# TELIX: An RDF-Based Model for Linguistic Annotation

Emilio Rubiera[1], Luis Polo[1], Diego Berrueta[1], and Adil El Ghali[2]

[1] Fundación CTIC
name.surname@fundacionctic.org
[2] IBM CAS France
adil.elghali@fr.ibm.com

**Abstract.** This paper proposes the application of the RDF framework to the representation of linguistic annotations. We argue that RDF is a suitable data model to capture multiple annotations on the same text segment, and to integrate multiple layers of annotations. As well as using RDF for this purpose, the main contribution of the paper is an OWL ontology, called TELIX (Text Encoding and Linguistic Information eXchange), which models annotation content. This ontology builds on the SKOS XL vocabulary, a W3C standard for representation of lexical entities as RDF graphs. We extend SKOS XL in order to capture lexical relations between words (e.g., synonymy), as well as to support word sense disambiguation, morphological features and syntactic analysis, among others. In addition, a formal mapping of feature structures to RDF graphs is defined, enabling complex composition of linguistic entities. Finally, the paper also suggests the use of RDFa as a convenient syntax that combines source texts and linguistic annotations in the same file.

## 1 Introduction

A linguistic annotation is a descriptive or analytic mark dealing with raw language data extracted from texts or any other kind of recording. A large and heterogeneous number of linguistic features can be involved. Typically linguistic annotations include part-of-speech tagging, syntactic segmentation, morphological analysis, co-references marks, phonetic segmentation, prosodic phrasing and discourse structures, among others.

There is an increasing need for vendors to interchange linguistic information and annotations, as well as the source documents they refer to, among different software tools. Text analysis and information acquisition often require incremental steps with associated intermediate results. Moreover, tools and organizations make use of shared resources such as thesauri or annotated corpus. Clearly, appropriate standards that support this open information interchange are necessary. These standards must provide the means to model and serialize the information as files.

In [16], the following requirements for a linguistic annotation framework are identified: expressive adequacy, media independence, semantic adequacy, uniformity, openness, extensibility, human readability, processability and consistency.

E. Simperl et al. (Eds.): ESWC 2012, LNCS 7295, pp. 195–209, 2012.

We postulate that the RDF framework features these properties, and therefore constitutes a solid foundation. RDF graphs make use of custom vocabularies defined by ontologies. Therefore, we introduce TELIX, a lightweight ontology that provides comprehensive coverage of linguistic annotations and builds on previous resources, such as feature structures and SKOS concept schemes. TELIX takes advantage of the RDF/OWL expressive power and is compatible with legacy materials. Moreover, translations from traditional linguistic annotation formats to RDF is possible as shown in [6].

This paper is organized as follows. The next section revises the RDF framework and discusses the advantages of representing linguistic annotations as RDF graphs. The main contribution of this paper is TELIX, an OWL ontology which is described in Section 3. Details on how to embed linguistic annotations using the RDFa syntax are given in Section 4. Finally, Section 5 examines previous initiatives, and conclusions and connections to ongoing, similar proposals are presented in Section 6.

## 2 Linguistic Annotations as RDF Graphs

Effective and open communication between independent parties requires an agreement on shared standards. This paper proposes the application of the RDF framework to linguistic annotations. We sustain that RDF, in combination with the TELIX ontology which is discussed in the next section, facilitate the exchange of expressive linguistic annotations among software tools.

RDF is the W3C recommended framework for making resource descriptions available on the web. RDF graphs can be authored, stored, and web-published. They can also be queried by means of the SPARQL query language, which also defines a web service protocol (SPARQL endpoints) to enable queries on remote graphs. In fact, RDF graphs capturing the linguistic annotations of a given text can be located anywhere on the web, and retrieved as needed, as long as a simple set of principles known as “linked data” are adopted [14]. RDF graphs can be split into many interlinked files, or combined into a single one. Using the RDFa syntax, it is even possible to combine the RDF graph and the source text document into a single file, as it will be discussed in Section 4. This flexibility satisfies several exchange scenario requirements. For instance, a reduced number of files eases management, minimizes the risk of inconsistencies and simplifies internal references. On the other hand, some scenarios demand fine-grained separation of aspects and annotation layers into multiple files.

One of the advantages of RDF is its ability integrate multiple annotation layers in the same framework. The heterogeneity of linguistic analysis is reconciled thanks to the versatility of the RDF graph-based data model. For instance, a single RDF graph may include both syntactic and discourse structures without conflict or interference. Moreover, uniform identifiers (URIs) make it possible to link linguistic resources across multiple RDF graphs, even if they belong to alternative annotation layers. This gluing power is a notable advantage over other annotation formats, such as the XML-based alternatives.

The introduction of URIs as a means to identify and make reference to structures, and particularly text fragments, is a notable departure from more traditional techniques based on positions and offsets, i.e., counting the number of preceding structures or characters. These location-based references are sometimes multi-dimensional (e.g., "the token starting at character 23 of sentence 8"), and are dependent on assumptions about text segmentation. For instance, it is implicit that the reference producer and consumer have previously agreed on the sentence segmentation algorithm. Sometimes, these references are unreliable due to text encoding divergences, e.g., the number of bytes used to represent non-ASCII characters and line breaks are an historical source of interoperability issues. Resolving location-based references is computationally expensive because it requires repeating the segmentation of the text. Moreover, location-based references are extremely sensitive to changes in the source document: even slight modifications of the text render the references invalid, forcing a recalculation. URIs do not present any of these issues, and can be used to make unambiguous, maintainable and easy to resolve references to structures.

## 3 TELIX, an Ontology of Linguistic Information

This section introduces TELIX (Text Encoding and Linguistic Information eXchange), an OWL vocabulary designed to permit the representation linguistic information as RDF graphs. It extends SKOS XL, which allows capturing lexical entities as RDF resources, and it overcomes SKOS limitations of expressiveness. TELIX introduces a number of classes and properties to provide natural language acquisition and extraction tools with interchangeable, multilingual lexical resources such as dictionaries or thesauri, as well as representing the outcomes of text analyses, i.e., annotations content. The reader is invited to read the TELIX specification [11] where complete details about the ontology and modeling decisions are provided.

The TELIX namespace is `http://purl.org/telix/ns#`, although for the sake of brevity, in this paper it is assumed to be the default namespace. The namespace URL uses HTTP content negotiation to redirect to the OWL specification or the HTML documentation. The OWL file can be downloaded from `http://purl.org/telix/telix.owl`.

### 3.1 Text Segmentation

TELIX provides machinery to describe a given piece of text. More precisely, TELIX types the document and the corpus with Dublin Core concepts, namely `dctype:Text` and `dctype:Collection`. A set of textual units to grasp the text structure are also defined: `Section`, `Paragraph`, `Sentence` and `Token`. In addition, inspired by LAF annotations [16], TELIX introduces the superconcept `Segment` to capture any fragment of choice that does not match any of the former. These entities can be combined by annotation tools to segment the primary data.

Multiple segmentations of the same textual fragment are possible. For example, tokens are assumed to be auxiliary entities, defined as contiguous string of alphabetic or numerical characters and separated by separation characters such as whitespaces. Note that punctuation is included in the resulting list of tokens in some parsers and discourse analysis tools. Tokens enable tools to provide divergent lexical understandings of a given piece of text. The same bunch of words can be interpreted differently depending on the focus of the analysis. Consider, for instance, the string "maraging steel", composed of two tokens ($t1$, $t2$). It can be seen either as a composition of two single words "[maraging$_{t1}$] [steel$_{t2}$]" or as the collocation"[maraging$_{t1}$ steel$_{t2}$]" making the whole string a single lexical unit. These lexical issues are critical when dealing with technical terminology, where term boundaries are fuzzy and disputed. TELIX does not enforce a concrete analysis regarding the lexical disambiguation of texts. The concept `Token` provides a free-focus word segmentation of the text, over which upper segmentation layers (such as term identification) can be built.

More refined textual units, such as title, chapter, itemized list, etc., are not part of TELIX. However, as TELIX is an OWL ontology, it can be extended or combined with other ontologies to fit the specific requirements of a particular application.

### 3.2 Words and Senses in RDF

The W3C SKOS vocabulary [3] is a lightweight OWL ontology created to facilitate web-oriented taxonomies and thesauri. SKOS supports multilingual information by means of three labeling properties: `skos:prefLabel`, `skos:altLabel` and `skos:hiddenLabel`. However, SKOS lacks the expressiveness to fully describe the labels of concepts as it treats them as RDF literals, which cannot play the role of subject in a triple. Labels cannot then be further detailed nor linked to other labels. To overcome this limitation, W3C recommends SKOS eXtension for Labels (SKOS XL) [20], which is an extension to the vanilla SKOS vocabulary that provides mechanisms for identifying and describing lexical entities. Fundamentally, a new class `skosxl:Label` is introduced to deal with lexical entities as RDF resources. A label in SKOS-XL can be either a single word or a multiword expression [22], such as a collocation.

As natural languages are inherently complex, in linguistics, three complementary entities, with different natures and properties, are distinguished: concepts, lexemes and words (occurrences). For the sake of precision, TELIX refines SKOS XL to seamlessly meet the requirements of linguistic text analysis as explained above. Table 1 sums up the TELIX proposal. Note that the entities are described according to their bound to languages and texts.

**Concepts**, abstract ideas formed in mind, can be extrapolated, to a greater or lesser extent, from language to language. Concepts are also called *meanings*. In order to be treated as single resources, concepts are represented in TELIX both as instances of `skos:Concept` or elements of a domain ontology.

A language captures these concepts by means of words or sets of words. However, a distinction must be made between the physical realization of the words

**Table 1.** TELIX proposal to represent concepts, lexemes and words (occurrences)

| Linguistic Entity | OWL class | Language-dependence | Text-dependence |
|---|---|---|---|
| Concept | `skos:Concept` | - | - |
| Lexeme | `skosxl:Label` | + | - |
| Word (occurrence) | `LabelOccurrence` | + | + |

(in a speech or written down in a document) and their abstract interpretation. The latter is often called a **lexeme**, i.e., a meaningful linguistic unit belonging to the vocabulary of a language. A lexeme is merely a theoretical notion not traceable in actual textual samples. TELIX put lexemes at the same level as `skosxl:Label`, thus restricting the interpretation of lexical entities under the specification. Canonical forms of lexemes (*lemmas*) are the values of the property `skosxl:literalForm` in a SKOS XL terminology. TELIX also refines the generic property `skosxl:labelRelation` in order to capture lexical relationships between lexemes, for instance synonymy (for *synsets*), homonymy and hyponymy. The connections between lexemes and concepts are borrowed from the relationship between SKOS and SKOS XL.

Finally, **words** occur in natural language materialization, typically being part of a communicative act. For instance, a text piece such as "Bronze, an alloy of copper and tin, was one of the first alloys discovered" contains 14 words including the words "alloy" and "alloys" (i.e., two occurrences of the lexeme "alloy"). This interpretation of a word is not a theoretical entity, but real, concrete realization of the natural language. Therefore, TELIX introduces the class `LabelOccurrence` to capture physical realizations[1]. For each term identified in a given sentence, a new RDF resource (typed as a `LabelOccurrence`) is created and linked to its corresponding `skosxl:Label` by means of the property `realizes`. Morphosyntactic information of word forms is captured by RDF-based feature structures, as described below and illustrated in the example of Figure 2.

The word sense annotation of the term occurrence may involve connecting the lexical entities (`skosxl:Label` instances) to domain ontologies and SKOS thesauri concepts. Note that, in this situation, links between the occurrence and its word senses are carried out by the indirect mediation of `skosxl:Label` entities. Complementarily, TELIX also provides the property `sense` to directly relate the label occurrence and its meaning. Thus, it is not necessary to go from word occurrences to word senses through word definitions. Figure 1 illustrates both mediated and direct links. Note that the resource used to disambiguate the lexical entities is drawn from DBpedia [2].

### 3.3 Linguistic Feature Structures

Lexical items descriptions can be enriched by means of feature structures that capture their grammatical properties. Feature structures are a recursive

[1] At the time of writing, TELIX only covers written realizations in texts. In the future, it is planned to extend the ontology to capture other forms of word materializations, such as sounds/phonemes.

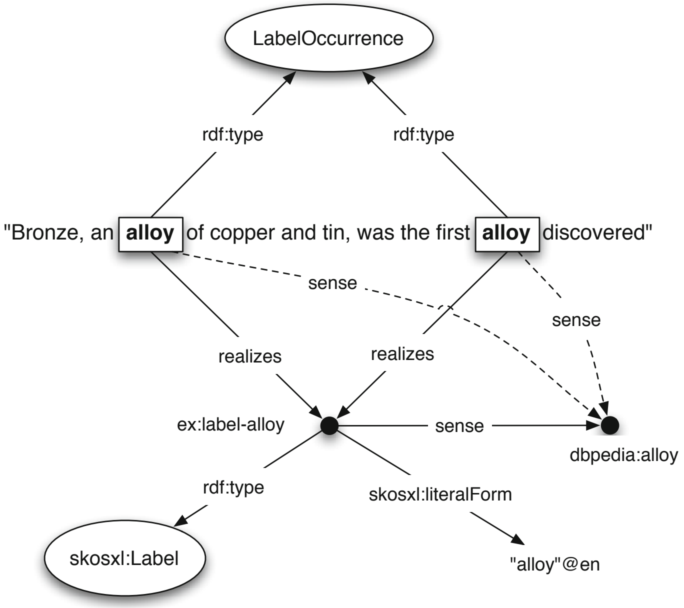

**Fig. 1.** This figure shows the links between label occurrences in a text and lexical entities in the lexicon. In addition, the property `sense` attaches meanings to both words and their occurrences.

representation of linguistic information. They are sets of feature-value pairs, where features stand for atomic grammatical properties, and values are either atomic symbols or other feature structures.

A common mathematical representation of feature structures is a direct acyclic graph [21,18]. Let us assume there are two disjoint, finite sets $\mathcal{F}$ of *feature names* and $\mathcal{S}$ of *species names*. In linguistics, $\mathcal{S}$ interprets the sorts (types) of entities according to a grammatical theory (for instance, cases, verbs, types of phrases, etc.), and $\mathcal{F}$ interprets the grammatical features, such as agreement or tenses of verbs. A *feature graph* is an ordered triple $\mathcal{G} = \langle \mathcal{V}, \phi, \psi \rangle$, where: $\mathcal{V}$ is the set of vertices of $\mathcal{G}$; $\phi$ is a function which maps each feature name $f \in \mathcal{F}$ to a partial function $\phi(f)$ from $\mathcal{V} \times \mathcal{V}$; and $\psi$ is a function which maps each species name $s \in \mathcal{S}$ a subset of $\mathcal{V}$.

We define directed graph as an ordered pair $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V}$ is again the set of vertices (nodes) of the graph and $\mathcal{E}$ a subset of $\mathcal{V} \times \mathcal{V}$, called the edges of $\mathcal{G}$. The node $n_j$ is *accessible* from $n_i$, where $n_i, n_j \in \mathcal{V}$, if there is a path (finite sequence of edges) from $n_i$ to $n_j$. A feature graph is then reinterpreted as a directed graph. Given the feature graph $\mathcal{G} = \langle \mathcal{V}, \phi, \psi \rangle$, the relation $\mathcal{E}_\phi \subseteq \mathcal{V} \times \mathcal{V}$ is defined as the union of the interpretation of the feature names: $\bigcup\{\phi(f) : f \in \mathcal{F}\}$. Therefore $\langle \mathcal{V}, \mathcal{E}_\phi \rangle$ is a directed graph.

Finally, we define a *feature structure* as a quadruple $\langle n_0, \mathcal{V}, \phi, \psi \rangle$, where: $\langle \mathcal{V}, \phi, \psi \rangle$ is a feature graph; and $\langle n_0, \mathcal{V}, \mathcal{E}_\phi \rangle$ is a directed graph where $n_0$ is the root of the structure, as every node $n_k \in \mathcal{V}$ is accessible from it.

### 3.4 Feature Structures as RDF Graphs

Given the definitions above, the translation of feature structures to RDF is straightforward because both data structures are directed graphs. The RDF language distinguishes three sets of disjoint syntactic entities. Let $U$ denote the set of *URI references* and $Bl$ the set of *blank nodes*, i.e., variables. Let $L$ be the set of *literals*, i.e., data values such as floats or strings. An RDF graph $G$ is a set of *triples*, where the tuple $\langle s\ p\ o \rangle \in (U \cup Bl) \times U \times (U \cup Bl \cup L)$ is called an RDF triple. In the tuple, $s$ is the subject, $p$ the predicate and $o$ the object.

The vocabulary of an RDF graph $G$, denoted by $V(G)$, is the set of names that occur as subject, predicate or object of a triple in $G$ [13]. A *simple interpretation* $I$ of a vocabulary $V$ is a 6-tuple $I = (R_I, P_I, E_I, S_I, L_I, LV_I)$, where $R_I$ is a non-empty set, called the set of resources or the universe, $P_I$ is the set of properties, $LV_I$ is the set of literal values, which is a subset of $R_I$ that contains at least all the plain literals in $V$, and where $E_I$, $S_I$ and $L_I$ are functions:

- $E_I$: $P_I \to \wp(R_I \times R_I)$, where $\wp(X)$ denotes the power set of the set X. The function $E_I$ defines the extension of a property as a set of pairs of resources.
- $S_I$: $(V \cap U) \to (R_I \cup P_I)$ defines the interpretation of URI references.
- $L_I$: $(V \cap L) \to (L \cup R_I)$ defines the interpretation of literals.

If $t = \langle s\ p\ o \rangle$ is an RDF triple, then a simple interpretation $I$ of a vocabulary $V$ is said to satisfy $t$ if $s, p, o \in V$, $I(p) \in P_I$, and $(I(s), I(o)) \in E_I(I(p))$. We extend this interpretation with $Class$ and $type$ from the RDFS vocabulary, where $I(type) \in P_i$, and the set $C_I$ of classes of $I$ is defined as: $C_I = \{c \in R_I : (c, I(Class)) \in E_I(I(type))\}$. Thus, given a feature structure $\mathcal{G} = \langle n_0, \mathcal{V}, \phi, \psi \rangle$ and an augmented RDF vocabulary $V \cup \{type, Class\}$, we define a *mapping* $\pi = \langle \pi_1, \pi_2, \pi_3, \pi_4, \pi_5 \rangle$ to transform $\mathcal{G}$ to an RDF graph $G$ given an interpretation $I$ as follows:

- $\forall n \in \mathcal{V} : \pi_1(n) = \eta$, where $\eta \in U$ and $I(\eta) \in R_I$. This mapping function introduces a new node $\eta$ in $G$. In other words, $\pi_1$ is a labeling function, providing URIs for each node of the feature structure. Note that the root node $n_0$ is also included in the transformation.
- $\forall f \in \mathcal{F} : \pi_2(f) = p$, where $p \in U$, and $I(p) \in P_I$. Observe that $p$ is a property defined in the TELIX vocabulary.
- $\forall s \in \mathcal{S} : \pi_3(s) = c$, where $c \in U$, and $I(c) \in (R_I \cap C_I)$. Note that $c$ is a class defined in the TELIX vocabulary.
- $\pi_4(\phi) = E_I$, where for each pair $\langle n_i, n_j \rangle \in \phi(f)$ in $\mathcal{G}$, the application of $\pi_4$ returns $(I(\pi_1(n_i)), I(\pi_1(n_j)) \in E_I(I(\pi_2(f))) = (I(\eta_i), I(\eta_j)) \in E_I(I(p))$ in $G$. This mapping retains the feature names interpretation of $\mathcal{G}$, $\phi : \mathcal{F} \mapsto \mathcal{V} \times \mathcal{V}$, in $G$. In other words, this mapping is an isomorphism between the

$$\begin{bmatrix} \textit{word} & \\ \text{POS} & \text{noun} \\ \text{AGR} & \begin{bmatrix} \text{GEND} & \text{masc} \\ \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix}$$

(a) AVM representation

```
[ rdf:type telix:LabelOccurrence ;
  telix:value "alloy"@en ;
  telix:pos telix:NNS ;
  telix:agreement [ telix:number telix:Singular ;
                    telix:gender telix:Masculine ] ] .
```

(b) RDF-based TELIX representation

**Fig. 2.** Feature structure of the word "alloy"

original feature structure and the resultant RDF graph, where each pair of nodes of $\mathcal{G}$, connected by a grammatical feature, is translated to an RDF triple.

- $\pi_5(\psi) = CE_I$, where $CE_I$ is the class *extension function* of $I$, defined by: $CE_I(c) = \{a \in R_I : (a, c) \in E_I(I(\mathit{type}))\}$ in $\mathcal{G}$. Therefore, for each $n_i \in \psi(s)$, the application of $\pi_5$ returns $(I(\pi_1(n_i), I(\pi_3(s)) \in E_I(I(\mathit{type})) = (I(\eta_i), I(c)) \in E_I(I(\mathit{type}))$ in $G$. This mapping retains the species names interpretation of $\mathcal{G}$ (types of nodes) in the RDF graph: $\psi : \mathcal{S} \mapsto \mathcal{V}$.

TELIX introduces a core set of concepts and properties that represent $\mathcal{S}$ and $\mathcal{F}$ respectively. This vocabulary permits the application of mappings $\pi_1$, $\pi_2$ and $\pi_3$ to a given linguistic theory in order to express feature structures using the RDF data model. Part of these grammatical features refer to morpho-syntactic information, such as number (`telix:number`), person (`telix:person`), gender (`telix:gender`) and tense (`telix:tense`) for agreement, or part-of-speech information (`telix:pos`). Furthermore, TELIX provides collections of values over which these properties range. Some of these collections are based on existing linguistic classifications. This is the case of part-of-speech tags, adapted from the list used in the Penn Treebank Project (which can be extended to deal with other languages). The purpose is to facilitate the exchange and integration of linguistic information by reusing resources widely-adopted by the community.

As an example, Figure 2 illustrates the outcome produced by the application of $\pi$ mappings to a feature structure. The left-side of the Figure is the feature structure, represented here with the graphical Attribute-Value Matrix notation, which captures the grammatical analysis of the word "alloy". The right-side shows the resulting RDF graph (in N3 syntax).

TELIX also covers other aspects of text analysis, such as syntactic and discourse structures. RDF translations are provided for both constituent parse trees (as partially illustrated in Figure 4) and dependency graphs. Furthermore, discursive entities are defined. With regards to referring expressions, TELIX introduces properties: `correfers`, `antecedent` and `anaphora`, to express different correference nuances. Rhetorical relations are also supplied to represent the underlying structure at the discourse level of a given text.

It is worth mentioning that although TELIX provides machinery to represent feature structures as RDF graphs, it does not cover complex constraints or feature structures operations (such as unification). In other words, TELIX permits rich, interchangeable descriptions of linguistic entities, but does not represent grammars such as HPSG, LFG or other linguistic theory. To this end, more expressive, rule-based formalisms on top of TELIX are necessary.

### 3.5 Annotation Support

TELIX also introduces the concept `Annotation` to capture linguistic annotations as entities within the model. This makes it possible to describe the annotation itself, for instance by means of the Dublin Core vocabulary to express authorship (`dc:creator`), date (`dc:date`) and the source of the annotation (`dc:source`).

Another strong point of RDF is that it natively supports both the combination of complementary linguistic analysis and multi-authored annotations over the same text fragment, as segments are univocally identified by URIs. An annotation needs only to link a given fragment in order to describe it.

Firstly, RDF facilitates the amalgam of multiple annotation layers. For instance, Figure 4 contains an example of an annotation which merges the parse tree of the nominal phrase “the first alloy” enriched with morpho-syntactic and lexical information of the terminal node “alloy” (Figure 2). Both analyses, even coming from different NLP analyzers and platforms, are easily integrated in the same graph. In the case of the syntactic analysis, the parsing of constituents was performed by the Stanford parser (Figure 3) and subsequently translated to an RDF graph using the TELIX vocabulary. Although for the sake of readability, relations that capture the order of the terminal nodes of the parse tree in Figure 4 have been omitted, TELIX introduces the property `telix:precedes` with this purpose. Moreover, the new property navigation feature defined in SPARQL 1.1 [24] is particularly useful for querying this kind of tree-shaped graph structure. For instance, the property path `telix:childNode*` traverses the edges of the derived parse tree.

Secondly, TELIX takes advantage of *named graphs* [24,5] to handle multi-authored annotations over the same piece of text. If analyses performed by different NLP tools at a given linguistic layer (such as word sense disambiguation) are mixed into a single annotation or graph, tracking them is practically impossible. Therefore, a set $\mathcal{A}$ of annotations is defined as an RDF dataset, where: $\mathcal{A} = \{G_0, \langle u_1, G_1\rangle, \ldots, \langle u_n, G_n\rangle\}$. $G_0$ and each $G_i$ are graphs, and $u_1, \ldots, u_n$ are distinct IRIs. The pairs $\langle ui, Gi\rangle$ are named graphs, where $G_i$ an RDF graph

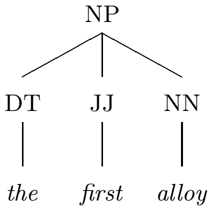

**Fig. 3.** Constituents parsing performed by the Stanford parser

```
[ rdf:type telix:NounPhrase ;
  telix:childNode [ rdf:type telix:DT ;
        telix:childNode [ rdf:type telix:LabelOccurrence ;
              telix:value "the"@en ] ] ;
  telix:childNode [ rdf:type telix:JJ ;
        telix:childNode [ rdf:type telix:LabelOccurrence ;
              telix:value "first"@en ] ] ;
  telix:childNode [ rdf:type telix:NN ;
        telix:childNode [ rdf:type telix:LabelOccurrence ;
              telix:value "alloy"@en ;
              telix:realizes ex:label−alloy ;
              telix:sense dbpedia:alloy ;
              telix:agreement [ telix:number telix:Singular ;
                                telix:gender telix:Masculine ] ] ] .
```

**Fig. 4.** RDF graph combining multiple annotation layers

containing the annotation data (potentially featuring multiple layers) and $u_i$ the name of the annotation. $G_0$ is the default graph of the RDF dataset and it provides the definition attached to each annotation $u_i$, where $u_i \in \texttt{Annotation}$.

## 4 Adding Annotations to Structured Text Documents Using RDFa

Among the RDF syntaxes mentioned in Section 2, RDFa [4] seems particularly well suited for adding linguistic annotations to source documents. RDFa is a W3C standard that can weave RDF graphs within a host mark-up language, typically (X)HTML. One of the advantages of RDFa is that it is possible to combine the original content (the source text) and the resulting segmented corpus, in a single file. As the number of files to be exchanged decreases, benefits appear in the form of easier management, simplified change tracking and consistency maintenance. Moreover, RDFa does not alter the appearance of a document, which preserves its original styling information.

The essence of RDFa is a set of coordinated attributes that are attached to elements of the XML infoset [25]. In terms that are more familiar to HTML developers, this means that attributes are appended to opening *tags* such as `<p>`. It is not possible to annotate units of text that are not delimited by tags, but this is not an issue because ad hoc mark-up can be added to the document, usually by means of invisible `<div>` and `<span>` tags (for units larger and smaller than paragraphs, respectively).

### 4.1 Document Preparation

Prior to annotation, some preparation is often required. Firstly, the original document must be converted into XHTML. Plain text documents can be trivially upgraded to XHTML documents by simply enclosing the text in an XHTML

```
<body about="#document1" typeof="dctype:Text" datatype=""
      property="telix:value" rel="dct:hasPart" id="document1">
<p about="#para1" typeof="telix:Paragraph" datatype=""
   property="telix:value" rel="dct:hasPart" id="para1">
  <span about="#sent11" typeof="telix:Sentence" datatype=""
        property="telix:value" id="sent11">Steel is an alloy that consists
        mostly of iron and has a carbon content between 0.2% and 2.1% by weight,
        depending on the grade.</span>
  <span about="#sent12" typeof="telix:Sentence" datatype=""
        property="telix:value" id="sent12">Carbon is the most common alloying
        material for iron, but various other alloying elements are used, such as
        manganese, chromium, vanadium, and tungsten.</span>
</p>
</body>
```

**Fig. 5.** Document body annotated with RDFa attributes. Namespaces are omitted

template. Then, additional `<p>`, `<div>` and `<span>` tags must be introduced as required until the document mark-up structure matches the text segmentation. At this point, the document looks like the example in Figure 5. Note that sentences are delimited by `<span>` tags nested inside the `<p>` tags. Tag nesting captures multiple levels of structure (sections, paragraphs, sentences, parts of sentences, words. . . ). Due to the tree-based model of XML documents, it is not possible to build structures that overlap without one being contained within the other. However, the RDFa document can be combined with other RDF documents sharing the same URIs

Note that tags make sentence segmentation explicit in the document. Therefore, it is no longer necessary that the producer and the consumer of the document implement the same segmentation algorithm in order to unequivocally agree on the scope of each sentence. As the boundaries of each sentence are explicitly marked in the document, and the number of sentences is unambiguous, location-based references have a solid ground to build on. TELIX supports location-based references as a fallback option to be backward compatible with legacy tools.

### 4.2 In-place Document Annotation

The simplest RDFa annotation involves attributes `about` and `typeof`, which introduce identifiers (URIs) for structures of the document and specify their type (as explained in Section 3.1).

Even if the relationship between the text structure and the text content is implicit due to the mark-up nesting, RDFa parsers do not automatically convert it into RDF triples. To this effect, the pair of attributes `property="telix:value"` and `datatype=""` must be added, as will be shown in the final example.

The hierarchy of structures implicit by the mark-up nesting (e.g., the sentences are contained in the paragraphs) must be explicitly named in order to be captured in the RDF graph. A pair of inverse properties (`dct:hasPart` and `dct:isPartOf`) can be used for this purpose, in combination with the `rel` and `rev` attributes of RDFa. In fact, just one of them is enough to express the hierarchy, and the choice depends only on syntactic convenience. Figure 5 contains the final result of annotating the document body.

### 4.3 Separate Annotations

Although it is a convenient choice in many scenarios, there are a number of reasons that may render RDFa unsuitable to embed complex RDF graphs into source documents. These reasons include RDFa limitations regarding annotation of non-contiguous text fragments, its inability to capture coexistent but divergent text segmentations, and its verbosity (when compared to other RDF syntaxes). Moreover, some scenarios simply require separating life-cycles for the source document and its annotations.

For these scenarios, we suggested decoupling the linguistic annotations and the source document. A basic, uncontroversial and shared segmentation of the source text may be added using RDFa, while the other annotation layers are kept in separate files (possibly using other RDF syntaxes). Even in this case, annotations can still univocally point to the corresponding text fragments by means of their URIs identifiers.

## 5 Previous Work

TELIX builds on the experience of a chain of proposed languages, ontologies and frameworks that have previously addressed the effective exchange of textual resources in order to facilitate automated processing.

Most notably, TEI (Text Encoding Initiative) [1] is an XML-based encoding scheme that is specifically designed for facilitating the interchange of data among research groups using different programs or application software. TEI provides an exhaustive analysis about how to encode the structure of textual sources, feature structures or graphics and tables in XML format. Although TEI defines a very detailed specification of linguistic annotations, its XML syntax does not facilitate the integration of heterogeneous layers of annotations. Since most of the linguistic workflows (UIMA, Gate, etc.) rely on multiple modules covering different layers of annotations, an RDF-based format to represent the annotations, such as TELIX, is more suitable to be used by these systems. More concretely, TELIX offers some advantages over TEI, derived from the more flexible nature of RDF graphs with respect to XML trees, permitting the description of several layers of annotations linked to the source document.

GrAF [17] is another graph-based format for linguistic annotation encoding, although it does not rely on RDF but on an ad-hoc XML syntax. As it is based on RDF, our proposal elegantly solves the graph merging problem. Moreover, GrAF annotations can be translated into RDF [6], thus existing GrAF annotations can easily be translated into TELIX. Furthermore, another advantage of using the RDF framework is the availability of a standard query language, namely SPARQL. Both GrAF and TELIX are motivated by LAF (Linguistic Annotation Framework [16]), which identifies the requirements and defines the main decision principles for a common annotation format. TELIX supports integrated multilayered annotations and enables multiple annotations to the same text fragment. However, although TELIX includes support for stand-off annotations (based on offsets), it discourages them. Instead TELIX proposes a combination of URI identifiers and RDFa annotations in mark-up documents.

LMF (Lexical Markup Framework, ISO 24613:2008) is a model of lexical resources. It is suitable for the levels of annotations that are attached to a lexical entry, but not for syntactic annotations in the case of non-lexicalized grammars. Being an XML format, it lacks the advantages of RDF discussed in this paper.

Other ontologies have been proposed to represent linguistic information. The most noteworthy one is GOLD [12], which is specified in OWL and provides a vocabulary to represent natural languages. GOLD is designed as a refined extension of the SUMO ontology. TELIX and GOLD have some resemblances, although they diverge in their goals: TELIX is more annotation-oriented, while GOLD aims to provide the means to describe natural languages formally.

The OLiA ontologies [7] provide OWL vocabularies to describe diverse linguistic phenomena, from terminology and grammatical information to discourse structures. TELIX and OLiA take different approaches to similar goals, in particular regarding constituent-based syntactic trees. TELIX also contributes a formal foundation to translate feature structures in RDF.

The Lemon model [19] proposes its own vocabulary to model words and senses in RDF. However, TELIX prefers to take advantage of (and extend) the SKOS framework for modeling lexical entities, as discussed in Section 3.2. Regarding WordNet [23], there are some overlaps with TELIX regarding lexical entities treatment. Nevertheless, they are potentially complementary, e.g., WordNets synsets can be used as values of TELIX's `sense` property.

## 6 Conclusions

This paper proposes the use of the RDF framework in combination with an ontology (TELIX) for linguistic annotation. Despite the considerable body of previous and current proposals with similar goals, the authors believe that TELIX sits in a previously unoccupied space because of its comprehensiveness and its orientation to the information exchange on the web of data. A comprehensive evaluation of TELIX with respect to the related works is planned for the coming months. Among the works that are concurrently being developed and that are closely tied to TELIX, POWLA [8] is a recent proposal of an OWL/DL formalization to represent linguistic corpora based on the abstract model PAULA [10], a complete and complex XML model of linguistic annotations. Another ongoing initiative is NIF [15], also based on OLiA. NIF and TELIX coincide in the use of URIs to univocally identify text fragments, enabling the handling of multiply-anchored annotations over them. The main difference between NIF and TELIX is that the latter offers a corpus level that is not provided by the former.

TELIX is driven by the goal to provide usable, expressive linguistic annotations. Being able to query and to reason on these annotations is a key requirement for a widely-adopted annotation format. Moreover, the success of SKOS as a web-oriented standard for concept schemes inspires confidence in the possibility of a web-oriented standard for linguistic annotations.

For TELIX to be successful, it must be embraced by the community and implemented by NLP tools. The authors are working in both fronts. Firstly, plans are in place to submit the specification of TELIX to W3C, either as a Member Submission or as a contribution to the Ontology-Lexica Community Group. Secondly, prototype implementations of TELIX in some NLP tools, such as an UIMA workflow for rules extraction [9], are being produced in the context of the ONTORULE project [11].

**Acknowledgment.** The work described in this paper has been partially supported by the European Commission under ONTORULE Project (FP7-ICT-2008-3, project reference 231875).

## References

1. TEI P5: Guidelines for Electronic Text Encoding and Interchange. Technical report, TEI Consortium (2012), `http://www.tei-c.org/Guidelines/P5/`
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
3. Bechhofer, S., Miles, A.: SKOS Simple Knowledge Organization System Reference. W3C recommendation, W3C (August 2009), `http://www.w3.org/TR/2009/REC-skos-reference-20090818/`
4. Birbeck, M., Adida, B.: RDFa primer. W3C note, W3C (October 2008), `http://www.w3.org/TR/2008/NOTE-xhtml-rdfa-primer-20081014/`
5. Carroll, J.J., Bizer, C., Hayes, P., Stickler, P.: Named graphs, Provenance and Trust. In: WWW 2005: Proceedings of the 14th International Conference on World Wide Web, pp. 613–622. ACM, New York (2005)
6. Cassidy, S.: An RDF realisation of LAF in the DADA annotation server. In: Proceedings of ISA-5, Hong Kong (2010)
7. Chiarcos, C.: An Ontology of Linguistic Annotations. LDV Forum 23(1), 1–16 (2008)
8. Chiarcos, C.: POWLA: Modeling Linguistic Corpora in OWL/DL. In: Simperl, E., et al. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 225–239. Springer, Heidelberg (2012)
9. Derdek, S., El Ghali, A.: Une chaîne UIMA pour l'analyse de documents de réglementation. In: Proceeding of SOS 2011, Brest, France (2011)
10. Dipper, S.: XML-based stand-off representation and exploitation of multi-level linguistic annotation. In: Proceedings of Berliner XML Tage 2005 (BXML 2005), pp. 39–50 (2005)
11. Lévy, F. (ed.): D1.4 Interactive ontology and policy acquisition tools. Technical report, Ontorule project (2011), `http://ontorule-project.eu/`

12. Farrar, S., Langendoen, T.: A Linguistic Ontology for the Semantic Web. GLOT International 7, 95–100 (2003)
13. Hayes, P.: RDF semantics. W3C recommendation. W3C (February 2004), http://www.w3.org/TR/2004/REC-rdf-mt-20040210/
14. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space, 1st edn. Morgan & Claypool (2011)
15. Hellmann, S.: NLP Interchange Format (NIF) 1.0 specification, http://nlp2rdf.org/nif-1-0
16. Ide, N., Romary, L.: International Standard for a Linguistic Annotation Framework. Journal of Natural Language Engineering 10 (2004)
17. Ide, N., Suderman, K.: GrAF: a graph-based format for linguistic annotations. In: Proceedings of the Linguistic Annotation Workshop, LAW 2007, Stroudsburg, PA, USA, pp. 1–8. Association for Computational Linguistics (2007)
18. King, P.J.: An Expanded Logical Formalism for Head-Driven Phrase Structure Grammar. Arbeitspapiere des SFB 340 (1994)
19. McCrae, J., Spohr, D., Cimiano, P.: Linking Lexical Resources and Ontologies on the Semantic Web with Lemon. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 245–259. Springer, Heidelberg (2011)
20. Miles, A., Bechhofer, S.: SKOS Simple Knowledge Organization System eXtension for Labels (SKOS-XL). W3C recommendation, W3C (August 2009), http://www.w3.org/TR/2009/REC-skos-reference-20090818/skos-xl.html
21. Pollard, C.: Lectures on the Foundations of HPSG. Technical report, Unpublished manuscript: Ohio State University (1997), http://www-csli.stanford.edu/~sag/L221a/cp-lec-notes.pdf
22. Sag, I.A., Baldwin, T., Bond, F., Copestake, A., Flickinger, D.: Multiword Expressions: A Pain in the Neck for NLP. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 1–15. Springer, Heidelberg (2002)
23. Schreiber, G., van Assem, M., Gangemi, A.: RDF/OWL representation of WordNet. W3C working draft, W3C (June 2006), http://www.w3.org/TR/2006/WD-wordnet-rdf-20060619/
24. Seaborne, A., Harris, S.: SPARQL 1.1 query. W3C working draft, W3C (October 2009), http://www.w3.org/TR/2009/WD-sparql11-query-20091022/
25. Tobin, R., Cowan, J.: XML information set, W3C recommendation, W3C, 2nd edn. (February 2004), http://www.w3.org/TR/2004/REC-xml-infoset-20040204