An intelligent authoring environment for abstract semantic representations of cultural object descriptions

Stasinos Konstantopoulos and **Vangelis Karkaletsis** and **Dimitris Bilidas**
Institute of Informatics & Telecommunications
NCSR 'Demokritos', Greece
{konstant,vangelis,dbilid}@iit.demokritos.gr

Abstract

In this paper we describe an authoring environment for the creation of cultural-domain ontologies and the associated linguistic and profile annotations, for dynamically generating adaptable natural-language descriptions of the cultural objects in the ontology. Adaptation is achieved at the expense of considerable authoring effort, since it relies on providing numerical parameters for each ontological entity. To assist the authoring process, we provide an intelligent authoring back-end that completes manually authored models by inferring missing values. This intelligent authoring support facility, combined with immediate previews, can considerably reduce the effort required to create a fully functional model as the author can iterate through cycles of providing information, previewing the generated text, and only elaborating the model where the text is unsatisfactory.

1 Introduction

Cultural heritage organizations create and maintain repositories of (digital representations of) artifacts, including extensive semantic knowledge and meta-data about the cultural objects in the collection. Such semantic repositories are typically seen as an opportunity to catalogue, index, and classify the cultural content, for the purpose of providing semantic *searching* and *browsing* facilities to professional users as well as to the general public.

In this article we discuss another unique opportunity that cultural heritage repositories offer: the opportunity to automatically generate adaptable and customizable *textual descriptions* of the cultural objects for a variety of audiences and purposes.

More specifically, we present ELEON, an authoring environment for creating abstract conceptual representations of cultural heritage object descriptions, as well as the linguistic and profiling models necessary to realize those into concrete natural-language descriptions exploiting natural language generation technology. The advantages of this approach, as opposed to directly authoring natural language descriptions, are manifold:

- Abstract descriptions constitute machine-readable and reusable models of the cultural heritage collection. Besides deriving natural language descriptions, such models can be used for the semantic indexing and searching of the collection. This can also be seen from the reverse perspective: the natural language descriptions can be derived from existing conceptual models created for the purpose of semantic indexing and searching.

- The conceptual descriptions are realized using domain-independent, reusable linguistic models. By clearly separating the conceptual and linguistic models, the same conceptual descriptions can be realized in different languages and the same linguistic models can be used to realize descriptions of different collections.

- The dynamic generation of the description is driven by profiles that personalize the descriptions for different audiences, but also adapt them to different contexts and situations.

ELEON provides great flexibility in finely parametrizing how the generated descriptions are adapted to different audiences and situations. Furthermore, the authoring environment is backed by Artificial Intelligence tools that assist the author

Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education –LaTeCH – SHELT&R 2009, pages 10–17,
Athens, Greece, 30 March 2009. ©2009 Association for Computational Linguistics

by automatically inferring missing profile parameters, alleviating the burden of explicitly providing all necessary details for large numbers of objects.

Although the system can be used in a variety of domains and Human-Computer interaction applications, it is particularly pertinent to cultural heritage content, which is interesting for wide ranges of age groups, levels of expertise, cultural and educational backgrounds, situations and contexts, emphasising the need for personalized and custom-tailored text.

In the rest of this article we first set the background by describing the authoring environment and particularly the way in which it can be used to create the conceptual model of the collection and populate it with data (Section 2) and then proceed to to describe how adaptation parameters are represented and used by human-computer interaction systems (Section 3). We then focus on the main contribution of this paper by describing the intelligence mechanism behind the environment (Section 4), discuss related work (Section 5), and conclude (Section 6).

2 Authoring Domain Ontologies

ELEON enables its authors—i.e., persons that have domain expertise but no technological expertise—to create a new application domain, defining the ontology of the new domain, as well as the corresponding language resources and the profiling models. All these elements are used by a natural language generation (NLG) engine in order to derive natural language descriptions from the conceptual representations in the ontology. The environment also enables authors to generate text previews using the NLG engine in order to examine the effect of their updates to the domain ontology, the language resources and the profiling parameters.

Concerning the language resources, these affect the content and the surface form of the derived texts, leading to more varied texts and contain entries for nouns and verbs for each supported language. With regard to the profiling parameters, these are used by ELEON to adapt the generated descriptions to the users' preferences and knowledge.

ELEON ontologies encode domain knowledge in the form of concepts, instances of concepts, (entity types and entities respectively in ELEON terminology), and relations between concepts and instances.

Figure 1 illustrates part of such an ontology that encodes knowledge about the ancient Agora of Athens. This ontology is used in the INDIGO project,[1] implementing a use case where the system guides visitors through an exhibition on the ancient Agora of Athens, introducing the buildings to them before they attend a virtual 3D tour of the Agora hosted at the Foundation of the Hellenic World. The examples used in this paper are drawn from this domain.

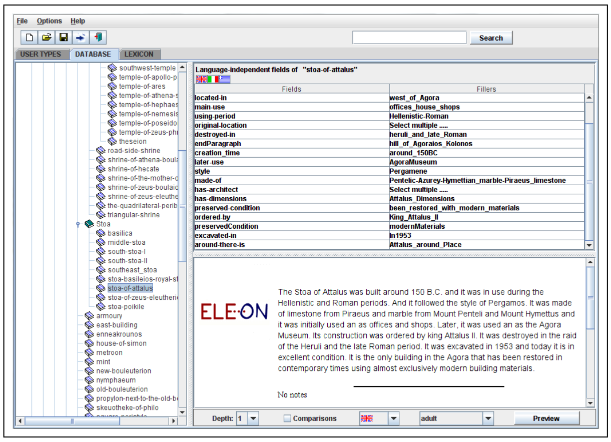
In the example of Figure 1, 'stoa-of-attalus', is an instance of the entity type Stoa, a subtype of Building which is a sub-type of ArchitecturalConstruction, a sub-type of PhysicalObject. Properties and relationships are expressed using fields. For any entity type, it is possible to introduce new fields which then become available to all the entities that belong to that type and its subtypes. In Figure 1, the field locatedIn is introduced at the ArchitecturalConstruction entity type and is defined as a relationship between ArchitecturalConstruction and Place, while the using-period field defines a property of the PhysicalObject entity type. Consequently, all entities of type PhysicalObject and its subtypes, i.e. ArchitecturalConstruction and ArtObject inherit these fields. Furthermore, all the instances of these entity types and their subtypes also inherit these fields.

The proposed system expresses such ontological conceptual models in OWL [11], an ontology representation language that is one of the core semantic web technologies. OWL models can be created from scratch in the authoring tool or imported, facilitating the use of well-established conceptual models in the cultural heritage domain, as almost all can be (or already are) expressed as ontologies. The CIDOC conceptual reference model, for example, also provides an official OWL version.[2] Most other cultural heritage vocabularies, thesauri, and classification schemes using XML or relational database data models are consistent with the *Simple Knowledge Organization System* (SKOS) and can be automatically converted to ontologies.[3]

[1] See http://www.ics.forth.gr/indigo/ and Acknowledgements section.

[2] http://cidoc.ics.forth.gr/official_release_cidoc.html

[3] See http://www.w3.org/2004/02/skos/ about SKOS. A variety of tools exist for converting SKOS data models to, or aligning with,



Language-independent fields of "stoa-of-attalus"

Fields	Fillers
located-in	west_of_Agora
main-use	offices_house_shops
using-period	Hellenistic-Roman
original-location	Select multiple .....
destroyed-in	heruli_and_late_Roman
endParagraph	hill_of_Agoraios_Kolonos
creation_time	around_150BC
later-use	AgoraMuseum
style	Pergamene
made-of	Pentelic-Azurey-Hymettian_marble-Piraeus_limestone
has-architect	Select multiple .....
has-dimensions	Attalus_Dimensions
preserved-condition	been_restored_with_modern_materials
ordered-by	King_Attalus_II
preservedCondition	modernMaterials
excavated-in	In1953
around-there-is	Attalus_around_Place

The Stoa of Attalus was built around 150 B.C. and it was in use during the Hellenistic and Roman periods. And it followed the style of Pergamos. It was made of limestone from Piraeus and marble from Mount Penteli and Mount Hymettus and it was initially used an as offices and shops. Later, it was used an as the Agora Museum. Its construction was ordered by king Attalus II. It was destroyed in the raid of the Heruli and the late Roman period. It was excavated in 1953 and today it is in excellent condition. It is the only building in the Agora that has been restored in contemporary times using almost exclusively modern building materials.

No notes

Depth: 1 ☐ Comparisons 🇬🇧 adult

Figure 1: ELEON screen, showing the class hierarchy and the individuals of each class (left), the properties of the currently selected individual (right top), and a preview of the description of the individual (right bottom). The preview language and profile can be seen on (and selected from) the bar at the bottom of the screen.

3 Description Adaptation

Besides modelling the cultural heritage domain itself, ELEON supports annotating the objects, classes, and properties of the domain with adaptation and linguistic information. Such information is used by NLG engines to (a) plan the description that will be generated, adapting it to the current audience and circumstance, and (b) realize the planned description in a particular language.

Realization is based on clause plans (microplans) that specify how an ontological property can be expressed in each supported natural 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. Similar annotations for instances and classes specify how they should be realized as noun phrases that fill slots in the property-generated clauses. Micro-plan annotations also comprise several other language-specific parameters, such as whether the resulting clause can be aggregated into a longer sentence or not, its voice and tense, and so on, as described in more detail by Androutsopoulos et al. [1], Sect. 3.

Adaptive planning, on the other hand, operates at the abstract level and does not involve specifics of the target language. It is rather aimed at reflecting a *synthetic personality* in the description, as well as *personalizing* it for a particular audience. Adaptation parameters are provided in the form of *profile attributes* that control aspects of the text plan such as how many and which of the facts known about an object should be used to describe it, as discussed in more detail below.

3.1 Personalization and personality

The system supports authoring the *adaptation profiles* that control the *dynamic adaptation* of the generated descriptions. Profiles permit the author to specify, for example, that technical vocabulary be used when generating for experts, or that shorter and simpler sentences are generated for children. This is achieved by providing a variety of generation parameters though user profiles, including a numerical *interest* attribute of the properties of the ontology.

Isard et al. [7] describe how interest is used to impose a preference ordering of the properties of ontological entities, controlling which facts will be used when describing each entity. In the work described here, we have extended profiles in two respects:

- by generalizing *interest* into arbitrary, author-defined *profile attributes*; and
- by permitting profile attributes to apply not only to ontological properties, but also to individuals and classes.

Using these extensions, authors can define *personality profiles* for generating text, managing dialogue, and simulating emotional variation in a way that reflects a certain *personality* on behalf of the system.

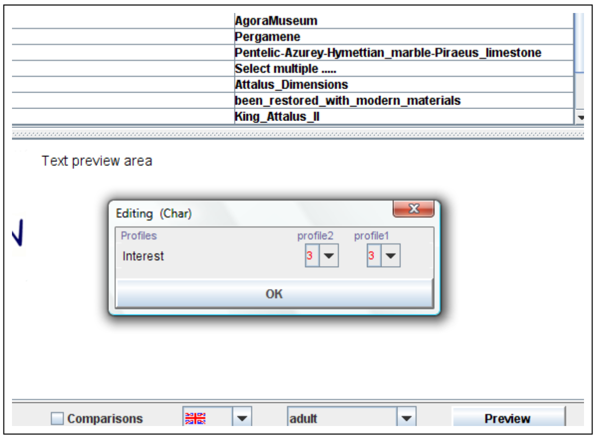
In the INDIGO project we use these profiles in a human-robot interaction application, where a robotic tour guide that gives the impression of empathizing with the visitor is perceived as more natural and user-friendly. But the methodology is generally interesting in any context of generating descriptions of cultural heritage content, especially if the individual descriptions are aggregated in a tour of the collection. In such contexts, dialogue-management adaptivity can vary the exhibits included in personalized tours and emotional state variation can match the described content and make the tour more engaging and lively.

The way in which personality profiles are used to parametrize dialogue management and simu-

ontological models. See, for example, http://www.heppnetz.de/projects/skos2gentax/ and http://annocultor.sourceforge.net/

lated emotions are discussed in more detail elsewhere [9], so we shall only briefly outline it here. Emotional variation is achieved by using the personality profile to estimate the *emotional appraisal* of dialogue acts and update the mood and emotional state of artificial agents. Dialogue management is affected both directly, by taking exhibit preference into account when deliberating over dialogue acts, and indirectly, by being influenced by the artificial agent's current mood; and, as already mentioned above, NLG is adapted by using property preference to plan a description

In the Konstantopoulos et al. [9] model, preference is calculated based on a logic model of the robot's personality traits and also on ground facts regarding objective attributes of the content—such as the *importance* of an exhibit—but also subjective attributes that reflect the robot's perception of the content—such as how *interesting* an exhibit is. With the work described here, we alleviate the burden of manually providing all the ground parameters, exploiting the fact that these parameters are strongly inter-related and can, to a large extend, be automatically inferred. More specifically, ELEON backs the profile authoring process by reasoning over manually provided exhibit attributes in order to infer what the values of the missing attributes should be. The author can inspect the explicitly provided as well as the automatically inferred values and make corrections where necessary (Figure 2). Manual corrections trigger a re-estimation of the missing values, so that after each round of corrections the overall model is a closer approximation of the author's intention.



AgoraMuseum
Pergamene
Pentelic-Azurey-Hymettian_marble-Piraeus_limestone
Select multiple .....
Attalus_Dimensions
been_restored_with_modern_materials
King_Attalus_II

Text preview area

Editing (Char)

Profiles	profile2	profile1
Interest	3	3

□ Comparisons

adult

Figure 2: Screen fragment, showing the pop-up window for providing profile attribute values for an exhibit. Automatically inferred attribute values are displayed in red, to stand out from explicitly provided ones which are displayed in black.

3.2 Representation and interoperability

Linguistic and profile annotations are represented in RDF, the *Resource Description Framework* (RDF) [5]. RDF is a knowledge representation technology built around the concept of using subject-predicate-object triples to describe abstract entities, *resources*. RDF triples assign to their subject resource the property of being related to the object through the predicate resource. Predicates can be *data properties*, in which case their objects are concrete values (numbers, strings, time periods, and so on), or *object properties*, in which case their objects are abstract resources.

Although OWL is not formally defined in RDF, it is defined in such a way that it can be represented within RDF. In fact, the OWL specification itself provides a serialization of OWL ontologies as RDF for transport and data interchange purposes. Since ELEON uses this OWL/RDF representation for the domain ontology, linguistic and profile annotations can be directly represented as RDF triples of extra-ontological properties of the ontological instances.

The RDF vocabulary used defines a property that relates ontological entities (individuals, classes, and properties) with profile attribute nodes that involve:

- the profile to which they are pertinent, e.g., 'expert';
- the attribute, e.g., 'interest' or 'importance'; and
- the numerical value of the attribute for this entity in this profile.

When applied to ontology properties, profile attribute nodes can be further elaborated to apply only to properties of instances of a particular class. For example, one can express that users find it more interesting to know the architectural style when discussing temples than when discussing stoas.

Using RDF is motivated by the usage of OWL to represent the domain ontology as well as the availability of natural language generation (NLG) engines that support it. More specifically, as already discussed, OWL ontologies and RDF annotations can be easily merged in a combined model since OWL ontologies can be expressed in RDF.

An alternative approach would have been to incorporate profile attributes in the OWL ontology. Since profile attributes refer to classes and properties as well as individuals, profile attributes would, then, have to be interpreted as second-order ontological properties. Although second-order constructs can be represented in OWL-Full (the most expressive 'dialect' of OWL), logical inference over OWL-Full ontologies is a challenging and computationally inefficient task. In fact, second-order inference is only supported by research prototypes and only for restricted fragments, often excluding binary second-order predicates (second-order properties).

By contrast, the chosen approach restricts the ontology within the computationally efficient OWL-DL dialect, for which multiple stable and highly optimized inference engines have been developed. Profile attributes are provided as extra-ontological properties, without tying them to a particular logical interpretation. We shall revisit this point in the following section.

The second motivating factor behind RDF profile attributes is interoperability with NLG engines. The RDF vocabulary used to assign linguistic and profile attributes is understood by the NATURALOWL [6] and METHODIUS [10] generation engines (Figure 3).

4 Intelligent Authoring Support

We have previously discussed how profile attributes were not directly incorporated in the domain ontology as second-order statements, but are rather represented as extra-logical RDF annotations. While avoiding forcing a second-order interpretation of profile attributes is a definite advantage from a computational-complexity point of view, this choice leaves profile attributes outside the scope of OWL reasoning tools.

In order to be able to efficiently reason over and draw inferences about profile attributes, we have chosen to interpret profile attributes within *many-valued description logics*. Using description logics has the advantage of direct access to the domain ontology; using many-valued valuations has the advantage of providing a means to represent and reason over numerical values.

This section describes this interpretation and how it is used, after first introducing description logics and many-valued valuations.

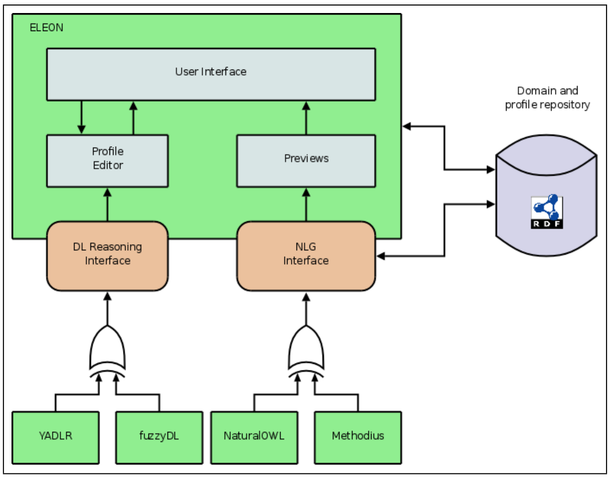


Figure 3: System architecture showing the interfacing with NLG and Inference engines

4.1 Integrating reasoning in ELEON

ELEON specifies a Java interface through which inference results can be requested and retrieved: OWL domain models and RDF profile annotations are passed to an inference engine, which responds with the numerical values of profile attributes for all ontological entities in the domain (individuals, classes, and properties).

ELEON also extends and uses the TransOnto semantic knowledge migration system[4] to perform all the necessary transformations for using many-valued DL reasoners, i.e., transforming the OWL and RDF models into many-valued DL assertions, as well as transforming logical query answers into the numerical profile-attribute values.

Furthermore, as depicted in Figure 3, the implementation includes the necessary API calls for using either of two many-valued DL reasoners, YADLR [8] or FUZZYDL [4]. Support for alternative many-valued DL reasoners can be easily added, by using such reasoners to implement the reasoning API expected by the authoring system.

4.2 Many-valued DL Reasoning

Description Logics (DL) [2] are a family of first-order logics; their main characteristic is *decidability*, attained by being restricted to *concepts* (unary predicates, sets of individuals) and *relations* (binary predicates, sets of pairs of individuals). Of particular importance is the DL called $\mathcal{SHOIN}$, which covers OWL-DL.

[4] See http://transonto.sourceforge.net/

DL statements, *concept descriptions*, use logical connectives to define concepts by combining (a) other concepts, and (b) relation constructs that describe the set of individuals that have a certain relation with a certain set of *fillers* (relation objects). Relation descriptions are not supported, and membership in a relation can only be explicitly asserted, except for a limited set of relation axioms such as inversion, subordination, and transitivity. Most DL reasoners also provide limited support for reasoning over concrete domains (numbers, strings, etc.) through *data properties* that relate abstract individual subjects with concrete value objects.

Many-valued logics in general, and consequently many-valued DLs, extend the binary true-false valuations of logical formulae into many-valued numerical valuations, denoting the *degree* to which formulae hold. Such many-valued models receive their semantics not from set theory, as is the case with binary valuations, but from algebraic *norms* that assign semantics to the logical connectives. These norms are used to calculate the degree at which complex logical propositions hold, given the degrees of their constituent elementary propositions.

In the work described here we use *Łukasziewicz-Tarski algebra* to provide many-valued semantics [9, Sect. 3]. Although there is nothing in ELEON itself that forces this choice, Łukasiewicz-Tarski algebra is well-suited to inferring profile attribute values, as it is founded on neither probability nor uncertainty, which would be inappropriate in our case, but on the notion of *relevance*.

4.3 Inferring missing attribute values

Profile attributes of individuals are captured by normalizing in the $[0, 1]$ range and then using the normalized value as a class membership degree. So, for example, if `interesting` is such an attribute of individual exhibits, then an exhibit with a (normalized) interest level of 0.7 is an instance of the Interesting class at a degree of 0.7.

Attributes of classes are reduced to attributes of the members of the class, expressed by a class subsumption assertion at the degree of the attribute. So, if the class of stoas is interesting at a degree of 0.6, this is expressed by asserting that being a member of Stoa implies being a member of Interesting. The implication is asserted at a degree of 0.6, which, under Łukasiewicz-Tarski semantics, means that being a stoa implies being interesting at a loss of 0.4 of a degree. Thus individuals that are members of the Stoa class at a degree of 1.0, are implicitly interesting at a degree of 0.6. Although this is not identical to saying that the class itself is interesting, it clearly captures the intention behind the original RDF annotation.

Ontology Instance	Interesting membership
Doric style	0.8
Ionic style	0.7
Pergamene style	0.3
Attalus	0.9

Table 1: Profile fragment.

Resource	Property	Value
Stoa of Attalus	style	Doric
Stoa of Attalus	style	Ionic
Stoa of Attalus	style	Pergamene
Stoa of Attalus	orderedBy	Attalus

Table 2: Ontology fragment, showing the properties of the ‘Stoa of Attalus’ instance.

Profile attributes can also characterize properties, like orderedBy, creationEra or style, encoding the information that it might, for example, be more interesting to describe the artistic style of an exhibit rather than provide historical data about it. This is interpreted as the strength of the connection between how interesting an exhibit is, and how interesting its properties are. In other words, if having an interesting filler for style also makes the exhibit interesting, this is taken as an indication that the style relation itself is an interesting one. Formulated in logical terms, having interesting relation fillers implies being interesting, and the implication holds at a degree provided by the interest level of the relation itself.

For example, consider the assertion at 0.8 that the class of things that are related to at least one Interesting instance with the style property, are themselves Interesting and the assertion at 0.4 that the class of things that are related to at least one Interesting instance with the orderedBy property, are themselves Interesting.

Given a profile fragment like the one in Table 1 and a domain ontology including the factual information in Table 2, ‘Stoa of Attalus’ has an interesting style at a degree of 0.8, which is the maximum among the three architectural styles found in the stoa (Doric, Ionic, and Pergamene). Since style fillers transfer interest at a loss of 0.2, style contributes 0.6 to the stoa's Interesting-ness. By contrast, the filler of orderedBy (which is more interesting in this profile than any of the architectural styles) only contributes 0.3 of a degree, because orderedBy is annotated as uninteresting and interest transfers across it at a heavy loss.

We have so far discussed how to infer profile attribute values for the individuals of the domain. Classes and relations receive the value of the minimal instance of the class (relation). That is to say, the individual (pair of individuals) for which nothing else is known, except that it is a member of the class (relation).

As an example, consider a DoricBuilding class which is a subclass of Building that only admits instances that have a style relation with 'Doric'. The minimal instance of this class is a member of Interesting through having an interesting property as discussed above, even though nothing else is known about it. This membership degree in Interesting is taken to be an attribute of the class itself rather than any one of its members, and is used as the attribute value for the class itself.

For relations, two minimal instances of the relation's domain and range are created. The attribute value for the property is the degree of the implication that having this property makes the domain individual have the attribute. For example, in order to infer how interesting the property devotedTo is, we first observe that it relates Temple instances with MythicalPerson instances, and create bare instances of these two classes. The implication that having a devotedTo relation to an Interesting individual leads to being member of Interesting holds to a degree that can be calculated, given the Interesting degrees of the Temple and MythicalPerson instances involved in the relation. The degree of the implication is then used as the value of the `interesting` attribute.

5 Related Work

ELEON is based on the authoring tool described by Androutsopoulos et al. [1], which was also targeted at creating ontologies for generating personalized descriptions of the individuals in the ontology. ELEON inherits from that tool the idea of separating the abstract ontological relations from the concrete linguistic information, facilitating the easy reuse of the ontological information to generate descriptions in multiple languages, as well as using an external NLG engine to provide previews of the descriptions from within the authoring environment.

The system presented here extends a previous version of ELEON [3], which supports using an external DL reasoner to catch logical errors by checking the consistency of the authored ontology. In the work described here, the intelligence behind the tool is substantially extended by using logical inference to *predict* values that have not been explicitly entered by the user, alleviating the need to manually provide large volumes of numerical data.

A parallel line of development of the original Androutsopoulos et al. tool is based on the Protégé ontology authoring and management environment.[5] Galanis and Androutsopoulos [6] developed a Protégé plug-in that builds upon the extensive ontology authoring features of Protégé to provide an environment for creating cultural heritage ontologies and the associated linguistic and profiling annotations. It does not, however, offer the flexibility to define new profile attributes as ELEON does, and is restricted to specifying the level of interest of the various ontological entities. Furthermore, it only uses logic inference to catch ontological inconsistencies in a manner similar to that described by Bilidas et al. [3] without any prediction facilities.

6 Conclusion

In this article we have presented an authoring environment for the creation of domain ontologies and the associated linguistic and profile annotations. Annotated ontologies can be used to automatically generate natural-language descriptions of the entities of the ontology, dynamically adapting the generation engine to the audience and context of the description.

The advantages of using ELEON instead of generic knowledge tools, such as Protégé, stem from the ability to couple ELEON with external engines that provide important conveniences to the author. More specifically, ELEON can invoke a natural language generation engine in order to display previews of the description based on the information currently provided about an object. Furthermore, logical inference is used to provide an intel-

[5] See `http://protege.stanford.edu/`

ligent authoring back-end that completes the current model by inferring missing values based on what has already been provided.

This intelligent authoring support facility, combined with the immediate previews, can considerably reduce the effort required to create a fully functional model as the author can iterate through cycles of providing information, previewing the generated text, and only elaborating the model where the text is unsatisfactory. This iterative process converges to satisfactory descriptions much faster than having to manually enter all adaptation parameters, especially for large and complex domains.

In the context of the XENIOS project,[6] the previous version of ELEON has been evaluated by curators of the Foundation of the Hellenic World, who used it to create an ontology of the buildings, rooms, and exhibitions of the Foundation. In the context of creating the Agora of Athens ontology and annotations for INDIGO, we are planning to extend this evaluation to include the new intelligent authoring features.

Acknowledgements

The work described here is supported by the FP6-IST project INDIGO[7] which aims to advance human-robot interaction methodology and technology, enabling robots to perceive natural human behaviour, as well as making them act in ways that are more familiar to humans. The authors also wish to acknowledge the help of the colleagues from the Foundation of the Hellenic World who used ELEON to create the initial version of the ontology for the ancient Agora of Athens.

References

[1] Ion Androutsopoulos, Jon Oberlander, and Vangelis Karkaletsis. 2007. Source authoring for multilingual generation of personalised object descriptions. *Journal of Natural Language Engineering*, 13(3):191–233.

[2] Franz Baader, Ian Horrocks, and Ulrike Sattler. 2003. Description logics as ontology languages for the semantic web. In Steffen Staab and Rudi Studer, editors, *Lecture Notes in Artificial Intelligence*. Springer Verlag.

[3] Dimitris Bilidas, Maria Theologou, and Vangelis Karkaletsis. 2007. Enriching OWL ontologies with linguistic and user-related annotations: the ELEON system. In *Proc. 19th IEEE Intl. Conf. on Tools with Artificial Intelligence (ICTAI-2007), Patras, Greece, Oct. 2007*. IEEE Computer Society.

[4] Fernando Bobillo and Umberto Straccia. 2008. fuzzyDL: an expressive fuzzy Description Logic reasoner. In *Proceedings of the 2008 International Conference on Fuzzy Systems (FUZZ-08)*.

[5] Dan Brickley and R. V. Guha. 2004. RDF Primer. W3C Recommendation.

[6] Dimitris Galanis and Ion Androutsopoulos. 2007. Generating multilingual descriptions from linguistically annotated OWL ontologies: the NaturalOWL system. In *Proceedings of the 11th European Workshop on Natural Language Generation (ENLG 2007), Schloss Dagstuhl, Germany*, pages 143–146.

[7] Amy Isard, Jon Oberlander, Ion Androutsopoulos, and Colin Matheson. 2003. Speaking the users' languages. *IEEE Intelligent Systems*, 18(1):40–45.

[8] Stasinos Konstantopoulos and Georgios Apostolikas. 2007. Fuzzy-DL reasoning over unknown fuzzy degrees. In *Proc. Intl. IFIP Workshop of Semantic Web and Web Semantics (IFIP-SWWS 07)*, Nov. 2007, Vilamoura, Portugal. LNCS 4806, Springer, Berlin/Heidelberg.

[9] Stasinos Konstantopoulos, Vangelis Karkaletsis, and Colin Matheson. 2008. Robot personality: Representation and externalization. In *Proceedings of Intl. Workshop on Computational Aspects of Affective and Emotional Interaction (CAFFEi 08)*, Patras, Greece, July 2008.

[10] Matthew Marge, Amy Isard, and Johanna Moore. 2008. Creation of a new domain and evaluation of comparison generation in a natural language generation system. In *Proceedings of the Fifth International Language Generation Conference (INLG08), June 2008, Salt Fork, Ohio, USA*.

[11] Michael K. Smith, Chris Welty, and Deborah L. McGuinness. 2004. OWL web ontology language. W3C Recommendation.

[6] See `http://www.ics.forth.gr/xenios/` (in Greek)

[7] See `http://www.ics.forth.gr/indigo/`