KNOWLEDGE-BASED MULTIMODAL DATA REPRESENTATION AND QUERYING

Julien Seinturier, Elisabeth Murisasco and Emmanuel Bruno
LSIS umr CNRS 6168, Université du Sud Toulon Var, Avenue de l'université, 83957 La Garde, France

Keywords: Ontology, Knowledge representation, Querying, Social sciences, Applications.

Abstract: This paper focuses on the representation and querying of knowledge-based multimodal data. Our work stands in the multidisciplinary project OTIM (Tools for Multimodal Annotation) dedicated to the development of tools for multimodal annotation of french conversational data. OTIM aims at encoding and manipulating annotations from all the linguistic domains in an unique framework. Defining a data model suited to the concurrent representation of these annotations involve to be able to analyze and to query them in order to help to determinate correlations between the linguistic domains. Linguists commonly use Typed Feature Structures (TFS) to provide an uniform view of multimodal annotations but such a representation cannot be used within an applicative framework. Moreover TFS expressibility is limited to hierarchical and constituency relations and does not suit to any linguistic domain that needs for example to represent temporal relations. To overcome these limits, we propose an ontological approach based on Description logics (DL) for the description of linguistic knowledge and we provide an applicative framework based on OWL DL (Ontology Web Language) and the query language SPARQL.

1 INTRODUCTION

The OTIM (Tools for Multimodal Annotation[1]) project aims at developing conventions and tools for multimodal annotation of a large conversational french speech corpus. The idea is to encode and to manipulate all the linguistic domains (from prosody to gesture (et alii, 2010)) in an unique framework. For that, it has to be possible to bring together and align all the different pieces of information (called annotations) associated to a corpus. This multidisciplinary project is funded by the French ANR agency, it groups together Social Sciences and Computer Science researchers.

The objectives of the OTIM project can be summarized in two main steps:

1. the multimodal annotation of a conversational speech between two persons.
2. the representation and manipulation of multimodal annotation.

Step 1. Annotation is done according to different levels of linguistic analysis (morpho-syntax, prosody, gesture and posture, discourse, disfluencies...). The qualifier *multimodal* is due to the nature of the studied corpus which is composed of text, sound, video. The creation of the corpus is under the responsibility of linguists; Each expert has to annotate the same data flow according to its knowledge domain and the nature of the signal on which he annotates (signal transcription or signal). Experts generally use dedicated tools (e.g. Praat[2], Anvil[3], Elan[4], ...).

Step 2. To analyze and find correlations between annotated linguistic domains, it is necessary to consider them grouped together: it requires the definition of a formal model for describing and manipulating them in a concurrent way. The main difficulty in defining a data model comes from the heterogeneity of the domains and media and from the distribution of the resources. Concurrent manipulation consists in querying annotations belonging to two or more modalities or in querying the relationships between modalities. For instance, we want to be able to express queries over gestures and intonation contours (what kind of intonational contour does the speaker use when he looks at the listener ?) and to query temporal rela-

[1] http://aune.lpl.univ-aix.fr/ otim/

[2] http://www.fon.hum.uva.nl/praat/

[3] http://www.anvil-software.de/

[4] http://www.lat-mpi.eu/tools/elan/

Seinturier J., Murisasco E. and Bruno E..
KNOWLEDGE-BASED MULTIMODAL DATA REPRESENTATION AND QUERYING.
DOI: 10.5220/0003627901520158
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2011), pages 152-158
ISBN: 978-989-8425-80-5

tionships (in terms of anticipation, synchronization or delay) between both gesture strokes and lexical affiliates. The results of queries could be useful to help in constructing new annotations or to extend existing ones.

In this paper, we focus on this last step considering semantic web technologies for the development of a linguistic Knowledge-based Information System. Each annotator using his own tool, our objective is to propose a common underlying data model and an architecture dedicated to the multimodal exploitation of the data. Our theoretical standpoint being to share data and resources, we will use open standards from the XML (Bray et al., 1998) universe.

1.1 Context and Motivation

Within the project OTIM, linguists propose an encoding for annotating spoken language data, with the acoustic signal, the video signal as well as its orthographic transcription. They have chosen to use *Typed Feature Structures* (Carpenter, 1992) (TFS) to represent in an unified view the knowledge and the information they need for annotation. TFS representation is usual for linguists: it aims at normalizing, sharing and exchanging annotation schemas between experts. Linguistic knowledge is captured by means of three types of information:

- *properties*: the set of characteristics of an object. An object is a type of information to be annotated in the corpus
- *relations*: the set of relations that an object has with other objects
- *constituents*: complex objects are composed of other objects called constituents

TFS proposes a formal presentation of each object in terms of feature structures and type hierarchies : properties are encoded by features, constituency is implemented with complex features, and relations make use feature structure indexing; each linguistic domain is represented as a hierarchical model.

For example, Figure 1 graphically describes TFS representation of the prosodic domain. Notice that every feature of the domain related to signal is a sub-feature of the *OtimObject* that is constituted of an *INDEX* feature in order to be referred and a *LOCALISATION* feature that represents an interval, which boundaries are defined by the features *START* and *END*, with temporal value (usually milliseconds). Prosodic phrases are of two different types: *ap* (accentual phrases) and *ip* (intonational phrases). Accentual phrases is constituted of two appropriate features: the *LABEL*, which value is simply the name of the corresponding type, and the list of *CONSTITUENTS*, in this case a list of syllables. The features of type *ip* contain the list of its *CONSTITUENTS* (a set of *ap*) as well as the description of its *CONTOUR* which is a prosodic event, situated at the end of the *ip* and is usually associated to an *ap*. The prosodic phrases are formally defined as set of syllables. A syllable (*syl*) is constituted of features: *STRUCT* that describes the syllable structure (for example *CVC*, *CCVC*, etc.), the position of the syllable in the word (*POSITION*), its possibility to be accented or prominent (resp. *ACCENTUABLE*, *PROMINENCE*). Features of type *const_syl*, contains two different features: a set of phonemes, denoted *PHON*, and the type of the constituent (*onset*, *nucleus* and *coda*), denoted *CONST_TYPE*. Note that each syllable constituent can contain a set of phonemes.

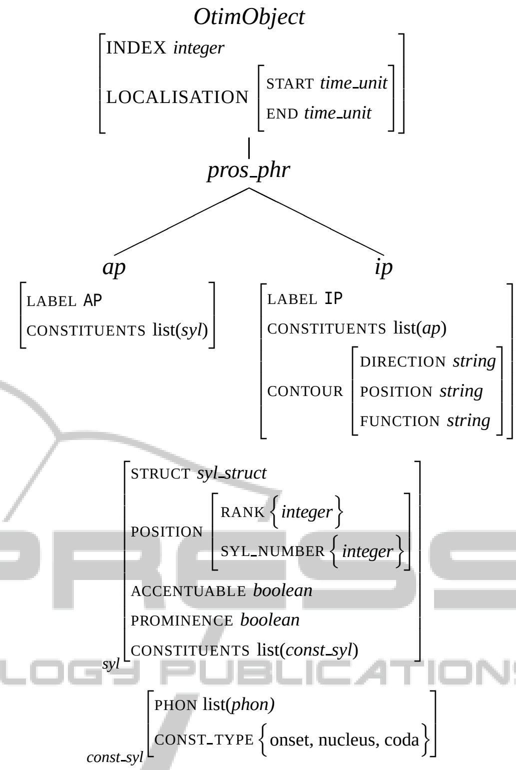

Figure 1: TFS representation of the prosodic domain.

TFS is well suited to take into account the heterogeneous characteristics of annotated data. Nevertheless, due to its theoretical nature, such a representation cannot be used within an applicative framework and has to be implemented into other formalisms. These remarks on TFS limits are not recents. In 1994, (Maitre et al., 1994) have proposed the use of the *O*2 object oriented data model (Lécluse et al., 1992) to implement and query dictionaries represented with

TFS. Moreover, TFS expressivity is limited, for example for temporal relations. Object anchoring is absolute and it would be useful to make it relative. We shall see another limit due to the underlying model of TFS which is a Directed Acyclic Graph (DAG). When linguists need to annotate coreferences or disfluences (lenghtenings, silent and filled pauses,....) which are organized around objects, it would be useful to have an object anchoring which is conflicting with the underlying acyclic graph.

1.2 Objectives

Our intention is to propose a knowledge representation formalism which be an alternative to TFS : an ontological approach based on Description Logics (Baader et al., 2003) (DL) and on semantic web technologies for the development of a linguistic Knowledge-based Information System.

Ontology will enable experts to share and annotate information in their respective knowledge domain. Ontological representation will both represent semantic descriptions of linguistic domains and data.

In this context, our contribution is twofold:

- the definition of a linguistic ontology from the TFS provided by linguists
- the definition of an applicative framework by means of semantic web proposals such as OWL-DL (Ontology Web Language[5]) for the representation of this ontology and SPARQL[6] the querying language of semantic web for its manipulation.

Our knowledge-based Information System will rely on the linguistic ontology and its individuals. Some linguistic projects have a similar objective than OTIM, for instance NITE[7], AGTK[8], PAULA[9], XStandoff (Sthrenberg and Jettka, 2009). Our approach differs from them because we focus on an ontological contribution. Moreover, we only use open standards from the XML universe (OWL, SPARQL). Indeed, we want that standards tools remain available and that evolutivity be guaranteed. Moreover, linguistic annotation tools rely on native and not often open formats which are not directly interoperable. Encoding annotation using a high level formalism independent from coding languages and tools is an element of answer to the question of interoperability. Such a question has been discussed in (Schmidt et al., 2009) but it focuses on tools interoperability only and does not aim to provide independence from coding and semantic.

The paper is organized as follows. Section 2 studies TFS and DL in order to prove their theoretical correspondence (TFS and DL both enable to represent Directed acyclic graph (DAG)); this study relies on a third formalism of knowledge representation : Conceptual Graphs (CG). Section 3 deals with the RDF/OWL representation and the manipulation of the linguistic ontology. Section 4 describes the current implementation and Section 5 concludes.

2 FROM TYPED FEATURE STRUCTURE TO ONTOLOGY

In this section we propose a formal and automatic transformation from a linguistic specific knowledge representation based on TFS to a standard representation within Ontology Web Language framework (OWL-DL). This transformation use two transitory formalisms as Description Logics which is OWL-DL underlying formalism and the Conceptual Graphs as they are suitable to represent TFS. Moreover, the link between Conceptual Graphs and Description logic has been already proved (Coupey and Faron, 1998).

2.1 Linguistic Representation: Typed Feature Structures (TFS)

The Typed Feature Structures (TFS) (Carpenter, 1992) is a knowledge representation formalism based on hierarchical graph used within linguistic domain. It enables to make a graphical and suitable representation from a textual description as described in section 1.1 and illustrated in figure 1.

Beside the graphical representation, a formal definition of TFS has been given in (Copestake, 2003): A TFS is defined on a finite set of features *Feat* and a type hierarchy $(Type, \subseteq)$. It is a tuple (Q, r, δ, θ), where:

- Q is a finite set of nodes
- $r \in Q$ is the root node
- $\theta : Q \rightarrow Type$ is a partial typing function
- $\delta : Q \times Feat \rightarrow Q$ is a partial feature value function

subject to the following conditions:

1. r is not a δ-descendant.
2. all members of Q except r are δ-descendants of r. Some systems add an extra condition:

[5] http://www.w3.org/TR/2004/REC-owl-features-20040210/

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

[7] http://groups.inf.ed.ac.uk/nxt/

[8] http://weblex.ens-lsh.fr/projects/xitools/logiciels/AGTK/agtk.htm

[9] http://www.sfb632.uni-potsdam.de/∼d1/paula/doc/

3. there is no node n or path π such that $\delta(n, \pi) = n$.

The type hierarchy and the condition 3 enable to consider TFS as Direct Acyclic Graphs (DAG).

2.2 From TFS to Conceptual Graphs

Conceptual Graphs, denoted CG (Sowa, 1992), are a knowledge representation formalism close to TFS on some characteristics (hierarchy, relations). The Simple Conceptual Graphs (Chein, 1997) are a subfamily of CG and have some properties that enable to represent the same knowledge described by TFS. We focus on a typed extension of the SGs given by (Leclère, 1997) that extends the SGs with typing capability and defined as follows:

Let $(T_c, \leq_c)$ and $(T_r, \leq_r)$ two finite partially preordered sets denoted *concept types* and *relation types* respectively. Let $M = \{\{*\} \cup \{m_1, \ldots, m_n\}\}$ a finite set of tags where $*$ is the universal tag and m_i, $1 \leq i \leq n$ is an individual tag. The set $S = \{T_c \cup T_r \cup M\}$ is called *support* of the graph. A Simple Conceptual Graph, denoted SG is a tuple $SG = (C, R, \gamma, \varepsilon)$ such that:

- C is a finite set of concepts
- R is a finite set of relations
- $\gamma : R \rightarrow C_a \subset C$ associate to each relation $r \in R$ its arguments $C_a = \{c_1, \ldots c_k \mid \forall 1 \leq i \leq k, c_i \in C\}$.
- $\varepsilon : C \cup R \rightarrow (T_c \times M \cup T_r$ such that $\forall c \in C, \varepsilon(c) = (t, m), t \in T_c, m \in M$ and $\forall r \in R, \varepsilon(r) = t \in T_r$. Value t is called type of the concept (resp. relation) and value m is called tag of the concept. If m is equals to $*$, the concept is generic else the concept is individual.

We can represent a TFS with a Simple Conceptual Graph by following the steps below:

1. The type hierarchy $(Type, \subseteq)$ of the TFS is represented by a concept types hierarchy $(T_c, \leq_c)$ of the SG where $T_c = Type$ and $\leq_c$ is such that $\forall t_i, t_j \in T_c, t_i \leq_c t_j \leftrightarrow t_i \subseteq t_j$.
2. The set of features *Feat* of the TFS is represented by *relation types* $(T_r, \leq_r)$ where $T_r = Feat$ and $\leq_r$ is such that $\forall t_i, t_j \in T_r, t_i \leq_r t_j$ does not exist. The pre-order is not defined as there is no hierarchy on the features. As within TFS formalism the relations have not type, the set T_r of relation types and the set R of relations can be considered as equals. Otherwise, the set R can be defined as bijective set from T_r.
3. The set of tags M is defined by $M = \{*\}$ as TFS only represents terminology (generic knowledge).
4. The node set Q of the TFS can be associated to the set of concepts C with $Q = C$ as only concepts are nodes within the TFS formalism.
5. The partial typing function δ that associates to each node of Q a type of *Type*, is represented by the function ε.
6. The θ function of the TFS represents the relations between nodes by accessing a feature and it is assimilated to the SG γ.

This method enables to automatically construct a Simple Conceptual Graph from a TFS. Figure 2 illustrates the CG representation of prosodic phrases obtained from the TFS representation given in figure 1. The type hierarchy is explicit (*Is a* relation). Concepts *Contour* and *Pos* are artificially added to make explicit features that are implicitly declared because of the TFS representation (anonymous features). Relations *Const* have the same name for the *AP* and *IP* concepts but are different. We chose to keep the names of the original TFS features for the sake of simplicity. We can now focus on the transformation from SGs to Description Logic as it is the base on the ontological representation we need.

2.3 From Conceptual Graphs to Description Logics

Description Logics, denoted DL, are formalisms that enable to represent a domain related knowledge using "descriptions". These descriptions are concepts, roles and individuals (Baader et al., 2003). Concepts represent sets of individuals (also called classes) and roles represent relations between concepts. We focus in this work on the well known ALEOI Description Logic (Attributive Langage with Complement, with cardinality constraints) as it is the formal base of OWL-DL ontology language and its characteristics are suited to the representation of the SGs stemming from TFS. A transformation between SGs and ALEOI DL has been given in (Coupey and Faron, 1998)

3 ONTOLOGICAL (RDF/OWL) REPRESENTATION

One of the goals of the OTIM project is to provide tools for representing, querying and sharing linguistic knowledge. The ontological approach comes from the need of more expressiveness than the limited TFS representation. Formal justification has shown that the use of DL based ontology is efficient regarding the representation of the target linguistic domains terminology. Moreover, ontological representation enables

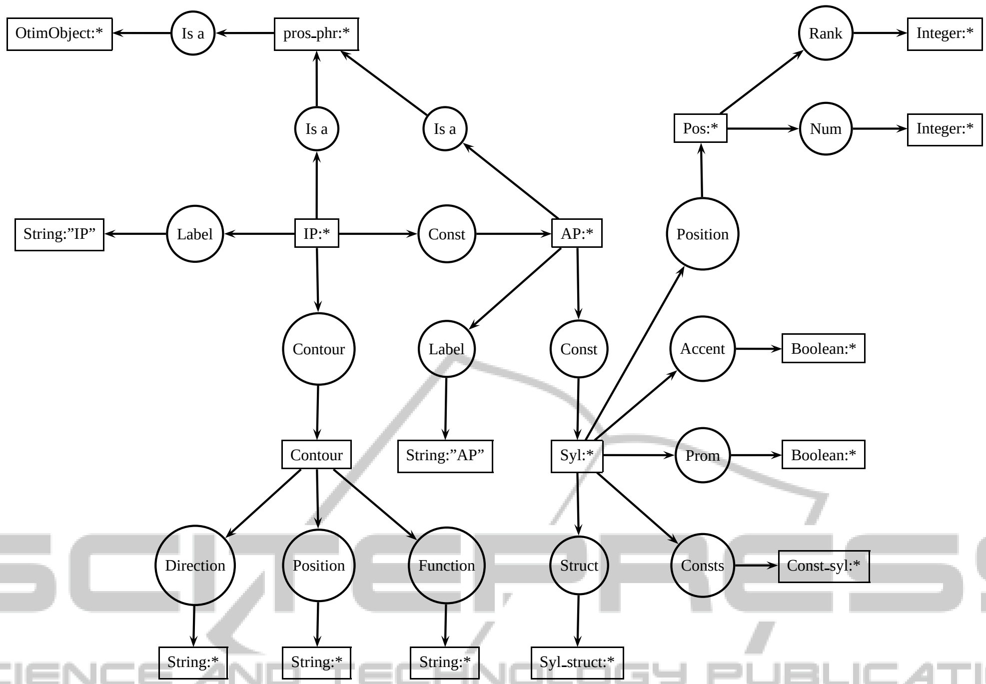

Figure 2: Representation in CG of the Prosodic Phrase.

to represent individuals and so, to represent the linguistic data.

Querying and sharing linguistic knowledge involve to implement the ontology. We choose OWL-DL as framework because:

- OWL-DL relies on DL and can represent the knowledge as TFS can do and even more
- the language is a standard and its use answers the need of linguistic knowledge sharing
- the SPARQL querying language enables to make complex queries on the ontology and its individuals
- there are various tools maintained for creating, managing and querying OWL ontologies

We now present the applicative work that leads from an abstract TFS representation to a complete OWL ontological representation and its querying.

3.1 Creating OWL Ontology

Creation of the OWL ontology follows two steps. First of all, the terminological knowledge from the TFS is implemented into OWL using the Protege[10] ontology editor. The Protege framework was initially designed for biologists and biochemists. This characteristic is quite interesting because this is not a computer scientist tool and so there is no need of a specific knowledge in computer science to use it.

The user interface relies on a graphical and textual description of the concepts, relations and individuals. Within the OTIM project, the ontology has been hand maded using Protege instead of processing TFS. This choice comes from the fact that we use the OWL-DL expressiveness to integrate descriptions that was impossible to represent (for example time relations or cyclic references). At this time, a complete ontology including prosody, phonetics and lexical domains terminology is available. Figure 3 shows the ontology of the prosodic domain. This ontology is linked with two other domains: the phonetics domain, which is a part of the OTIM knowledge representation framework, and the time domain given by a standard ontology of the W3C.

3.2 Managing Data and Querying with SPARQL

Management and querying of OWL data relies on the standard SPARQL (Prud'hommeaux and Seaborne, 2007) querying language. SPARQL enables to match graph pattern against the graph of RDF/OWL triple (*WHERE* clause) and identifies values to be returned (*SELECT* clause). The *FROM* clause enables to identify the data sources to query. The *FILTER* clause add constraints to the matching pattern and give more filtering capabilities. By convention, vari-

[10]http://protege.stanford.edu/

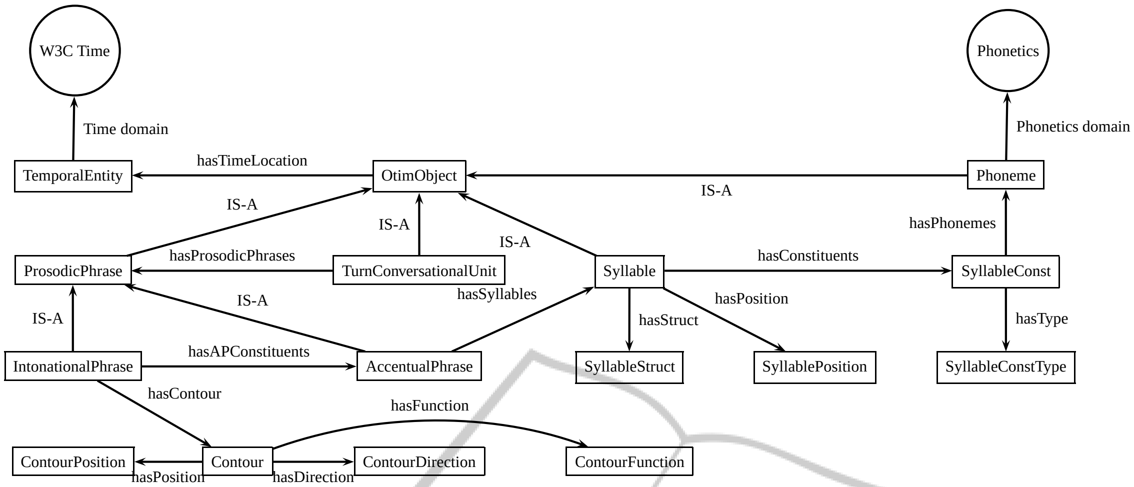

Figure 3: Ontological representation of the prosodic domain.

ables declared within a query are marked with a '?'. Notice that by default the graph pattern is a conjunction of triple. Each triple (subject, predicate, object) represents a piece of knowledge and means the subject has a predicate with object as value.

We express within the OTIM project the linguistic inter domain queries designed on TFS by SPARQL queries on the OWL representation. Linguists queries are expressed in natural language and a sample query is:

"We need the list of phonemes that are associated with the accentual phrases stated between the second 35 and the second 55 of the speech."

This query takes into account the prosodic domain (accentual phrase), the phonetic domain (phoneme) and the time. Such a query is represented in SPARQL by:

```
1.  SELECT      ?phoneme
2.  FROM        otim-prosody.owl, otim-phonetics.owl
3.  WHERE {     ?const rdf:type prosody:SyllableConst
4.            . ?const hasPhonemes ?phoneme
5.            . ?syl rdf:type prosody:Syllable
6.            . ?sc hasConstituents ?const
7.            . ?ap rdf:type prosody:AccentualPhrase
8.            . ?ap hasSyllables ?syl
9.            . ?t rdf:type time:TemporalEntity
10.           . ?ap hasTimeLocation ?t
11.           . ?tref time:contains ?t }
```

We assume in the sample that the time bounds given are represented as a *TemporalEntity* named *tref*. The *SELECT* clause specifies that the result to build is made of phonemes. The clause *FROM* contains the two data sources on which the query is processed. These sources represent the two target domains (prosody and phonetics). The *WHERE* clause describes the patterns for a phoneme to match. The *WHERE* clause is a logical conjunction (symbolized by .) of 9 triples. The first 6 triples (lines 3 to 8) describe the structure of the data and how to get a phoneme list from an accentual phrase. The last 3 triples (line 9 to 11) describe what are the selected accentual phrases regarding the time criterion. The relation *contains* applied to the variables *t* and *tref* represents the *contains* relation of the Allen Algebra (Allen, 1991) which is implemented within the W3C time ontology.

When this query is processed, all the instances on the phonemes composing the result are returned. Post processing can be done by linguists by making another query on the result or by exporting these instances to their specific tools.

4 IMPLEMENTATION AND RESULTS

The OTIM framework for linguistic multimodal annotations management has been implanted within a Java/OWL framework. The OWL standard used is OWL-DL as this is the specification that gives all the expressiveness we need and guarantees some calculability results that are critical for querying data. The Java framework is based on two packages:

- A specific OTIM package that enables to deal with linguistic tools and data.
- The Jena[11] package that provides robust OWL capabilities as SPARQL querying and logic reasoning.

[11] http://openjena.org/

The OTIM package has been developed for interfacing with widely used linguistic tools and data repository (the tools that are the most used within the project are PRAAT and ANVIL). The Jena package is developed by the Open Jena project and provides advanced OWL processing methods that can be embedded within a Java application. Jena also provides relational mapping of OWL data that makes optimal SPARQL queries by translating them into relational queries. These characteristics guarantee that the use of the developed Java/OWL is efficient.

5 CONCLUSIONS AND PERSPECTIVES

In this paper, our intention was to propose a framework for representing, querying and sharing linguistic knowledge. Our work stands in the multidisciplinary project OTIM dedicated to the creation (made by experts), the encoding and the manipulation of multimodal annotations associated to a audio video corpus. We have chosen an ontological approach based on Description logics (DL) for the description of linguistic knowledge and we have represented it by means of semantic web technologies. We have provided a set of tools relying on well defined or standard formalisms in order to enable to both query data and knowledge. This is the foundation of a multimodal Knowledge-based Information System. Our perspectives are the following:

- at this time, an ontology including prosody, phonetics and lexicals domains is available. Gesture and discourse have to be added. These are domains for which TFS expressiveness is limited and for which we have to work with linguists in order to capture their semantic description
- it is possible querying linguistic ontology by means of the query langage SPARQL. For instance we can query annotations belonging to two or more modalities or query the relationships between modalities. We need now to focus on the computational properties of our ontological approach for study reasoning systems

REFERENCES

Allen, J. (1991). Time and time again : The many ways to represent time. *International Journal of Intelligent Systems*, 6(4):341–355.

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Bray, T., Paoli, J., and Sperberg-McQueen, C.-M. (1998). Extensible Markup Language (XML) 1.0. Recommendation, W3C.

Carpenter, R. L. (1992). *The Logic of Typed Feature Structures*, volume 32 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, The Edinburgh Building, Shaftesbury Road, Cambridge CB2 8RU, United Kingdom.

Chein, M. (1997). The corali project: From conceptual graphs to conceptual graphs via labelled graphs. In Lukose, D., Delugach, H., Keeler, M., Searle, L., and Sowa, J., editors, *Conceptual Structures: Fulfilling Peirce's Dream*, volume 1257 of *Lecture Notes in Computer Science*, pages 65–79. Springer Berlin / Heidelberg. 10.1007/BFb0027909.

Copestake, A. (2003). *Collaborative Language Engineering: A Case Study in Efficient Grammar-based Processing*, chapter Definitions of Typed Feature Structures. CSLI Publications, Ventura Hall, Stanford University, Stanford, CA 94305-4115.

Coupey, P. and Faron, C. (1998). Towards correspondence between conceptual graphs and description logics. In *Proceedings of the 6th International Conference on Conceptual Structures: Theory, Tools and Applications*, ICCS '98, pages 165–178, London, UK. Springer-Verlag.

et alii, P. B. (2010). Multimodal annotation of conversational data. In *Proceedings of the fourth linguistic annotation workwhop (LAW)*, pages 186–191. Assocation for computational Lunguistics (ACL).

Leclère, M. (1997). Reasoning with type definitions. In *Proceedings of the Fifth International Conference on Conceptual Structures: Fulfilling Peirce's Dream*, ICCS '97, pages 401–415, London, UK. Springer-Verlag.

Lécluse, C., Richard, P., and Vélez, F. (1992). O2, an object-oriented data model. In *Building an Object-Oriented Database System, The Story of O2*, pages 77–97.

Maitre, J. L., Ide, N., and Veronis, J. (1994). Modélisation et interrogation de bases de données lexicales. *Ingńierie des systèmes dinformation (ISI)*, 2(1):57–82.

Prud'hommeaux, E. and Seaborne, A. (2007). Sparql query language for rdf (working draft). Technical report, W3C.

Schmidt, T., Duncan, S., Ehmer, O., Hoyt, J., Kipp, M., Loehr, D., Magnusson, M., Rose, T., and Sloetjes, H. (2009). Multimodal corpora. chapter An exchange format for multimodal annotations, pages 207–221. Springer-Verlag, Berlin, Heidelberg.

Sowa, J. F. (1992). *Conceptual graphs summary*, pages 3–51. Ellis Horwood, Upper Saddle River, NJ, USA.

Sthrenberg, M. and Jettka, D. (2009). A toolkit for multi-dimensional markup - the development of sgf to xstandoff. In *Proceedings of Balisage: The Markup Conference 2009*. Assocation for computational Lunguistics (ACL), Balisage Series on Markup Technologies, vol. 3.