LIRICS

Deliverable 3.3B

Interim Report: WD of syntactic annotation standard CD ballot

Project reference number	e-Content-22236-LIRICS
Project acronym	LIRICS
Project full title	Linguistic Infrastructure for Interoperable Resource and Systems
Project contact point	Laurent Romary, INRIA-Loria 615, rue du jardin botanique BP101. 54602 Villers lès Nancy (France) romary@loria.fr
Project web site	http://lirics.loria.fr
EC project officer	Erwin Valentini
Document title	Interim Report: CD document, 1st version
Deliverable ID	3.3B
Document type	Report
Dissemination level	Confidential
Contractual date of delivery	M27 for D3.3B
Actual date of delivery	09.02.2007
Status & version	Draft
Work package, task & deliverable responsible	DFKI
Author(s) & affiliation(s)	Thierry Declerck, Mirjam Kessler, Ulrich Krieger, Tania Avgustionva and Valia Kordoni (DFKI)
Additional contributor(s)	Many experts from national standardisation bodies and ISO.
Keywords	Syntax, Annotation, Standards, Tree-Banks

Document evolution

Version	date	version	date
0.9	22nd Dec. 2006		
1.0	31st Jan 2007		
2.0	30th August 2007		

ISO 24615:2007

Introduction

We present in this an updated version of the deliverable D3.3A the actual state of work of SynAF (Syntactic Annotation Framework), which will be soon submitted as a CD (end of September). SynAF has been reviewed in the line of joint discussions with the editors of LAF, LMF and MAF, discussions which started at the plenary ISO TC37 in Beijing (21-26 August 2006), and here we include some modifications suggested by participants to discussion on SynAF at a ISO meeting in Paris (Mai 2007) and at the plenary ISO TC37 meeting in Provo (August 2007), which we decided to included in the version to be submitted for a CD ballot, postponing thus the submission to September 2007.

.

Content:

Reference number of working document: ISO/TC 37/SC 4 N??? Rev.4

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

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 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 CD 24613, *Lexical Markup Framework (LMF)*, and ISO CD 24611, *Morphosyntactic Annotation Framework (MAF)*, and ISO NP 2461x-1, *Semantic Annotation Framework - Part 1: Time and events (SemAF-Time)*.

Annexes A forms an integral part of this International Standard.

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.

The document proposes a metamodel 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).

1 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.
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] 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).

2 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.

3 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:

3.1

Annotation: Some code associated with parts of text and providing for additional information about this part of text. 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.

Category: a feature value providing the content of a node.

Clause: a group of *phrases*, usually containing a verb, which valency also determines the number of obligatory clause elements (*phrases*). 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.
Constituent: all types of nodes we find in the syntactic annotation are building a constituent (to be revised)
Constituency relation: a syntactic grouping of words *(i*nto *phrases)*, *phrases* (into *clauses*) or *clauses (*into a *sentence)* on the base of structural (or hierarchical) properties
Dependency relation: a 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: a triplet with a source node, a target node, and a label. Non-Terminal nodes have an outgoing constituency edges (to be defined)

Grammatical function: Constituents can have a grammatical function within their embedding syntactic environment. So a 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.

Graph: a well understood model for representing objects that can be viewed as a connected set of more elementary sub-objects
Head: the most important word in a constituent; the word that carries the main meaning of the phrase. The head of a constituent cannot be left out.

Hierarchy: the relative position of constituents in a syntactic tree.
Human language technology: technology as applied to natural languages

Label: a feature value providing the content of an edge.
Main clause: a finite clause, which can act on its own as a complete sentence. Example: *The train has some delay.*
Modifier: a modifier is a part of the constituent which ascribes a property to the head of the constituent. A modifier may be placed before or after the head of the phrase (pre-modifier or post-modifier). Modifiers are optional in a constituent

Natural language processing NLP: field covering knowledge and techniques involved in the processing of linguistic data by a computer

Node: pair consisting of a (possibly multiple) span and a category,

Non-Terminal nodes have an outgoing constituency edges (to be defined)

Phrases: a word or group of words which can fulfil a *grammatical function* in a clause. 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 phrases, verb phrases, adjective phrases, adverb 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.

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

Span: a pair of points identifying a segment of the document submitted to syntactic annotation. 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.

Specifier: a specifier in a constituent specifies the head (or the combination of modifier and head) with information about number, definiteness, proximity and ownership

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

Sucategorization 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 relation for a sentence.
Syntax: The way words are grouped together in linguistically meaningful units, and the relations that exist between those units.
Syntactic argument: one of the essential and functional elements in a clause that identifies the participants in the process referred to by a verb

Syntax Tree: a syntactic graph in which each node has a single parent.
Terminal node: refers to a single wordFrom/lexical unit or a span with lentgh=0, and the node and the wordForm/lexical unit have identical span.

4 Key standards used by SynAF

4.1 Unicode

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

4.2 ISO 12620 Data Category Registry (DCR)

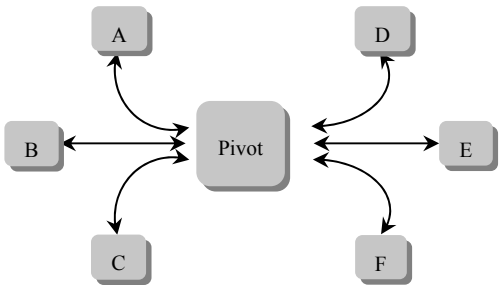
The designers of an SynAF conformant annotation shall use data categories from the ISO 12620 Data Category Registry (DCR), or a tagset that can be mapped onto the data categories.

4.3 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).

5 Embedding SynAF in the LAF model[6]

We want to embed the meta-model of SynAF in the more generic Linguistic AnnotationFramework (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 an abstract model for annotations instantiated by a *pivot format*, into and out of which annotations are mapped for the purposes of exchange.



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

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).

6 The SynAF Metamodel

6.1 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

[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.

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).

6.2 The SynAF diagram (to be represented in UML)

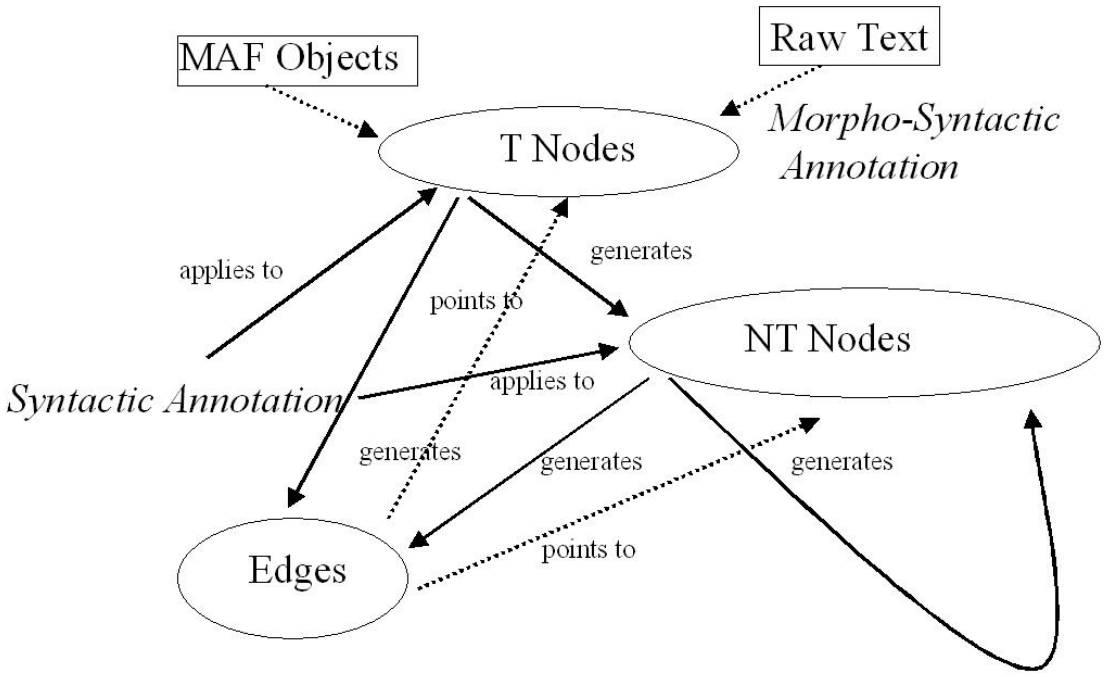


Figure 1: The SynAF metamodel

6.2.1 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.

6.2.2 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).

6.2.3 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.

6.2.4 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.1Constituency

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.2Dependency

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 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. The Grammaticl Functions (dependencies) are encoded here within the "clause" XML elements. The dependencies within constituents are not annotated explicitly in this example.

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<document id="ww92-short.xml" lang="de">
<paragraph id="p1" corresp="">
 <sentence id="s4" corresp="">
  <clauses>
    <clause id="cl1" from="c1" to="c22" pred_struct="c18+1"
pred_subcat_stem="sei" GF_Subj="c19+1" NP_List="c19+1"
VG_List="c18+1"/>
    <clause id="cl2" from="c23" to="c26" pred_struct="c22+1"
pred_subcat_stem="find" GF_Subj="c24+1" GF_Acc_Obj="c23+1 "
NP_List="c23+1 c24+1" VG_List="c22+1"/>
  </clauses>
  <chunks>
    <chunk id="c1" from="1" to="1" type="VG"/>
    <chunk id="c2" from="2" to="3" type="NP"/>
    <chunk id="c3" from="4" to="4" type="W"/>
    <chunk id="c4" from="5" to="5" type="W"/>
    <chunk id="c5" from="6" to="6" type="VG"/>
    <chunk id="c6" from="7" to="7" type="NP"/>
    <chunk id="c7" from="8" to="10" type="NP"/>
    <chunk id="c8" from="11" to="11" type="W"/>
  </chunks>
  <text>
    <token id="1" infl="[92 93 94 95]" pos="3" lemma="sei"
tc="22">Sind</token>
    <token id="2" infl="[2 5 20 6 13 23 9 16]" pos="8"
lemma="kein" tc="21">keine</token>
    <token id="3" infl="[6 7 8 9]" pos="1" lemma="angabe"
tc="22">Angaben</token>
    <token id="4" infl="[25]" pos="5" lemma="erhaeltlich"
tc="21">erhaeltlich</token>
    <token id="5" infl="[439]" pos="21" lemma=","
tc="1">,</token>
    <token id="6" infl="[204 205 209 206]" pos="2"
lemma="find" tc="21">findet</token>
    <token id="7" infl="[423 431 428 432 424 433 430 434]"
pos="11" lemma="sich" tc="21">sich</token>
    <token id="8" infl="[10 12]" pos="7" lemma="d-det"
tc="21">das</token>
    <token id="9" infl="[10 11 12 13 14 16]" pos="1"
lemma="kuerzel" tc="22">Kuerzel</token>
    <token id="10" tc="19">KA</token>
    <token id="11" infl="[440]" pos="21" lemma="."
tc="1">.</token>
```

```
  </text>
 </sentence>
 <sentence id="s5" corresp="">
  <clauses>
    <clause id="cl1" from="c1" to="c30" pred_struct="c27+1"
pred_subcat_stem="bedeut" GF_Subj="c26+1" NP_List="c26+1"
VG_List="c27+1"/>
  </clauses>
  <chunks>
    <chunk id="c1" from="1" to="2" type="NP"/>
    <chunk id="c2" from="3" to="3" type="VG"/>
    <chunk id="c3" from="4" to="14" type="SUBORD_CLAUSE"/>
    <chunk id="c4" from="15" to="15" type="W"/>
  </chunks>
  <text>
    <token id="1" infl="[17 10 12]" pos="8" lemma="ein"
tc="22">Ein</token>
    <token id="2" infl="[17 18 19]" pos="1" lemma="strich"
tc="22">Strich</token>
    <token id="3" infl="[204 205 209 107 206]" pos="2"
lemma="bedeut" tc="21">bedeutet</token>
    <token id="4" tc="1" lemma="," infl="[439]"
pos="21">,</token>
    <token id="6" tc="21" lemma="dass" pos="20">dass</token>
    <token id="8" tc="21" lemma="zutreff" infl="[204]"
pos="2">zutrifft</token>
    <token id="9" tc="21" lemma="nicht" pos="22">nicht</token>
    <token id="11" tc="22" lemma="wert" infl="[18 20 21 23]"
pos="1">Werte</token>
    <token id="12" tc="21" lemma="d-det" infl="[2 5 20 6 13 23
9 16]" pos="7">die</token>
    <token id="14" tc="21" lemma="nicht"
pos="22">nicht</token>
    <token id="16" tc="21" lemma="vergleichbar" infl="[25]"
pos="5">vergleichbar</token>
    <token id="18" tc="21" lemma="sei" infl="[92 93 94 95]"
pos="3">sind</token>
    <token id="20" tc="21" lemma="oder" pos="19">oder</token>
    <token id="22" tc="22" lemma="kriterium" infl="[10 11 12]"
pos="1">Kriterium</token>
    <token id="23" tc="21" lemma="d-det" infl="[10 12]"
pos="7">das</token>
    <token id="25" tc="22" lemma="spalt" infl="[18 20 21 23]"
pos="1">Spalte</token>
    <token id="26" tc="21" lemma="dies" infl="[17 3 4 21 7
14]" pos="7">dieser</token>
    <token id="28" tc="21" lemma="d-det" infl="[10 12]"
pos="7">das</token>
    <token id="29" tc="21" lemma="fuer" infl="[102]"
pos="23">fuer</token>
    <token id="30" tc="22" lemma="unternehmen" infl="[10 11 12
13 14 15 16]" pos="1">Unternehmen</token>
    <token id="31" infl="[440]" pos="21" lemma="."
tc="1">.</token>
  </text>
 </sentence>
```

```
 <sentence id="s6" corresp="">
  <clauses>
    <clause id="cl1" from="c1" to="c37" pred_struct="c31+1"
pred_subcat_stem="bezieh" GF_Subj="c30+1" GF_Acc_Obj="c32+1 "
PP_Adjunkt="c35+1" NP_List="c30+1 c32+1" PP_List="c33+1 c34+1
c35+1" VG_List="c31+1"/>
  </clauses>
  <chunks>
    <chunk id="c1" from="1" to="3" type="NP"/>
    <chunk id="c2" from="4" to="4" type="VG"/>
    <chunk id="c3" from="5" to="5" type="NP"/>
    <chunk id="c4" from="6" to="8" type="PP"/>
    <chunk id="c5" from="9" to="10" type="PP"/>
    <chunk id="c6" from="11" to="11" type="PP"/>
    <chunk id="c7" from="12" to="12" type="W"/>
  </chunks>
  <text>
    <token id="1" infl="[2 5 20 6 13 23 9 16]" pos="7"
lemma="d-det" tc="22">Die</token>
    <token id="2" infl="[41 42 43 44 45 46 47 48 49 50 51 52
53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73
74 75 76 77 78 79 80 81 82 83 84]" pos="6" lemma="angeb"
tc="21">angegebenen</token>
    <token id="3" infl="[18 20 21 23]" pos="1" lemma="wert"
tc="22">Werte</token>
    <token id="4" infl="[92 93 94 95 96 97 98 99 100 101]"
pos="2" lemma="bezieh" tc="21">beziehen</token>
    <token id="5" infl="[423 431 428 432 424 433 430 434]"
pos="11" lemma="sich" tc="21">sich</token>
    <token id="6" infl="[24 102]" pos="23" lemma="in"
tc="21">in</token>
    <token id="7" infl="[17 3 4 21 7 14]" pos="7" lemma="d-
det" tc="21">der</token>
    <token id="8" infl="[2 3 4 5]" pos="1" lemma="regel"
tc="22">Regel</token>
    <token id="9" infl="[24 102]" pos="23" lemma="auf"
tc="21">auf</token>
    <token id="10" infl="[20 21 23]" pos="1" lemma="abschluss"
tc="22">Abschluesse</token>
    <token id="11" tc="11">31.</token>
    <token id="12" tc="11">1991</token>
    <token id="13" tc="21" lemma="zum" infl="[24]"
pos="23">zum</token>
    <token id="14" tc="22" lemma="dezember" infl="[17 18 19 20
21 22 23]" pos="1">Dezember</token>
    <token id="15" infl="[440]" pos="21" lemma="."
tc="1">.</token>
  </text>
 </sentence>
 <sentence id="s7" corresp="">
  <clauses>
   </clauses>
  <chunks>
    <chunk id="c1" from="1" to="8" type="PP"/>
    <chunk id="c2" from="9" to="9" type="VG"/>
    <chunk id="c3" from="10" to="15" type="PP"/>
```

```
    <chunk id="c4" from="16" to="16" type="VG"/>
    <chunk id="c5" from="17" to="17" type="W"/>
  </chunks>
  <text>
    <token id="1" infl="[24 102]" pos="23" lemma="in"
tc="22">In</token>
    <token id="2" infl="[41 42 43 44 45 46 47 48 49 50 51 52
53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73
74 75 76 77 78 79 80 81 82 83 84]" pos="5" lemma="einig"
tc="21">einigen</token>
    <token id="3" infl="[22]" pos="1" lemma="fall"
tc="22">Faellen</token>
    <token id="4" pos="20" lemma="wie" tc="21">wie</token>
    <token id="5" pos="22" lemma="beispielsweise"
tc="21">beispielsweise</token>
    <token id="6" tc="22" lemma="siemens" infl="[10 0 11 12 13
14 16]" pos="1">Siemens</token>
    <token id="7" pos="19" lemma="oder" tc="21">oder</token>
    <token id="8" tc="19">MAN</token>
    <token id="9" infl="[204]" pos="2" lemma="werd"
tc="21">wird</token>
    <token id="10" infl="[102]" pos="23" lemma="per"
tc="21">per</token>
    <token id="11" infl="[10 11 12]" pos="1" lemma="ende"
tc="22">Ende</token>
    <token id="12" infl="[17 18 19 20 21 23]" pos="1"
lemma="september" tc="22">September</token>
    <token id="13" pos="17" lemma="beziehungsweise"
tc="21">beziehungsweise</token>
    <token id="14" infl="[10 11 12]" pos="1" lemma="ende"
tc="22">Ende</token>
    <token id="15" infl="[17 18 19]" pos="1" lemma="juni"
tc="22">Juni</token>
    <token id="16" infl="[204 205 107 206]" pos="2"
lemma="bilanzier" tc="21">bilanziert</token>
    <token id="17" infl="[440]" pos="21" lemma="."
tc="1">.</token>
  </text>
 </sentence>
…
</Paragraph>
</document>
```

Bibliography

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

Calzolari N., Mc Naught J., Zampolli A. 1996 Eagles, editors introduction. http://www.ilc.cnr.it/EAGLES96/edintro/edintro.html

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

Rumbaugh J., Jacobson I., Booch G. The unified modeling language reference manual, second edition Addison Wesley 2004

S. Montemagni, 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. Building the Italian Syntactic-Semantic Treebank. In Anne Abeillé, editor, Building and Using syntactically annotated corpora, pages 189--210. Kluwer, Dordrecht, 2003.

Nancy Ide 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.

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

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

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

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/