Multilayer annotations in Parmenides

Fabio Rinaldi[*], James Dowdall[*], Michael Hess[*], Jeremy Ellman[††]
Gian Piero Zarri[§], Andreas Persidis[†], Luc Bernard[‡], Haralampos Karanikas[‡]

ABSTRACT

Most of the thrust in the semantic web movement comes from the observation that existing NLP tools are not sophisticated or efficient enough to process the full richness of Natural Language, and therefore Machine Understandable annotations need to be added to Web Resources in order to make them accessible by remote agents. However, when the target application is not required to handle a huge amount of documents, but more limited sets, it is conceivable and practical to take advantage of NLP tools to pre-process textual documents in order to generate annotations (to be verified by human editors).

We discuss an approach based on a combination of various Natural Language Processing techniques that addresses this issue. Documents are analized fully automatically and converted into a semantic annotation, which can then be stored together with the original documents. It is this annotation that constitutes the machine understandable resource that remote agents can query.

1. INTRODUCTION

There is a wealth of research efforts focusing on the foundations of the semantic web [18], and in particular on the problem of how to represent the semantic information carried by web resources (be they structured databases or unstructured natural language documents, or a combination of both). The XML-based Resources Description Framework [21] is the standardized Semantic Web language, however it is really meant for use by computers, not humans. The same applies to all the extensions that have been proposed, such as RDF Schema [4], which provides a basic type system for use in RDF models, or DAML+OIL [9] and OWL [3], which both provide languages with well-defined semantics for the specification of Ontologies.

Lamentably, there seems to be significantly less interest in the problem of how to help users in the transition from conventional web pages to richly annotated semantic web resources. Current efforts to tackle this problem seem to focus on the development of user-friendly editors for semantic annotations (e.g. OntoMat [8]). The general approach is that details of XML/RDF should be hidden behind GUI authoring tools, as users do not need (and do not want) to get in contact with XML/RDF. However, according to [19] the benefits of the semantic web should come for free to the vast majority of the users: semantic markup should be a by-product of normal computer use.

As a very large proportion of the existing web resources are represented by human-readable documentation, we believe that a promising approach is to use existing tools to extract information from documents and enrich them with automatically generated annotations. In this paper we propose an approach based on a combination of various Natural Language Processing (NLP) techniques geared towards the creation of semantic annotations, starting from the available textual documents. A possible alternative approach is to integrate NLP tools in the new generation of web editors, so that they'll become capable of semi-automatically generating the annotations that are needed to make the Semantic Web a reality. In other words, shifting the problem from information consumers to information producers (e.g.[20]).

In a recently started EU project (*Parmenides*)[1] focusing on the integration of Information Extraction and Data Mining techniques, we aim at exploring the ideas discussed above within a few well-defined application domains (e.g. biotechnology). One of our first goals has been to define a project internal annotation scheme, compatible with W3C standards (e.g. RDF), which is going to be the main focus of this paper.

As Ontologies play a key role in the project, we first outline (section 2) their interaction with different NLP tools to be integrated in the system. Section 3 then focuses on the Annotation Scheme, which is designed to achieve all the necessary and sufficient expressive power for the requirements of Parmenides and at the same time be easy to create and maintain. However, no inferential capabilities are associate with it. In order to allow reasoning and query answering, the approach currently being considered is to export the annotations into a more powerful Knowledge Representation tool, which is presented in section 4. A natural question that a reader familiar with Semantic Web developments might ask while reading this paper is why not use RDF as the language of the annotations. This topic will be discussed in section 5. Finally, in section 6, we consider relations with other current research activities.

---

[*] Institute of Computational Linguistics, University of Zurich, Switzerland; [†] Biovista, Athens, Greece; [‡] Centre for Research in Information Management, UMIST, Manchester, UK; [§] CNRS, Paris, France; [††] Wordmap Ltd., Bath, UK;

[1] See `http://www.ifi.unizh.ch/cl/Parmenides`

2. THE ROLE OF ONTOLOGIES

Following [17] we define an Ontology as being "a shared and common understanding of some domain that can be communicated between people and application systems". Ontologies are key aspects of conceptual information system and have long been claimed as being essential for knowledge reuse [17, 13]. Ontologies may be used in Knowledge Management for indexing organisational data through the definition of related metadata vocabularies ([26]), in Information Retrieval through the augmentation of user queries and text analysis [29], in Information Extraction [12], and in supporting Semantic Web applications [10].

The overall objective of the Parmenides project is to develop a systematic approach to the integration of information gathering, processing and analysis. This process is based on Ontologies. In this section we will look briefly at some of the applications to be included in the project, and describe their dependencies on ontologies. These include Information Extraction, Lexical Chaining, Text Mining, and data storage and retrieval using a Document Warehouse.

Information Extraction is a robust approach to natural language processing that combines pattern definitions as combinations of keywords with low-level part of speech tagging in order to identify predetermined document concepts [1]. An example would be the names of organizations, individuals, and actions, such as a company takeover, or terrorist threat. There are two means through which an Ontology can improve Information Extraction. Firstly, it can act as a repository of synonyms that extend the applicability of IE patterns and secondly, it may be used to restrict the application of IE patterns to the most appropriate situations. If we consider a business take-over event for example, we could define a rule as:

```
Take_over(X,Y)->Company(X) &
    Company(Y) & Buy(X,Y);
```

By storing synonyms for *company*, such as *business*, *firm*, *market leader*, or even *player*, our rule can find many more take-over events. However, the advantage of increased recall is lower precision, as many of these terms are ambiguous leading to inappropriate rule application. Precision may be increased however by disambiguating the candidate synonyms terms in our 'take-over' rule to determine if they are being used in an appropriate sense. [32] has for example discussed how Ontologies may be used in word sense disambiguation.

Lexical chains are contiguous sequences of words in a text that are related in meaning [24]. They appear as part of the cohesive nature of text, and may be identified using ontologies or structured thesauri such as WordNet, or Roget's thesaurus. Lexical chains may also be used to disambiguate text [11], and will consequently assist in local ontology based word sense disambiguation. In Parmenides, the lexical chains identified using Ontologies will also be used to find connections between events that are identified using the Information Extraction component. These lexical chains are seen as intrinsic components of temporal sequence data.

Text mining is the application of data mining techniques to text (e.g. [31]). Here, it is first necessary to derive semantic annotations for text units and relationships among them. Next, data mining algorithms such as clustering, association rules' discovery and sequence mining may be applied. These mining algorithms are used to discover unknown annotation labels, and relationships. This process may be applied to structured data, unstructured documents and semi-structured documents. An ontology provided by a domain expert is a key component here to enable the recognition of basic annotation labels.

Parmenides adopts an ontology-based approach in order to facilitate the integration of information from different sources into a **document warehouse**. This will store documents that appropriately annotated with meta data to both improve document retrieval, and as a step towards identify inter-document temporal events.

This brief sketch of the role of Ontologies in Parmenides has primarily been focused on the applications they service. However, there are many open research questions regarding the most desirable characteristics of Ontologies for real world applications. These include:

1. Is one large ontology preferable to several smaller partitions?
2. Will text need to be multiply tagged if several ontologies are used?
3. How rich need the ontologies' set of relations be for these applications?
4. What degree of granularity (or depth) does an ontology require to be useful?
5. What counts as an adequate ontology?

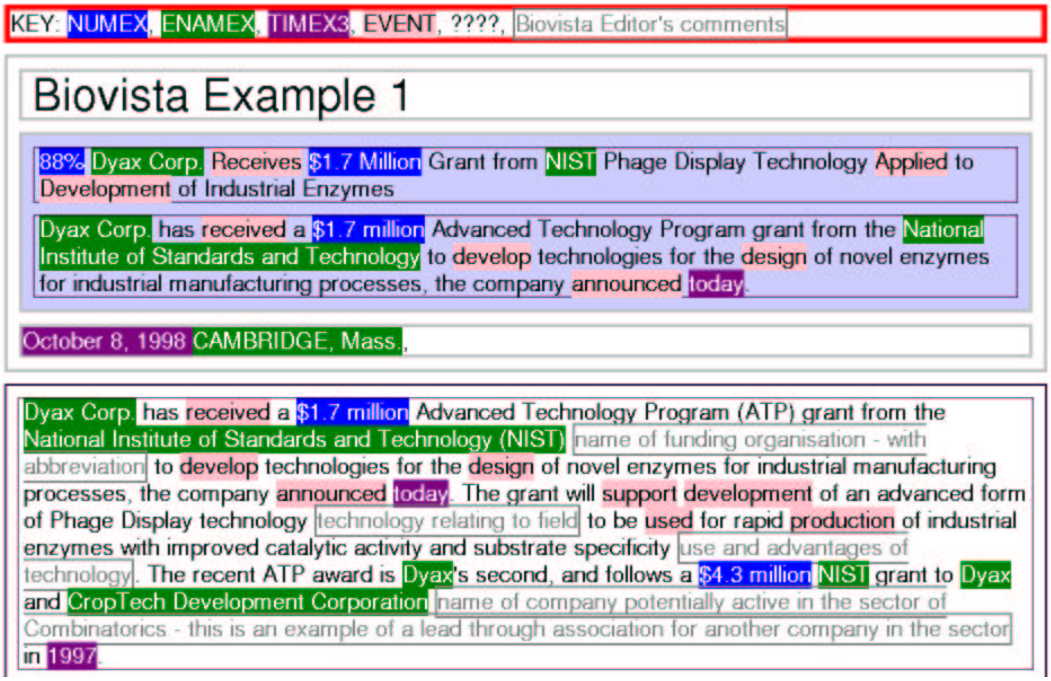
Whilst we may not be able to answer these questions within the scope of the current work there is certainly one clear requirement. That is, a mark up scheme that will be able to tag content sufficiently richly with ontological data so as to be able to pose these questions.

3. PARMENIDES ANNOTATIONS

It is by now widely accepted that some W3C standards (such as XML and RDF) provide a convenient and practical framework for the creation of field-specific markup languages (e.g. MathML, VoiceXML). However XML provides only a common "alphabet" for interchange among tools, the steps that need to be taken before there is any real sharing are still many (just as many human languages share the same alphabets, that does not mean that they can be mutually intelligible) [18]. A minimal approach is to create a common data model, which can directly increase interoperability among different tools. It is not enough to have publicly available APIs to ensure that different tools can be integrated. In fact, if their representation languages (their "data vocabulary") are too divergent, no integration will be possible (or at least it will require a considerable mapping effort).

In this section we will briefly introduce the XML-based annotation scheme developed for the Parmenides project. The annotation scheme is intended to work as the projects' *lingua franca*: all the modules are required to be able to accept as input and generate as output documents conformant to the (agreed) annotation scheme. The specification will be used to create data-level compatibility among all the tools involved in the project.

There are currently three methods of viewing the annotated documents which offer differing ways to visualize the annotations. These are all based on transformation of the same XML source document, using XSLT and CSS (see example in figure 1), and some Javascript for visualization of attributes.



KEY: NUMEX, ENAMEX, TIMEX3, EVENT, ????, Biovista Editor's comments

Biovista Example 1

88% Dyax Corp. Receives $1.7 Million Grant from NIST Phage Display Technology Applied to Development of Industrial Enzymes

Dyax Corp. has received a $1.7 million Advanced Technology Program grant from the National Institute of Standards and Technology to develop technologies for the design of novel enzymes for industrial manufacturing processes, the company announced today.

October 8, 1998 CAMBRIDGE, Mass.,

Dyax Corp. has received a $1.7 million Advanced Technology Program (ATP) grant from the National Institute of Standards and Technology (NIST) name of funding organisation - with abbreviation to develop technologies for the design of novel enzymes for industrial manufacturing processes, the company announced today. The grant will support development of an advanced form of Phage Display technology technology relating to field to be used for rapid production of industrial enzymes with improved catalytic activity and substrate specificity use and advantages of technology. The recent ATP award is Dyax's second, and follows a $4.3 million NIST grant to Dyax and CropTech Development Corporation name of company potentially active in the sector of Combinatorics - this is an example of a lead through association for another company in the sector in 1997.

Figure 1: Basic Annotation Viewing

3.1 Layers of Annotation

The set of Parmenides annotations is organized into three layers:

- **Structural Annotations**
  Used to define the physical structure of the document, it's organization into head and body, into sections, paragraphs and sentences.
- **Lexical Annotations**
  Associated to a short span of text (smaller than a sentence), and identify lexical units that have some relevance for the Parmenides project. They could be referred to also as *Textual Annotations*.
- **Semantic Annotations**
  Not associated with any specific piece of text and as such could be free-floating within the document, however for the sake of clarity, they will be grouped into a special unit at the end of the document. They refer to lexical annotations via co-referential Ids. They (partially) correspond to what in MUC7 was termed 'Template Elements' and 'Template Relations'.

Structural annotations apply to large text spans, lexical annotations to smaller text spans (sub-sentence). Semantic annotations are not directly linked to a specific text span, however, they are linked to text units by co-referential identifiers.

All annotations are required to have an unique ID and thus will be individually addressable, this allows semantic annotations to point to the lexical annotations to which they correspond. Semantic Annotations themselves are given a unique ID, and therefore can be elements of more complex annotations ("Scenario Template", in MUC parlance).

Lexical Annotations are used to mark any text unit (smaller than a sentence), which can be of interest in Parmenides. They include (but are not limited to): Named Entities in the classical MUC sense, New domain-specific Named Entities, Terms, Temporal Expressions, Events, Descriptive phrases (chunks).

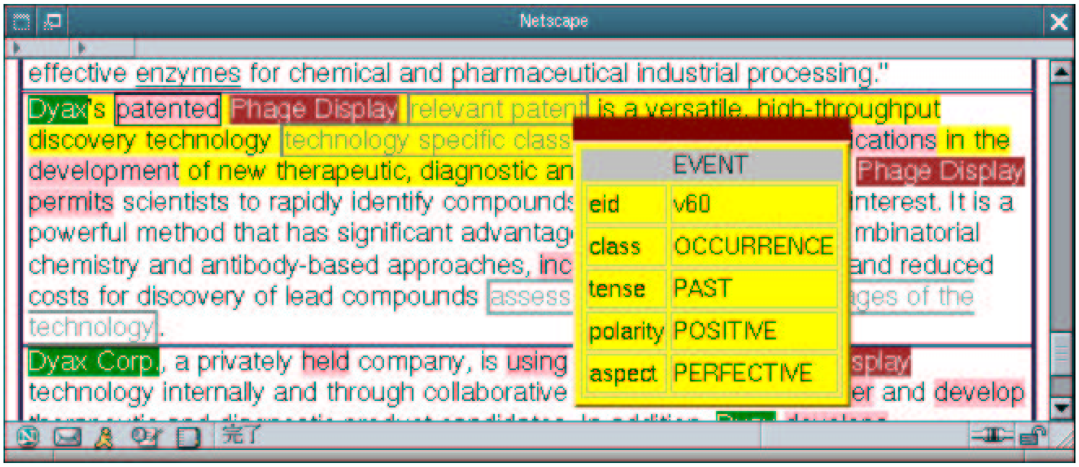
Essentially Lexical Annotations correspond to traditional markup, as exemplified for instance by the MUC Named Entities, with the caveat that old MUC-style elements are replaced by PNAMEX. However, while the name of the tag has changed from ENAMEX and NUMEX to PNAMEX, the function of the tags has not changed: they can still be identified using the type attribute. It is important to stress, that while in the examples the values of the type attributes are plain strings (for readability reasons), in the actual Parmenides system the value of this attribute is actually a pointer into a domain-specific Ontology, thus allowing more sophisticated markup. When visualizing the set of Lexical Tags in a given annotated document, clicking on specific tags displays the attribute values (see figure 2).

The relations that exist between lexical entities are expressed through the semantic annotations. So lexically identified people can be linked to their organisation and job title, if this information is contained in the document, as we will illustrate in the following section. In terms of temporal annotations, it is the explicit time references and events which are identified lexically, the temporal relations are then captured through the range of semantic tags.

3.2 Example

While the structural annotations and lexical annotations should be easy to grasp as they correspond to accepted notions of document structure and of conventional span-based annotations, an example might help to illustrate the role of semantic annotations.

(1) The recent ATP award is
`<PNAMEX id="e8" type="ORGANIZATION">`



effective enzymes for chemical and pharmaceutical industrial processing."

Dyax's patented Phage Display relevant patent is a versatile, high-throughput
discovery technology technology specific class ications in the
development of new therapeutic, diagnostic an Phage Display
permits scientists to rapidly identify compounds interest. It is a
powerful method that has significant advantag mbinatorial
chemistry and antibody-based approaches, inc and reduced
costs for discovery of lead compounds assess ages of the
technology.

Dyax Corp., a privately held company, is using splay
technology internally and through collaborative er and develop

EVENT	
eid	v60
class	OCCURRENCE
tense	PAST
polarity	POSITIVE
aspect	PERFECTIVE

Figure 2: Visualization of attributes

Dyax
</PNAMEX>
's second, and follows a
<PNAMEX id="n5" type="MONEY">
$4.3 million
</PNAMEX>
<PNAMEX id="e9" type="ORGANIZATION">
NIST
</PNAMEX>
grant to
<PNAMEX id="e10" type="ORGANIZATION">
Dyax
</PNAMEX>
and
<PNAMEX id="e11" type="ORGANIZATION">
CropTech Development Corporation
</PNAMEX>
in
<TIMEX3 tid="t4" type="DATE" value="1997">
1997
</TMEX3>

There are two occurrences of Dyax in this short text: the two Lexical Entities `e8` and `e10`, but clearly they correspond to the same Semantic Entity. To capture this equivalence, we could use the syntactic notion of co-reference (i.e. identify the two as co-referent). Another possible approach is to make a step towards the conceptual level, and create a semantic entity, of which both `e8` and `e10` are lexical expressions (which could be different, e.g. "Dyax", "Dyax Corp.", "The Dyax Corporation"). The second approach can be implemented using an empty XML element, created whenever a new entity is mentioned in text. For instance, in (2) we can use the tag <`PEntity`> (which stands for Parmenides Entity).

(2) `<PEntity peid="obj1" type="ORGANIZATION"`
`mnem="Dyax" refid="e1 e3 e6 e8 e10 e12"/>`

The new element is assigned (as usual) a unique identification number and a type. The attribute `mnem` contains just one of the possible ways to refer to the semantic entity (a mnemonic name, possibly chosen randomly). However, it also takes as the value of the `refid` attribute as many coreferent ids as are warranted by the document. In this way all lexical manifestations of a single entity are identified. All the lexical entities which refer to this semantic entity, are possible ways to 'name' it.

Notice that the value of the 'type' attribute has been represented here as a string for readability purposes, in the actual specification it will be a pointer to a concept in a domain-specific Ontology.

Other semantic entities from (1) are:

(3) `<PEntity peid="obj2" type="ORGANIZATION"`
`mnem="NIST" refid="e2 e4 e7 e9"/>`
`<PEntity peid="obj3" type="ORGANIZATION"`
`mnem="CropTech" refid="e11"/>`

The newly introduced semantic entities can then be used to tie together people, titles and organizations on the semantic level. Consider for example the text fragment (4), which contains only Lexical Annotations.

(4) ... said
<PNAMEX id="e17" type="PERSON">
Charles R. Wescott
</PNAMEX>
, Ph.D.,
<ROLE type='x' id="x5">
Senior Scientist
</ROLE>
at
<PNAMEX id="e60" type="ORGANIZATION">
Dyax Corp
</PNAMEX>

The Lexical Entity `e17` requires the introduction of a new semantic entity, which is given the arbitrary identifier 'obj5':

(5) `<PEntity peid="obj5" type="PERSON" mnem="Charles`
`R. Wescott" refid="e17"/>`

In turn, this entity is linked to the entity `obj1` from (1) by a relation of type 'workFor' (PRelation stands for Parmenides Relation):

(6) `<PRelation prid="rel2" source="obj5" target="obj1"`
`type="worksFor" role="Senior Scientist"`
`evidence="x5"/>`

3.3 Temporal Annotations

Given the foremost importance of temporal information in the Parmenides project, a detailed analysis of existing annotation schemes for temporal information has been carried out. TIDES [14], developed at the MITRE Corporation, can be considered as an extension of the MUC7 Named Entity Recognition [7]. It aims at annotating and normalizing explicit temporal references. STAG [28], developed at the University of Sheffield, has a wider focus than TIDES in the sense that it combines explicit time annotation, event annotation and the ability to annotate temporal relations between events and times.

TimeML [27] stands for "Time Markup Language" and represents the integration and consolidation of both TIDES and STAG. It was created at the TERQAS Workshop[2] and is designed to combine the advantages of the previous temporal annotations schemes. It contains a set of tags which are used to annotate events, time expressions and various types of event-event, event-time and time-time relations. TimeML is specifically targeted at the temporal attributes of events (time of occurrence, duration etc.).

As the most complete and recent, TimeML will be adopted for the temporal annotations in Parmenides. Broadly, its organization follows the Parmenides distinction between lexical/semantic annotations. Explicit temporal expressions and events receive an appropriate (text subsuming) lexical tag. The temporal relations existing between these entities are then captured through a range of semantic (non-text subsuming) tags.

4. NKRL

As already stated before, no reasoning and inferential capability is associated per se with Parmenides Annotations. In our project, these tasks are then entrusted to NKRL (Narrative Knowledge Representation Language), see [35]. This provides a standard, language independent description for the semantic content of narrative documents, in which information content consists of the description of events that relate the real or intended behaviour of some actors.[3] These actors try to attain a specific result, experience particular situations, manipulate some (concrete or abstract) materials, send or receive messages, buy, sell, deliver etc. All the NKRL knowledge representation tools are structured into four connected components:

The descriptive component concerns the tools used to produce the formal representations, called (NKRL) templates, of some general narrative classes of real world events, like moving a generic object, formulate a need, starting a company, obtained by abstraction/generalisation from sets of concrete, elementary narrative events. Templates are inserted into an inheritance hierarchy (a tree) that is called H_TEMP (hierarchy of templates).

The factual component provides the formal representation of the different, possible elementary events characterised, at least implicitly, by precise spatial and temporal coordinates under the form of instances of the templates of the descriptive component. These formal representations are called (NKRL) predicative occurrences. A predicative occurrence is then the NKRL representation of elementary events like Mr. Smith has fired Mr. Brown.[4]

The definitional component concerns the formal representation of the general notions like human_being, taxi_ (the general class referring to all the possible taxis, not a specific cab), etc. that must be represented for taking into account the events proper to a specific application domain. Their NKRL representations are called concepts , and correspond quite well to the concepts of the usual, formal ontologies of terms. NKRL concepts are inserted into a generalisation/ specialisation directed graph structure, called H_CLASS(es).

The enumerative component concerns the formal representation of the instances (concrete examples) of the general notions (concepts) pertaining to the definitional component; the NKRL formal representations of such instances take the name of individuals. Therefore, individuals are created by instantiating (some of) the properties of the concepts of the definitional component. Individuals are characterised by the fact of being countable (enumerable), of being associated with a spatio-temporal dimension, and of possessing unique conceptual labels (smith_, general_motors): two individuals associated with the same NKRL description but having different labels will be different individuals.

The frames of the definitional and enumerative components are tripartite structures (symbolic label-attribute-value). The descriptive and factual components, however, are characterised by the association of quadruples connecting together the symbolic name of the template/occurence, a predicate and the arguments of the predicate introduced by named relations, the roles. The quadruples have in common the name and predicate components. If we denote then with Li the generic symbolic label identifying a given template/-occurrence, with Pj the predicate used (like MOVE, PRODUCE, RECEIVE etc.), with Rk the generic role (slot, case, like SUBJ(ect), OBJ(ect), SOURCE, DEST(ination)) and with Ak the corresponding argument (concepts, individuals, or associations of concepts or individuals), the NKRL data structures for the descriptive and factual components have the following general format:

```
(Li (Pj (R1 A1) (R2 A2) & (Rn An)))
```

We can then say that, in NKRL, the intrinsic properties of concepts and individuals are described as frame-like structures, and that the mutual relationships which can be detected between those concepts or individuals when describing real-world events or classes of events are represented as case grammar-like structures.

Templates are inserted into the H_TEMP(lates) hierarchy, where each node represents a template object, producing a taxonomy of events. This enlarges the traditional interpretation of ontologies where only taxonomies of concepts are taken into consideration. Analogously, all the NKRL concepts are inserted into the H_CLASS(es) generalisation/specialisation hierarchy. At the difference of H_TEMP, which is simply a tree, H_CLASS admits in general multiple inheritance and is, in formal terms, a lattice or DAG, Directed Acyclic Graph.

Individuals (enumerative component), and predicative occurrences (factual component), are linked as well, in a way,

---

[2] http://www.cs.brandeis.edu/~jamesp/arda/time

[3] The term event is taken here in its more general meaning, covering also strictly related notions like fact, action, state, situation etc.

[4] Note that the usual ontological languages like description logics (see [5]) do not consider any sort of descriptive or factual structure, which constitute then the main conceptual innovation introduced by NKRL.

with the H_CLASS and H_TEMP hierarchies, where they appear as the leaves of particular concepts and templates. As instances of concepts, individuals share the same basic format (frames) of these last ones; analogously, occurrences are characterised by the same case grammar format of templates. The main reason for keeping the enumerative and factual components separate from the definitional and descriptive ones is linked with the very different epistemological status of, e.g., concepts vs. individuals. Other Knowledge Representation tools used in NKRL for, e.g., representing temporal data, are described in [34]. The richness and variety of the knowledge representation paradigms used by NKRL - compared with the standard taxonomic (description logic) one used in DAML+OIL, OWL etc. - allows the implementation and use of a variety of reasoning and inference mechanisms neatly more general and powerful than the usual "rule languages" used in the traditional, ontological approach. We will only mention here the possibility of implementing, in NKRL, rules of the "hypothesis" type (automatic construction of causal explanations), of the "transformation" type (allowing to find semantically similar answers also in the absence, in a knowledge base, of the information originally searched for), of powerful (positive and negative) filtering strategies, of case based reasoning (CBR) procedures, etc. Information on these topics can be found, e.g., in [35].

5. NOTES ON RDF

RDF provides a standardized syntax that allows the definition of Resources and Properties. The properties are resources used as predicates of triples; the semantics of a triple clearly depends on the property used as predicate. Two things are very important with the concept of property. First, RDF considers properties as first class object, unlike most object modeling languages, where properties are attributes of a class (this is equivalent to what we do with the Prelation statement). Even though the concept of class exists in RDF, properties can be defined and used independently of classes. Secondly, the fact that properties are resources allows them to be described with RDF itself. RDF is about describing resources; according to [21], "resources are always named by URIs" and "anything can have a URI". So RDF can theoretically be used to describe anything. Yet it was mainly designed to handle "network retrievable" resources.

However, the purpose of the Common Annotation Scheme is manyfold. It is not only a language for representing the conceptual content of documents, it is also an interchange format for the Parmenides tools. As such it needs to fulfill different targets, as we tried to model with the three-level partition presented in section 3. The need to capture the organization of the Parmenides documents is fulfilled by the structural annotations. Although it would be possible to map them into RDF (as anything can be mapped into RDF), we think it would be gross overkill.

The distinction between lexical and semantic-conceptual annotations is essentially an epistemological distinction between *"strings in text"* and *"object from the external world that those strings represent"*. It has a very practical implication in the sense that different lexical entities might be coreferent to the same conceptual entities, however this information might become available only at later stages of processing (or not at all in some case, e.g. due to shortcomings of the anaphora resolution algorithm). Because lexical annotations should be considered only as substrings of the original document (although typed), we think a representation using a conceptual framework like RDF would not be warranted.

However, at the level of semantico-conceptual annotations, the argument for using RDF as the representation language becomes more compelling. Once references are resolved into Parmenides Entities and relations are asserted among them, we are left with a representation of the content of the document, which can be mapped directly into RDF. For example, the headline: *"Torben Svejgaard, President of Emulsifiers at Denmark's ingredients giant Dansico, launched the company's fat replacer "Salatrim" today."* includes the Entities: `Torben Svejgaard` (a Person), `President of Emulsifiers` (a Role), `Danisco` (an Organization), `Salatrim` (a Product).

Once the classes of **Person**, **Role**, **Organization** and **Product** have been defined in an RDF Schema specification [4] (essentially, an Ontology), the relationships between these entities can be captured in rdf as below.

```
<Person rdf:ID="Torben_Svejgaard"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xml:base="http://www.parmenides.org/NE#>
 <WorksFor> Danisco </WorksFor>
</Person>
<Title rdf:ID="President_of_Emulsifiers"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xml:base="http://www.parmenides.org/NE#>
 <heldBy> Torben_Svejgaard </heldBy>
</Title>
<Organization rdf:ID="Danisco"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xml:base="http://www.parmenides.org/NE#>
  <IsLocated> Denmark </IsLocated>
</Organization>
<Product rdf:ID="Salatrim"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xml:base="http://www.parmenides.org/NE#>
 <ProducedBy> Danisco </ProducedBy>
</Product>
```

Further, by pointing to RDF resources as the value of properties (rather than the literals as above) an important gap can be bridged between the entities as described in a single document and the entities as they function in the real world. Essentially, RDF can facilitate the aggregation of information within the framework of a standardized, well established format. Using the Parmenides annotations, the above RDF statements would look as follows:

```
<PEntity  id="obj1" type="person"
   mnem="Torben_Svejgaard" refid="..."/>
<PEntity  id="obj2" type="role"
   mnem="President_of_Emulsifiers " refid="..."/>
<PEntity  id="obj3" type="organization"
   mnem="Danisco" refid="..."/>
<PEntity  id="obj4" type="product"
   mnem="Salatrim" refid="..."/>
<PRelation  id="r1" type="worksFor"
   source="obj1" target="obj3 "/>
<PRelation  id="r2" type="helbBy"
   source="obj2" target="obj1"/>
<PRelation  id="r3" type="isLocated"
   source="obj3" target=".... "/>
<PRelation  id="r4" type="producedBy"
  source="obj4" target="obj2"/>
```

The information represented in the two formats is the same, with the exception that the Parmenides annotations include a list of the *lexical annotations* from which they were derived (the 'refid' attribute). This has however just a practical purpose to allow the annotations to be *traced back* into the document. The other difference is that we propose to use an indexical name for entities and relations (equivalent to an index in a Database), to avoid any *naming* conflict. XML ids are unique within the document and, if combined with an unique document identifier, can be made unique across the entire Parmenides Document Warehouse. Using a human-level identifier (e.g. the name of a company or a person) would surely result ambiguous within any collection. This approach however is not very different from the usage of Uniform Resource Identifiers (URIs) in RDF.

The format that we suggest appears to be easier to map into a DB representation, however RDF statements such as those shown above can easily be converted into DB entries. There is a proposal for non-XML serialization of RDF (called Notation3), which might be closer to our approach. In conclusion, except for the refid attributes, we think that the Parmenides conceptual annotations can be mapped directly into RDF (and viceversa). Given this equivalence, why then not choose RDF as the representation language?

We think that there are two main reasons to prefer (in the context of the Parmenides project) the Parmenides-style conceptual annotations. The first one is uniformity and simplicity of the markup (intended as *interlingua* within the Parmenides tools). The second one is that it might be convenient to maintain an epistemological distinction between the Parmenides annotations (which always are based upon a source document) and RDF annotations (which, in theory at least, refer to resources which exist independently of a document which mentions them).

This might be made more clear with an example, suppose that in the Parmenides document number 12566 the company Dyax is mentioned a couple of times. As we explained before, each occurrence of the string `Dyax` would give rise to a different Parmenides lexical entity, like:

`<PNAMEX id='e34' type='organization' tokid='t45'/>`

Which could be glossed as *"The token t45 of the Parmenides document 12566 represents an object of type Organization"*. All the co-referring lexical entities would then be merged into a conceptual entity like the following:

`<PEntity id='p23' type='organization' refid='..e34..'/>`

This could be glossed as *"In the Parmenides document number 12566 there are various mentions of an object of type organization, one of them is e34"*. By following the references we could further assert that such company is referred to as `Dyax`, `Dyax Corp.`, and so on....

However, we still do not know anything about the *"real"* Dyax company. Whatever we learn in document 12566 might be contradicted by another document in the collection. Or we might have references to an object which shares its name with the one found, but it is a distinct one. So what the Parmenides annotation above says, should be glossed as *"Dyax as mentioned in the Parmenides document 12566"*. If we had to translate this into an RDF-style URI, it would look like the following:

`http://www.parmenides.org/docbase/pardoc12566#p23`

An aggregator tool should be able to detect the existence of the same company within different documents, and thus create a new, document-independent resource, to which all the individuals mention of `Dyax` in different documents point to, such as:

`http://www.parmenides.org/organization#Dyax`

In any case, it is possible (and certainly useful) to provide an export utility that converts the Parmenides semantico-conceptual entities into RDF.

6. RELATED WORK

Parmenides aims at using consolidated Information Extraction techniques, such as Named Entity Extraction, and therefore this work builds upon well-known approaches, such as the Named Entity annotation scheme from MUC7 [7]. Crucially, attention is also given to temporal annotations, with the intention of using extracted temporal information for detection of trends (using Data Mining techniques). Therefore we have investigated all the recently developed approaches to such a problem, and have decided for the adoption of the TERQAS tagset [27]. Other annotation sources that have been considered are the GENIA tagset [16], TEI [33] and the GDA[5] tagset.

The goals of the project are similar in kind to many efforts to use semantic annotations to increase the computational information content of documents. However, the increasing popularity of RDF(S) and DAML+OIL within the community is clear. From the development of annotation tools to describe the semantic content of documents within this formalism [2], to demonstrable enhancements in information retrieval over research data [23], to machine learning algorithms for generating RDF document annotations[22].

Whilst NKRL is compatible with RDF, it possess greater inference capabilities, especially in terms of temporal reasoning. The four components of NKRL are highly homogeneous, given that templates, occurrences, concepts and individuals are all implemented as structured objects identified by a symbolic label.

More precisely, the definitional and enumerative data structures that support the concepts and the individuals are built up in a frame-like fashion, i.e., as bundles of attribute/value relations where neither the number nor the order of the attributes is fixed. NKRL frames conform to the general requirements of the Open Knowledge Base Connectivity (OKBC), see [6], and are quite similar to the object structures used in tools like Protégé-2000 [25]. The structured objects of the descriptive and factual components (templates and occurrences), on the other hand, are more original, and make use of data structures which are not dissimilar to the case grammars used in Linguistics and Computational Linguistics, see [15, 30].

7. CONCLUSION

In this paper we have presented ongoing work within the scope of the European project *Parmenides*, focusing in particular on the Annotation Scheme developed with the dual aim to serve as data integration support and as a text annotation tool. We have also introduced the important role played by other components of the Parmenides system, such as the Knowledge Representation Tool NKRL.

We hope that this paper will provide a useful input to ongoing discussion on the role of Knowledge Markup and Semantic Annotations, especially related to developments in the Semantic Web movement.

---

[5] http://www.i-content.org/GDA/tagset.html

Acknowledgments

The Parmenides project is funded by the European Commission (contract No. IST-2001-39023) and by the Swiss Federal Office for Education and Science (BBW/OFES).

8. REFERENCES

[1] Appelt, D., and Israel, D. Introduction to Information Extraction Technology. In *Proc. of 16th International Joint Conference on Artificial Intelligence, IJCAI-99* (Sweden, 1999).

[2] Bechhofer, S., and Goble, C. Towards Annotation using DAML+OIL. In *Proc. of Workshop on Knowledge Markup and Semantic Annotation, K-CAP01* (BC, Canada, 2001).

[3] Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., and Stein, L. A. OWL Web Ontology Language Reference. W3C Candidate Recommendation, 2003. `http://www.w3.org/TR/owl-ref/`.

[4] Brickley, D., and Guha, R. RDF vocabulary description language 1.0: RDF Schema. Tech. rep., W3C working draft, World Wide Web Consortium, April 2002. A reference for RDFS.

[5] Buchheit, M., Donini, F., and Schaerf, A. Decidable reasoning in terminological knowledge representation systems. *Journal of Artificial Intelligence Research 1* (1993), 109–138.

[6] Chaudhri, A., Fikes, R., Karp, P., and Rice, J. OKBC: A Programmatic Foundation for Knowledge Based Interoperability. In *Proc. of the National Conference on Artificial Intelligence, AAAI98* (Cambridge(MA), 1998).

[7] Chinchor, N. MUC-7 Named Entity Task Definition, Version 3.5, 1997. `http://www.itl.nist.gov/iaui/894.02/related\_projects/muc/proceedings/n%e\_task.html`.

[8] Cimiano, P., and Handschuh, S. Ontology-based linguistic annotation. In *The ACL-2003 workshop on Linguistic Annotation, July 2003, Sapporo, Japan.* (2003).

[9] DAML+OIL, 2001. http://www.daml.org/.

[10] Ding, Y. A review of ontologies with the Semantic Web in view. *Journal of Information Science 27*, 6 (2001), 377–384.

[11] Ellman, J. *Using Roget's thesaurus to determine the similarity of text.* PhD thesis, University of Sunderland, 2000.

[12] Embley, D., Campbell, D., Liddle, S., and Smith, R. Ontology-Based Extraction and Structuring of Information from Data-Rich Unstructured Document. In *Proc. of 7th International Conference on Information Knowledge Management, CIKM-98* (Maryland (USA), 1998).

[13] Farquhar, A., Fikes, R., and Rice, J. The Ontolingua Server: A Tool for Collaborative Ontology Construction. Tech. rep., Stanford KSL, 1996.

[14] Ferro, L., Mani, I., Sundheim, B., and Wilson, G. Tides temporal annotation guidelines, version 1.0.2. Tech. rep., The MITRE Corporation, 2001.

[15] Fillmore, C. *Universals in Linguistic Theory.* Holt, Rinehart and Winston, 1968, ch. The Case for Case.

[16] GENIA. Genia project home page, 2003. http://www-tsujii.is.s.u-tokyo.ac.jp/˜genia.

[17] Gruber, T. R. *The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Base.* Morgan Kaufmann, 1991.

[18] Guarino, N. Formal ontologies in information systems. In *Proceedings of FOIS'98* (Trento, June 1998), N. Guarino, Ed., IOS Presss, Amsterdam, pp. 3–15.

[19] Hendler, J. Agents and the semantic web. *IEEE Intelligent Systems 16*, 2 (2001), 30–37.

[20] Kogut, P., and Holmes, W. AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages. In *Proceedings of the K-CAP 2001 Workshop on Knowledge Markup and Semantic Annotation* (2001).

[21] Lassila, O., and Swick, R. R. Resource description framework (RDF) model and syntax specification. Tech. rep., W3C, 1999. `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222`.

[22] Li, J., Zhang, L., and Yu, Y. Learning to Generate Semantic Annotation for Domain Specific Sentences. In *Proc. of Workshop on Knowledge Markup and Semantic Annotation, K-CAP01* (BC, Canada, 2001).

[23] Lopatenko, A. S. Information retrieval in Current Research Information Systems. In *Proc. of Workshop on Knowledge Markup and Semantic Annotation, K-CAP01* (BC, Canada, 2001).

[24] Morris, J., and Hirst, G. Lexical Cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics 17*, 1 (1991), 21–48.

[25] Noy, N., Fergerson, R., and Musen, M. The Knowledge Model of Protege-2000: Combining Interoperability and Flexibility. In *Knowledge Acquisition, Modeling and Management - Proc. of the European Knowledge Acquisition Conference, EKAW2000* (Berlin, 2000).

[26] Patel, M. Concensus based ontology harmonisation. Poster, International Semantic Web Conference, 2002.

[27] Pustejovsky, J., Sauri, R., Setzer, A., Gaizauskas, R., and Ingria, B. TimeML Annotation Guideline 1.00 (internal version 0.4.0), July 2002. `http://www.cs.brandeis.edu/~jamesp/arda/time/documentation/TimeML-Draft%3.0.9.html`.

[28] Setzer, A. *Temporal Information in Newswire Articles: An Annotation Scheme and Corpus Study.* PhD thesis, University of Sheffield, 2001.

[29] Smeaton, A. F. Using nlp or nlp resources for information retrieval tasks. In *Natural Language Information Retrieval*, T. Strzalkowski, Ed. Kluwer Academic Publishers, 1997, pp. 99–111.

[30] SparkJones, K., and Boguraev, B. A Note on a Study of Cases. *Computational Linguistics 13* (1997), 65–68.

[31] Spiliopoulou, M., and Pohle, C. Modelling and Incorporating Background Knowledge in the Web Mining Process. In *Proc. of ESF Explanatory Workshop on Pattern Detection and Discovery* (2002), pp. 154–169.

[32] Stevenson, M., and Wilks, Y. The interaction of knowledge sources in word sense disambiguation. *Computational Linguistics 27*, 3 (2001).

[33] TEI Consortium. The text encoding initiative, 2003. `http://www.tei-c.org/`.

[34] Zarri, G. Representation of temporal knowledge in events: The formalism, and its potential for legal narratives. *Information and Communications Technology Law - Special Issue on Models of Time, Action, and Situations 7* (1998), 213–241.

[35] Zarri, G. A conceptual model for representing narratives. In *Innovations in Knowledge Engineering*, R. Jain, A. Abraham, C. Faucher, and B. van der Zwaag, Eds. Advanced Knowledge International, Adelaide (Aus.), 2003.

---

[6] Available at `http://www.cl.unizh.ch/CLpublications.html`