOTATION_VALUE/>
      </REF_ANNOTATION>
    </ANNOTATION>
    <ANNOTATION>
      <REF_ANNOTATION ANNOTATION_ID="a121" ANNOTATION_REF="a24">
        <ANNOTATION_VALUE>pro</ANNOTATION_VALUE>
      </REF_ANNOTATION>
    </ANNOTATION>
  </TIER>
  <LINGUISTIC_TYPE GRAPHIC_REFERENCES="false" LINGUISTIC_TYPE_ID="utterance" TIME_ALIGNABLE="true"/>
  <LINGUISTIC_TYPE CONSTRAINTS="Time_Subdivision" GRAPHIC_REFERENCES="false"
    LINGUISTIC_TYPE_ID="words" TIME_ALIGNABLE="true"/>
  <LINGUISTIC_TYPE CONSTRAINTS="Symbolic_Association" GRAPHIC_REFERENCES="false"
    LINGUISTIC_TYPE_ID="phonetic_transcription" TIME_ALIGNABLE="false"/>
  <!-- [...] -->
  <CONTROLLED_VOCABULARY CV_ID="POS" DESCRIPTION="Part of Speech">
    <CV_ENTRY DESCRIPTION="noun">n</CV_ENTRY>
    <CV_ENTRY DESCRIPTION="verb">v</CV_ENTRY>
    <CV_ENTRY DESCRIPTION="interjection">int</CV_ENTRY>
    <!-- [...] -->
  </CONTROLLED_VOCABULARY>
</ANNOTATION_DOCUMENT>
```

Figure 46: Excerpt of an ELAN annotation file

6.3.1.4. EXMARaLDA (Extensible Markup Language for Discourse Annotation)

Developers:	Thomas Schmidt, Kai Wörner, SFB Multilingualism, Hamburg
URL:	http://www.exmaralda.org
File format documentation:	Example corpus on the tool's website, DTDs for file formats at http://www.exmaralda.org/downloads.html#dtd, data model and format motivated and explained in (Schmidt, 2005; Schmidt, 2005).

EXMARaLDA's core area of application are different types of spoken language corpora (for conversation and discourse analysis, for language acquisition research, for dialectology), but the system is also used for phonetic and multimodal corpora (and for the annotation of written language). EXMARaLDA defines three inter-related file formats – Basic-Transcriptions, Segmented-Transcriptions and List-Transcriptions. Only the first of these two are relevant for interoperability issues. A Basic-Transcription is an annotation graph with a single, fully ordered timeline and a

partition of annotation labels into a set of tiers (aka the "Single timeline multiple tiers" data model: STMT). It is suitable to represent the temporal structure of transcribed events, as well as their assignment to speakers and to different levels of description (e.g. verbal vs. non-verbal). A Segmented-Transcription is an annotation graph with a potentially bifurcating time-line in which the temporal order of some nodes may remain unspecified. It is derived automatically from a Basic-Transcription and adds to it an explicit representation of the linguistic structure of annotations, i.e. it segments temporally motivated annotation labels into units like utterances, words, pauses etc. EXMARaLDA's data model can be viewed as a special type of an annotation graph. It is largely similar to the data models underlying ANVIL, ELAN, FOLKER, Praat and TASX.

```
<basic-transcription>
  <head>
    <meta-information>
      <transcription-name>PearStory</transcription-name>
      <referenced-file url="PearStory.mp3"/>
      <!-- [...] -->
    </meta-information>
    <speakertable>
      <speaker id="SPK0">
        <abbreviation>X</abbreviation>
        <sex value="f"/>
        <!-- [...] -->
      </speaker>
      <speaker id="SPK1">
        <abbreviation>Y</abbreviation>
        <!-- [...] -->
      </speaker>
    </speakertable>
  </head>
  <basic-body>
    <common-timeline>
      <tli id="T0" time="0.0" type="intp"/>
      <tli id="T1" time="1.9000"/>
      <!-- [...] -->
      <tli id="T5" time="5.0930"/>
      <tli id="T6" time="9.2000"/>
    </common-timeline>
    <tier id="TIE0" speaker="SPK0" category="sup" type="a" display-name="">
      <event start="T1" end="T3">louder </event>
      <!-- [...] -->
    </tier>
    <tier id="TIE1" speaker="SPK0" category="v" type="t" display-name="X [v]">
      <event start="T0" end="T1">So it starts out with: A </event>
      <event start="T1" end="T2">roo</event>
      <event start="T2" end="T3">ster crows. </event>
      <event start="T3" end="T4">((1,4s)) </event>
      <event start="T4" end="T5">((breathes in)) </event>
      <event start="T5" end="T6">And theen you see ehm a maan in maybe </event>
      <!-- [...] -->
    </tier>
    <tier id="TIE2" speaker="SPK0" category="nv" type="d" display-name="X [nv]">
      <event start="T0" end="T1">rHA on rKN, lHA on lSH </event>
      <event start="T1" end="T3">rHA up and to the right </event>
      <!-- [...] -->
    </tier>
  </basic-body>
</basic-transcription>
```

Figure 47: Excerpt of an EXMARalDA Basic Transcription

6.3.1.5. FOLKER (FOLK Editor)

Developers:	Thomas Schmidt, Wilfried Schütte, Martin Hartung
URL:	http://agd.ids-mannheim.de/html/folker.shtml
File format documentation:	Example files, XML Schema and (German) documentation of the data model and format on the tool's website

FOLKER was developed for the construction of the FOLK corpus, a spoken language corpus, predominantly addressing researchers in conversation analysis. Being built in parts on

EXMARaLDA technology, FOLKER uses a data model based on STMT. However, the FOLKER XML file format stores transcriptions in a format in which the tier/annotation hierarchy is transformed into an ordered list of speaker contributions, thus bringing the format closer to structures typically used for written language. Optionally, transcriptions can be parsed according to the GAT transcription conventions. In addition to speaker contributions and temporally anchored segments, the file format will then also contain explicit markup for specific discourse entities like words, pauses, breathing etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<folker-transcription>
   <head/>
   <speakers>
      <speaker speaker-id="CLA">
         <name>Clara</name>
      </speaker>
      <speaker speaker-id="JES">
         <!-- [...] -->
      </speaker>
   </speakers>
   <recording path="block.wav"/>
   <timeline>
      <timepoint timepoint-id="TLI_0" absolute-time="0.0"/>
      <timepoint timepoint-id="TLI_1" absolute-time="2.44443"/>
      <timepoint timepoint-id="TLI_2" absolute-time="3.17776"/>
      <timepoint timepoint-id="TLI_3" absolute-time="3.50253"/>
      <timepoint timepoint-id="TLI_4" absolute-time="3.75553"/>
      <timepoint timepoint-id="TLI_5" absolute-time="4.18886"/>
      <timepoint timepoint-id="TLI_6" absolute-time="5.35552"/>
      <timepoint timepoint-id="TLI_7" absolute-time="5.75552"/>
      <timepoint timepoint-id="TLI_8" absolute-time="6.277745"/>
      <timepoint timepoint-id="TLI_9" absolute-time="6.799965"/>
      <!-- [...] -->
   </timeline>
```

```
   <contribution speaker-reference="JES" start-reference="TLI_0"
      end-reference="TLI_4" parse-level="2">
      <uncertain>
         <w>it</w>
      </uncertain>
      <w>doesn</w>
      <w transition="assimilated">t</w>
      <w>matter</w>
      <w>he</w>
      <w transition="assimilated">s</w>
      <w>the</w>
      <w>boyfriend</w>
      <w>of</w>
      <w>one</w>
      <w>of</w>
      <w>your</w>
      <w>friends</w>
      <time timepoint-reference="TLI_1"/>
      <w>and</w>
      <w>as</w>
      <w>long</w>
      <w>as</w>
      <w>they</w>
      <w transition="assimilated">re</w>
      <time timepoint-reference="TLI_2"/>
      <w>atta<time timepoint-reference="TLI_3"/>ched</w>
   </contribution>
   <contribution speaker-reference="CLA" start-reference="TLI_2"
      end-reference="TLI_3"  parse-level="2">
      <w>what</w>
   </contribution>
   <contribution start-reference="TLI_4" end-reference="TLI_5"
      parse-level="2">
      <pause duration="0.43"/>
   </contribution>
   <!-- [.. ]-->
</folker-transcription>
```

Figure 48: Excerpt of a parsed FOLKER transcription file

6.3.1.6. Praat / TextGrid

Developers: Paul Boersma/David Weenink

URL: http://www.fon.hum.uva.nl/praat/

File format documentation: (Sparse) description of the file format inside the tool's help database

Praat is a very widely used piece of software for doing audio annotation and phonetic analysis and thus for creating phonetic corpora. Praat reads and writes several audio formats and several text formats (all based on the same principle) for storing annotation data, acoustic measurements (pitch, intensity), etc. The file format relevant for this section is that of a Text Grid. The textGrid-file format is a plain text format. Different encodings, UTF-8 and UTF-16 among them, can be used. Annotations are organized into tiers and refer to the recording via timestamps. The data model is thus largely similar to the data models underlying ANVIL, ELAN, EXMARaLDA, FOLKER, and TASX.

```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0.0
xmax = 47.01143378716898
tiers? <exists>
size = 6
item []:
  item [1]:
    class = "IntervalTier"
    name = ""
    xmin = 0.0
    xmax = 47.01143378716898
    intervals: size = 9
    intervals [1]:
      xmin = 0.0
      xmax = 1.900016816780248
      text = ""
    intervals [2]:
      xmin = 1.900016816780248
      xmax = 3.2510766568811755
      text = "louder "
    intervals [3]:
      xmin = 3.2510766568811755
      xmax = 31.04000569670313
      text = ""
    intervals [4]:
      xmin = 31.04000569670313
      xmax = 31.500004203597936
      text = "louder "
  item [2]:
    class = "IntervalTier"
    name = "X [v]"
    xmin = 0.0
    xmax = 47.01143378716898
    intervals: size = 53
    intervals [1]:
      xmin = 0.0
      xmax = 1.900016816780248
      text = "So it starts out with: A "
    intervals [2]:
      xmin = 1.900016816780248
      xmax = 2.0931989342595405
      text = "roo"
    intervals [3]:
      xmin = 2.0931989342595405
      xmax = 3.2510766568811755
      text = "ster crows. "
    intervals [4]:
      xmin = 3.2510766568811755
      xmax = 4.646368334964649
      text = "((1,4s)) "
    intervals [5]:
      xmin = 4.646368334964649
      xmax = 5.09300632412194
      text = "((breathes in)) "
    intervals [6]:
      xmin = 5.09300632412194
      xmax = 9.200016816639748
      text = "And theen you see ehm a maan in maybe "
    intervals [7]:
      xmin = 9.200016816639748
      xmax = 10.072686591293524
      text = "his fifties. "
```

Figure 49: Excerpt of a Praat TextGrid

6.3.1.7. Transcriber

Developers: Karim Boudahmane, Mathieu Manta, Fabien Antoine, Sylvain Galliano, Claude Barras

URL: http://trans.sourceforge.net/en/presentation.php

File format documentation: DTD inside the source distribution, demo files inside the binary distribution

Transcriber was originally developed for the (orthographic) transcription of broadcast speech. It uses an XML format which organizes a transcription into one or several sections. Each section consists of one or several speech turns, and each speech turn consists of one or several transcription lines. Background noise conditions can be transcribed independently of the section/turn/line organization of the transcript. All of these units can be timestamped.

```
<Trans version="1" version_date="981211" audio_filename="frint980428" scribe="YM" xml:lang="fr">
 <Topics>
  <Topic id="to1" desc="les titres"/>
 </Topics>
 <Speakers>
  <Speaker id="sp1" name="Simon Tivolle" type="male"/>
  <Speaker id="sp2" name="Patricia Martin" type="female"/>
 </Speakers>
 <Episode program="France Inter" air_date="980428:0700">
  <Section type="filler" startTime="0.000" endTime="4.736">
   <Turn speaker="sp1 sp2" startTime="0.000" endTime="0.387">
    <Sync time="0.000"/>
    <Who nb="1"/> ouais . <Who nb="2"/> sûr ? </Turn>
   <Turn speaker="sp1" startTime="0.387" endTime="4.736">
    <Sync time="0.387"/> ah bon ? <Event desc="rire"/> non . blague , blague de Patricia . <Sync
     time="3.008"/>
    <Event desc="i"/> France-Inter , <Event desc="rire" type="noise" extent="begin"/> il est 7
    heures <Event desc="rire" type="noise" extent="end"/> . </Turn>
```

```
 </Section>
 <Section type="nontrans" startTime="4.736" endTime="9.609">
  <Turn startTime="4.736" endTime="9.609">
   <Sync time="4.736"/>
   <Background time="4.736" type="music" level="high"/>
   <Background time="9.609" type="other" level="off"/>
  </Turn>
 </Section>
 <Section type="filler" startTime="9.609" endTime="10.790">
  <Turn speaker="sp2" startTime="9.609" endTime="10.790">
   <Sync time="9.609"/> le journal , Simon Tivolle : </Turn>
 </Section>
 <Section type="report" topic="to1" startTime="10.790" endTime="20.000">
  <Turn speaker="sp1" startTime="10.790" endTime="20.000">
   <Sync time="10.790"/>
   <Event desc="i"/> bonjour ! <Sync time="11.781"/>
   <Background time="11.781" type="music" level="high"/>
   <Sync time="12.237"/> mardi 28 avril . <Sync time="13.344"/> la consultation nationale sur les
   programmes des lycées : <Sync time="16.236"/>
   <Event desc="i"/> grand débat aujourd'hui et demain à Lyon <Sync time="18.521"/> pour tirer les
   enseignements du </Turn>
 </Section>
 </Episode>
</Trans>
```

Figure 50: Transcriber file

6.3.1.8. Other tools[44]

There are numerous other tools for doing media annotation most of which use their own format. Since we believe the afore-mentioned tools to be the most relevant ones, we restrict ourselves to a short overview of the others here.

- **EMU Speech Database System** [http://emu.sourceforge.net/] – EMU is "a collection of software tools for the creation, manipulation and analysis of speech databases. At the core of EMU is a database search engine which allows the researcher to find various speech segments based on the sequential and hierarchical structure of the utterances in which they occur. EMU includes an interactive labeller which can display spectrograms and other speech waveforms, and which allows the creation of hierarchical, as well as sequential, labels for a speech utterance." (quote from the website) EMU reads and writes ESPS formatted label files as produced by ESPS and Waves+ software from Entropic. The system can also import Praat TextGrids.
- **Wavesurfer** (Developers: Kåre Sjölander and Jonas Beskow) [http://www.speech.kth.se/wavesurfer/] – Wavesurfer is a tool for sound visualization and manipulation, mainly used for the construction of speech corpora. It reads several formats commonly used for such corpora, namely HTK/MLF, TIMIT, ESPS/Waves+, and Phondat. Wavesurfer supports different encodings, Unicode encodings among them.
- **Phon** (Developers: Greg Hedlund and Yvan Rose) [http://childes.psy.cmu.edu/phon/] – Phon is a relatively new software "designed to facilitate phonological and phonetic analysis of data transcribed in CHAT" (quote from the website). The tool uses its own XML-based format, but should be largely compatible with the CHAT format.
- **XTrans** [http://www.ldc.upenn.edu/tools/XTrans/] – XTrans is "a next generation multi-platform, multilingual, multi-channel transcription tool developed by Linguistic Data Consortium (LDC) to support manual transcription and annotation of audio recordings" (quote from the website). It reads and writes a tabular separated text format.

[44] **AnColin** and **iLex** should probably also be mentioned here as tools used for constructing sign language corpora. However, I could not find sufficient information on the tools' data formats – references are (Vauquois, 1968) and (Hanke and Storz, 2008).

- **TASX Annotator** (Developer: Jan-Torsten Milde, no URL available anymore) – The TASX annotator is a tool similar in design to ANVIL, ELAN and EXMARaLDA. It uses an XML-based data format representing a multi-layered annotation graph. Development of the tool was abandoned some time ago. The tool itself is not offered for download anymore, but some corpora created with it are available.
- **WinPitch** [http://www.winpitch.com/] – WinPitch is a windows based speech analysis tool, comparable in its functionality to (but probably not as widely used as) Praat. It uses an XML-based data format representing a multi-layered annotation graph.
- **Annotation Graph Toolkit** [http://agtk.sourceforge.net/] – The Annotation Graph Toolkit (AGTK) comprises four different tools all of which are based on the same software library and tool architecture. **TableTrans** is for observational coding, using a spreadsheet whose rows are aligned to a signal. **MultiTrans** is for transcribing multi-party communicative interactions recorded using multi-channel signals. **InterTrans** is for creating interlinear text aligned to audio. **TreeTrans** is for creating and manipulating syntactic trees. The tools are intended as a proof-of-concept for the annotation graph framework (see below). Their data format is the XML-based ATLAS interchange format (see below). Development of the AGTK was abandoned at a relatively early stage – the tools have probably not been widely used in practice.
- **Transana** (Developers: Chris Fassnacht and David K. Woods) [http://www.transana.org/] – Transana is a tool for managing, transcribing and analyzing digital audio and video recordings. Transcripts, keywords, video segmentations etc. are stored internally in a MySQL database. An XML export is provided for these data. However, in this export, transcripts are represented as one big stretch of character data, interspersed with RTF formatting instructions. Thus, there is no real content-oriented markup of the transcription. Moreover, when the transcription contains timestamps, the tool produces a non-well-formed XML export. Transana data are therefore rather problematic in terms of exchangeability and interoperability.
- **F4** [http://www.audiotranskription.de/f4.htm] – F4 may be seen as a typical exponent of a further class of annotation tools, namely simple combinations of a text editor with a media player. Other tools belonging to this class are the **SACODEYL Transcriptor** (http://www.um.es/sacodeyl/), **Casual Transcriber** [https://sites.google.com/site/casualconc/utility-programs/casualtranscriber/], or **VoiceScribe** [https://www.univie.ac.at/voice/page/voicescribe]. Such tools are widely used for quickly creating text transcripts linked to media files. However, they have in common that they do not produced structured data which could be systematically exploited for further automatic processing. Rather, they use some free (plain or rich) text format with special constructs for linking into media. For the purposes of CLARIN, such data will not be usable without further curation.

6.3.2. Generic formats and frameworks

6.3.2.1. TEI transcriptions of speech

Chapter 8 of the TEI Guidelines is dedicated to the topic of "Transcriptions of Speech". The chapter defines various elements specific to the representation of spoken language in written form, such as

- utterances,
- pauses,
- vocalized but non-lexical phenomena such as coughs,
- kinesic (non-verbal, non-lexical) phenomena such as gestures, etc.

These elements can be used together with elements defined in other chapters to represent multimedia annotations. In particular, elements from chapter 16 (“Linking, Segmentation, and Alignment”) can be used to point from the annotation into a recording, to represent simultaneity of events, etc. Furthermore, elements defined in chapter 3 (“Elements Available in All TEI Documents”) and chapter 17 (“Simple analytic mechanisms”) can be relevant for multimedia annotations.

The entirety of the elements and techniques defined in the TEI guidelines is certainly suited to deal with most issues arising in multimedia annotation. A problem with the TEI approach is rather that (in the general form as formulated the guidelines at least) it contains too many degrees of freedom. For most phenomena and for synchronisation of text and media in particular, there are always several options to express one and the same fact (e.g. the fact that two words are uttered simultaneously). Therefore, if the TEI guidelines are to be used with an annotation tool, more specific rules will have to be applied. However, few tools or corpus projects so far have developed such more specific TEI based formats for multimedia annotation.[45] Among the existing examples are:

- the French CLAPI database [http://clapi.univ-lyon2.fr/analyse_requete.php] which offers a TEI-based export of transcription files,
- the Modyco project [http://www.modyco.fr/corpus/colaje/viclo/] which uses a TEI-based format as an interlingua between ELAN and CHAT annotations (see (Parisse and Morgenstern, 2010)),
- EXMARaLDA which contains options for importing and exporting TEI-conformant data (see (Schmidt, 2005)),

Besides proving the general applicability of TEI to multimedia annotations, these examples also demonstrate that being TEI-conformant alone does not lead to improved interoperability between annotation formats – none of the TEI examples mentioned is compatible with any of the others in the sense that data could be exchanged between the tools or databases involved.

These difficulties notwithstanding, the TEI guidelines certainly contain important observations which may become useful when defining a comprehensive standard for multimedia annotations. In particular, the fact that they are part of a framework which also (even: chiefly) deals with different types of written language data, is a good (and probably the only) point of contact for bringing together written corpora and multimedia corpora.

6.3.2.2. Annotation Graphs / Atlas Interchange Format / Multimodal Exchange Format

Annotation graphs (AGs, (Bird and Liberman, 2001)) are an algebraic formalism intended as “a formal framework [doing] for linguistic annotation”. The authors explicitly state that, in a three-layer-architecture as commonly assumed for databases, AGs belong to the logical, not to the physical level. They thus formulate a data model rather than a data format. In principle, one and the same AG can be represented and stored in various physical data structures, e.g. an XML file, a text file or a relational database.

[45] (Bird and Liberman, 2001), p. 26, say:

> The TEI guidelines for 'Transcriptions of Speech' […] offer access to a very broad range of representational techniques drawn from other aspects of the TEI specification. The TEI report sketches or alludes to a correspondingly wide range of possible issues in speech annotation. All of these seem to be encompassed within our proposed framework [i.e. Annotation Graphs, T.S.], but it does not seem appropriate to speculate at much greater length about this, given that this portion of the TEI guidelines does not seem to have been used in any published transcriptions to date.

The basic idea of AGs is that "all annotations of recorded linguistic signals require one unavoidable basic action: to associate a label or an ordered set of labels, with a stretch of time in the recording(s)". Bird and Liberman's suggestion is therefore to treat annotations as Directed Acyclic Graphs (DAGs) whose nodes represent (or point to) timepoints in a recording, and whose arcs carry the non-temporal information, i.e. the actual text, of the annotation. They also allude to various ways of adding additional structure to such a DAG (e.g. by typing arcs or partitioning arcs into equivalence classes), but consciously leave the issue open to be resolved by concrete applications. In that sense, many of the data models described in the previous section (e.g. ANVIL, ELAN, EXMARaLDA) can be understood as applications of the AG framework, which specify and restrict a general AG to a subclass which can be efficiently handled by the respective tool.

One way of physically representing an annotation graph is level 0 of the ATLAS Interchange Format (AIF, DTD available from http://xml.coverpages.org/aif-dtd.txt)

AIF defines an XML format which represents all the components of an AG as XML elements. The format, or a subset thereof, is used by different tools from the Annotation Graph Toolkit (AGTK, see above).

Taking the idea of AGs as a starting point, the developers of several of the tools described in the previous section devised a multimodal annotation format in which the common denominator information shared by the tools is represented in an AIF file. The format and the underlying analysis of the respective tool formats are described in (Schmidt et al., 2008; Schmidt et al., 2009). The format is supported by various (built-in or stand-alone) import and export filters. Although it is in practice less useful than a direct exchange method between any two tools, it can be seen as a first step towards a standardisation of different AG-based tool formats.

6.3.2.3. NXT (NITE XML Toolkit)

NXT (Carletta et al., 2003) is a set of libraries and tools designed to support the representation, manipulation, query and analysis of multimedia language data. NXT is different from the afore-mentioned formats and frameworks because its focus is not on transcription or direct annotation of media data, but rather on the subsequent enrichment of data which has already been transcribed with some other tool. Part of NXT is the NITE Object Model, a data model specifying how to represent data sets with multiple, possibly intersecting hierarchies. Serialization of the data model is done through a set of interrelated XML files in which annotations can point to other annotations in different files (i.e. NXT uses standoff annotation).

6.3.3. Other formats

There are at least two other formats, which are neither associated with a specific tool nor intended as generic formats, but which are widely cited in the literature and have been applied to larger bodies of multimedia data and may therefore be relevant for the scope of this document:

- **BAS Partitur Format** (Schiel et al., 1998) – The BAS (Bavarian Archive of Speech Signals) has created the Partitur format based on their experience with a variety of speech databases. The aim has been to create "an open (that is extensible), robust format to represent results from many different research labs in a common source." The Partitur-Format is probably relevant only for speech corpora, not for the other types of corpora described above.
- **TIMIT** (Fisher et al., 1986) – TIMIT is a corpus of phonemically and lexically transcribed speech of American English speakers of different sexes and dialects, i.e. a phonetic corpus in

the sense defined above. Phonetic (phoneme-wise) and orthographic (word- and utterance-wise) transcriptions are stored in individual tabular-separated text files. Timestamps, i.e. start and end offsets for each transcription unit, can be used to refer the different text files to one another.

6.3.4. Interoperability of tools and formats

Interoperability between tools and formats, at this point in time, usually means that a converter exists to directly transform one tool's format into that of another (i.e. interoperability is usually not achieved via a pivot format or a tool-external standard). In most cases, such converters are built into the tools in the form of an import or an export filter. Filters may be implemented as XSLT stylesheets or as pieces of code in some other programming language. The following table provides an overview of import and export filters integrated in the most widely used tools:

Tool	Imports	Exports
ANVIL	ELAN, Praat	---
CLAN	ELAN, Praat	ELAN, EXMARaLDA, Praat
ELAN	CHAT, Praat, Transcriber, (ShoeBox, Toolbox, FLeX)	CHAT, Praat, (ShoeBox, Toolbox, TIGER)
EXMARaLDA	ANVIL, CHAT, ELAN, FOLKER, Praat, Transcriber, (TASX, WinPitch, HIAT-DOS, syncWriter, TEI, AIF)	CHAT, ELAN, FOLKER, Praat, Transcriber, (TASX, TEI, AIF)
FOLKER	EXMARaLDA	ELAN, EXMARaLDA (TEI)
Praat	---	---
Transcriber	CHAT (ESPS/Waves, Timit, several NIST formats)	Several formats, but none of the ones discussed here

The following matrix pairs off the different tools, showing where a direct interoperability in the form of an import or export filter exists.

	Imports							Exports						
	ANVIL	CHAT	ELAN	EXMARaLDA	FOLKER	Praat	Transcriber	ANVIL	CHAT	ELAN	EXMARaLDA	FOLKER	Praat	Transcriber
ANVIL		-	+	-	-	+	-		-	-	-	-	-	-
CLAN	-		+	-	-	+	-	-		+	+	-	+	-
ELAN	-	+		-	-	+	-	-	+		-	-	+	-
EXMARaLDA	+	+	+		+	+	+	-	+	+		+	+	-
FOLKER	-	-	-	+		-	-	-	-	+	+		-	-
Praat	-	-	-	-	-		-	-	-	-	-	-		-
Transcriber	-	+	-	-	-	-		-	-	-	-	-	-	

Taking transitivity into account (if tools A and B are interoperable, and tools B and C are interoperable, then A and C are interoperable via B), there seems to be, in principle, a way of exchanging data between any two of these seven tools. Furthermore, for some pairs of tools, there is more than one way of exchanging data (e.g. ELAN imports CHAT, and CLAN also exports ELAN). In practice, however, interoperability in its present form has to be handled with great care for the following reasons:

- Information may be lost in the conversion process because the target format has no place for storing specific pieces of information contained in the source format (e.g. when exporting Praat from ELAN, information about speaker assignment will be lost).
- For similar reasons, information may be reduced or structurally simplified in the conversion process (e.g. EXMARaLDA transforms structured annotations into simple annotations when importing certain tier types from ELAN).
- Some converters rely on simplified assumptions about the source or target format and may fail when faced with their full complexity (e.g. CLAN's EXMARaLDA export will fail when the source transcriptions are not fully aligned).
- Since there is no common point of reference for all formats and data models, different conversion routes between two formats will usually lead to different results (e.g. importing Praat directly in ANVIL will not result in the same ANVIL file as first importing Praat in ELAN and then importing the result in ANVIL).

Lossless roundtripping between tools is therefore often not possible, and any researcher working with more than one tool or format must handle interoperability issues with great care. Thus, although the existing interoperability of the tools is useful in practice, a real "standardization" would still be an important improvement.

6.3.5. Transcription conventions / Transcription systems

6.3.5.1. Systems for phonetic transcription

- **IPA** (International Phonetic Alphabet) – The International Phonetic Alphabet can be regarded as one of the longest-standing standards in linguistics. Its development is controlled by the International Phonetic Association. IPA defines characters for representing distinctive qualities of speech, i.e. phonemes, intonation, and the separation of words and syllables. IPA extensions also cater for additional speech phenomena like lisping etc. According to Wikipedia, there are 107 distinct letters, 52 diacritics, and four prosody marks in the IPA proper. Unicode has a code page (0250-02AF) for IPA symbols (IPA symbols that are identical with letters of the Latin alphabet, are part of the respective Latin-x codepages).
- **SAMPA, X-SAMPA** ((Extended) Speech Assessment Methods Phonetic Alphabet, (Gibbon et al., 1997; Gibbon et al., 2000)) – SAMPA and X-SAMPA are mappings of the IPA into a set of symbol included in the 7-bit-ASCII set. The mapping is isomoprhic so that a one-to-one transformation in both directions can be carried out (see, for instance, http://www.theiling.de/ipa/). Many speech corpora and pronunciation lexicons have been transcribed using SAMPA. As Unicode support in operating systems and applications gains ground, SAMPA and X-SAMPA will probably become obsolete over time.
- **ToBi** (Tones and Break Indices) – „ToBI is a framework for developing community-wide conventions for transcribing the intonation and prosodic structure of spoken utterances in a language variety. A ToBI framework system for a language variety is grounded in research on the intonation system and the relationship between intonation and the prosodic structures of the language (e.g., tonally marked phrases and any smaller prosodic constituents that are distinctively marked by other phonological means).” (quote from http://www.ling.ohio-state.edu/~tobi/). ToBi systems are available or under development for different varieties of English, German, Japanese, Korean, Greek, different varieties of Catalan, Portuguese, Serbian and different varieties of Spanish.

6.3.5.2. Systems for orthographic transcription

There are numerous systems for doing orthographic transcription. Many of them are language specific or have at least been used with one language only. Also, many of them are documented only sparsely or not at all. The following list therefore includes only such systems for which an accessible documentation exists and which are known to be used by a larger community and/or for larger bodies of data.

- **CA** (Conversation Analysis, (Sacks et al., 1978)) – In an appendix of (Sacks et al., 1978), the authors sketch a set of conventions for notating transcripts to be used in conversation analysis. The conventions consist of a set of rules about how to format and what symbols to use in a type-written transcript. They have been transferred later to be used with text-processors on computers, but there is no official documentation of a computerized CA, let alone a document specifying the symbols to be used as Unicode characters. Although never formulated in a more comprehensive manner, the CA conventions have been widely used and have inspired or influenced some of the systems described below.
- **CHAT** (Codes for the Human Analysis of Transcripts, (MacWhinney, 2000)) – Besides being a text-based data format (see above), CHAT is also a transcription and coding convention. Analogous to the CLAN tool, it was originally developed for the transcription

and coding of child language data, but now also contains a CA variant for use in conversation analysis (in a way, this could be seen as the (or one) computerized variant of CA – see above). Since it is so closely tied to the CHAT format and the CLAN tool, many aspects relevant for computer encoding (e.g. Unicode compliancy) have been treated in sufficient detail in the conventions. A special system for the transcription of bilingual data, LIDES (Barnett et al. 2000), was developed on the basis of CHAT.

- **DT/DT2** (Discourse Transcription, (DuBois et al., 1993)) – DT is the convention used for transcription of the Santa Barbara Corpus of Spoken American English. It formulates rules about how to format and what symbols to use in a plain text transcription, including timestamps for relating individual lines to the underlying recording. DT2 is an extension of DT. It contains a table which specifies Unicode characters for all transcription symbols.
- **GAT/GAT2/cGAT** (Gesprächsanalytisches Transkriptionssystem, (Selting et al., 2009)) – GAT is a convention widely used in German conversation analysis and related fields. It uses many elements from CA transcription, but puts a special emphasis on the detailed notation of prosodic phenomena. The original GAT conventions explicitly set aside all aspects of computer encoding of transcriptions. To a certain degree, this has been made up for in the recently revised version, GAT 2. cGAT is based on a subset of the GAT 2 conventions and formulates explicit rules, including Unicode specifications of all transcription symbols, for computer-assisted transcription in the FOLKER editor (see above).
- **GTS/MSO6** (Göteborg Transcription Standard, Modified Standard Orthography, (Nivre, 1999; Nivre et al., 2005)) – According to its authors, GTS is a "standard for machine-readable transcriptions of spoken language first used within the research program Semantics and Spoken Language at the Department of Linguistics, Göteborg University." It consists of two parts, one language independent part called GTSG (GTS General), and one language dependent part. The MSO. GTS, however does not necessarily require MSO. GTS in combination with MSO is the basis for the Göteborg Spoken Language Corpus.
- **HIAT** (Halbinterpretative Arbeitstranskriptionen, (Ehlich and Rehbein, 1976; Ehlich, 1993; Rehbein et al., 2004)) – HIAT is a transcription convention originally developed in the 1970s for the transcription of classroom interaction. The first versions of the system (Ehlich and Rehbein, 1976) were designed for transcription with pencil or typewriter and paper. HIAT's main characteristic is the use of so-called Partitur (musical score) notation, i.e. a two-dimensional transcript layout in which speaker overlap and other simultaneous actions can be represented in a natural and intuitive manner. HIAT was computerized relatively early in the 1990s in the form of two computer programs – HIAT-DOS for DOS (and later Windows) computers, and syncWriter for Macintoshes. However, standardization and data exchange being a minor concern at the time, these data turned out to be less sustainable than their non-digital predecessors. The realization in the HIAT community that data produced by two functionally almost identical tools on two different operating systems could not be exchanged and, moreover, the prospect that large existing bodies of such data might become completely unusable on future technology was one of the major motivations for initiating the development of EXMARaLDA. The most recent version of the conventions (Rehbein et al., 2004) therefore contains explicit instructions for carrying out HIAT transcription inside EXMARaLDA (or Praat).
- **ICOR** (Interaction Corpus, http://icar.univ-lyon2.fr/documents/ICAR_Conventions_ICOR_2007.doc) – ICOR is the transcription convention used for transcriptions in the French CLAPI database. As formulated in the cited

document, it is a convention for producing plain text files. However, the fact that CLAPI offers TEI versions of all ICOR transcripts shows that there is a conversion mechanism for turning ICOR text transcriptions into XML documents.

6.3.5.3. Systems for sign language transcription

- HamNoSys (Hamburger Notationssystem für Gebärdensprachen, (Prillwitz and others, 1989))
- Stokoe notation

6.3.5.4. Commonly used combinations of formats and conventions

Although, in principle, most of the tools described in section 6.3.1 could be used with most transcription systems described in this section, most communities in the humanities have an outspoken preference towards a specific combination of tool and transcription system. The following lists some widely used such combinations:

- CLAN + CHAT is widely used for doing transcription and coding of child language,
- CLAN + CA is widely used for doing transcription in conversation analysis,
- EXMARaLDA + HIAT is widely used for doing transcription in functional-pragmatic discourse analysis,
- FOLKER + GAT is widely used for doing transcription in interactional linguistics, German conversation analysis and related fields,
- ICOR + TEI (though strictly speaking not a combination of a convention and a tool) is the basis of the French CLAPI database.

6.4. Summary / Recommendations

Author: Thomas Schmidt

6.4.1. Media Formats

At this point in time, no clear recommendation can be given what kind of media encoding formats will be supported in CLARIN. As a general rule, data contributors should be encouraged to:

1. record audio and video in a quality that is sufficiently high to capture all aspects of the signal that you may need in your analyses (e.g. for speech recordings, use 16 bit 44.1/48 kHz)
2. prefer uncompressed formats over compressed ones if possible – if it is not possible, lossless compressions should be preferred over lossy ones,
3. avoid proprietary formats wherever possible,
4. keep and archive the original recording even if the actual processing (annotation etc.) is done on a compressed or otherwise transcoded version,
5. document editing (cutting etc.) and transcoding operations that were carried out on the original data.

6.4.2. Media Annotation

There is, to date, no widely dominant method, let alone an official standard, for doing media annotation. Considering the heterogeneity of the corpus types and research communities involved, this is probably not too surprising. However, the number of tools and formats actually used has a manageable proportion. Moreover, there are obvious conceptual commonalities between them, and they interoperate reasonably in practice. Using these (i.e. the first seven discussed in section 3.1.) tools as a basis, and combining the approaches of AG and TEI, a real standard, suitable to be used in a digital infrastructure, does not seem to be an unrealistic goal.

Until such a standard is defined, however, researchers doing media annotation need recommendations about which tools and formats to use in order to maximize their chances of being standard compliant eventually. The following criteria may serve as guidelines for such a recommendation:

- The format should be XML based, since XML has become the virtually unrivalled solution for representing structured documents and further processing will be easier if files are in XML.
- The tool and format should support Unicode, since only this – equally unrivalled standard – ensures that data from different languages can be reliably exchanged.
- The format should be based on a format-independent data model (or it should be possible to relate the format to such a data model) since this greatly facilitates the integration of data into an infrastructure which may require diverse physical representations of one and the same piece of data.
- The tool should be stable, its development should be active, since the definition of a standard or the integration of a tool (format) into an infrastructure may require adaptations of the tool or at least information from its developer(s).
- The tool should run on different platforms, since the targeted user community of the infrastructure will also have diverging platform preferences.
- It should be possible to import the tool format into or export it to other tools, since this demonstrates a basic ability to interoperate.

According to these criteria, there are at least four tools in the above list which can be **recommended without restrictions**, i.e. they meet all of the criteria:

- **ANVIL**
- **ELAN**
- **EXMARaLDA**
- **FOLKER**

From the more widely used tools, three remain which do not meet all of the criteria, but which can still be **recommended with some restrictions**:

- **Praat** – the only objection to Praat is that its data format is not XML based. This makes an initial transformation of Praat data somewhat more difficult. However, since the format itself is rather simple and the underlying data model well understood, and since, furthermore several tools provide converters for transforming Praat data into XML, this objection is a minor one.

- **CHAT** – CHAT's main drawback is the lack of an explicit data model, making CHAT data somewhat more closely tied to, and more dependent on, the application they were created with. However, considering CHAT's wide user base and the huge amounts of CHAT data available in CHILDES and Talkbank, it would probably not be a good idea to exclude CHAT from a standardization effort. Some of the problems arising from the lack of a data model can be alleviated by making full use of the CLAN tool's internal checking mechanisms (see below) and maybe also by transforming CHAT data into the Talkbank XML format.
- **Transcriber** – Transcriber's main problem is that the development is not active anymore. A new version (based on annotation graphs) has been announced for quite some time now, but the envisaged release data (2nd quarter of 2009) has long passed without a new version having been released. However, the tool has been sufficiently used in practice and its format seems to be sufficiently well understood also by other applications to make it worthwhile considering Transcriber in a standardization effort.

For similar reasons (i.e. development not active), some of the tools can only be **recommended with severe restrictions**:

- **TASX**'s development seems to have been abandoned, the tool itself not officially available anymore
- **AG toolkit**'s development was abandoned at a relatively early stage
- **WinPitch**'s development status is unclear

Otherwise, however, these three tools meet most of the criteria specified above.

A few of the tools mentioned in 6.3.1.8 **disqualify** for a standardization effort, because their file formats lack a sound structural basis. This is the case for **Transana** as well as for **F4** and similar tools.[46]

Two further recommendations can be made to data creators or curators:

First, they should be encouraged to use the full palette of tool-internal mechanisms for ensuring consistency, validating data structures, etc., since this is likely to increase the reliable processability and exchangeability of data. More specifically, this means:

- In **ANVIL**, specification files should be carefully designed and used consistently.
- In **CLAN**, the check command should be used on completed data, and a Talkbank XML version should be generated for completed data.
- In **ELAN**, linguistic types for tiers should be carefully designed and used consistently. If possible, linguistic types should be associated with categories of the ISOCat registry.
- In **EXMARaLDA**, basic transcriptions should be checked for structure errors and segmentation errors. A segmented transcription should be generated for completed basic transcriptions.

[46] The remaining tools described in 6.3.1.8, but not yet touched on in this section (Wavesurfer, EMU, Phon and XTrans), are probably all more on the "recommendable" side of the scale, but I (TS) do not feel competent to judge this. It should maybe also be mentioned that there are some legacy tools around whose development was abandoned 10 years or more ago and which can, for all practical purposes, be regarded as obsolete. However, larger bodies of data still exist which were created with these tools and which may have to be curated for integration into a digital infrastructure. **syncWriter**, **HIAT-DOS** and **MediaTagger** are three such tools (see (Schmidt and Bennöhr, 2008) for a description of the first two).

- In **FOLKER**, the mechanism for checking the syntax and temporal structure of transcriptions should be enabled.

Second, they should be encouraged to use an established (i.e. tested and document) combination of tools and conventions as described in section 6.3.5.4. Where none of the established combinations meets their demands, new conventions should be developed as extensions or modifications of existing ones.

7. Translation

Authors: Ineke Schuurman, Inguna Skadina and Jan Odijk

In the following we first present an overview of the topics we want to address, mentioning in a separate paragraph those topics related to **translation** that are likely to be addressed in other parts of this deliverable.

In the remainder of this section we will present short characteristics of the standards[47] and best practices involved. While the focus is on HSS researchers with little or no computer literacy, tools and resources requiring more knowledge of the field will also be mentioned.

We will only address standards and best practices with respect to **machine translation** (MT), and therefore we will deal with MT engines, evaluation algorithms, translation memory (TM) data exchange formats, etc. (see Table 1).

The following issues are outside the scope of this chapter:

- Terminology (like TBX) and multilingual dictionaries
- Alignment (several levels)
- Annotation (several levels)
- Text segmentation (SRX, GRX, …)
- Tools for processing of bi/multilingual corpora (Paraconc, ...)
- Input format/text encoding (UTF-8, UTF-16, Latin-1)

Our starting point was FlaReNet deliverable D4.1, *Identification of problems in the use of LR standards and of standardization needs* (esp. section 3.2), by Gerhard Budin. His audience, however, differs from the academic one addressed in CLARIN. Moreover, several issues addressed in this section were not covered in the FlaReNet deliverable.

[47] There are no official standards yet for (machine) translation, translation memories and/or evaluation metrics. Most are either *de facto* standards or best practices.

Name	Best practice / Standard	Recommendation	Function	User	Comment
Machine translation					
Moses	+	++	SMT Engine with example full systems (SMT)	comp.lit[48]	phrase (non ling) based SMT and factored
Pharaoh		+	SMT engine	comp.lit	phrase-based
GIZA++	+	+	Training of translation model	comp.lit	
SDL Trados	+	+	Translation Memory	comp.illit	
Joshua			SMT tool kit	comp.lit	statistical hierarchical phrase-based machine translation
Evaluation metrics					
BLEU	+	+		comp.lit	
NIST	+	+		comp.lit	
TER	+			comp.lit	
PER	+			comp.lit	
ROUGE			Evaluation, widely used in summarization	comp.lit	
METEOR	+			comp.lit	
WER		+		comp.lit	
Data exchange					
TMX	++	+	Translation memory exchange format	comp.illit	

Table 1. Overview of standards and best practices in automated translation

+ means *partly*, ++ *strong* in column *best practice / standard*; + means "recommended" in column *recommendation*

[48] 'Comp.illit' is used as abbreviation for computer-illiterate, 'comp.lit' for computer literate.

7.1.1. Machine Translation

There are several approaches in Machine Translation (MT)

1. Rule-Based Machine Translation (RBMT), example: Systran, PROMT
2. Statistical Machine Translation (SMT), example: Google Translate, MOSES
3. Example-Based Machine Translation (EBMT)
4. Hybrid approaches

Rule-Based Machine Translation (RBMT), the oldest approach, is characterized by the use of (mostly hand-made) linguistic rules in the translation process. The source language text is analyzed on different levels (e.g. morphological, syntactic and sometimes semantic). Rules are written to convert source language representations into target language ones. Two methods are used for doing this: the interlingua method, and the transfer method (see Figure 1).

When using an interlingua, the sentence to be translated is first analyzed and represented in an abstract, language-independent way (for example in terms of semantic notions), and from there a target language sentence is generated.

In a transfer-based system, the source language sentence is analyzed (rule-based). Rules are written to convert structures in language A directly into language B, so-called transfer rules.

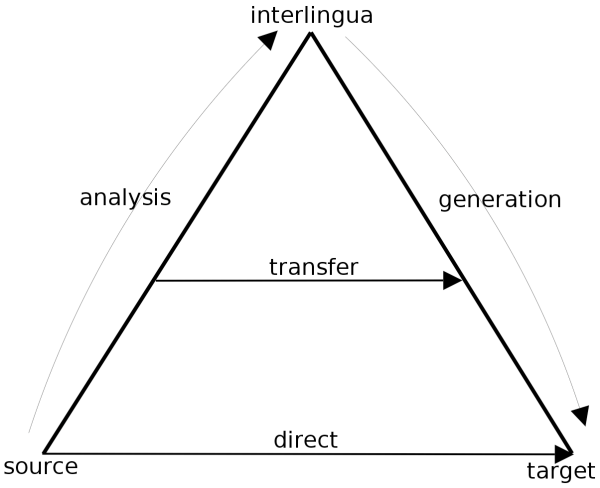


Figure 51: MT pyramid suggested by (Vauquois, 1968)

Adding a new language-pair to an RBMT system is very time-consuming. An advantage, however, is that one does not need a (huge) parallel corpus (which are not available for all language pairs and for all domains).

Most pure RBMT systems are nowadays the commercial ones; in research statistical and hybrid approaches are in vogue.

Statistical Machine Translation systems (SMT) make use of huge parallel and monolingual corpora with aligned source and target language information. A translation model is learnt from a parallel corpus, and a language model from a target language corpus. Statistics (maximizing probabilities) play a key role. Currently phrase-based models, i.e. any sequence of words (where "phrase" has a different meaning than the same concept in linguistics), are dominant. However for morphologically rich language usually factored models are used.

Adding a new language pair/domain is relatively easy for people with the relevant technical background, provided that a large enough (huge) parallel corpus is available. This is the weak point of the SMT approach, as such corpora are relatively scarce.

Example-Based Machine Translation (EBMT) makes use of techniques from both RBMT and SMT. Like SMT, EBMT makes use of a parallel, aligned corpus and a database of translated (parts of) sentences (as in a TM). Sometimes rules are used to reason with similarities between an example and a sentence to be translated.

Recently, more and more hybrid systems are being developed, for example deriving transfer rules automatically from a parallel corpus, using fully parsed corpora in SMT, adding translated (and post-edited) sentences into the system as a new parallel corpus, etc. etc.

A full system based on SMT, e.g. example MT-systems based on MOSES[49], can be used by HSS researchers, especially since training for domain and language pair involved has been done

7.1.2. Evaluation Methods and Metrics

For an overview of Evaluation in general, see the ELRA HLT Evaluation Portal.

FEMTI

FEMTI, a Framework for Machine Translation Evaluation within the ISLE initiative (International Standards for Language Engineering), is "an attempt to organize the various methods that are used to evaluate MT systems, and to relate them to the purpose and context of the systems". In order to do this, FEMTI provides two interrelated classifications or taxonomies. The first classification enables evaluators to define an intended context of use for the MT system to evaluate. Each feature is then linked to relevant quality characteristics and metrics, defined in the second classification.

The aim of FEMTI is to help two types of users:

- Final users of MT systems. They can select the quality characteristics that are most important to them and thereby choose the MT system that best suits these characteristics.
- Designers and developers of MT systems. They can browse and select the characteristics that best reflect their circumstances, and thereby find associated evaluation measures and tests. They can also learn about the needs of users and find niche applications for their system.

The automated evaluation metrics described below are not meant to address all issues mentioned in FEMTI.

Automated Evaluation

The trend to automate evaluation in the area of MT was initiated by seminal work by (Papineni et al., 2002). They introduced BLEU as an evaluation metrics. BLEU makes it possible to automatically measure (aspects of) translation quality of a candidate translation against a reference corpus which contains one or many reference translations, and it is claimed that it yields results that correlate highly with human judgments of quality at the corpus level.

The NIST metric is derived from the BLEU metric, with assignment of more weight to n-grams that are rarer (Doddington, 2002).

[49] http://www.statmt.org/moses/?n=Public.Demos

Position-independent word error rate (PER), is based on the Word Error Rate (WER), familiar from speech recognition evaluation. WER is a measure to indicate the number of differences between two texts in terms of insertions, deletions and substitutions. PER, in contrast to WER, allows for re-ordering of words and sequences of words between a translated text and a reference translation.

The Translation Edit Rate (TER), also (erroneously) called Translation Error Rate, metrics measures the number of edits needed to change a system output so that it exactly matches a given reference. It differs from WER in that it counts a shift of any sequence of words over any distance as a single edit; hence this has the same costs as an insertion, deletion or substitution (Snover et al., 2006). TERp (or TER-Plus) is a promising variant of TER, cf. (Snover et al., 2010).

The METEOR metric is designed to address some of the deficiencies inherent in the BLEU metric which focuses on precision. METEOR also includes recall aspects, by using the weighted harmonic mean of unigram precision and unigram recall (Lavie 2004). It is claimed to achieve a higher correlation than BLEU and NIST which are based on precision alone. METEOR also includes a mechanism for synonym matching so that also synonyms for words yield a match.

Many other metrics, usually small variants of the metrics mentioned above, have been proposed and are being tested.

Evaluation metrics inspired by BLEU have also been used for domains other than MT, e.g. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and its variants for text summarization (Lin, 2004). Variants of ROUGE (ROUGE-L, ROUGE-S and ROUGE-W) were especially proposed for the evaluation of MT, cf. (Lin and Och, 2004).

Automated MT evaluation tools, such as BLEU or NIST, initially proposed for assessing the progress in the development of individual MT systems, have also been used in other evaluation scenarios, e.g., for comparing translation quality of different MT systems, or for assessing the difficulty of specific genres and text types for MT. (Babych et al., 2004), for predicting human evaluation scores and acceptability thresholds for MT systems on the basis of automated evaluation scores. However, some of such scenarios were later shown to be problematic. For instance, it is not possible to make a meaningful comparison of Rule-Based vs. Statistical MT systems using BLEU, because this metric over-rates the quality of SMT (Callison-Burch et al., 2006). Also, absolute values of BLEU-type scores are not meaningful without comparison with the values obtained under the same experimental conditions, because they cannot predict human evaluation scores without setting specific calibration parameters (e.g., slope and the intercept of the regression line), but these parameters are different for different text types and target languages (Babych et al., 2005). Therefore, MT systems cannot be meaningfully assigned BLEU scores for comparison with other MT systems across the board, and need to be evaluated on the case-by-case basis under controlled experimental conditions. As a result, the validity of many automated MT evaluation metrics for industrial use has been questioned (Thurmair, 2007).

7.1.3. Usage scenarios for automated evaluation metrics

The Centre for Translation Studies of the University of Leeds focuses on the tasks of developing usage scenarios for automated MT evaluation tools. These scenarios will be focused on possible industrial applications. In particular, in previous work researchers from the University of Leeds showed that N-gram based evaluation scores like BLEU loose sensitivity for higher-quality MT systems, while performance-based metrics, such as metrics based on Information Extraction tasks from MT output, don't show the loss of sensitivity across a wider quality spectrum (Babych and Hartley, 2007). These researchers also developed a method for automated error-analysis on the basis

of concordance evaluation using BLEU (Babych and Hartley, 2008), which is useful for the developers of industrial MT systems. Leeds University's current work concentrates on creating new types of usage scenarios for existing MT evaluation metrics, where the scores can have meaningful interpretation, and possible limitations on the use of automated MT evaluation tools can be avoided.

7.1.4. Useful resources for MT

Taus Data Association (TDA, http://www.tausdata.org/) is organization which provides platform to share parallel language data to facilitate automation of translation activities. The main principle is: "Share your translation memories and in return get access to the data of all other members." (http://www.tausdata.org/). TDA hosts translation memories and glossaries in many languages structured by industry domains and company indexes. Free access is provided to its databases for the look-up of translations of terms and phrases. Members of TDA can select and pool data. TDA is using TMX format for data exchange.

JRC Acquis (http://wt.jrc.it/lt/Acquis/) – is the largest multilingual parallel corpus. It contains selected EU legal texts, the so called *Acquis Communautaire*. Current version (3.0) contains texts in all EU official languages (Bulgarian, Czech, Danish, Dutch, German, Greek, English, Spanish, Estonian, Finnish, French, Hungarian, Italian, Lithuanian, Latvian, Maltese, Polish, Portuguese, Romanian, Slovak, Slovene, Swedish), except Irish. Data are provided in XML format and are encoded with UTF-8. Texts are aligned with Vanilla and Hunalign aligners (in total 231 language pairs) and are ready for exploitation in SMT systems.

DGT Multilingual Translation Memory (http://langtech.jrc.it/DGT-TM.html) is translation memories of EC Directorate General for Translations for the *Acquis Communautaire*. The Translation Memory is available for the same language pairs as JRC Acquis corpus. Data are provided in TMX format.

Europarl parallel corpus (http://www.statmt.org/europarl/) contains texts from European Parliament proceedings in 11 languages (French, Italian, Spanish, Portuguese, English, Dutch, German, Danish, Swedish, Greek and Finnish). These texts are aligned and are provided in XML format.

OPUS (http://urd.let.rug.nl/tiedeman/OPUS/) provides access to several multilingual corpora, such as EMEA (European Medicines Agency documents), KDE (KDE manual corpus), SETIMES (parallel corpus of news in the Balkan languages) corpora are freely available. Corpus data are available in MOSES/GIZA++ and TMX formats.

7.1.5. Projects related to MT

TC-STAR

The EU FP6 TC-STAR project (2004-2007) was envisaged as a long-term effort to advance research in the core technologies of Speech-to-Speech Translation (SST). To assess the advances in SST technologies, annual competitive evaluations were organized (e.g. 2006, 2007). The aim of the evaluation campaigns were to measure the progress made during the life of the project in Automatic Speech Recognition (ASR), Spoken Language Translation (TTS), Text-To-Speech and in the whole end-to-end Speech-to-Speech system. In addition to the measure performance, the infrastructure built in TC-STAR was also evaluated. For Automatic Speech Recognition, systems were evaluated automatically for English, Spanish and Mandarin Chinese by computing Word Error Rates and Character Error Rates. For SLT, evaluation carried out for English-to-Spanish, Spanish-to-English

and Mandarin Chinese-to-English. In addition to automatic metrics such as BLEU, NIST, WER, PER, subjective evaluations were organized with hundreds of evaluators to assess the quality of SLT systems. For TTS, systems were evaluated for Chinese, Spanish and English. MOS subjective tests were organized to assess various aspects such as *overall voice quality, listening effort, comprehension, pronunciation, naturalness, etc.* In addition to subjective tests, individual module evaluations were carried out.

EUROMATRIX and EUROMATRIXPlus

EuroMatrix (running from 2006 to 2009, funded by the EU under FP7) aims at a major push in Machine Translation (MT) technology by applying the most advanced MT technologies systematically to all pairs of EU languages with special attention for the languages of the new and near-term prospective member states. It designs and investigates novel combinations of statistical techniques and linguistic knowledge sources as well as hybrid MT architectures.

EuroMatrix aims at enriching the statistical MT approach with novel learning paradigms and experiment with new combinations of methods and resources from statistical MT, rule-based MT, shallow language processing and computational lexicography/morphology. With respect to evaluation, EUROMATRIX aims to organize a competitive annual international evaluation of MT with a strong focus on European economic and social needs

EuromatrixPlus is the successor of EuroMatrix, will run from 2009 to 2012 and is funded by the EU under FP7. Regarding evaluation, the project aims to

- Organize an annual evaluation campaign on European language translation on large-scale open domain tasks such as translation of news stories or Wikipedia articles.
- Prepare ready-for-use training and test sets, and annotate them with additional linguistic markup.
- Develop manual and automatic evaluation metrics and validate these metrics.
- Pose special challenges that arise from work on the user-centric work packages, for instance improved interactive machine translation.

MetaNet (http://www.meta-net.eu/**)**

META-NET is a Network of Excellence dedicated to fostering the technological foundations of a multilingual European information society. Language Technologies will:

- enable communication and cooperation across languages,
- secure users of any language equal access to information and knowledge,
- build upon and advance functionalities of networked information technology.

A concerted, substantial, continent-wide effort in language technology research and engineering is needed for realizing applications that enable automatic translation, multilingual information and knowledge management and content production across all European languages. This effort will also enhance the development of intuitive language-based interfaces to technology ranging from household electronics, machinery and vehicles to computers and robots.
To this end META-NET is building the Multilingual European Technology Alliance (META). Bringing together researchers, commercial technology providers, private and corporate language technology users, language professionals and other information society stakeholders. META will prepare the necessary ambitious joint effort towards furthering language technologies as a means towards realising the vision of a Europe united as one single digital market and information space.

- Building a community with a shared vision and strategic research agenda
- Building META-SHARE, an open resource exchange facility
- Building bridges to neighbouring technology fields

META-NET is supporting these goals by pursuing three lines of actions:

- fostering a dynamic and influential community around a shared vision and strategic research agenda (META-VISION),
- creating an open distributed facility for the sharing and exchange of resources (META-SHARE),
- building bridges to relevant neighboring technology fields.

The third action line is most relevant in this context.

Four themes have been selected in this line of work, all of them with the application to the machine translation domain in mind: adding semantics to machine translation, hybridization of machine translation, using wider context and improving the empirical base (the latter having implications for multilingual natural language processing even beyond machine translation). Each of these themes will be pursued through exchange and cooperation with other disciplines that offer methods, knowledge and resources urgently needed for progress in MT.

ACCURAT project (www.accurat-project.eu)

Since applicability of current data-driven methods directly depends on the availability of very large quantities of parallel corpus data, the translation quality of data-driven MT systems varies dramatically from being quite good for language pairs with large corpora available to being almost unusable for under-resourced languages and domains. Therefore ACCURAT project aims to find, analyze and evaluate novel methods how to exploit comparable corpora and to achieve a significant increase in translation quality for under-resourced languages and narrow domains.

The key innovation of ACCURAT will be the creation of methodology and tools to measure, to find and to use comparable corpora to improve the quality of MT for under-resourced languages and domains. ACCURAT will make its novel methodology for under-resourced areas of MT openly accessible in respect to comparability metrics, methods and techniques of alignment for comparable corpora, methods and techniques of information extraction from aligned comparable corpora at different levels (document, paragraph, phrase / word), methods and techniques of collecting comparable corpora from the Web as well as collections of comparable corpora for the project language.

LetsMT!

The LetsMT! project (http://www.letsmt.eu/) aims to build an online collaborative platform for data sharing and MT building. The platform will support upload of monolingual and parallel corpora for public as well as proprietary MT training data. The HSS research community can easily build MT services using their specific content as well as evaluate them. Building of a custom SMT system will be as simple as selecting training corpus and pressing “Train SMT system” button. Of course, more advanced customization options will be provided to advanced users, for example, representing HSS research community.

One of SMT system usage issues LetsMT! will address is slow speed of SMT system training. LetsMT! will span SMT training tasks between several server instances, thus shortening SMT system training time. It is also planned to develop incremental training support of SMT systems as it

is expected that SMT systems will often be re-trained after adding additional corpus to training data set.

LetsMT! systems will allow to run several trained SMT systems simultaneously thus allowing quick comparison of different SMT systems and parallel use of those systems using various user interfaces – web page and integration in translation and localization tools (for example, SDL Trados). Open API will ensure extendibility of a system and possibility to integrate any trained LetsMT! SMT system in various applications using web-services.

8. Conclusion

This deliverable contains the efforts of WP 5.7, resp. of its Working Groups. The aim was to specify the requirements for the registries of resources and tools for the representational standards for the various types of resources. Based on these specifications, the WP has studied generic frameworks and has given recommendations for converters that transform existing language resources into the CLARIN representational standards and formats. Generic frameworks and the conversion tools required are described in detail.

In Section 1, the general standards Unicode, XML and TEI have been introduced. The standardization work present in this deliverable is largely based on a combination of these three standards. Section 2 was concerned with Lexica and Terminology Standards. The Lexical Markup Framework (LMF) and the Terminology Markup Framework (TMF) have been introduced. Furthermore, various lexicon formats which now use LMF have been presented. Section 3 was concerned with Ontologies. Apart from a presentation of ontology-related terminology, ontology definition languages, rule definition languages, query definition languages and mapping definition languages have been introduced. Section 4 was dedicated to written corpora. Apart from the terminology related with written corpora, the standards TEI and CES/XCES were treated. TEI and XCES, in particular, were explained by the example of the National Corpus of Polish. Section 5 was dedicated to annotation-related matters. Annotation standards for syntactic and semantic annotation have been presented. The syntactic annotation standards include SynAF (the Syntactic Annotation Framework), KAF (the Kyoto Annotation Format), the NeGra format, the Prague Markup Language and the Penn Treebank format. Standards for semantic annotation include SemAF (the Semantic Annotation Framework), TimeML for the annotation of events and time expressions, the MATE annotation scheme for coreference annotation, and an overview of the annotation of Named Entities. Section 6 was devoted to multimedia encoding and annotation. Apart from an extensive overview on terminology, the biggest part of the section consists of a presentation of current formats and frameworks for media annotation, along with recommendations. Section 7, finally, was concerned with Machine Translation. Current evaluation standards were presented and explained. Furthermore, active projects concerned with Machine Translation were introduced.

The ultimate objective of CLARIN is to create a European federation of existing digital repositories that include language-based data, to provide uniform access to the data, wherever it is, and to provide existing language and speech technology tools as web services to retrieve, manipulate, enhance, explore and exploit the data. This deliverable in particular, prepared during the CLARIN Preparatory Phase Project (2008-2010), provides the requirements for the registries of resources and tools for the representational standards for the various types of resources and will therefore hopefully be a valuable resource for the subsequent construction and exploitation phases of CLARIN beyond 2010.

Bibliography

Baayen, R. H., R. Piepenbrock and L. Gulikers (2005). The CELEX Lexical Database (Release 2) CD-ROM. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.

Babych, B., D. Elliott and A. Hartley (2004). Extending MT evaluation tools with translation complexity metrics. Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004). Barcelona, Spain**:** 106-112.

Babych, B. and A. Hartley (2007). Sensitivity of automated models for MT evaluation: proximity-based vs. performance-based methods. Machine Translation Summit XI: Automatic procedures in MT evaluation**:** 10-14.

Babych, B. and A. Hartley (2008). Automated MT evaluation for error analysis: automatic discovery of potential translation errors for multiword expressions. ELRA Workshop on Evaluation Looking into the Future of Evaluation: When automatic metrics meet task-based and performance-based approaches at LREC 2008. Marrakech, Morocco**:** 6-11.

Babych, B., A. Hartley and D. Elliott (2005). Estimating the predictive power of n-gram MT evaluation metrics across language and text types. Machine Translation Summit X**:** 13-15.

Banski, P. and A. Przepiórkowski (2009). Stand-off TEI Annotation: the Case of the National Corpus of Polish. Proceedings of the Third Linguistic Annotation Workshop (LAW III) at ACL-IJCNLP 2009. Singapore**:** 64-67.

Beckwith, R., G. A. Miller and R. Tengi (1993). Design and implementation of the WordNet lexical database and searching software, Cognitive Science Laboratory, Princeton University.

Bird, S. and M. Liberman (2001). "A formal framework for linguistic annotation." Speech Communication **33**: 23-60.

Bruneseaux, F. and L. Romary (1997). Codage des Références dans les Dialogues Homme-machine. ACH/ALLC. Kingston, Ontario.

Burchardt, A., S. Padó, D. Spohr, A. Frank and U. Heid (2008). "Constructing Integrated Corpus and Lexicon Models for Multi-Layer Annotations in OWL DL." Linguistic Issues in Language Technology **1**: 1-33.

Burnard, L. and S. E. Bauman (2008). TEI P5: Guidelines for electronic text encoding and interchange, Text Encoding Initiative.

Callison-Burch, C., M. Osborne and P. Koehn (2006). Re-evaluating the role of BLEU in machine translation research. 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL). Trento, Italy**:** 249-256.

Carletta, J., S. Evert, U. Heid, J. Kilgour, J. Robertson and H. Voormann (2003). "The NITE XML Toolkit: flexible annotation for multi-modal language data." Behavior Research Methods, Instruments, and Computers, special issue on Measuring Behavior **35**(3): 353-363.

CLARIN:CM (2009). Component metadata: A CLARIN short-guide. http://www.clarin.eu/documents.

Clément, L. and É. Villemonte de la Clergerie (2005). MAF: a Morphosyntactic Annotation Framework. Proceedings of the 2nd Language and Technology Conference. Poznan, Poland**:** 90-94.

Derwojedowa, M., S. Szpakowicz, M. Zawisławka and M. Piasecki (2008). Lexical units as the centrepiece of a wordnet. Proceedings of the Intelligent Information Systems XVI (IIS 2008). Zakopane, Academic Publishing House Exit.

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. Proceedings of the Second International Conference on Human Language Technology Research. San Diego, CA**:** 128-132.

DuBois, J., S. Schuetze-Coburn, S. Cumming and D. Paolino (1993). Outline of Discourse Transcription. Talking Data: Transcription and Coding in Discourse Research. J. A. Edwards and M. D. Lampert. Hillsdale, NJ, Lawrence Erlbaum**:** 45-89.

Ehlich, K. (1993). HIAT: A Transcription System for Discourse Data. Talking Data: Transcription and Coding in Discourse Research. J. A. Edwards and M. D. Lampert. Hillsdale, NJ, Lawrence Erlbaum**:** 123-148.

Ehlich, K. and J. Rehbein (1976). "Halbinterpretative Arbeitstranskriptionen (HIAT)." Linguistische Berichte **45**: 21-41.

Fellbaum, C. (1998). WordNet: An electronic lexical database. Cambridge, MA, MIT press.

Fisher, W., G. Doddington and K. Goudie-Marshall (1986). The DARPA Speech Recognition Research Database: Specifications and Status. Proceedings of DARPA Workshop on Speech Recognition**:** 93-99.

Fligelstone, S. (1992). Developing a scheme for annotating text to show anaphoric relations. New directions in English language corpora. Methodology, results, software development. G. Leitner. Berlin, Mouton de Gruyter.**:** 153-170.

Francopoulo, G., M. George, N. Calzolari, M. Monachini, N. Bel, M. Pet and C. Soria (2006). Lexical markup framework (LMF). Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006). Genoa, Italy**:** 233-236.

Gibbon, D., I. Mertins and R. Moore (2000). Handbook of multimodal and spoken dialogue systems: resources, terminology and product evaluation. Norwell, MA, Kluwer Academic Publishers.

Gibbon, D., R. Moore and R. Winski (1997). Handbook of Standards and Resources for Spoken Language Systems. Berlin, de Gruyter.

Gruber, T. R. (1993). "A translation approach to portable ontology specifications." Knowledge Acquisition **5**: 199-220.

Gruber, T. R. (1995). "Toward Principles for the Design of Ontologies Used for Knowledge Sharing." International Journal of Human and Computer Studies **43**(5/6): 907-928.

Guarino, N. (1994). The Ontological Level. Philosophy and the Cognitive Science. R. Casati, B. Smith and G. White. Vienna, Hölder-Pichler-Tempsky**:** 443-456.

Guarino, N. (1998). Formal Ontology and Information Systems. Proceedings of FOIS'98, Trento, Italy. N. Guarino. Amsterdam, IOS Press**:** 3-15.

Guarino, N. (2000). Invited Mini-course on Ontological Analysis and Ontology Design. Proceedings of the First Workshop on Ontologies and lexical Knowledge Bases - OntoLex 2000. Sozopol, Bulgaria.

Guarino, N. and C. Welty (2002). "Evaluating Ontological Decisions with OntoClean." Communications of the ACM **45**(2): 61-65.

Gundel, J., N. Hedberg and R. Zacharski (1993). "Cognitive Status and the Form of Referring Expressions in Discourse." Language **69**: 274-307.

Hanke, T. and J. Storz (2008). iLex - a database tool for integrating sign language corpus linguistics and sign language lexicography. Proceedings of the Third Workshop on the Representation and Processing of Sign Languages: Construction and Exploitation of Sign Language Corpora at LREC 2008. Marrakesh, Morroco.

Heim, I. (1982). The semantics of definite and indefinite noun phrases, University of Massachusetts at Amherst.

Henrich, V. and E. Hinrichs (2010). GernEdiT - The GermaNet Editing Tool. Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10). Valetta, Malta**:** 2228-2235.

Henrich, V. and E. Hinrichs (2010). Standardizing Wordnets in the ISO Standard Wordnet-LMF: The Case of GermaNet. Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), main conference. Beijing, China**:** 456-464.

Hirschman, L. and N. Chinchor (1997). MUC-7 coreference task definition. http://www-nlpir.nist.gov/related_projects/muc/proceedings/co_task.html.

Hirst, G. (2004). Ontology and the lexicon. Handbook on Ontologies. S. Staab and R. Studer. Berlin, Springer Verlag**:** 209-229.

Horák, A., K. Pala, A. Rambousek and M. Povolny (2006). DEBVisDic -- First Version of New Client-Server Wordnet Browsing and Editing Tool. Proceedings of the Third Global WordNet Conference (GWC 2006). Masaryk University**:** 325-328.

Horák, A. and P. Smrž (2004). VisDic -- WordNet Browsing and Editing Tool. Proceedings of the Second Global WordNet Conference (GWC 2004)**:** 136-141.

Ide, N. (1998). Corpus Encoding Standard: SGML guidelines for encoding linguistic corpora. Proceedings of the First International Language Resources and Evaluation Conference (LREC'01)**:** 463-470.

Ide, N. and L. Romary (2004). A registry of standard data categories for linguistic annotation. Proceedings of the Fourth Language Resources and Evaluation Conference (LREC'04)**:** 135-139.

Ide, N. and K. Suderman (2007). GrAF: A graph-based format for linguistic annotations. Proceedings of the Linguistic Annotation Workshop at ACL 2007. Prague, Czech Republic**:** 1-8.

Ide, N. and J. Véronis (1995). "Encoding dictionaries." Computers and the Humanities **29**(2): 167-179.

Kamp, H. and U. Reyle (1993). From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory. Dordrecht, Kluwer.

Klappenbach, R. and W. Steinitz (1962-1977). Wörterbuch der deutschen Gegenwartssprache (=WDG) Berlin.

Koeva, S. (2008). Derivational and Morphosemantic Relations in Bulgarian Wordnet. Proceedings of Intelligent Information Systems 2008**:** 359-368.

König, E., W. Lezius and H. Voormann (2003). TIGERSearch user's manual.

Kunze, C. and L. Lemnitzer (2002). GermaNet - representation, visualization, application. Third International Conference on Language Resources and Evaluation. Las Palmas, Gran Canaria, ELRA**:** 1485-1491.

Kunze, C. and L. Lemnitzer (2002). Standardizing Wordnets in a Web-compliant Format: The Case of GermaNet. Workshop on Wordnet Structures and Standardisation, and how these affect Wordnet Applications and Evaluations at LREC 2002. Las Palmas, Gran Canaria**:** 24-29.

Lee, L., S. Hsieh and C. Huang (2009). CWN-LMF: Chinese WordNet in the lexical markup framework. Proceedings of the 7th Workshop on Asian Resources at ACL-IJCNLP 2009. Singapore**:** 123-130.

Lemnitzer, L. and C. Kunze (2002). Adapting GermaNet for the Web. First Global Wordnet Conference. Central Institute of Indian Languages, Mysore, India**:** 174-181.

Lemnitzer, L., L. Romary and A. Witt (2010). Representing human and machine dictionaries in Markup languages. Dictionaries. An International Encyclopedia of Lexicography. Supplementary volume: Recent developments with special focus on computational lexicography. Berlin, New York, de Gruyter/Max Niemeyer.

Lenci, A., F. Busa, N. Ruimy, E. Gola, M. Monachini, N. Calzolari, A. Zampolli, E. Guimier, G. Recourc¸àö¬©, L. Humphreys, U. von Rekovsky, A. Ogonowski, C. McCauley, W. Peters, I. Peters, R. Gaizauskas and M. Villegas (2000). SIMPLE Work Package 2 - Linguistic Specifications, Deliverable D2.1. ILC-CNR, Pisa, Italy.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004) at ACL 2004. Barcelona, Spain**:** 25-26.

Lin, C.-Y. and F. J. Och (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004). Barcelona, Spain**:** 606-613.

MacWhinney, B. (2000). The CHILDES project: tools for analyzing talk. Mahwah, NJ, Lawrence Erlbaum.

Młodzki, R. and A. Przepiórkowski (2009). The WSD development environment. Proceedings of the 4th Language & Technology Conference (LTC 2009). Poznan, Poland**:** 185-189.

MUC-6: Sixth Message Understanding Conference (1995). Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995.

Nirenburg, S. and V. Raskin. (2004). Ontological Semantics, MIT Press.

Nivre, J. (1999). Modifierad Standardortografi (MSO6), Department of Linguistics, Göteborg University.

Nivre, J., J. Allwood, L. Grönqvist, E. Ahlsén, M. Gunnarsson, J. Hagman, S. Larsson and S. Sofkova (2005). Göteborg Transcription Standard. Version 6.3 - DRAFT, Department of Linguistics, Göteborg University.

Noy, N. F. and D. L. McGuiness (2001). Ontology Development 101: A Guide to Creating Your First Ontology, Stanford University.

Pala, K. and D. Hlaváčková (2007). Derivational Relations in Czech WordNet. Proceedings of the Workshop on Balto-Slavonic Natural Language Processing at ACL 2007. Prague, Czech Republic**:** 75-81.

Papineni, K., S. Roukos, T. Ward and W. J. Zhu (2002). BLEU: a method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002). Philadelphia, PA**:** 311-318.

Parisse, C. and A. Morgenstern (2010). A multi-software integration platform and support for multimedia transcripts of language. Proceedings of the Workshop 'Multimodal Corpora: Advances in Capturing, Coding and Analyzing Multimodality' at LREC 2010**:** 106-110.

Passonneau, R. (1997). Instructions for applying discourse reference annotation for multiple applications (DRAMA).

Piasecki, M., S. Szpakowicz and B. Broda (2009). A Wordnet from the Ground Up, Oficyna Wydawnicza Politechniki Wrocławskiej.

Poesio, M. (2000). Annotating a corpus to develop and evaluate discourse entity realization algorithms: issues and preliminary results. Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00). Athens, Greece.

Poesio, M., F. Bruneseaux and L. Romary (1999). The MATE meta-scheme for coreference in dialogues in multiple languages. Proceedings of the Workshop on Standards for Discourse Tagging at ACL 1999. University of Maryland**:** 65-74.

Prillwitz, S. and others (1989). HamNoSys. Version 2.0; Hamburger Notationssystem fur Gebardensprache. Eine Einfuhrung. Hamburg, Signum.

Przepiórkowski, A. and P. Bański (2009). Which XML standards for multilevel corpus annotation? Proceedings of the 4th Language & Technology Conference. Poznan, Poland.

Przepiórkowski, A. and P. Bański (2009). XML Text Interchange Format in the National Corpus of Polish. Practical Applications in Language and Computers (PALC 2009). S. Goźdź-Roszkowski, Peter Lang.

Przepiórkowski, A., R. L. Górski, M. Łaziński and P. Pęzik (2010). Recent Developments in the National Corpus of Polish. Seventh International Conference on Language Resources and Evaluation (LREC'10). Valletta, Malta, ELRA.

Przepiórkowski, A., R. L. Górski, B. Lewandowska-Tomaszczyk and M. Łaziński (2008). Towards the National Corpus of Polish. 6th Language Resources and Evaluation Conference (LREC 2008). Marrakesh, Morocco**:** 827-830.

Quochi, V., R. Del Gratta, E. Sassolini, R. Bartolini, M. Monachini and N. Calzolari (2009). A Standard Lexical-Terminological Resource for the Bio Domain. Third Language and Technology Conference (LTC 2007) Poznan, Poland, October 5-7, 2007. Revised Selected Papers Human Language Technology. Challenges of the Information Society. Z. Vetulani and H. Uszkoreit. Berlin/Heidelberg, Springer. **5603/2009:** 325-335.

Quochi, V., M. Monachini, R. D. Gratta and N. Calzolari (2008). A lexicon for biology and bioinformatics: the BOOTStrep experience. Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08). Marrakech, Morocco.

Rehbein, J., T. Schmidt, B. Meyer, F. Watzke and A. Herkenrath (2004). Handbuch für das computergestützte Transkribieren nach HIAT, SFB 538, University of Hamburg.

Romary, L. (2009). Questions & Answers for TEI Newcomers. Jahrbuch für Computerphilologie, Mentis Verlag. **10**.

Romary, L. (2010). Standardization of the formal representation of lexical information for NLP. Dictionaries. An International Encyclopedia of Lexicography. Supplementary volume: Recent developments with special focus on computational lexicography. Berlin, New York, de Gruyter/Max Niemeyer.

Romary, L., S. Salmon-Alt and G. Francopoulo (2004). Standards going concrete: from LMF to Morphalou. EletricDict '04: Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries. Geneva, Switzerland**:** 22-28.

Sacks, H., E. Schegloff and G. Jefferson (1978). A simplest systematics for the organization of turn taking for conversation. Studies in the Organization of Conversational Interaction. J. Schenkein. New York, Academic Press**:** 7-56.

Schiel, F., S. Burger, A. Geumann and K. Weilhammer (1998). The Partitur Format at BAS. Proceedings of the First International Conference on Language Resources and Evaluation (LREC'98). Paris, France, ELRA**:** 1295-1301.

Schmidt, T. (2005). Computergestützte Transkription -- Modellierung und Visualisierung gesprochener Sprache mit texttechnologischen Mitteln. Frankfurt a. M., Peter Lang.

Schmidt, T. (2005). Time-based data models and the Text Encoding Initiative's guidelines for transcription of speech, SFB 538, University of Hamburg.

Schmidt, T. and J. Bennöhr (2008). "Rescuing Legacy Data." Language Documentation and Conservation **2**: 109-129.

Schmidt, T., S. Duncan, O. Ehmer, J. Hoyt, M. Kipp, M. Magnusson, T. Rose and H. Sloetjes (2008). An exchange format for multimodal annotations. Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08). Marrakech, Morocco**:** 359-365.

Schmidt, T., S. Duncan, O. Ehmer, J. Hoyt, M. Kipp, M. Magnusson, T. Rose and H. Sloetjes (2009). An Exchange Format for Multimodal Annotations. Multimodal Corpora: from models of natural interaction to systems and applications. M. Kipp, J. Martin, P. Paggio and D. Heylen. Berlin/Heidelberg, Springer. **5509:** 207-221.

Selting, M., P. Auer, D. Barth-Weingarten, J. Bergmann, P. Bergmann, K. Birkner, E. Couper-Kuhlen, A. Deppermann, P. Gilles, S. Gänthner, M. Hartung, F. Kern, C. Mertzlufft, C. Meyer, M. Morek, F. Oberzaucher, J. Peters, U. Quasthoff, W. Schütte, A. Stukenbrock and S. Uhmann (2009). "Gesprächsanalytisches Transkriptionssystem 2 (GAT 2)." Gesprächsforschung **10**: 353-402.

Snover, M., B. Dorr, R. Schwartz, R. Micciulla and J. Makhoul (2006). A study of translation edit rate with targeted human annotation. 7th Conference of the Association for Machine Translation in the Americas (AMTA-2006). Cambridge, MA**:** 223-231.

Snover, M. G., N. Madnani, B. Dorr and R. Schwartz (2010). "TER-Plus: paraphrase, semantic, and alignment enhancements to Translation Edit Rate." Machine Translation **23**(3): 117-127.

Soria, C. and M. Monachini (2008). Kyoto-LMF --- Wordnet representation format.

Soria, C., M. Monachini and P. Vossen (2009). Wordnet-LMF: fleshing out a standardized format for wordnet interoperability. Proceedings of ACM Workshop on Intercultural Collaboration. Palo Alto, CA, ACM**:** 139-146.

Suárez-Figueroa, M. C., G. Aguado de Cea, C. Buil, C. Caracciolo, M. Dzbor, A. Gómez-Pérez, G. Herrrero, H. Lewen, E. Montiel-Ponsoda and V. Presutti (2007). NeOn: Lifecycle Support for Networked Ontologies, Integrated Project (IST-2005-027595), D5.3.1 NeOn Development Process and Ontology Life Cycle.

Thurmair, G. (2007). Automatic evaluation in MT system production. Machine Translation Summit XI workshop: Automatic Procedures in MT Evaluation. Copenhagen, Denmark.

Tufiş, D., D. Cristea and S. Stamou (2004). "BalkaNet: Aims, Methods, Results and Perspectives. A General Overview." Special Issue on BalkaNet. Romanian Journal on Science and Technology of Information **7**(1-2): 9-43.

The Unicode Standard / the Unicode Consortium. Version 5.2 (2009). Mountain View, CA., Unicode Consortium.

Vauquois, B. (1968). A survey of formal grammars and algorithms for recognition and transformation in machine translation. IFIP Congress-68. Edinburgh**:** 254-260.

Velten, E. (1968). "A laboratory task for induction of mood states." Behavior Research & Therapy **6**: 473–482.

Vossen, P. (2002). EuroWordNet General Document. Amsterdam, The Netherlands, University of Amsterdam.

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 and J. VanGent (2008). KYOTO: a system for mining, structuring, and distributing knowledge across languages and cultures. Proceedings of the Fourth Global WordNet Conference. University of Szeged, Szeged, Hungary**:** 474-484.

Webber, B. (1979). A formal approach to discourse anaphora, Harvard University.

Witt, A., G. Rehm, E. Hinrichs, T. Lehmberg and J. Stegmann (2009). "SusTEInability of linguistic resources through feature structures." Literary and Linguistic Computing **24**(3): 363-372.

Wright, S. E. (2004). A global data category registry for interoperable language resources. Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04). Lisbon, Portugal**:** 123-126.