Generating Multilingual Personalized Descriptions from OWL Ontologies on the Semantic Web: the NaturalOWL System*

Dimitrios Galanis and Ion Androutsopoulos
Department of Informatics
Athens University of Economics and Business
Patission 76, GR-104 34 Athens, Greece

August 2007

Abstract

We present NaturalOWL, an open-source multilingual natural language generation system that produces personalized descriptions of individuals and classes, starting from a linguistically annotated ontology. The generator is heavily based on ideas from ILEX and M-PIRO, but it is in many ways simpler and it provides native support for OWL DL ontologies, which we annotate with linguistic and user modelling resources in RDF. NaturalOWL is written in Java and it is supported by M-PIRO's authoring tool, as well as an alternative authoring plug-in for the Protégé ontology editor.

1 Introduction

In recent years, considerable effort has been devoted to the Semantic Web [5, 3], which can be thought of as an attempt to establish mechanisms that will allow computer applications to reason more easily about the semantics of the Web's resources (documents, services, etc.), enabling them to share, locate, and integrate the resources more easily. Domain ontologies play a central role in this endeavour. They establish domain-dependent semantic vocabularies and background knowledge; for example, definitions of classes and subclasses of individuals, possible relations between the individuals of the various classes, properties of particular individuals etc. These can be used to mark up Web resources with machine-readable meta-data describing aspects of their semantics and to carry

*This document is part of Deliverable 2.1 of project XENIOS, which is co-funded by the European Union and the Greek General Secretariat of Research and Technology; see `http://www.ics.forth.gr/xenios/` for further detalis. A shorter version of this document has also been published [11].

out supporting inferences; for example, to perform consistency checks when integrating information, or to deduce missing information. Currently, the standard formalism for specifying ontologies of the Semantic Web is OWL (Ontology Web Language), a language built on top of RDF (Resource Description Framework) and RDF SCHEMA.[1]

The Web is currently dominated by documents containing mostly natural language texts. Documents of this type can be marked up with semantic annotations via mechanisms like RDFa, which allows RDF statements to be embedded in HTML; the semantic annotations can then be extracted from the documents by using mechanisms such as GRDDL.[2] Annotating documents manually, however, is time-consuming and tedious; consequently, manual document annotations, if at all present, typically convey very limited information, such as the authors and main topics of the documents, leaving most of the semantic content of the documents inaccessible to computer applications that cannot interpret natural language texts. Some semantic annotations can be produced automatically via ontology-aware information extraction [9]. Information extraction, however, can currently provide reliably only relatively simple types of semantic information, mostly by identifying and classifying named entities (individuals) mentioned in the texts, and, less reliably, relations between the named entities; hence, most of the semantic content of the documents remains inaccessible to computer applications.

A viable alternative in many application domains is to generate the documents automatically from an underlying ontology or other formal knowledge representation [25]; in this case, the generator can easily annotate the resulting documents with much richer information, including full representations of the semantics of their texts. In fact, an entire strand of work in natural language generation (NLG) has focused on generating textual descriptions of an ontology's classes or individuals.[3] A well-known example of such work is ILEX [24], which was demonstrated mostly with ontologies of museum exhibits. More recently, the M-PIRO project [17, 2] developed a multilingual extension of ILEX, which was tested in several domains, including museum exhibits and computing equipment. In this type of work, the ontology's role is no longer simply to provide a semantic vocabulary and background knowledge; the ontology acts as the main knowledge repository, and parts of its knowledge (e.g., information pertaining to particular individuals or classes) can be rendered automatically in any of several natural languages or in a machine-readable form that carries the same semantic content. For example, M-PIRO's generator can deliver a description like the following to an English-speaking human on-line shopper,

> `A110`: This is a laptop, manufactured by Toshiba. It has a Centrino Duo processor, 512 MB RAM, and an 80 GB hard disk. Its speed is 1.7 GHz and it costs 850 Euro.

and the following description to a Greek speaker; M-PIRO also supports Italian.

[1] Consult `http://www.w3.org/RDF/` and `http://www.w3.org/TR/owl-features/`.

[2] `http://www.w3.org/TR/xhtml-rdfa-primer/`, `http://www.w3.org/2001/sw/grddl-wg/`.

[3] Individuals are also called "instances" (of classes) or "entities".

A110: Αυτός είναι ένας φορητός υπολογιστής, κατασκευασμένος από την Toshiba. Διαθέτει επεξεργαστή Centrino Duo, 512 MB RAM και σκληρό δίσκο 80 GB. Η ταχύτητά του είναι 1.7 GHz και κοστίζει 850 Euro.

A semantically equivalent machine-readable representation of the above information could be easily generated when interacting with a computer agent: in effect, the generator would simply report the output of its content selection stage, the stage that determines which parts of the ontology's knowledge to convey. In our example, an appropriate machine-readable representation in OWL would look like the following.[4] Alternatively, each individual sentence of the natural language texts could have been marked up with a machine readable representation of its semantics.

```
<Laptop rdf:ID="A110">
 <manufacturedBy rdf:resource="#toshiba" />
 <hasProcessor rdf:resource="#centrinoDuo" />
 <hasMemory rdf:datatype="http://.../XMLSchema#string">512 MB</hasMemory>
 <hasDisk rdf:datatype="http://.../XMLSchema#string">80 GB</hasDisk>
 <hasSpeed rdf:datatype="http://.../XMLSchema#string">1.7 GHz</hasSpeed>
 <hasCost rdf:datatype="http://.../XMLSchema#string">850 Euro</hasCost>
</Laptop>
```

There are application domains (e.g., on-line shops, virtual museums) where one can envisage future Semantic Web sites that will maintain and publish their content entirely in the form of OWL ontologies. Then, NLG technology embedded in server or browser plug-ins could be used to render parts of the OWL ontologies in multiple natural languages or equivalent machine-readable representations on demand. NLG can, thus, be seen as a key technology of the Semantic Web, which makes knowledge fully accessible to both humans and computers. Furthermore, NLG can help keep Web content up to date across multiple languages, by allowing sites to focus on updating only their ontologies, from which texts are generated automatically. Other well-known benefits of NLG include the ability to tailor the resulting texts per user type (e.g., different content conveyed to experts vs. non-expert visitors, different natural language expressions used when interacting with adults vs. children), and to adapt the texts depending on the interaction history (e.g., avoid repeating the same facts, include comparisons to previously described individuals) [24, 2, 23].

In this paper, we introduce NaturalOWL, a prototype open-source natural language generator intended to demonstrate what NLG can offer to the Semantic Web. NaturalOWL is heavily based on ideas from ILEX and M-PIRO, but unlike its predecessors it provides native support for OWL DL, the most principled version of OWL that corresponds to description logic [4]. Previous attempts to support OWL in M-PIRO's generator ran into problems, because of incompatibilities between OWL and M-PIRO's ontological model [1]. Compared to ILEX and

[4] NaturalOWL currently outputs this information in a proprietary XML format, but it would be straightforward to produce OWL output like the one shown here instead. For simplicity, we show scalar values as strings that include the units of measurements; more principled, language-independent representations of these values are also possible.

M-PIRO, NaturalOWL is also simpler; for example, it is entirely template-based, as opposed to the systemic grammars [13] its predecessors employed.[5] Although future work may well enhance some of NaturalOWL's components, the simplicity of the current system makes it easier to master, as it does not require particular expertise in NLG. It also facilitates extending the system to support additional natural languages; for example, there is no need to develop systemic grammars for new natural languages. NaturalOWL currently supports English and Greek.

It has been argued [22] that in most OWL ontologies, classes and properties are given names that are either English words (e.g., `Laptop`, `cost`) or concatenations of English words (e.g., `manufacturedBy`, `hasMemory`). Based on this observation, Sun and Mellish [26] generate texts from RDF descriptions, without any domain-dependent linguistic resources. They use WordNet to tokenize the names of classes and properties, as well as to assign part-of-speech (POS) tags to tokens, and this allows them to guess that a class name like `Laptop` above is in fact a noun that can be used to refer to that class, or that `<manufacturedBy rdf:resource="#toshiba"/>` should be expressed in English as "[This laptop] was manufactured by Toshiba". Hewlett *et al.* [14] adopt very similar techniques. Similar in spirit, although much simpler, mechanisms appear to be used in the tooltips of classes and properties in Protégé, an ontology editor most Semantic Web researchers will be familiar with.[6] This approach, however, is problematic when texts have to be generated in multiple languages; for example, English-like class and property names do not show how the corresponding concepts could be rendered in other natural languages. Even in the monolingual case, there are significant problems: for example, a POS-tagger is often needed to distinguish between noun and verb uses of the same token, and morphological or even syntactic analysis is needed (especially in highly inflected languages) to extract natural language tokens from class and property names and convert them into grammatical phrases; in effect, this re-introduces to some extent the need to interpret natural language automatically, a problem that computer applications face in today's Web. Furthermore, our experience is that generating high-quality texts often requires linguistic information that is not present, not even indirectly, in OWL ontologies, nor can it be embedded conveniently in them [1]; we return to this point in following sections.

We, therefore, propose annotating OWL ontologies with stand-off RDF markup that associates elements of the ontologies (e.g., classes, properties) with domain-dependent linguistic resources (e.g., nouns that can be used to refer to classes, natural language templates that can be used to express properties). Apart from allowing parts of the ontology to be presented to end-users in multiple natural languages, this kind of linguistic annotation facilitates presenting ontologies to domain experts for validation; and the annotations can also be useful when querying or extending ontologies via natural language [18, 6]. Consequently, we believe that linguistic annotation should become a standard part of ontology

[5] Consult `http://www.ltg.ed.ac.uk/methodius/` for information on METHODIUS, another offspring of M-PIRO's generator, which uses CCG grammars and can handle larger ontologies and databases.

[6] Consult `http://protege.stanford.edu/`.

engineering for the Semantic Web. To demonstrate the benefits and viability of this recommendation, NaturalOWL is accompanied by a plug-in for Protégé.[7] The plug-in allows the ontology engineers to define all of the necessary linguistic annotations of an OWL DL ontology, as well as user modelling annotations to exploit the other benefits of NLG, and preview the resulting textual descriptions. M-PIRO's authoring tool [2], now called ELEON, has also been extended to support NaturalOWL [7]. For users that are not familiar with Protégé and OWL, ELEON is a more user-friendly alternative, but it is limited to OWL LITE, a subset of OWL DL. Consult also Bontcheva [8] for related work on authoring tools for NLG.

NaturalOWL's RDF linguistic annotations will hopefully contribute towards a cross-disciplinary discussion involving the communities of Semantic Web and NLG researchers, on how to annotate OWL ontologies with linguistic and user modelling information. This may eventually produce standards that will allow alternative NLG components to render OWL ontologies in natural language, in the same way that alternative browsers can be used to view HTML pages.

Section 2 below presents the overall architecture of NaturalOWL, its processing stages, as well as the format and purpose of the linguistic and user modelling annotations that are used in each stage. Section 3 then provides more information on the use of the Protégé plug-in. Finally, section 4 concludes and proposes directions for further work. Readers familiar with XML should be able to follow most of the discussion in this paper, but familiarity with RDF and OWL is necessary to grasp the full details of some of the examples that will be presented. For simplicity, we focus mostly on descriptions of individuals. NaturalOWL can also describe classes, but it currently conveys only information that is explicit in the definition of each class, excluding inherited and inferred restrictions, unlike the work of Mellish and Sun [21], where class descriptions also convey inferred facts.

2 NaturalOWL's processing stages

NaturalOWL adopts a pipeline architecture, which is typical in many natural language generators [25]. It generates texts in three stages: document planning, micro-planning, and surface realization. In document planning, the system selects from the ontology the logical facts it will convey to the reader (a sub-stage known as content selection) and it specifies the overall structure of the text (text planning), including the order in which the facts will be conveyed. In micro-planning, the system constructs an abstract sentence specification for each fact to be conveyed (a sub-stage known as lexicalization), it aggregates the sentence specifications to produce longer periods, and it generates appropriate referring expressions (e.g., pronouns vs. using proper names). Finally, during surface realization, the abstract sentence specifications of micro-planning are

[7] NaturalOWL and its Protégé plug-in are freely available with a GPL license; they can be downloaded from `http://www.aueb.gr/users/ion/publications.html`.

transformed into the final text, and appropriate markup (e.g., HTML tags or semantic markup) are added, if necessary. We discuss the three stages in turn in the following subsections. As in Wilcock's work [28], in NaturalOWL the stages communicate in XML, but they are implemented in Java, instead of XSLT, and there is a clearer separation between processing code and linguistic resources.

2.1 Document planning

2.1.1 Content selection

When instructed to produce a natural language description of an individual, NaturalOWL first selects all the logical facts of the ontology that are directly relevant to that individual. For example, when describing the laptop of section 1, whose identifier is `A110`, it would select the fact that `A110` is a `Laptop`, the fact that its manufacturer is Toshiba, etc. As in RDF, each fact of an OWL ontology can be thought of as a $\langle subject, property, object \rangle$ triple; for example, `<A110, type, Laptop>`, `<A110, manufacturedBy, toshiba>`. Similarly, when asked to describe the individual `exhibit24`, for which the following information is available in the OWL version of M-PIRO's museum ontology, NaturalOWL first selects the directly relevant facts, i.e., that the individual is an aryballos (a kind of vase), that its creation time is the seventh century B.C., that its current location is the Archaeological Museum of Delos, etc.

```
<aryballos rdf:ID="exhibit24">
  <creation-time rdf:resource="#sevencenturylateBC"/>
  <current-location rdf:resource="#archaelogical-delos"/>
  <exhibit-style rdf:resource="#corinthian-style"/>
  <location-found rdf:resource="#Iraion-Delos"/>
  <painting-technique-used rdf:resource="#black-figure-technique"/>
  <exhibit-story rdf:resource="#exhibit24str"/>
  <creation-period rdf:resource="#archaic-period"/>
</aryballos>
```

The ontology, however, will often contain further information on the individuals or classes that the individual being described is connected to. In our example, `exhibit24` is connected to the individual `archaic-period` through the property `creation-period`; intuitively, `exhibit24` was created during the archaic period. Through the property `covers`, the ontology further connects `archaic-period` to another individual, called `archaic-period-duration`, as shown below, which represents the time-span of `archaic-period`; we can ignore for the moment exactly how this time-span is represented in `archaic-period-duration`. Similarly, through the property `historical-period-story`, `archaic-period` is connected to a multilingual personalized canned text (MPCT) that provides a description of that period; we discuss MPCTs in following sections. The identifier of the MPCT is `archaic-story`.

```
<period rdf:ID="archaic-period">
  <covers rdf:resource="#archaic-period-duration"/>
  <historical-period-story rdf:resource="#archaic-story">
```

```
</period>
```

Figure 1 depicts as a graph some of the ontology's facts that relate to `exhibit24`. Each $\langle subject, property, object \rangle$ fact is shown as a directed edge, labeled *property*, that connects a node labeled *subject* to a node labeled *object*. Cyan nodes represent individuals, whereas green nodes represent classes. The facts that are directly relevant to `exhibit24` are the edges that start from the node of `exhibit24`; for example the fact that `exhibit24` was created in the archaic period, or that `exhibit24` is an aryballos. The facts that are indirectly relevant to `exhibit24` are all the other edges of all the paths that start from `exhibit24`; for example, the duration of the archaic period, the fact that an aryballos is a kind of vessel, and that a vessel is a kind of exhibit. NaturalOWL can be configured (through an initialization parameter) to select facts (edges) up to a maximum distance from the individual or class being described. The default distance is 2, which would eventually lead to a text like the following:

> `exhibit24`: This is an aryballos, a kind of vessel. An aryballos was a small spherical vessel with a narrow neck, in which the athletes kept the oil they spread their bodies with. This particular aryballos was found in the Heraion of Delos and it is currently exhibited in the Archaeological Museum of Delos. It was created during the archaic period. The archaic period was the time during which the Greek ancient city-states developed and it covers the time between 700 B.C. and 480 B.C. This aryballos was decorated with the black-figure technique. In the black-figure technique, the silhouetes are rendered in black on the pale surface of the clay, and details are engraved.

Actually, NaturalOWL may not select all of the facts within the maximum allowed distance from the individual or class being described. For example, it may discard the edge that links `archaic-period` to `archaic-period-duration`, causing the sentence "it covers the time between 700 B.C. and 480 B.C." to be excluded from the description, if it believes that the corresponding fact has already been assimilated by the reader. For example, it may be the case that the reader (visitor) has already encountered another exhibit of the same period, and the description of that exhibit mentioned the duration of the archaic period, in which case repeating the same information may be boring for the reader. As in ILEX and M-PIRO, NaturalOWL maintains a personal model for each reader, where it stores information showing which facts have already been conveyed to the particular reader in previous descriptions.[8] Each reader is given a unique identifier during a login phase, and this identifier is used to retrieve the corresponding personal model whenever necessary. During the login phase, each reader also selects their user type; for example, in M-PIRO's museum application

[8] As in M-PIRO, the personal models can be stored in the Personalization Server of NCSR "Demokritos", which provides persistent storage across multiple sessions (visits), or in an emulator of that server, whose contents are only stored in memory. An emulator of the latter type is included in the distribution of NaturalOWL. Contact NCSR "Demokritos" for information on how to obtain and use the Personalization Server.

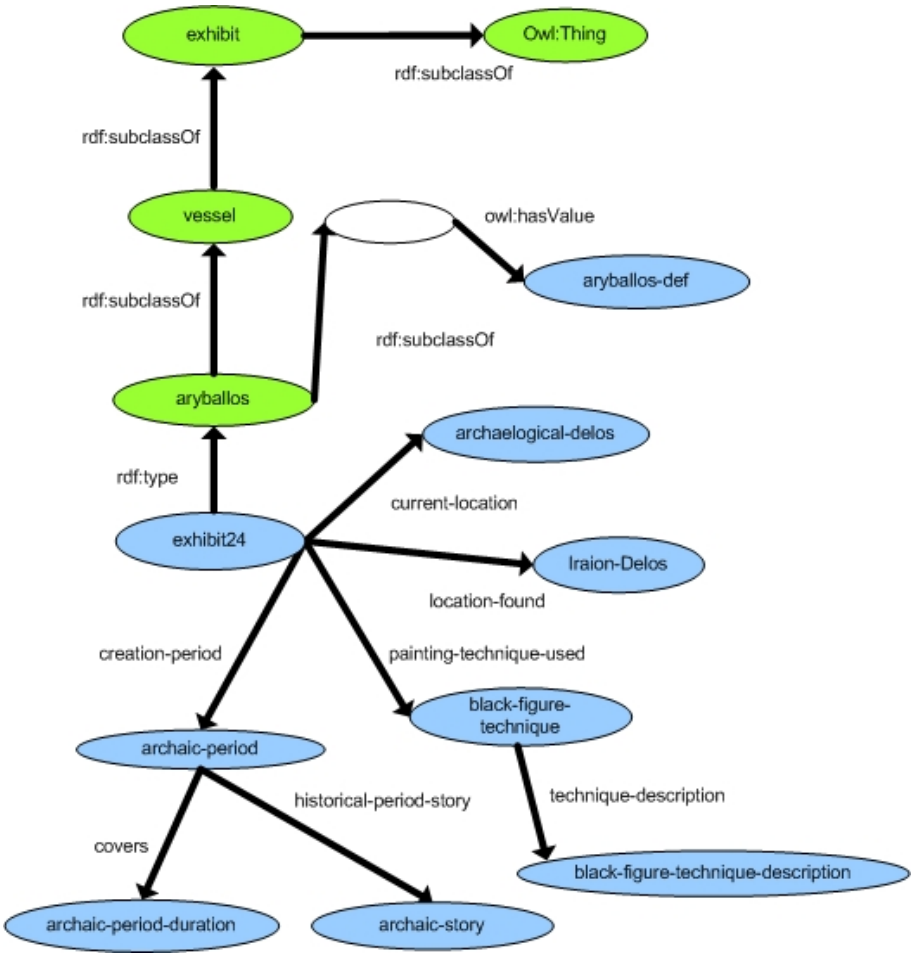

Figure 1: Graph representation of an OWL ontology.

there are three user types: child, non-expert adult, and expert adult. The user modelling annotations of the ontology show, among other things, how many times a fact has to be repeated when interacting with users of each type, before the system can assume that the facts has been assimilated.

Figure 2 illustrates NaturalOWL's user modelling annotations by showing the annotations of property `painting-technique-used`.[9] Each property can be annotated at multiple levels:

global level: This is the `DPInterestRepetitions` part of the annotations in figure 2. The annotations of this level apply by default whenever the property is used. In our example, the annotations of this level specify that mentioning a fact with property `painting-technique-used` once is enough to consider that the particular fact has been assimilated by the reader.

domain level: This is the `CDPInterestRepetitions` part. The annotations of this level, which override those of the global level, apply by default whenever the property is used in a fact with a *subject* from a particular domain. In our example, the annotations of this level specify that a fact whose property is `painting-technique-used` and whose *subject* is an individual of class `aryballos` should be mentioned twice for the fact to be considered assimilated.

individual level: This is the `IPInterestRepetitions` part. The annotations of this level, which override those of the previous levels, apply when the property is used in a fact with a particular individual as its *subject*. In our example, the annotations of this level specify that mentioning once a fact whose property is `painting-technique-used` and whose *subject* is `exhibit20` is enough for the fact to be considered assimilated.

The user modelling annotations also assign interest scores (integers in the range $[0, 19]$) to the various facts, as shown in figure 2. Higher scores correspond to more interesting facts. If a fact's interest is 0, then it is never conveyed; this can be used, for example, to make sure that the readers will never be told that an exhibit is a thing. Again, interest can be assigned at the global, domain, or individual level. This allows one, for example, to specify that facts about the materials of the exhibits are generally very interesting, but in the case of statues they are uninteresting (perhaps because most of the statues of the particular collection are made of marble), with the exception of a particular statue (perhaps because that statue is made of gold). If no user modelling annotations can be found for a fact, the default is 1 for both interest and repetitions.

During content selection, NaturalOWL first collects all the facts of the ontology that are within the maximum allowed distance in the ontology's graph from the individual or class being described. It then discards assimilated facts, it orders the remaining facts by interest, and it selects the m most interesting

[9] NaturalOWL's linguistic and user modelling annotations are kept in separate files from the OWL ontology, but they refer to the ontology's classes, properties and individuals via their unique identifiers; we abbreviate the identifiers to save space.

```
<owlnl:Property rdf:about="http://.../mpiro.owl#painting-technique-used">
  <owlnl:DPInterestRepetitions rdf:parseType="Resource">
    <owlnl:forUserType rdf:resource="#Child"/>
    <owlnl:InterestValue>6</owlnl:InterestValue>
    <owlnl:Repetitions>1</owlnl:Repetitions>
  </owlnl:DPInterestRepetitions>
  ...
  <owlnl:CDPInterestRepetitions rdf:parseType="Resource">
    <owlnl:forOwlClass rdf:resource="http://.../mpiro.owl#aryballos"/>
    <owlnl:forUserType rdf:resource="#Child"/>
    <owlnl:InterestValue>2</owlnl:InterestValue>
    <owlnl:Repetitions>2</owlnl:Repetitions>
  </owlnl:CDPInterestRepetitions>
  ...
  <owlnl:IPInterestRepetitions rdf:parseType="Resource">
    <owlnl:forUserType rdf:resource="#Child"/>
    <owlnl:forInstance rdf:resource="http://.../mpiro.owl#exhibit20"/>
    <owlnl:InterestValue>0</owlnl:InterestValue>
    <owlnl:Repetitions>1</owlnl:Repetitions>
  </owlnl:IPInterestRepetitions>
  ...
</owlnl:Property>
```

Figure 2: User modelling annotations of properties.

ones, where m is a user modelling parameter (`FactsPerPage`, not shown in figure 2), specified per user type in NaturalOWL's user modelling annotations. This is very similar to ILEX's content selection, but without employing rhetorical relations.

2.1.2 Text planning

Once content selection has been completed, the selected facts at distance 1 from the individual or class being described are ordered by consulting ordering annotations (see `owlnl:order` in Figure 3), which specify a partial order of properties; is-a facts are always mentioned first. In the ontology of figure 1, the ordering annotations may specify that the original location (`location-found`) should be mentioned first, followed by the current location, the creation period, and the painting technique. Facts at distance 2 are always placed right after the corresponding facts of distance 1, with the facts at distance 2 ordered again according to the annotations, and so on for greater distances. This produces texts like the description of `exhibit24` in section 2.1.1. In the application domains we have considered, this ordering scheme was adequate, although in other domains more elaborate text planning approaches may be needed; consult, for example, Bontcheva and Wilks [10] for an application of text schemata to NLG from ontologies. The ordering annotations could also be made sensitive to user type and target language.

2.2 Micro-planning

2.2.1 Lexicalization

For each property (e.g., `location-found`), one or more micro-plans need to be defined per supported natural language in order to specify how facts that involve that property (e.g., `<exhibit24, location-found, Iraion-Delos>`) can be expressed in each language. NaturalOWL's micro-plans are templates, each consisting of a sequence of slots; the fillers of the slots are concatenated to create a sentence. Each slot can be filled by one of the following:

1. An automatically generated referring expression pointing to the *subject* of the fact; for example "it" or "this aryballos" to refer to `exhibit24` in the case of `<exhibit24, location-found, Iraion-Delos>`.
2. A fixed string; for example "it was found in".
3. An automatically generated referring expression pointing to the *object* of the fact, if the *object* is an individual; for example "the Heraion of Delos" in the case of `<exhibit24, location-found, Iraion-Delos>`.
4. The *object* itself, if it is an instance of a data-type, such as an integer; for example, "8203571" in `<employee1968, hasPhoneNumber, 8203571>`.
5. The string of the *object* that corresponds to the micro-plan's language and the reader's user type, if the *object* is an MPCT (multilingual personalized canned text); this will be discussed below.

In M-PIRO's terminology, where facts were thought of as fields, the *subject* of a fact was taken to be the `owner` of the field, whereas the *object* of the fact was thought of as the `filler` of the field. NaturalOWL still uses the terms `owner` and `filler` to refer to the *subject* and *object* of a fact, respectively. In cases 3–5 above, one actually simply specifies that the slot should be filled in by the *object* of the fact (`filler` of the field), and NaturalOWL figures out automatically if it should fill in the slot with a referring expression pointing to the *object*, the *object* itself, or the appropriate string of an MPCT object, depending on whether the *object* is an individual (instance of class), an instance of a data-type, or an MPCT, respectively. Figure 3 illustrates how NaturalOWL's micro-plans are represented in RDF; the example also shows the ordering annotations (`owlnl:order`) discussed in section 2.1.2.

The English micro-plan in figure 3, identified as `location-found-templ1-en`, consists of four slots. According to the specifications of the micro-plan, the first slot is to be filled in by a referring expression in nominal case referring to the *subject* of the fact (the `owner` of the field); for example, in the case of the fact `<exhibit24, location-found, Iraion-Delos>`, the referring expression could be "it", "this aryballos", or the natural language name of the exhibit, if there is one. The `RE_AUTO` in the first slot allows the system to select automatically among using a pronoun, a noun phrase, or the natural language name,

```
  <owlnl:Property rdf:about="http://.../mpiro.owl#location-found">
  <owlnl:Order>1</owlnl:Order>
  <owlnl:GreekMicroplans rdf:parseType="Collection">
  ...
  </owlnl:GreekMicroplans>
  <owlnl:EnglishMicroplans rdf:parseType="Collection">
    <owlnl:Microplan rdf:about="http://.../mpiro.owl#location-found-templ1-en">
      <owlnl:MicroplanName>Template 1</owlnl:MicroplanName>
      <owlnl:Used>true</owlnl:Used>
      <owlnl:AggrAllowed>true</owlnl:AggrAllowed>
      <owlnl:Slots rdf:parseType="Collection">
        <owlnl:Owner>
          <owlnl:case>nominative</owlnl:case>
          <owlnl:RETYPE>RE_AUTO</owlnl:RETYPE>
        </owlnl:Owner>
        <owlnl:Verb>
          <owlnl:voice>passive</owlnl:voice>
          <owlnl:tense>past</owlnl:tense>
          <owlnl:Val xml:lang="en">was found</owlnl:Val>
          <owlnl:pluralVal xml:lang="en">were found</owlnl:pluralVal>
        </owlnl:Verb>
        <owlnl:Text>
          <owlnl:Val xml:lang="en">in</owlnl:Val>
        </owlnl:Text>
        <owlnl:Filler>
          <owlnl:case>accusative</owlnl:case>
          <owlnl:RETYPE>RE_AUTO</owlnl:RETYPE>
        </owlnl:Filler>
      </owlnl:Slots>
    </owlnl:Microplan>
  </owlnl:EnglishMicroplans>
</owlnl:Property>
```

Figure 3: Specifying the order and micro-plans of a property.

depending on the context; the algorithm that we use to select the most appropriate referring expression will be discussed in section 2.2.3 below. The second slot contains the fixed string "was found", which is also marked up as being a past passive verb form whose plural is "were found"; this additional markup is needed when aggregating sentences to form longer sentences. The third slot contains the string "in", whereas the fourth one is to be filled by an accusative automatically selected referring expression corresponding to the *object* of the fact (the `filler` of the field).[10] This micro-plan may generate, for example, a phrase like "This aryballos was found in the Heraion of Delos". Note that although OWL properties (and other elements of OWL ontologies) can be associated with strings in multiple languages via `rdfs:label` tags, this mechanism is inadequate for specifying micro-plans; for example, it provides no principled way to indicate positions in the strings where referring expressions should be placed, or to annotate sub-strings with syntactic categories.

The `MicroplanName` element associates each micro-plan with a more easily readable name, compared to its identifier, which is used by the authoring facilities. The `Used` element shows whether or not the generator is allowed to use the micro-plan; during the authoring process, it is sometimes handy to temporarily disable a micro-plan, to force the generator to use alternative micro-plans for the same property. Finally, the `AggrAllowed` element indicates whether or not the sentence that will be produced by the micro-plan can be aggregated with other sentences to form longer periods; this feature will be discussed further in section 2.2.2.

The user modelling annotations assign an appropriateness score to each micro-plan per user type, as illustrated in figure 4; higher scores correspond to more appropriate micro-plans. This allows one, for example, to specify that a micro-plan that generates sentences like "This amphora depicts Miltiades" is less appropriate when interacting with children, compared to an alternative micro-plan that uses a more common verb and generates sentences like "This amphora shows Miltiades". As in M-PIRO, if more than one micro-plans have been specified for the same property, NaturalOWL prefers the most appropriate micro-plan the first time it needs to convey a fact with that property to a reader. The second time that a fact with the same property has to be conveyed to the same reader, NaturalOWL prefers the second most appropriate micro-plan, provided that its appropriateness is positive, and so on, until we are left with no unused micro-plan with positive appropriateness; at that point, the process restarts from using the first most appropriate micro-plan. Non-positive appropriateness scores are taken to indicate that the corresponding micro-plans should be avoided. Micro-plans with non-positive appropriateness scores are only used when there is no alternative micro-plan with a positive score; in that case, if there are several micro-plans with non-positive scores, only the micro-plan with

[10] Apart from verb forms, future versions of NaturalOWL will most likely also mark up strings of the micro-plans that contain prepositions, for compatibility with Edinburgh's METHODIUS generator. We also plan to move the specifications of the verb forms into the domain-dependent lexicon (see below), and use the identifiers of the corresponding lexicon entries in the micro-plans instead.

```
<owlnl:MicroplanApprop rdf:about="http://.../mpiro.owl#current-location-templ1-en">
  <owlnl:Approp rdf:parseType="Resource">
    <owlnl:forUserType rdf:resource="#Child"/>
    <owlnl:AppropValue>1</owlnl:AppropValue>
  </owlnl:Approp>
  <owlnl:Approp rdf:parseType="Resource">
    <owlnl:forUserType rdf:resource="#Adult"/>
    <owlnl:AppropValue>2</owlnl:AppropValue>
  </owlnl:Approp>
</owlnl:MicroplanApprop>
```

Figure 4: Specifying the appropriateness of a micro-plan.

the highest score is used.

2.2.2 Aggregation

Lexicalization, the previous sub-stage of micro-planning, produces a single sentence for each fact. These sentences are then aggregated into longer periods to improve the readability of the resulting text. The maximum number of sentences that can be aggregated into a single period is specified per user type in NaturalOWL's user modelling annotations (the relevant element is `MaxFactsPerSentence`, not shown in the examples of this paper); this allows, for example, specifying that shorter periods should be produced when interacting with children, and longer ones when interacting with adults. The `AggrAllowed` element of the micro-plan specifications (see figure 3) allows one to declare that the resulting sentence of a micro-plan should not be aggregated with other sentences; this is useful, for example, when generating sentences that contain long fixed strings, because aggregating these sentences is usually undesirable. We do not discuss here NaturalOWL's aggregation rules, because they are very similar to those of M-PIRO, for which extensive documentation is available [20].

2.2.3 Generating referring expressions

NaturalOWL currently employs a very simple algorithm to generate referring expressions, which uses a notion of *focus* borrowed from Centering Theory [12]. At the beginning of a description, the focus is taken to be the individual or class being described, and NaturalOWL refers to the focus by using its natural language name, if there is one, or by mentioning its class (e.g., "Bancroft Chardonnay is a wine.", "This is a vessel.", or "This vessel was found..."). In all subsequent references to the focus, a pronoun is used, until there is a shift of focus. A shift of focus happens when we encounter a sentence that expresses a fact at a different distance from the individual or class being described in the graph form of the ontology (see section 2.1.1). For example, in the following text, the focus is initially `exhibit101`, and the first two sentences express facts that are at distance 1 from `exhibit101`. The third sentence expresses a fact about the sculptor of `exhibit101`, i.e., a fact at distance 2 from `exhibit101`;

hence, there is a shift of focus, and the new focus is the sculptor (Nikolaou). Again, the system initially refers to the new focus (in the third sentence) by using its natural language name, and then by using a pronoun. In the penultimate sentence, there is another focus shift, and the focus returns to `exhibit101`. Once again, the system first refers to the new focus by its class, and then by a pronoun.

> `exhibit 101`: This is a vessel. It was sculpted by Nikolaou. Nikolaou was born in Athens. He was born in 1918 and he died in 1998. This vessel is now exhibited in the National Gallery. It is one of the best...

In the special case where a description of an individual is being generated and the focus has shifted from that individual to its class, when the focus returns to the individual being described NaturalOWL produces a demonstrative that includes the word "particular"; this seems to provide a cue that helps the reader realize more easily that the focus returns from the class to the original individual, as illustrated in the following example:

> `exhibit24`: This is an aryballos, a kind of vessel. An aryballos was a small spherical vessel with a narrow neck, in which the athletes kept the oil they spread their bodies with. This particular aryballos was found in the Heraion of Delos and it is currently exhibited in the Archaeological Museum of Delos.

More elaborate referring expression generation algorithms (e.g., [15, 19, 27]) may be added in future versions of NaturalOWL, although the current algorithm was sufficient in the application domains that we have considered so far.

To be able to generate expressions that refer to individuals or classes by their names (e.g., "Nikolaou", "Bancroft Chardonnay") or by appropriate nouns (e.g., "this vessel"), NaturalOWL uses a domain-dependent lexicon that contains natural language names and nouns (or more generally n-bars, noun phrases without their determiners) linked to individuals and classes of the ontology.[11] The entries of the lexicon are multilingual. For example, figure 5 shows the multilingual lexicon entry whose identifier is `vessel-NP`; this entry contains the English noun "vessel" and its corresponding Greek noun "αγγείο". The lexicon entry lists the various forms of the noun in the two languages, it provides information on gender etc. The lexicon entry is also linked to the class `vessel` via the `owlnl:hasNP` element; this licenses NaturalOWL to use that noun (in English or Greek) to refer to the particular class. NaturalOWL does not currently distinguish between natural language names and nouns (or n-bars). Hence, natural language names are stored in the domain-dependent lexicon in the same way as nouns, and they can be linked to classes or individuals again in the same way as nouns. When linking lexicon entries to individuals, rather than classes, the `owlnl:owlInstance` tag (not shown in the examples) is used instead of the `owlnl:owlClass` tag.

Moreover, the domain-dependent lexicon contains information on what we call multilingual personalized canned texts (MPCTs). Each MPCT is a pseudo-

[11] We have considered associating classes with WordNet (or EuroWordNet) synsets, but the domain ontologies we have experimented with contain highly technical concepts, which are not covered by WordNet. The Greek version of WordNet is also not freely available. Nevertheless, we plan to consider emerging standards for linguistic annotations [16], especially regarding the lexicon.

```
<owlnl:owlClass rdf:about="http://.../mpiro.owl#vessel">
  <owlnl:hasNP rdf:resource="#vessel-NP"/>
</owlnl:owlClass>

<owlnl:NP rdf:ID="vessel-NP">
    <owlnl:LanguagesNP rdf:parseType="Collection">
        <owlnl:GreekNP>
            <owlnl:countable>yes</owlnl:countable>
            <owlnl:num>singular</owlnl:num>
            <owlnl:gender>neuter</owlnl:gender>
            <owlnl:singular>
                <owlnl:singularForms>
                    <owlnl:nominative xml:lang="el">...</owlnl:nominative>
                    <owlnl:genitive xml:lang="el">...</owlnl:genitive>
                    <owlnl:accusative xml:lang="el">...</owlnl:accusative>
                </owlnl:singularForms>
            </owlnl:singular>
            <owlnl:plural>
                <owlnl:pluralForms>
                    <owlnl:nominativexml:lang="el">...</owlnl:nominative>
                    <owlnl:genitivexml:lang="el">...</owlnl:genitive>
                    <owlnl:accusativexml:lang="el">...</owlnl:accusative>
                </owlnl:pluralForms>
            </owlnl:plural>
        </owlnl:GreekNP>
        <owlnl:EnglishNP>
            <owlnl:countable>yes</owlnl:countable>
            <owlnl:num>singular</owlnl:num>
            <owlnl:gender>neuter</owlnl:gender>
            <owlnl:singular xml:lang="en">vessel</owlnl:singular>
            <owlnl:pluralxml:lang="en">vessels</owlnl:plural>
        </owlnl:EnglishNP>
    </owlnl:LanguagesNP>
</owlnl:NP>
```

Figure 5: A lexicon entry that is linked to a class.

```
<owlnl:owlInstance rdf:about="http://.../mpiro.owl#exhibit46str">
  <owlnl:hasCannedText rdf:resource="#exhibit46str-CT1"/>
  <owlnl:hasCannedText rdf:resource="#exhibit46str-CT2"/>
  <owlnl:hasCannedText rdf:resource="#exhibit46str-CT3"/>
</owlnl:owlInstance>

<owlnl:CannedText rdf:ID="exhibit8str-CT1">
  <owlnl:Val xml:lang="el">...</owlnl:Val>
  <owlnl:Val xml:lang="en">From 168 B.C., after the defeat of Perseus, Macedonia was
    controlled by the Romans. The Romans divided the region into fours districts, the
    so-called merides. This coin originates from the first merida of Macedonia.</owlnl:Val>
  <owlnl:forUserType rdf:resource="http://www.owlnl.com/NLG/UserModelling#Child"/>
  <owlnl:FillerAggrAllowed>false</owlnl:FillerAggrAllowed>
  <owlnl:FOCUS_LOST>false</owlnl:FOCUS_LOST>
</owlnl:CannedText>

<owlnl:CannedText rdf:ID="exhibit8str-CT2">
  <owlnl:Val xml:lang="el">...</owlnl:Val>
  <owlnl:Val xml:lang="en">From 168 B.C., after the defeat of Perseus, Macedonia devolved
    to Roman control. The Romans divided the region into fours districts, the so-called
    merides. This coin originates from the first merida of Macedonia.</owlnl:Val>
  <owlnl:forUserType rdf:resource="http://www.owlnl.com/NLG/UserModelling#Adult"/>
  <owlnl:FillerAggrAllowed>false</owlnl:FillerAggrAllowed>
  <owlnl:FOCUS_LOST>false</owlnl:FOCUS_LOST>
</owlnl:CannedText>

<owlnl:CannedText rdf:ID="exhibit8str-CT3">
  <owlnl:Val xml:lang="el">...</owlnl:Val>
  <owlnl:Val xml:lang="en">This coin originates from the first Roman merida of
  Macedonia.</owlnl:Val>
  <owlnl:forUserType rdf:resource="http://www.owlnl.com/NLG/UserModelling#ExpertAdult"/>
  <owlnl:FillerAggrAllowed>true</owlnl:FillerAggrAllowed>
  <owlnl:FOCUS_LOST>false</owlnl:FOCUS_LOST>
</owlnl:CannedText>
```

Figure 6: A multilingual personalized canned text.

entity of the ontology, standing for a canned text that can take different forms, depending on the type of the reader and the target natural language. Figure 6 shows an MPCT whose identifier is `exhibit46str`. According to the specifications of figure 6, `exhibit46str` can take three different forms, depending on the type of the reader; the three forms are identified as `exhibit46str-CT1`, `exhibit46str-CT2`, and `exhibit46str-CT2`. The first form, `exhibit46str-CT1`, is to be used when interacting with children (user type `Child`), whereas the other two forms are to be used when interacting with expert and non-expert adults (user types `Adult` and `ExpertAdult`), respectively. Each form is provided in both English and Greek. Notice that the children's form of the text uses slightly simpler expressions compared to the adults' form ("was controlled by the Romans" instead of "devolved to Roman control"), and that the experts' form assumes that experts already know what the Roman merides were.

The MPCT `exhibit46str` is intended to contain a canned narrative about

exhibit46. That is, exhibit46 would be associated with exhibit46str via a fact of the form <exhibit46, exhibit-story, exhibit46str>. A micro-plan for exhibit-story would typically contain only one slot, which would be filled by the appropriate (in terms of language and user type) form of the text of the MPCT in the *object* role the fact (here, exhibit46str). In other cases, however, a micro-plan may contain additional slots to generate, for example, a sentence like "It depicts a group of Nymphs dancing in a forest." from a fact of the form <exhibit201, depicts-text, exhibit201str>, where the pronoun fills the first slot of the micro-plan, it refers to exhibit201, and it has been generated automatically; "depicts" is a fixed string that fills the second slot of the micro-plan; and the remainder of the sentence fills the third slot of the micro-plan and originates from MPCT exhibit201str, which is the *object* of the fact.

Unlike ordinary strings, MPCTs allow us to specify different versions of a canned text per user type and natural language, and to refer to all of the versions collectively via the MPCT's identifier. Furthermore, MPCTs allow us to specify per user type whether or not a sentence that contains the corresponding form of the MPCT's canned text can be aggregated or not (tag FillerAggrAllowed); if the canned text is long, aggregating it is usually undesirable. MPCTs also allow us to specify (tag FOCUS_LOST) per user type whether or not including the canned text in a sentence causes a focus shift, i.e., whether or not the *subject* of the corresponding fact (in our example, the exhibit46 of fact <exhibit46, exhibit-story, exhibit46str>) is no longer the focus at the end of the canned text; if it is no longer the focus and we need to refer to it in the next sentence after the canned text, we have to use its name or a demonstrative that includes its class, instead of using a pronoun.

2.3 Surface realization

In more fully fledged language generators, the sentence specifications that are produced at the end of the micro-planning stage may be underspecified; for example, they may not specify the exact order of the syntactic constituents, the genders of the adjectives etc. During the surface realization stage, grammars are often used to fill in the missing information (e.g., enforce gender agreement) and produce the final text; ILEX and M-PIRO used Systemic Grammars for this purpose [13]. In NaturalOWL, there is no need for grammars, because the generation of texts is template-based, and the templates will have already produced an almost final version of the text at the end of micro-planning. Hence, the only purpose of surface realization in NaturalOWL is to add presentation markup (e.g., HTML tags) or markup for speech synthesizers, if necessary. Semantic markup could also be added easily during this stage, as discussed in section 1, although this is currently not implemented.

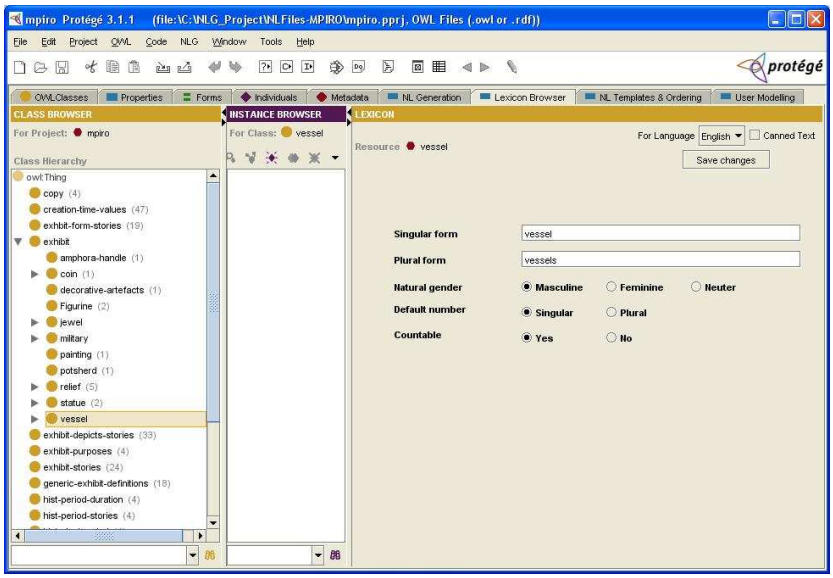

Figure 7: Creating a lexicon entry with the Protégé plug-in.

3 Source authoring

As already mentioned in section 1, NaturalOWL is supported by both M-PIRO's authoring tool, which has been extended to support NaturalOWL, and an authoring plug-in for Protégé; both provide very similar functionality. M-PIRO's original authoring tool has been documented extensively [2], and its new version, called ELEON, is described elsewhere [7]. Hence, here we only present briefly NaturalOWL's plug-in for Protégé. The plug-in adds the following widgets to Protégé:

Lexicon widget: This is used to create entries of the domain-dependent lexicon and link them to classes or instances of the ontology. Figure 7 illustrates the use of this widget; the author is editing the English part of a lexicon entry that is linked to the ontology class `vessel`.

Templates widget: This is used to create micro-plans. Its use is illustrated in figure 8, where the author is defining an English micro-plan for the property `made-of`. The micro-plan can generate sentences like "This statue is made of bronze".

Ordering widget: This is used to specify a partial order of the ontology's properties, which is in turn used to order the facts of the resulting texts, as explained in section 2.1.2. The use of this widget is illustrated in figure 9.

User modelling widget: Here the user can define the user types and their

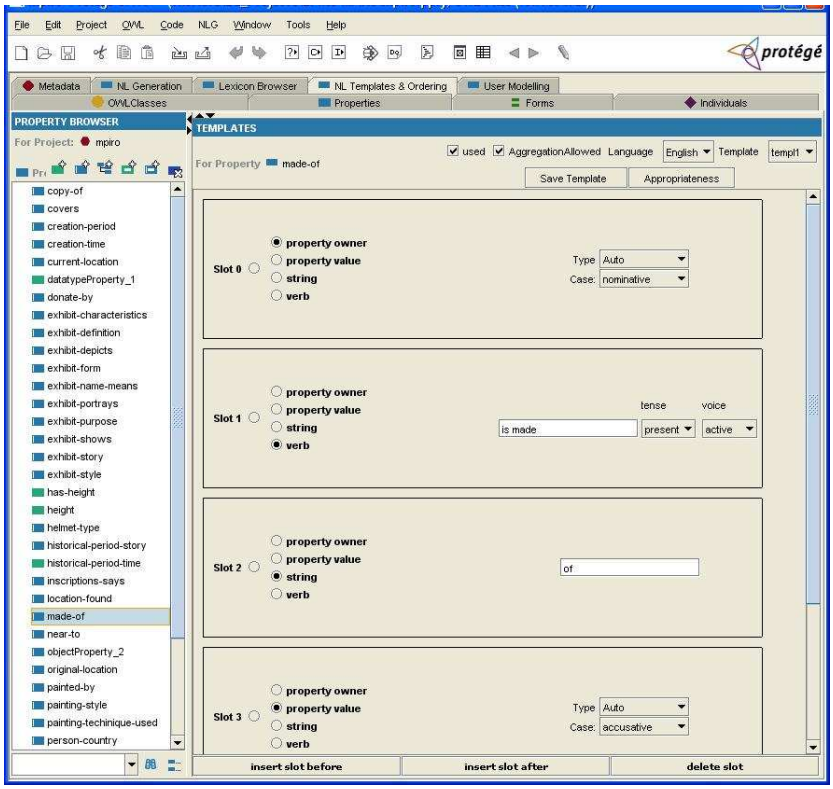

Figure 8: Defining a micro-plan with the Protégé plug-in.

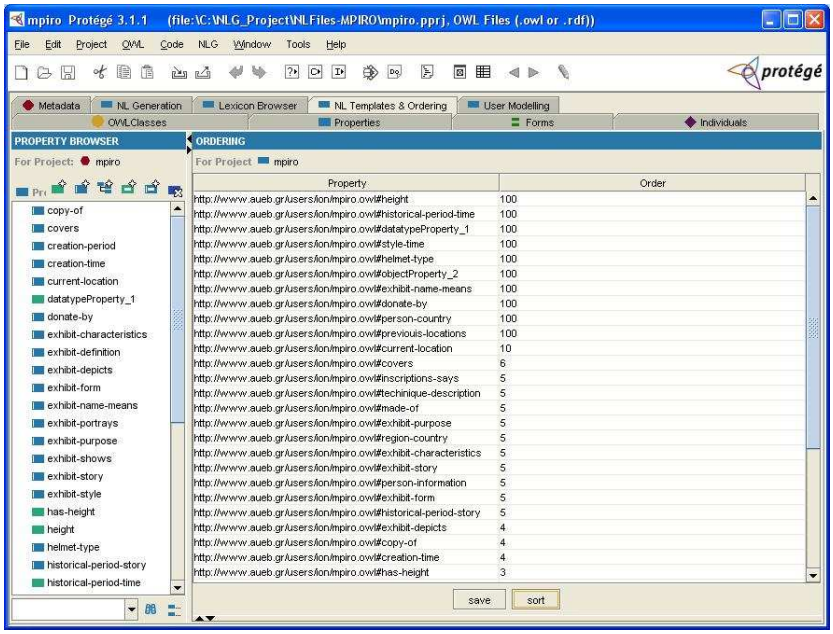

Figure 9: Setting the order of properties with the Protégé plug-in.

parameters (e.g., the maximum facts per aggregated period), as shown in figure 10. The same widget is used to specify the interest and repetitions of each property per user type (section 2.1.1); this is illustrated in figure 11.

NL generation widget: This is used to previewing the resulting texts, as illustrated in figure 12.

4 Conclusions and future work

We presented NaturalOWL, an open-source natural language generator for OWL DL ontologies that currently supports English and Greek. The system is intended to demonstrate the benefits of adopting NLG techniques on the Semantic Web, and to contribute towards a discussion on standards for annotating OWL ontologies with linguistic resources and user modelling information. NaturalOWL was partly developed and is being extended in project Xenios, where it is used by mobile robots acting as museum guides. Future work, in the context of project INDIGO, will explore how NaturalOWL can be embedded in a spoken dialogue system, and how the generated texts can be marked up with information showing where appropriate gestures and facial expressions have to be made.[12] The latter can be exploited in robots with synthetic heads and avatars.

[12] Consult `http://www.ics.forth.gr/indigo/`.

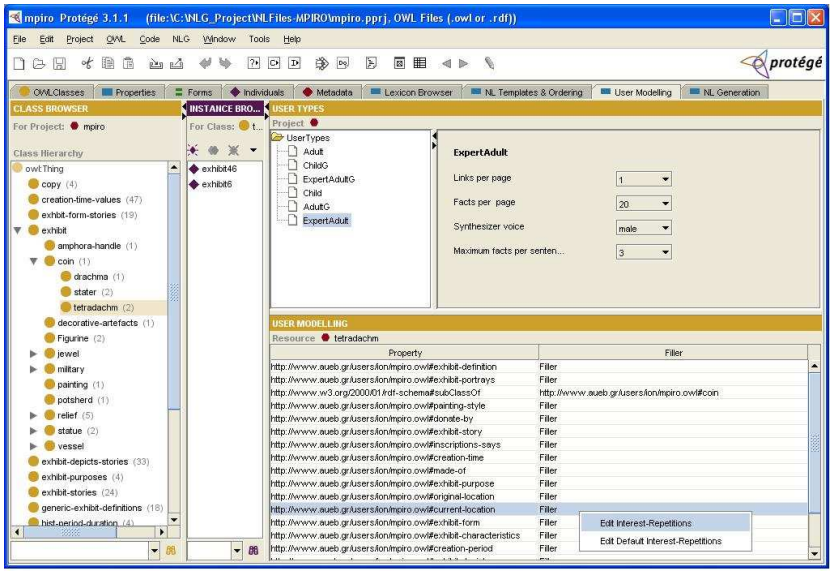

Figure 10: Specifying user types and their parameters with the Protégé plug-in.

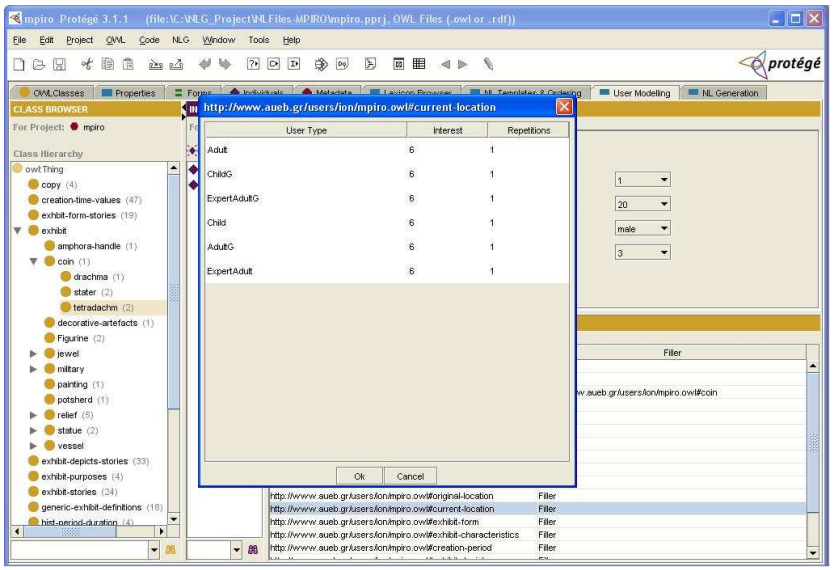

Figure 11: Specifying the interest and repetitions of properties with the Protégé plug-in.

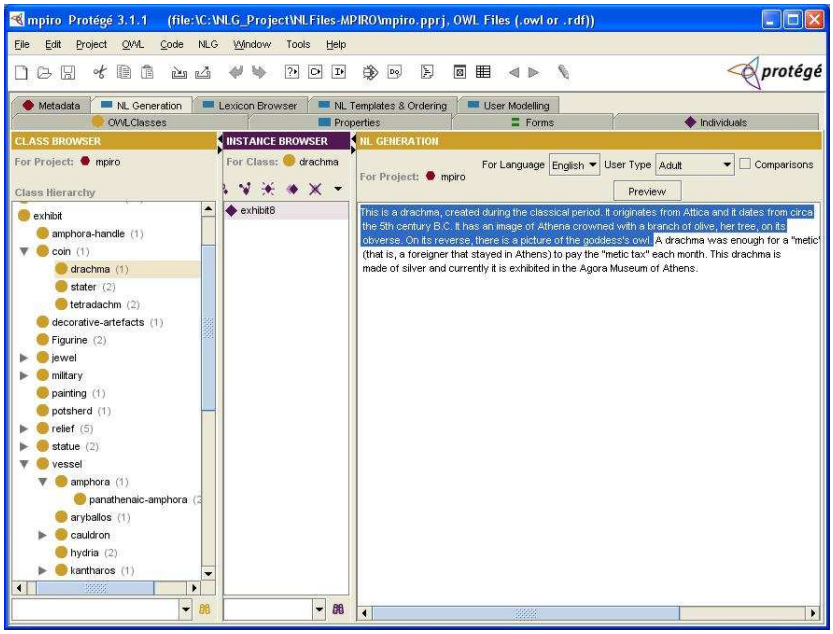

Figure 12: Previewing generated texts in the Protégé plug-in.

References

[1] I. Androutsopoulos, S. Kallonis, and V. Karkaletsis. Exploiting OWL ontologies in the multilingual generation of object descriptions. In *Proceedings of the 10th European Workshop on Natural Language Generation*, pages 150–155, Aberdeen, UK, 2005.

[2] I. Androutsopoulos, J. Oberlander, and V. Karkaletsis. Source authoring for multilingual generation of personalised object descriptions. *Natural Language Engineering*, 2007. In press, available on-line.

[3] G. Antoniou and F. van Harmelen. *A Semantic Web Primer*. MIT Press, 2004.

[4] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *The Description Logic handbook: theory, implementation and application*. Cambridge University Press, 2002.

[5] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, May 2001.

[6] A. Bernstein and E. Kaufmann. GINO – a guided input natural language ontology editor. In *Proceedings of the 5th International Semantic Web Conference*, pages 144–157, Athens, GA, 2006.

[7] D. Bilidas, M. Theologou, and V. Karkaletsis. Enriching OWL ontologies with linguistic and user-related annotations: the ELEON system. Technical report, Institute of Informatics and Telecommunications, National Centre for Scientific Research "Demokritos", Greece, 2007.

[8] K. Bontcheva. Open-source tools for creation, maintenance, and storage of lexical resources for language generation from ontologies. In *Proceedings of the 4th Conference on Language Resources and Evaluation*, Lisbon, Portugal, 2004.

[9] K. Bontcheva and H. Cunningham. The Semantic Web: a new opportunity & challenge for Human Language Technology. In *Proceedings of the Workshop on Human Language Technology for the Semantic Web and Web Services, 2nd International Semantic Web Conference*, Sanibel Island, FL, 2003.

[10] K. Bontcheva and Y. Wilks. Automatic report generation from ontologies: the MIAKT approach. In *Proceedings of the 9th International Conference on Applications of Natural Language to Information Systems*, pages 324–335, Manchester, UK, 2004.

[11] D. Galanis and I. Androutsopoulos. Generating multilingual descriptions from linguistically annotated owl ontologies: the naturalowl system. In *Proceedings of the 11th European Workshop on Natural Language Generation*, Schloss Dagstuhl, Germany, 2007.

[12] B.J. Grosz, A.K. Joshi, and S. Weinstein. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225, 1995.

[13] M. Halliday. *Introduction to Functional Grammar*. Edward Arnold, 2nd edition edition, 1994.

[14] D. Hewlett, A. Kalyanpur, V. Kolovski, and C. Halaschek-Wiener. Effective NL paraphrasing of ontologies on the Semantic Web. In *Proceedings of the Workshop on End-User Semantic Web Interaction, 4th International Semantic Web Conference*, Galway, Ireland, 2005.

[15] H. Horacek. A best-first search algorithm for generating referring expressions. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–106, Budapest, Hungary, 2003.

[16] N. Ide and L. Romary. International standard for a linguistic annotation framework. *Natural Language Engineering*, 10(3/4):211–225, 2004.

[17] A. Isard, J. Oberlander, I. Androutsopoulos, and C. Matheson. Speaking the users' languages. *IEEE Intelligent Systems*, 18(1):40–45, 2003.

[18] B. Katz, J. Lin, and D. Quan. Natural language annotations for the Semantic Web. In *Proceedings of the International Conference on Ontologies, Databases, and Application of Semantics*, University of California, Irvine, 2002.

[19] E. Krahmer, S. van Erk, and A. Verleg. Graph-based generation of referring expressions. *Computational Linguistics*, 2003.

[20] A. Melengoglou. Multilingual aggregation in the M-PIRO system. Master's thesis, School of Informatics, University of Edinburgh, UK, 2002.

[21] C. Mellish and X. Sun. Natural language directed inference in the presentation of ontologies. In *Proceedings of the 10th European Workshop on Natural Language Generation*, Aberdeen, UK, 2005.

[22] C. Mellish and X. Sun. The Semantic Web as a linguistic resource: opportunities for Natural Language Generation. *Knowledge Based Systems*, 19:298–303, 2006.

[23] M. Milosavljevic. *The Automatic Generation of Comparison in Descriptions of Entities*. PhD thesis, Department of Computing, Macquarie University, Australia, 1999.

[24] M. O'Donnell, C. Mellish, J. Oberlander, and A. Knott. ILEX: an architecture for a dynamic hypertext generation system. *Natural Language Engineering*, 7(3):225–250, 2001.

[25] E. Reiter and R. Dale. *Building Natural Language Generation Systems*. Cambridge University Press, 2000.

[26] X. Sun and C. Mellish. Domain independent sentence generation from RDF representations for the Semantic Web. In *Proceedings of the Combined Workshop on Language-Enabled Educational Technology and Development and Evaluation of Robust Spoken Dialogue Systems, European Conference on AI*, Riva del Garda, Italy, 2006.

[27] K. van Deemter. Generating referring expressions that involve gradable properties. *Computational Linguistics*, 32(2), 2006.

[28] G. Wilcock. Talking OWLS: towards an ontology verbalizer. In *Proceedings of the Workshop on Human Language Technology for the Semantic Web, 2nd International Semantic Web Conference*, pages 109–112, Sanibel Island, FL., 2003.