

Enriching OWL Ontologies with Linguistic and User-related Annotations: the ELEON system

D. Bilidas, M. Theologou, V. Karkaletsis

*Institute of Informatics and Telecommunications
National Centre for Scientific Research "Demokritos"
GR-153 10 Aghia Paraskevi Attikis, Greece
{dbilid, vangelis}@iit.demokritos.gr*

Abstract

This paper introduces ELEON, an editor that allows the enrichment of OWL ontologies with linguistic and user-related annotations. The enriched ontologies are used by natural language generation (NLG) engines to generate textual descriptions of the objects represented in the ontologies in the selected language and according to user's model. ELEON provides a well-defined interface that can be used by different NLG engines. The paper presents the relevant functionalities of ELEON, describes the provided interface to NLG engines and discusses the advantages of exploiting such enriched ontologies in NLG.

1. Introduction

Natural Language Generation (NLG) systems have found many applications, from generating software manuals from a single, symbolic source, like DRAFTER [5] and AGILE [8], to generating drug descriptions [9], and museum exhibit descriptions [1, 2]. All these systems can generate in multiple languages from a single, language-neutral symbolic source of information. A major problem is the difficulty of obtaining source symbolic information in forms compatible with the requirements of the language generators. This issue has mainly been addressed so far by developing authoring tools that help in the creation of source information and domain-dependent linguistic resources. Such tools were developed, for example, in GIST [7], DRAFTER [5], WYSIWYM [6], and M-PIRO [1]. M-PIRO's authoring environment allowed curators to fully define the domain ontology, create domain-dependent linguistic resources and link them to the ontology, and adjust parameters related to user modeling.

On the other hand, in recent years, considerable effort has been invested in the Semantic Web, aiming at the development of mechanisms that will allow computer applications to reason more easily about the semantics of the Web resources. A major target is the development of standard representation formalisms that will allow

ontologies to be published on the Web and be shared by different computer applications. The emerging standard for specifying ontologies is OWL¹, an extension of RDF². In NLG systems that describe objects, pre-existing OWL ontologies can provide much of the required source information, reducing the authoring effort and providing a common standard representation to generate from. This motivated us to extend M-PIRO authoring tool to support OWL. An initial version of the extended authoring tool is presented in [4]. The editor presented here (ELEON) extends this tool to enhance OWL support. It also provides a well-defined interface that can be used by different NLG engines.

Section 2 introduces the authoring functionalities of ELEON focusing on the editing and enriching of OWL ontologies. Section 3 presents the provided interface for NLG engines and discusses the advantages that the use of OWL ontologies offers to NLG. Section 4 concludes summarising our plans for future research.

2. ELEON authoring tool

M-PIRO project [1] developed the Exprimo NLG engine using the ILEX system [2] as a starting point. Exprimo generated personalized multilingual textual descriptions and supported three languages, English, Italian and Greek. M-PIRO developed an authoring tool to facilitate the creation of source information and domain-dependent linguistic resources. ELEON (Editor for Linguistically Enriched Ontologies) extends the authoring tool developed after the completion of the M-PIRO project [4]. Recently, it has been enhanced to cooperate with NaturalOWL [3] while it can also collaborate with the Methodius NLG engine³, the enhanced version of Exprimo. In general, it can easily communicate with various NLG engines due to its well-specified interface

¹ <http://www.w3.org/TR/>

² www.w3.org/RDF/

³ <http://www.ltg.ed.ac.uk/methodius/>

2.1. ELEON Basics

ELEON allows its authors – i.e. persons that have domain expertise but no technological expertise - to modify all the domain-dependent resources and to control the user modeling parameters. The domain-dependent resources contain not only the ontology that describes the knowledge base of the domain but also the language resources that are closely related to the domain in question.

ELEON generates texts from the ontology that encodes domain knowledge in the form of concepts, concepts' instances (entity types and entities respectively in ELEON's terminology), concepts' properties and relations between concepts. Entity types are organized in a taxonomy. Properties and relationships are expressed using fields. At any entity type, it is possible to introduce new fields, which then become available at all the entities that belong to that type and its subtypes. For each field of the ontology and each language, the authors have to specify at least one clause plan (micro-plan) that specifies how the field can be expressed in that language. The author specifies the clause to be generated in abstract terms, by specifying, for example, the verb to be used, the voice and tense of the resulting clause, etc. Furthermore, the author must insert nouns in the lexicon for each language and for each entity type and entity that he wants to generate text for.

ELEON also allows authors to specify the types of end-users (e.g., *'expert'*, *'average-adult'*, *'child'*, etc). For each user type, several parameters such as the desired length of the texts and paragraphs are specified. The capability of the aggregation of the clauses to form longer sentences is also enabled. In addition, authors can specify how interesting each field is for each user type, allowing the system to tailor the content of the descriptions to the users' interests. Finally, for each end-user, its interaction history with the system is stored allowing the generation of comparisons to previously seen objects.

2.2. Editing & Enriching OWL ontologies

Following the completion of M-PIRO, the authoring tool was enhanced to partially support the import and export of OWL ontologies. Building on this work and in order to bring NLG closer to the emerging Semantic Web, special effort was spent in ELEON to enhance OWL support. ELEON provides an ontology editor which enables the creation and maintenance of OWL ontologies. The OWL editor fulfils all the requirements of OWL Lite (the first of the three sub-languages of OWL) while it provides functionality that is almost compatible with OWL DL (the 2nd sub-language of OWL). Furthermore, ELEON provides authors with the ability to check the ontology's consistency.

ELEON's ontological assumptions are very similar to those of OWL. ELEON's entity type defines a group of entities and maps to *'Class'* OWL feature. Entity types can be organized to a taxonomy by defining for each entity type its subtypes (maps to *'rdfs:subClassOf'* OWL feature). Significant effort spent to support multiple inheritance at the level of the entity types i.e. an entity type may have more than one parent and subsequently an entity may belong to more than one entity type. Two entity types may be stated to be equivalent (maps to *'equivalentClass'* OWL feature). Equivalent entity types have the same entities. Relationships are expressed using fields (map to *'rdf:Property'* OWL feature). Relationship hierarchies may be created by stating that a relation is a sub-relation or super-relation of one or more other relations (map to *'rdfs:subPropertyOf'* OWL feature). Two relations may be stated to be equivalent relations (map to *'equivalentProperty'* OWL feature). ELEON's entities map to *'Individual'* OWL feature. Each entity may belong to more than one entity types and inherits all the relationships and properties defined to those entity types.

The author may also provide information concerning the characteristics of the relations and their values. One relation may be stated to be the inverse of another relation (map to *'inverseOf'* OWL feature). The author may also state that a relation is transitive (map to *'TransitiveProperty'* OWL feature) or symmetric (map to *'SymmetricProperty'* OWL feature). A relation may also be stated to have a unique value, i.e. to have either no more than one value or no value for each entity (map to *'FunctionalProperty'* OWL feature). Furthermore, relations may be stated to be inverse functional (map to *'InverseFunctionalProperty'* OWL feature).

ELEON provides restrictions to be placed on how relations can be used by the entities of an entity type. Two types of restrictions are supported. The first type of the restrictions (value restrictions) limits which values to be used by a relation. Thus, the author may state that all values of a relation belong to an entity type (map to *'allValuesFrom'* OWL feature). Authors may also state that at least one value of a relation is of certain type (map to *'someValuesFrom'* OWL feature) or a relation is required to have a certain entity as a value (map to *'hasValue'* OWL feature). The second type of the restrictions (cardinality restrictions) limits how many values can be used by a relation. Thus, the author may state that a relation is required to have at least N values for all entities (map to *'minCardinality'* OWL feature), to have no more than N values (map to *'maxCardinality'* OWL feature) or to have exactly N values (map to *'Cardinality'* OWL feature).

3. Technical implementation

In the current version, the domain ontology is exported in OWL, whereas the linguistic and user modelling elements are exported in RDF format. Each element in the RDF file is linked to a class, individual or property in the OWL ontology. By doing this, the author creates an ontology annotated and enriched with linguistic and user modelling elements.

3.1 Interfaces with NLG engines

The exported OWL and RDF files have convenient structure for easy communication with any NLG engine. Currently, they are taken as input from the NaturalOWL NLG Engine which is invoked by ELEON enabling the user to have a preview of the generated textual descriptions. ELEON's user has the ability to choose a specific entity from the ontology and preview the text produced for the entity in question. When the user makes some modifications, either in the ontology or in the linguistic and user modelling resources, ELEON exports these elements in the background, invokes the NaturalOWL for the specified language (English or Greek) and requires from the engine to generate the text for the pre-selected entity and for the pre-defined user type. In the context of INDIGO project⁴ and via the same interface, i.e., the OWL and RDF files, a second NLG engine, Methodius, is being embodied in the authoring tool. The user will have the ability to select one of these NLG engines and compare the texts generated by them.

Concerning the linguistic information, each noun in the RDF file corresponds to a class or individual in the OWL ontology. Likewise, the association of each property with the corresponding micro-plans and the form of these micro-plans are defined in specific entries in the RDF files.

3.2 Advantages from the usage of OWL in NLG

The use of OWL and RDF to represent all the information required by an NLG engine enables the re-use of this information by other applications through the semantic web. Furthermore, as OWL language is widely used, there are published ontologies for many different domains and subjects. As a result, if someone wants to create a new NLG application for a specific domain, most probably, he will find an existing relevant OWL ontology, import it in ELEON, modify it, if needed, and enrich it with linguistic and user modeling elements. Furthermore, the agreement on a common format for this information, can lead to the usage of more than one NLG engines. Apart from these advantages which are related to the fact that OWL language has become a standard in semantic

web, there are many other advantages deriving from the expressiveness of OWL language.

For example, in Figure 1, the generated text conveys information based on comparisons that the NLG engine has made during the generation process. The engine used in this example is NaturalOWL. In this case, the engine exploits the allocation of entities into classes in order to compare entities of the same class based on the common properties they have. As a result, the generated text can contain information such as *"It is the only amphora of the collection which was decorated with the red-figure technique"*.

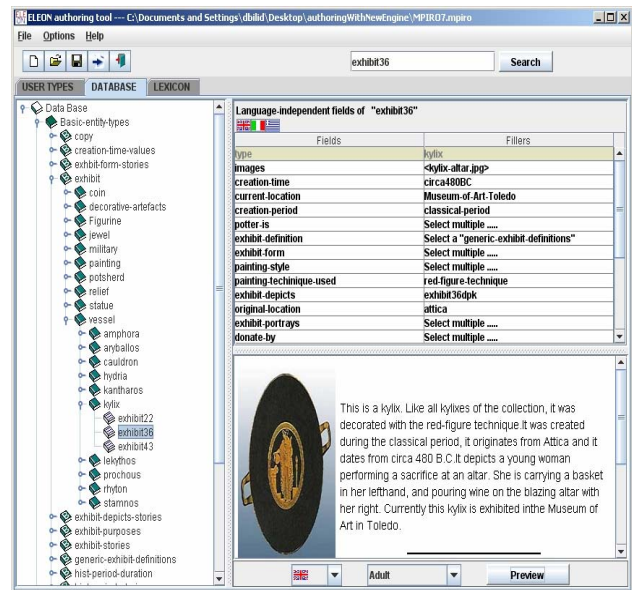


Figure 1. ELEON ontology

Another example is related to the multiple inheritance capability that OWL provides. This capability gives the opportunity to the NLG engine to produce texts like *"Xenokrates of Sicyon was a sculptor and writer"*. In this example, entity *"Xenokrates"* belongs both to classes *"Sculptor"* and *"Writer"*.

There are more features of OWL that can be exploited from an NLG engine. For example the sub-property declaration could be expressed using some linguistic elements which indicate particularization (for example in English: in particular, specifically, particularly, etc.). Thus, an author can declare the property *"isEmployee"* and its sub-property *"isAccountant"*. Then, the produced text for an accountant X of a company Y could be *"X is an employee of Y and specifically he is an accountant"*. Likewise, the declaration of a transitive property provides authors with using expressions such as *"subsequently"*, *"as a result"* etc.

Finally, OWL has some special properties, called annotation properties, which are used to annotate classes, properties, individuals and ontology headers. Two of the predefined annotation properties of OWL, namely

⁴ <http://www.ics.forth.gr/indigo>

rdfs:label and *rdfs:comment*, can be very useful in the description of an object, as they can be used to hold predefined canned text, which otherwise is difficult to be produced from an NLG engine using regular properties and clause-plans.

3.3 Usage of an inference mechanism

ELEON checks on-line its users' actions and displays error or warning messages in case of contradictions. These checks concern violation of cardinality restrictions, validation of the domain and range of a class, etc. Nevertheless, the increased complexity of the ontological models enforced us to provide authors with the capability to validate the consistency of the ontological model through a reasoning mechanism. ELEON is enhanced with an interface to the reasoning engine, RacerPro⁵. When the author selects to connect with the RacerPro server, the ontology is being exported in the background and sent to RacerPro. Then the author can inquire the RacerPro to check the consistency of the ontology, retrieve individuals of a particular concept, etc.

4. Future work

ELEON is used in the context of the R&D project XENIOS, by the curators of the Foundation of the Hellenic World (FHW), for the generation of descriptions of the areas in FHW buildings, the exhibition programmes and particular exhibits and technologies. The annotations added to the ontology by ELEON users concern the domain specific lexicons and grammars (in Greek and English) and the user stereotypes specified for the XENIOS application (adults, children, groups of visitors).

ELEON is currently being enhanced to become fully compliant with OWL Lite and OWL DL. Following these enhancements, ELEON will be extended in order to cover the requirements issued by another application in the recently started R&D project INDIGO. This application concerns a robotic guide interacting with museum visitors where the dialogue course will also be affected by the robotic personality. This requires the enrichment of the domain ontology with annotations related to the robotic personality.

Finally, it must be noted that following the completion of the XENIOS project, ELEON will be provided to the research community as an open source tool providing a detailed documentation of the interface for NLG engines.

Acknowledgements

This work was partially supported by the R&D project XENIOS which is co-funded by the Greek General

Secretariat of Research and Technology (GSRT). We would like to thank Ms. S. Valassopoulou and Ms Ch. Makri from the Foundation of the Hellenic World (FHW), for their suggestions.

References

- [1] Isard A., J. Oberlander, I. Androutsopoulos and C.Matheson, "Speaking the Users' Languages", *IEEE Intelligent Systems*, 18(1), 2003, pp. 40-45.
- [2] O'Donnell M., C. Mellish, J. Oberlander, and A. Knott, "ILEX: an architecture for a dynamic hypertext generation system", *Natural Language Engineering*, 7(3), 2001, pp. 225-250.
- [3] Galanis D. and Androutsopoulos I., "Generating multilingual descriptions from linguistically annotated OWL ontologies: the NaturalOWL system", *Proceedings of the 11th European Workshop on Natural Language Generation (ENLG'07)*, Dagstuhl, June 2007.
- [4] I. Androutsopoulos, S. Kallonis, V. Karkaletsis, "Exploiting OWL Ontologies in the Multilingual Generation of Object Descriptions", *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG-05)*, Aberdeen, August 2005.
- [5] A. Hartley and C. Paris. Multilingual document production – from support for translating to support for authoring. *Machine Translation*, 1997, 12(1-2):109-129.
- [6] K. Van Deemter and R. Power. High-level authoring of illustrated documents. *Natural Language Engineering*, 2003, 9(2):101-126
- [7] R. Power and N. Cavallotto. Multilingual generation of administrative forms. *Proceedings of the 8th International Workshop on Natural Language Generation*, Herstmonceux Castle, U.K., 1996, pp 17-19.
- [8] Hartley, A., Scott, D., Bateman, J. and Dochev, D. AGILE – A system for multilingual generation of technical instructions. *Proceedings of the 8th Machine Translation Summit*, Santiago de Compostella, Spain. 2001, pp. 145-150.
- [9] Brun, C., Dytelman, M. and Lux, V. Document structure and multilingual authoring. *Proceedings of the 1st International Natural Language Generation Conference*, Mitzpe Ramon, Israel. 2000, pp. 24-31.

⁵ <http://www.racer-systems.com>