Reference number of working document: ISO/TC 37/SC 4 N421

Date: 2007-08-22

ISO CD 24615:2007

Committee identification: ISO/TC 37/SC 4

Secretariat: KATS

Language resource management—Syntactic Annotation Framework (SynAF)

Gestion des ressources linguistiques — Cadre d' Annotation Syntactique —

Document type: International standard
Document subtype: if applicable
Document stage: 30.00
Document language: en

Table of contents

Warning 1
Copyright notice ii
Foreword iv
1 Introduction 5
2 Scope 6
3 Normative references 7
4 Terms and definitions 7
5 Key standards used by SynAF 11
5.1 Unicode 11
5.2 ISO 12620 Data Category Registry (DCR) 11
5.3 Unified Modeling Language (UML) 11
6 Embedding SynAF in the LAF model 11
7 The SynAF Metamodel 13
7.1 Introduction 13
7.2 The SynAF diagram (to be represented in UML) 13
7.2.1 T Nodes class 14
7.2.2 NT Nodes class 14
7.2.3 Edges class 14
7.2.4 Syntactic Annotation class 14
Annex A : (informative) Data Categories for SynAF 15
A.1 Constituency 15
A.2 Dependency 16
Annex B (informative) Annotation example 20

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

International Standard 24615 was prepared by Technical Committee ISO/TC 37, *Terminology and other language and content resources*, Subcommittee SC 4, *Language resource management*, in collaboration with the European eContent Project "LIRICS" (**L**inguistic **I**nf**r**astructure for **I**nteroperable Resour**c**es and **S**ystems), under the contract e-Content-22236-LIRICS.

ISO 24615 is designed to coordinate closely with ISO AWI 24612, *Linguistic Annotation framework (LAF)*, and ISO DIS 24613, *Lexical Markup Framework (LMF)*, and ISO CD 24611, *Morphosyntactic Annotation Framework (MAF)*, and ISO CD 24617-1, *Semantic Annotation Framework - Part 1: Time and events (SemAF/Time)*.

Annexes A forms an integral part of this International Standard.

1 Introduction

There have been in the past no thorough standardisation activities in the domain of syntactic annotation, despite the numerous projects (see Abeillé, 2003) that have designed ways to implement linguistic TreeBanks, i.e. syntactically annotated corpora. For several years the Penn Treebank initiatives have served as a de facto standard, but more recent work (e.g. the Negra/Tiger initiative[1] in Germany or the ISST initiative in Italy[2]) has shown that a more coherent framework could be designed to account for both (hierarchical) constituency and dependency phenomena in syntactic annotation.

Within the European eContent LIRICS project, a group of international experts has started the ISO process, called SynAF (Syntactic Annotation Framework). The actual document is a revision of ISO WD 24615, which is the result of a more extended discussion, including feedback and comments from ISO experts, and will be submitted for its acceptance as a CD and its first CD ballot.

The document proposes a meta-model for syntactic annotation and lists in the annex candidate data-categories for syntactic annotation, to be described in more details in ISO/TC 37/SC 4 Ad hoc Thematic Domain Group 4: Syntax (on syntactic data-categories). The establishment of this group has been resolved at the ISO TC37/SC4 annual meeting in Beijing (2006-08-21/25).

[1] See: http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus/

[2] See Montemagni (2003).

2 Scope

This International Standard describes the Syntactic Annotation Framework (SynAF), a high level model for representing the syntactic annotation of textual documents.

SynAF is building on the ISO MAF proposal (CD 24611). MAF (Morpho-Syntactic Framework) is dealing with the morpho-syntactic annotation of specific segments of textual documents. The morpho-syntactic annotation framework is about *part of speech* (noun, adjective, verb, etc.), *morphological* and *grammatical* features (such as number, gender, person, mood, verbal tense).

SynAF is about the annotation of the syntactic constituency of such (groups of) morpho-syntactically annotated fragments and the syntactic dependency relations existing between those (groups of) morpho-syntactically annotated fragments. We consider that the sentence will define the boundaries of the fragments of textual documents to which SynAF will apply.

As suggested just above, syntactic annotation has at least two functions in language processing:

1) To represent linguistic constituencies, like Noun Phrases (NP), describing a structured sequence of morpho-syntactically annotated items[3], where we consider also constituents built from non-contiguous elements, and

2) To represent dependency relations, like head-modifier relation[4]. The dependency information can exist between morpho-syntactically annotated items within a phrase (an adjective is the modifier of the head noun within an NP) or describe a specific relation between syntactic constituents at the clausal and sentential level (i.e. an NP being the "subject" of the main verb of a clause or sentence). The dependency relation can also be stated including empty elements (like the pro-drop property in romance languages[5])

SynAF is dealing with the description of a metamodel for syntactic annotation, which means that SynAF will describe elementary linguistic (in fact syntactic) abstractions that support the construction and the interoperability of (syntactic) annotations and resources. The Thematic Domain Group 4 (TDG 4) “Syntax” associated to SynAF will propose the definition of the related data categories, which will represent a point of reference for particular tagsets used for the syntactic annotation of various languages, also in the context of various application scenarios.

To summarize: SynAF is concerned with a metamodel that covers both dimensions of syntactic *constituency* and *dependency,* and SynAF will propose a multi-layered annotation framework that allows the combined and interrelated annotation of language data along both lines of consideration. Also the data-categories to be proposed within TDG4 will be about the basic annotation concerning both dimensions.

---

[3] But SynAF is also designed for dealing with like empty elements or traces generated by movements at the constituency level.

[4] Including also relations between same categories, like the head-head relation between nouns in appostions or nominal coordinations.

[5] This point has been particularily stressed by the authors of the ISST framework, showing here an advantage of the two-level approach, where the dependency information do not have to map entirely to the constituencey approach. In this sense, both levels of annotation have a certain independency in relation to each other (see Montemagni, 2003).

This standard is designed to be used in close conjunction with the metamodel presented in ISO AWI 24612, Linguistic resource framework (LAF) and with ISO 12620, Terminology and other language resources — Data categories.

3 Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of ISO 24615. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on ISO 24615 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 639-1:2002, Codes for the representation of names of languages – Part 1: Alpha-2 Code.

ISO 639-2:1998, Code for the representation of languages – Part 2: Alpha-3 Code.

ISO DIS 639-3:2005, Codes for the representation of languages – Part 3: Alpha-3 Code for comprehensive coverage of languages.

ISO 1087-1:2000, Terminology – Vocabulary – Part 1: Theory and application.

ISO 1087-2:1999, Terminology – Vocabulary – Part 2: Computer application.

ISO/IEC 10646-1:2003, Information technology – Universal Multiple-Octet Coded Character Set (UCS).

ISO/IEC 11179-3:2003, Information Technology – Data management and interchange – Metadata Registries (MDR) – Part 3: Registry Metamodel (MDR3)

ISO 12620:200?, Terminology and other language resources – Data Categories – Specification of data categories and management of a data category registry for language resources.

4 Terms and definitions

For the purposes of this International Standard, the terms and definitions given in ISO 1087-1, ISO 1087-2, ISO 12620:200? and the following apply:

4.1
Annotation

code associated with parts of text and providing for additional information about this part of text

NOTE In this document we use “**annotation**” as a short form for “linguistic” **annotation**, meaning the kind of textual enrichment that can be provided by linguistic information, which is here limited to morpho-syntax and **syntax**.

4.2
Category

feature value providing the content of a **node**

4.3
Chunk

4.4
constituent of a particular type, in the sense that it is not allow recursion within the **constituent**

NOTE So normally a noun **phrase** (NP) should not embed another one.

4.5
Clause

group of **phrases**, usually containing a verb, which valency also determines the number of obligatory **clause** elements (**phrases**)

NOTE A **clause** can be either a **main clause** or a **subordinated clause**. **Clauses** can be either finite or non-finite, in dependency of the mode of its verb. Usually, a finite clause contains at least a *subject* in addition to the verb. A **main clause** alone can build a complete **sentence**. In our model, a clause is a special case of a **constituent**.

4.6
Constituent

type of **nodes** we find in the syntactic **annotation** are building a **constituent** (to be revised)

4.7
Constituency relation

syntactic grouping of words *(i*nto **phrases**), **phrases** (into **clauses**) or **clauses** *(i*nto a **sentence***)* on the base of structural (or **hierarchical**) properties

4.8
Dependency relation

relation between **constituents** on the base of **grammatical functions constituents** plays in relation to each other within the larger **constituent** they are embedded in

Edge

4.9
triplet with a source **node**, a target **node**, and a **label**

NOTE **Non-terminal nodes** have an outgoing constituency **edges.**

Grammatical function

grammatical role of a **constituent** within its embedding syntactic environment

NOTE So an **NP** can act as a subject within a **sentence**. We speak here also of a **grammatical relation** between the subject-**NP** and the main verb in a **sentence**. We subsume all those **grammatical relations** (Subject-Predicate, **Head**-Modifier, etc.) under the concept of **dependency relations**.

4.10
Graph

model for representing objects that can be viewed as a connected set of more elementary sub-objects

4.11
Head

most important word in a **constituent** that carries the main meaning of the **phrase**

NOTE The **head** of a **constituent** cannot be left out.

4.12
Hierarchy

relative position of **constituents** in a **syntactic tree**

Human language technology

technology as applied to natural languages

4.13
Label

feature value providing the content of an **edge**

4.14
Main clause

finite **clause**, which can act on its own as a complete **sentence**

Example: *The train has some delay.*

4.15
Modifier

part of the **constituent** which ascribes a property to the **head** of the **constituent**

NOTE A **modifier** may be placed before or after the **head** of the **phrase** (pre-modifier or post-modifier). **Modifiers** are optional in a **constituent**

4.16
Natural language processing NLP

field covering knowledge and techniques involved in the processing of linguistic data by a computer

4.17
Node

pair consisting of a (possibly multiple) span and a category

NOTE **Non-Terminal nodes** have an outgoing constituency **edges**.

4.18
Phrases

word or group of **words** which can fulfil a **grammatical function** in a **clause**

NOTE But we allow empty **phrases** (as the example of the empty **NP** in Italian and Spanish, being no-realised pronouns and having the role of subjects in **clauses**). A **phrase** is typically named after the most important word in it (which we also call the **head**), so we have for example **noun phrase**s, verb **phrases**, adjective **phrases**, adverbial **phrases** and prepositional phrases.). **Phrases** have been informally described as "bloated words", in that the parts of the **phrase** that are added to the **head** elaborate and specify the reference of the **head word**. In our model, a **phrase** is a special case of a **constituent.**

4.19
Sentences

sequence of words, starting very often with a capital letter up to a final punctuation mark

NOTE But his definition is too restricted to layout property of certain language styles. A usage rule says that a complete **sentence** must contain a subject and a verb (in finite mode). A **sentence** consists of one or more **clauses**. In describing speech, it is common to talk about 'utterances' rather than **sentences**.

4.20
Span

pair of points identifying a segment of the document submitted to syntactic **annotation**

NOTE The first point is less or equal to the second point. A multiple **span** is sequence of **spans** where the ending point of each **span** is less or equal to the starting point of the subsequent **span**.

4.21
Specifier

part of a **constituent** that specifies the **head** (or the combination of **modifier** and **head**) with information about number, definiteness, proximity and ownership

4.22
Subordinated clause

clause which fulfils a **grammatical function** in a **phrase** (for example a relative **clause** modifying the **head** noun of a nominal **phrase**) or in another **clause**

NOTE A **subordinated clause** can not act on its own as a **sentence.**

4.23
Subcategorization frame

set of restrictions indicating the properties of the **syntactic arguments** that can or must occur with it

Example: Alfred (**syntactic argument**) read a book (**syntactic argument**) today (adjunct)

NOTE The subject, indirect object and direct object are possible **grammatical relations** for a sentence.

4.24
Syntax

way in which words are grouped together in linguistically meaningful units, thus capturing the relations that exist between those units

4.25
Syntactic argument

functionally essential element in a **clause** that identifies the participants in the process referred to by a verb

4.26
Syntactic tree

syntactic graph in which each node has a single parent

Terminal node

single wordFrom/lexical unit or a span with lentgh=0

NOTE The **terminal node** and the wordForm/lexical unit have an identical span.

5 Key standards used by SynAF

5.1 Unicode

SynAF is Unicode compliant and presumes that all data are represented using Unicode character encodings.

5.2 ISO 12620 Data Category Registry (DCR)

The designers of an SynAF conformant annotation shall use data categories from the ISO 12620 Data Category Registry (DCR), or a tagset that can be mapped onto the data categories.

5.3 Unified Modeling Language (UML)

SynAF complies with the specifications and modeling principles of UML as defined by the Object Management Group (OMG) [4]. SynAF uses a subset of UML that is relevant for linguistic description. (not done yet).

6 Embedding SynAF in the LAF model[6]

We want to embed the meta-model of SynAF in the more generic Linguistic Annotation Framework (LAF)l and reuse its annotation strategy. LAF provides a general framework for representing annotations that has been described elsewhere in detail (Ide and Romary, 2004, 2006). Its development has built on common practice and convergence of approach in linguistic annotation over the past 15-20 years. The core of the framework is specification of

[6] The whole section 5 is taken from (Ide, 2007).

an abstract model for annotations instantiated by a *pivot format*, into and out of which annotations are mapped for the purposes of exchange.

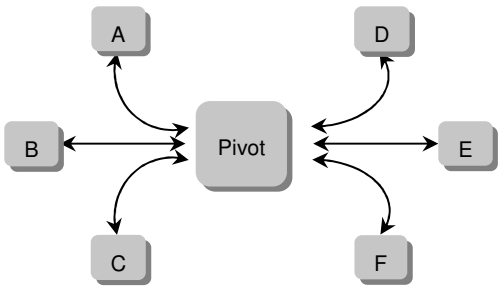

Figure 1: Use of the LAF pivot format

Figure 1 shows the overall idea for six different user annotation formats (labeled A – F), which requires two mappings for each scheme—one into and one out of the pivot format, provided by the scheme designer. The maximum number of mappings among schemes is therefore $2n$, vs. n^2-n mutual mappings without the pivot.

To map to the pivot, an annotation scheme must be (or be rendered via the mapping) isomorphic to the abstract model, which consists of (1) a *referential structure* for associating stand-off annotations with primary data, instantiated as a directed graph; and (2) a *feature structure representation* for annotation content. An annotation thus forms a directed graph referencing n-dimensional regions of primary data as well as other annotations, in which nodes are labeled with feature structures providing the annotation content. Formally, LAF consists of:

- A data model for annotations based on directed graphs defined as follows: A graph of annotations G is a set of vertices $V(G)$[7] and a set of edges $E(G)$. Vertices and edges may be labeled with one or more features. A feature consists of a quadruple (*G', VE, K, V*) where, *G'* is a graph, *VE* is a vertex or edge in *G'*, *K* is the name of the feature and *V* is the feature value.
- A *base segmentation* of primary data that defines edges between virtual nodes located between each "character" in the primary data.[8] The resulting graph *G* is treated as an *edge graph G'* whose nodes are the edges of *G*, and which serve as the leaf ("sink") nodes. These nodes provide the base for an annotation or several layers of annotation. Multiple segmentations can be defined over the primary data, and multiple annotations may refer to the same segmentation.
- Serializations of the data model, one of which is designated as the pivot.
- Methods for manipulating the data model.

Note that LAF does not provide specifications for annotation *content categories* (i.e., the labels describing the associated linguistic phenomena), for which standardization is a much trickier matter. The LAF architecture includes a *Data Category Registry* (DCR) containing pre-defined data elements and schemas that may be used directly in annotations, together with means to specify new categories and modify existing ones (see Ide and Romary, 2004).

---

[7] The word "vertice" is her esynonym to "node".

[8] A character is defined to be a contiguous byte sequence of a specified length .For text, the default is UTF-16.

7 The SynAF Metamodel

7.1 Introduction

While preparing SynAF, we identified some existing initiatives sharing a somehow common data model that seems to offer a good basis for the SynAF meta-model (Tiger and ISST for example, but also a longer list of corpora has been studied, see Deliverable D.3.1 of LIRICS). Base on this study we strongly suggest the adoption of a multi-layered annotation strategy interrelating syntactic annotation for both constituency and dependency in a sound representation scheme. The studied initiatives are also offering a quite complete list of descriptors, which we started to "merge" into a first list of candidate data-categories, to be extended by data categories covering syntactic phenomena (constituency and dependency) for other languages then German and Italian. Our list of candidate data categories is presented in Annex A. TIGER and ISST are summarized in Annexes

The SynAF model will be represented by UML classes and by a set of ISO 12620 data categories that function as UML attribute-value pairs. The data categories are used to decorate the UML classes that provide a high level view of the model. SynAF specifications in the form of textual descriptions that describe the semantics of the modeling elements provide more complete information about the SynAF classes, relationships, and extensions than can be included in the UML diagram. Developers shall define a data category selection (DCS) as specified for SynAF data category selection procedures (see below).

7.2 The SynAF diagram (to be represented in UML)

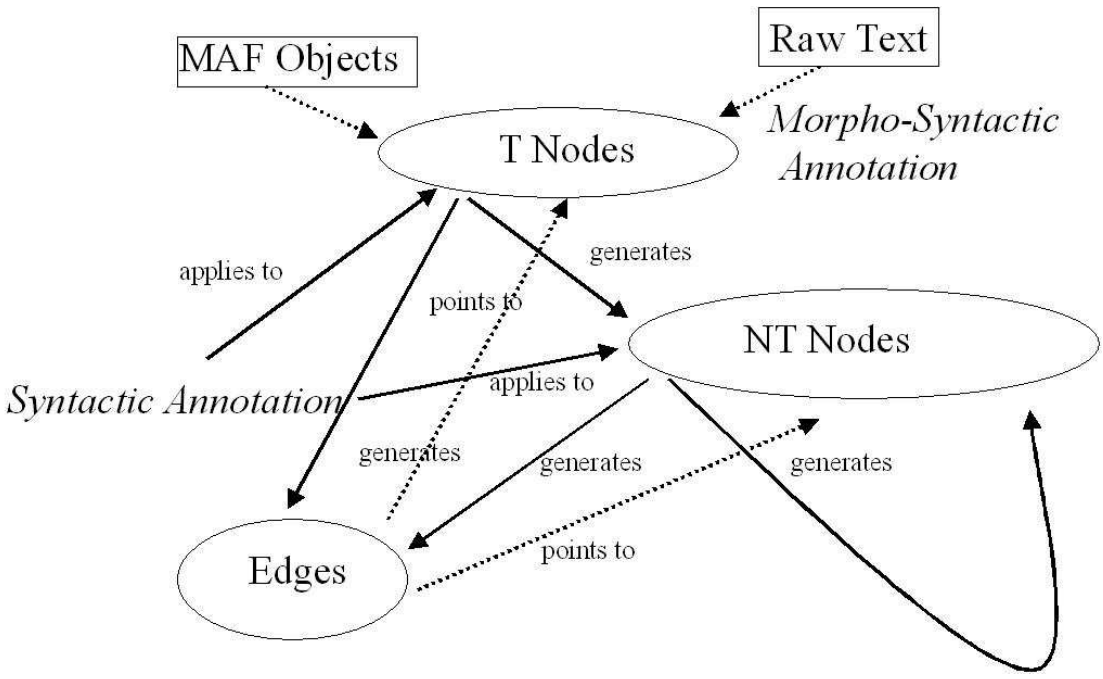

Figure 1: The SynAF metamodel

7.2.1 T Nodes class

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

7.2.2 NT Nodes class

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

7.2.3 Edges class

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

7.2.4 Syntactic Annotation class

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

Annex A: (informative) Data Categories for SynAF

Our strategy consisted in collecting some of the most consensual syntactic annotation definitions for gaining a list of data categories for constituency (node labels) and dependency (edge labels) annotation, which will be established in the document resulting from the work in ISO TC37/SC4 TDG 4 “Syntax”. In this document we present the actual list of candidates, as they have been detected in annotation initiatives like TIGER, ISST, Sparkle and EAGLES, and modified/harmonized for the purpose of this document. We do not quote the specific origin of each candidate data category. We indicate, where appropriate, language specific data catagories.

A.1 Constituency

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

	phrases
S	Sentence
VP	verb phrase (non-finite)
VZ	infinitive with zu (for German)

SPD	prepositional phrase *di* 'of' (for Italian)
SPDA	prepositional phrase *da* 'by, from' (for Italian)
IBAR	verbal nucleus with finite tense and all adjoined elements like clitics, adverbs and negation
SV2	infinitival clause
SV3	participial clause
SV5	gerundive clause
FAC	sentential complement
FS	subordinate sentence
FINT	+*wh* interrogative sentence
F2	relative clause
CP	dislocated or fronted sentential adjuncts
COMPC	copulative/predicative complement

A.2 Dependency

In the following we present the candidate data categories for dependency structures (the labels of edges in the annotation graph). Source of inspiration here were the Sparkle and the Tiger tagsets for dependency. We use also some examples taken from Sparkle (the short boxes below some data categories.)

mod: indicates the word introducing the dependent in a head-modifier relation

mod(of,gift,book)	the gift of a book
mod(by,gift,Peter)	the gift of a book by Peter

mod(of,examination,patient) the examination of the patient

mod('s,doctor,examination) the doctor's examination of the patient

cmod, xmod, ncmod: Clausal and non-clausal modifiers may (optionally) be distinguished by the use of cmod / xmod, and ncmod respectively, each with the same slots as **mod**. The GR cmod is for when the adjunct is controlled from within, and xmod for control from without the constituent under consideration.

cmod(because,eat,be) he ate the cake because he was hungry

xmod(without,eat,ask) he ate the cake without asking

subj: indicates the subject in the grammatical relation Subject-Predicate. The relation between a predicate and its subject; where appropriate, the **initial_gr** indicates the syntactic link between the predicate and subject before any GR-changing process.

subj(arrive,John,_) John arrived in Paris

subj(employ,Microsoft,_) Microsoft employed 10 C programmers

subj(employ,Paul,obj) Paul was employed by Microsoft

With pro-drop languages such as Italian, when the subject is not overtly realised the annotation is, for example, as follows:

subj(arrivare,Pro,_) arrivai in ritardo '(I) arrived late'

Where the dependent slot is filled by the abstract filler **pro**, which indicates that person and number of the subject can be recovered from the inflection of the head verb form.

csubj, xsubj, ncsubj: The Grammatical Realtions (RL) s **csubj** and **xsubj** may be used for clausal subjects, controlled from within, or without, respectively. **ncsubj** is a non-clausal subject.

csubj(leave,mean,_) that Nellie left without saying good-bye meant she was still angry

xsubj(win,require,_) to win the America's Cup requires heaps of cash

dobj: Indicates the object in the grammatical relation between a predicate and its direct object.

dobj(read,book,_) read books

dobj(mail,Mary,iobj) mail Mary the contract

iobj The relation between a predicate and a non-clausal complement introduced by a preposition; **type** indicates the preposition introducing the dependent.

iobj(in,arrive,Spain) arrive in Spain

iobj(into,put,box) put the tools into the box

iobj(to,give,poor) give to the poor

obj2: The relation between a predicate and the second non-clausal complement in ditransitive constructions.

obj2(head,dependent)

obj2(give,present) give Mary a present

obj2(mail,contract) mail Paul the contract

dependent: The most generic relation between a head and a dependent dependent(introducer,head,dependent)

dependent(in,live,Rome) Marisa lives in Rome

Dependency Rel	**ID**	**Definition**	**Parent**
Adpositional Case Marker	AC	Preposition/postposition in a PP, annotated as a sister constituent of the determiner, adjectives, noun etc	PP
Adjective Component	ADC	Component of a multi-token adjective (MTA)	MTA
Apposition	APP	"inserted" phrase, further specifying/modifying the entity described by the matrix NP.	NP PP
Adverbial phrase Component	AVC	Component of a head-less AVP	ADV
conjunct	CJ	Constituent participating in coordination	any
comparative conjunction	CM	Linguistic particles introducing a comparison in comparative constructions (for example “grosser als” in German)	

dative	DA	Dative object/`free dative' (for languages having this case in the morphology/syntax))	S VP AP AVP
head	HD	The main elements in all kind of consitutents	S VP AP AVP
postnominal modifier	MNR	Postnominal NP/PP modifier	NP PP
negation	NG	the negation particle `nicht' (also modified)	any
genitive object	OG	Genitive objects of verbs, participles and certain adjectives (for language having the geneitive case in the morphology/syntax)	
predicate	PD	Predicative AP/NP/PP, typically in a copular construction	S VP
morphological particle	PM	two cases: the infinitival `zu' (zu gehen) the adjectival (superlative) `am' (am besten)	VZ AA
relative clause	RC		NP PP S VP AP

Annex B (informative) Annotation example

The following example shows how a multi-layered approach to syntactic annotation can be encoded in XML. The tagset in use is not pointing yet to the data categories, but such a linking will be included in the next version of the document.

```
<?xml version="1.0" encoding="UTF-8" ?>

<SynAF>

  <head>

    <annotation>

    <nodelabel>

    <feature name="wordForm" domain="T" />

    <feature name="pos" domain="T" />

    <value name="adjective" />

    <value name="subordinatingConjunction" />

    <value name="particle" />

    <value name="pronoun" />

    <value name="reflexivePronoun" />

    <value name="verb" />

    <value name="auxiliaryVerb" />

    <value name="modalVerb" />

   <value name="coordinatingConjunction" />

   <value name="definiteDeterminer" />

   <value name="indefiniteDeterminer" />

   <value name="adverb" />

   <value name="prefix" />

   <value name="ordinalNumeral" />

   <value name="interjection" />

   <value name="person" />

   <value name="noun" />
```

```
<value name="conjunction" />
<value name="properNoun" />
<value name="punctuation" />
<value name="possessive" />
<value name="numeral" />
<value name="cardinal" />
<value name="preposition" />
<value name="relPronoun"/>
<feature name="wordForm" domain="T" />
<value name="NP" />
<value name="PP"/>
<value name="AP"/>
<value name="ADVP"/>
<value name="VG"/>
<value name="SUBORDCLAUSE"/>
<value name="C"/>
<value name="sentence"/>
<value name="clause"/>
<value name="S"/>
<value name="DA"/>
</nodelabel>
<edgelabel>
<value name="subject"/>
<value name="deepObject"/>
<value name="directObject"/>
<value name="indirectObject"/>
<value name="prepositionPhraseAdjunct"/>
<value name="predicativeAdverbial"/>
<value name="xComp"/>
```

```
    <value name="head"/>
    <value name="mod"/>
    <value name="spec"/>
    </edgelabel>
</annotation>
</head>
<body>
<graph>
<nonterminals>
      <sentence id="1">
      <edge id="s1_3" label="DEEP_OBJ" nt_node="8" />
      <edge id="s1_4" label="PP_ADJUNCT" nt_node="5" />
      <node id="s1_5" label="PP" from="14" to="24" />
      <node id="s1_6" label="VG" from="26" to="26" />
      <node id="s1_7" label="AdvP" from="28" to="28" />
      <edge id="s1_9" label="head" t_node="34" />
      <edge id="s1_10" label="spec" t_node="30" />
      <edge id="s1_11" label="mod" t_node="32" />
      <node id="s1_12" label="VG" from="36" to="38" />
      <node id="s1_13" label="S" from="39" to="39" />
      </sentence>
 </nonterminals>
<terminals>
      <t id="14" wordForm="Fuer" lemma=" fuer" pos=“preposition“ />
      <t id="18" wordForm="Angaben" lemma=" angabe" pos=“definiteDeterminer“ />
      <t id="20" wordForm="in" lemma=" in" pos=“preposition“ />
      <t id="22" wordForm="unseren" lemma=" unser" pos=“possessive“ />
      <t id="24" wordForm="Listen" lemma=" list" pos=“noun“ />
      <t id="26" wordForm="wurde" lemma=" werd" pos=“auxiliaryVerb“ />
```

<t id="28" wordForm="grundsaetzlich" lemma=" grundsaetzlich" pos=“adverb“ />

<t id="30" wordForm="die" lemma="" pos=“definiteDeterminer“ />

<t id="32" wordForm="weitestgehende" lemma=" weitestgehend" pos=“adjective“ />

<t id="34" wordForm="Bilanz" lemma=" bilanz" pos=“noun“ />

<t id="36" wordForm="zugrunde" lemma=" zugrunde" pos=“particle“ />

<t id="38" wordForm="gelegt" lemma=" leg" pos=“verb“ />

<t id="39" wordForm="." lemma=" ." pos=“punctuation“ />

</terminals>

</graph>

</body>

</SynAF>

Bibliography

Abeillé, A., S. Hansen-Schirra, and H. Uszkoreit (eds.), 2003. *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora* (LINC-03).

Calzolari, N., J. McNaught, and Zampolli A. (eds). 1996. *EAGLES: Introduction*. http://www.ilc.cnr.it/EAGLES96/edintro/edintro.html

Calzolari, N., F. Bertagna F., A. Lenci, and M. Monachini (eds). 2003. Standards and Best Practice for Multilingual Computational Lexicons. *MILE (The Multilingual ISLE Lexical Entry). ISLE CLWG Deliverable,* D2.2 & 3.2, Pisa.

Ide, Nancy, and Laurent Romary, 2004. A Registry of Standard Data Categories for Linguistic Annotation. *Proceedings of the Fourth Language Resources and Evaluation Conference* (LREC), Lisbon, 135-39.

Ide, Nancy, and Laurent Romary, 2004. International Standard for a Linguistic Annotation Framework. *Journal of Natural Language Engineering*, 10:3-4, 211-225.

Ide, Nancy, and Laurent Romary, 2006. Representing Linguistic Corpora and Their Annotations. *Proceedings of the Fifth Language Resources and Evaluation Conference* (LREC), Genoa, Italy.

Ide, Nancy. 2007. GrAF: A Graph-based Format for Linguistic Annotations. *Proceedings of the LAW Workshop at ACL 2007*, Prague.

Montemagni, S, F. Barsotti, M. Battista, N. Calzolari, A. Lenci O. Corazzari, A. Zampolli, F. Fanciulli, M. Massetani, R. Basili R. Raffaelli, M.T. Pazienza, D. Saracino, F. Zanzotto, F. Pianesi N. Mana, and R. Delmonte. 2003. Building the Italian Syntactic-Semantic Treebank. In Anne Abeillé (ed), *Building and Using Syntactically Annotated Corpora*, pages 189--210. Kluwer, Dordrecht.

Rumbaugh, J., Jacobson I., and G. Booch. 2004. *The Unified Modeling Language Reference Manual*, 2nd edition. Addison Wesley.

Project websites:

The EAGLES Initiative: http://www.ilc.cnr.it/EAGLES96/home.html

The LIRICS Project: http://lirics.loria.fr

The SPARKLE Project: http://www.ilc.cnr.it/sparkle/sparkle.htm

The TIGER Project: http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus/