# Quick User Guide to ELEON

A step by step approach will be followed to build an ontology which concerns an archeological site. The point of this guide is to get acquainted with ELEON. The ontology will consist of types and instances, and will be enriched with linguistic elements to allow text generation. Also, user models will be added. To complete this you will need the ELEON tool, and natural language generation engine Natural OWL , are all available from the web site of ELEON[1] . At the same location the reader can download the enriched ontology (*ShortOntology*).   A detailed description of ELEON is provided by the user's guide[1].

Task Description

Let us build a world for a virtual museum that includes ancient temples and relevant information such as architectural style, date of construction, etc. In particular, we shall create the ontology of the museum (types, subtypes, and entities), as well as relations between entities. Second, we shall create user models, so as to personalize the information. Third, we shall create microplans to instruct the NLG engine to generate text.

Create two user models

There is supposed to be two types of visitors in this museum, children and adults. Thus in user types we shall create a *child* and an *adult* type. We make the plausible assumption that children prefer shorter sentences, so we set *maximum facts per sentence* and *facts per page 2.* The relevant values for adults will be 10. The links per page is not applicable, as well the synthesiser voice (see next figure).
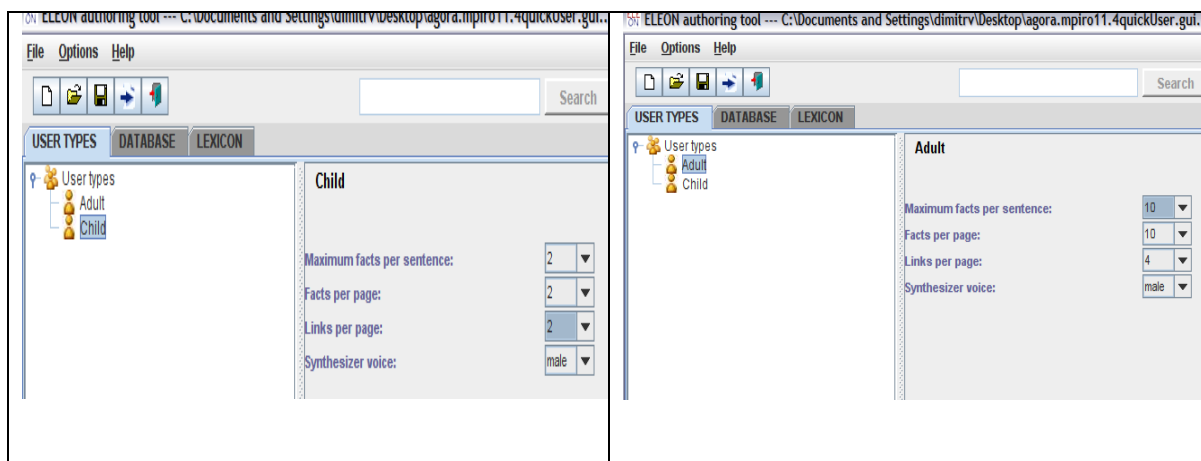
Add types and entities

We add the following basic types: `ArchitecturalConstruction, Direction, HistoricalPeriod,` and `Person.` The `ArchitecturalConstruction` will include the `Altar` type. The `Person` type will include the `Architect, God` and `Hero.` In addition the Altar will have the following entities: `altar-of-aphrodite-ourania` and `altar-of-ares.` Direction will include the following entities: north-side, southern, south-side, southeast and southern. Historical-period, will have the entities: `archaic` and `classical.` Material will have the entity `Cycladic-marble.` The Architect will have the entities `Callikrates, Ictinus.` God will include `Aphrodite, Apollo` and `ares.` Finally, Hero will have the entity `Aiakos.` (Right click on Basic-entity-types to add types or entities ---see next figure.)
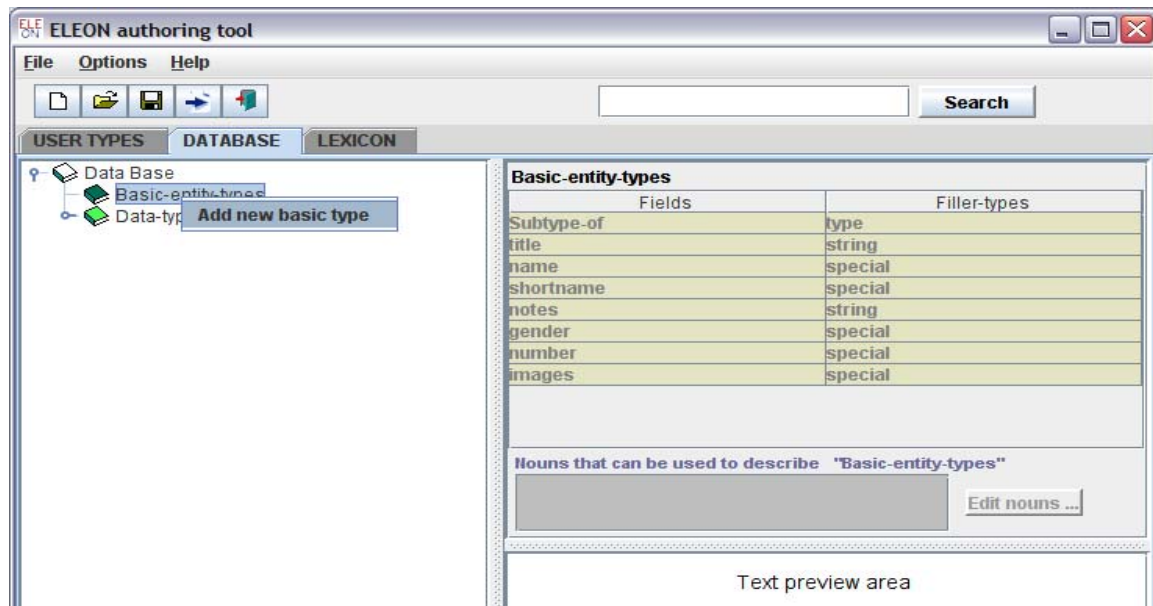
Defining Fields

To `ArchitecturalConstruction` we will add the fields, with the following filler (data) types:

ArchitecturalConstruction	
Fields	**Filler-types**
Subtype-of	Basic-entity-types
title	string
name	special
shortname	special
notes	string
gender	special
number	special
images	special
has-length	Number
has-architect	Person
made-of	Material
has-description	String
constructed-by	Person
has-width	Number
is-located-in-side	Direction
construction-date	Historical-period
used-for	String
dedicated-by	Person
dedicated-to	Person

The fields, appearing in a shaded background are predefined, set by the system. Next, some remarks that concern the filler types. `has-architect` derives its values from the Person data type. And is-located-in-side, derives its values from the Direction datatype. Similar remarks can be made for the other fields. (To add a new field, right click on the shaded area of the corresponding type).

| USER TYPES	DATABASE	LEXICON

Data Base
- Basic-entity-types
  - person
- Data-types

person

Fields	Filler-types
Subtype-of	Basic-entity-types
title	string
name	special
shortname	al
notes	string
gender	special
number	special
images	special

Add new field (end of table)

Nouns that can be used to describe "person"

Edit nouns ...

After adding all the types and entities the ontology will look like:
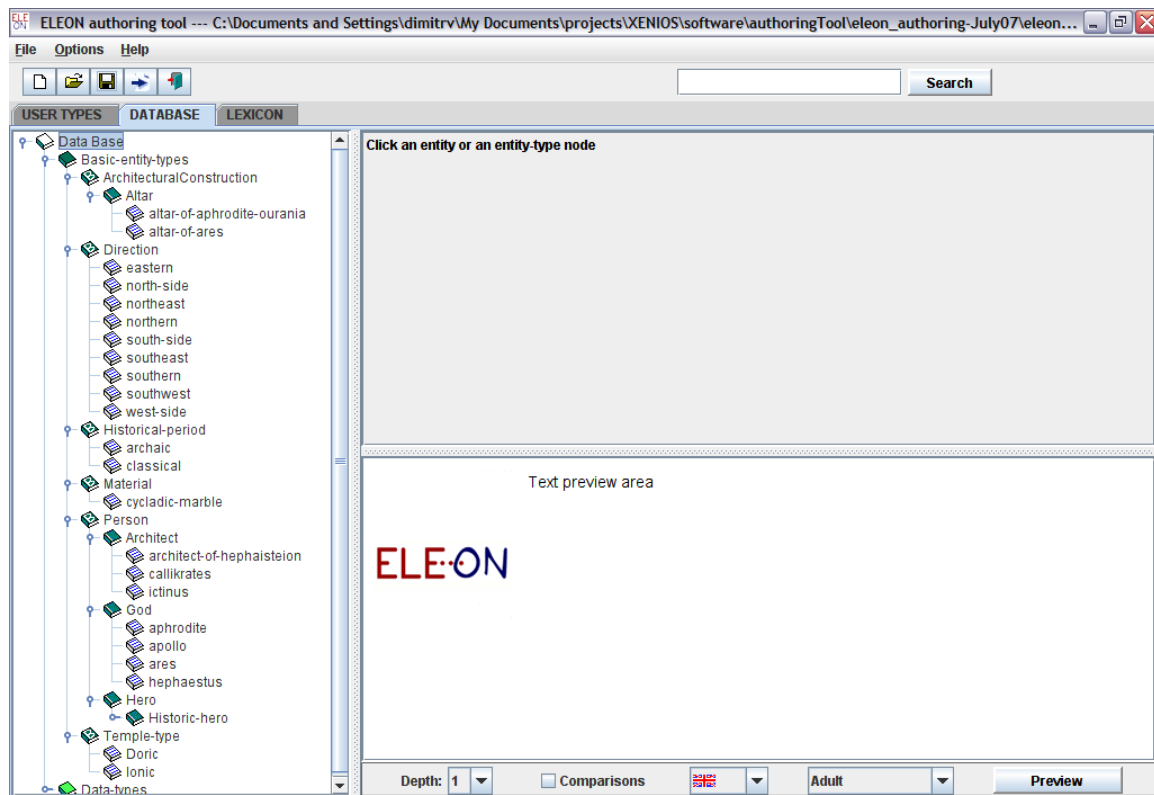
Inserting Restrictions

Next, we restrict the `has-architect` field of the `ArchitecturalContruction` type to assume all values from the architect type, and also to assume at least one value, reflecting the notion that each construction has at least one architect. (By right clicking on the `construction-data` field and selecting edit-restrictions from the pull down menu, the menu on the right appears which allows you
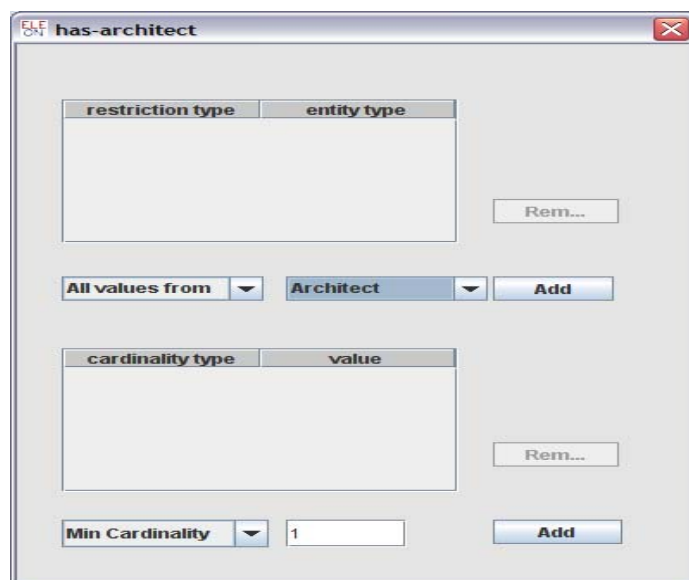
to set the appropriate values).

Relate entities to nouns

Associating a type of the ontology (and hence all relating entities) with a noun can provide some guidance to the NLG engine. For instance, if we associated the noun `altar` with the type `altar` in the ontology, then the NLG engine (should it support this function) might produce: "*another altar next to altar of Aprodite Ourania is the altar of Ares*". Thus, first we access the lexicon pane of the ELEON tool, and we right click on the *nouns* to add a new noun, which will be called altar-NP (See next figure)
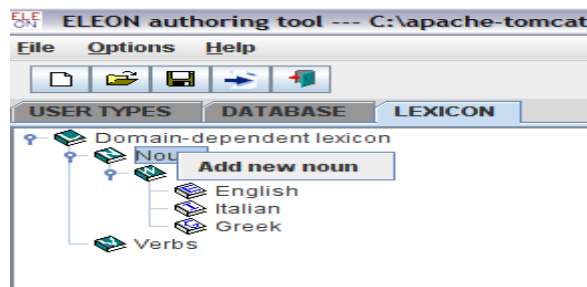
Next, the noun form for English will be defined as follows:
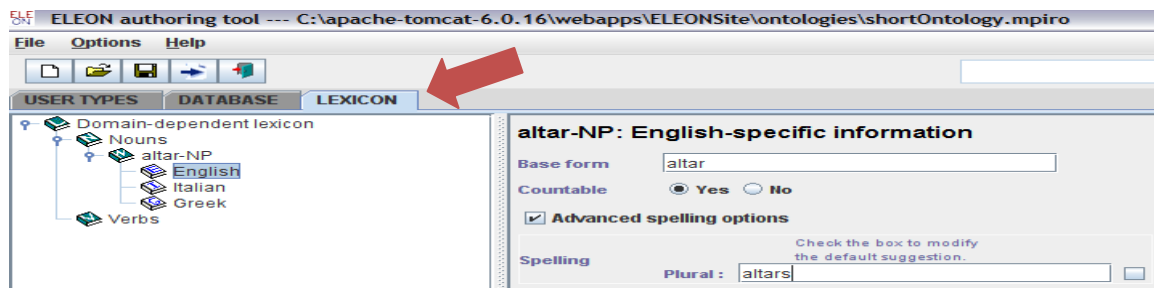
Then, we can associate the type "`Altar`" with the relevant noun by clicking on edit nouns.
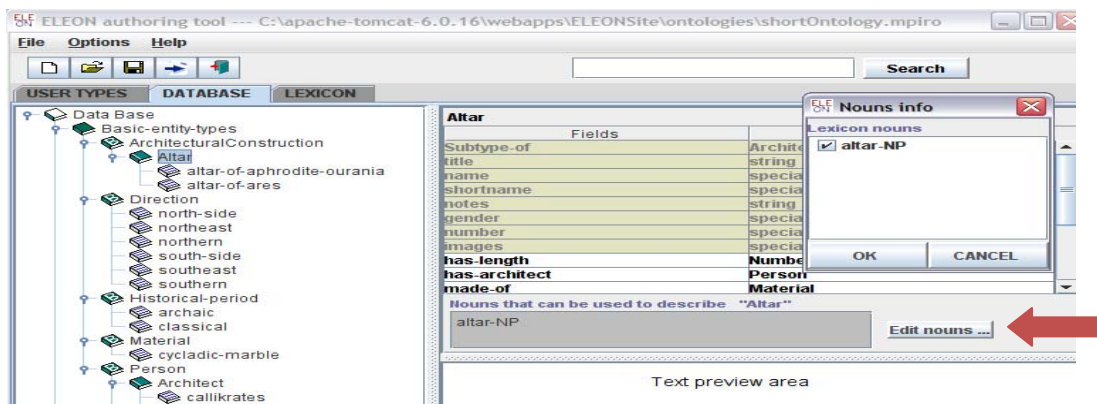
Create templates for natural language

The next step is to define templates (microplans), which are used primarily by the NLG engine to generate text. Templates are defined for fields of types (or subtypes), and are inherited by the relevant entities. We shall define a template in the `Altar` for the `has-architect field`. In this case, text will be generated for the two entities under the `Altar` type.


By left clicking on the `has-architect` field, a template can be defined which instructs a natural language generation engine produce text. (In particular, the NaturalOWL that is integrated with ELEON supports templates). For the `has-architect field`, we have four slots. In Slot 1, the "referring to owner expression" refers to the "name" field of the `altar-of-aphrodite-ourania`. Slots 2 and 3 contain strings, and slot 4 refers to the filler value of the "has-architect" field. In addition, for every slot there is linguistic information that will potentially be used by the attached NLG. Similarly, we can define templates for the other fields. Note also that more than one templates can be defined for every field.

The appropriateness button appearing on the above figure, makes sense when more than one templates have been defined for each field. Thus different microplans will presumably have different appropriateness values for different user types.

Importance/repetitions

It is possible that some fields of the `Altar` type are of lower importance to children than to adults. In particular let the fields: `has-length` and `has-width` are not important to children.

This can be denoted by setting the importance of the two fields at the `Altar` type. (Right click on the aforementioned fields, and select the next values).
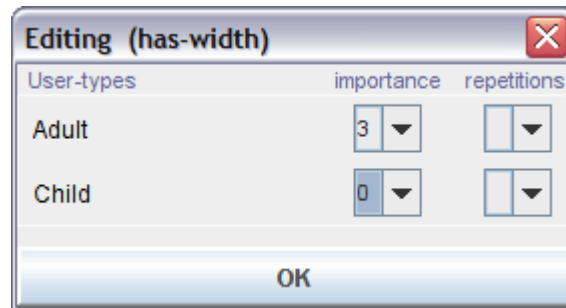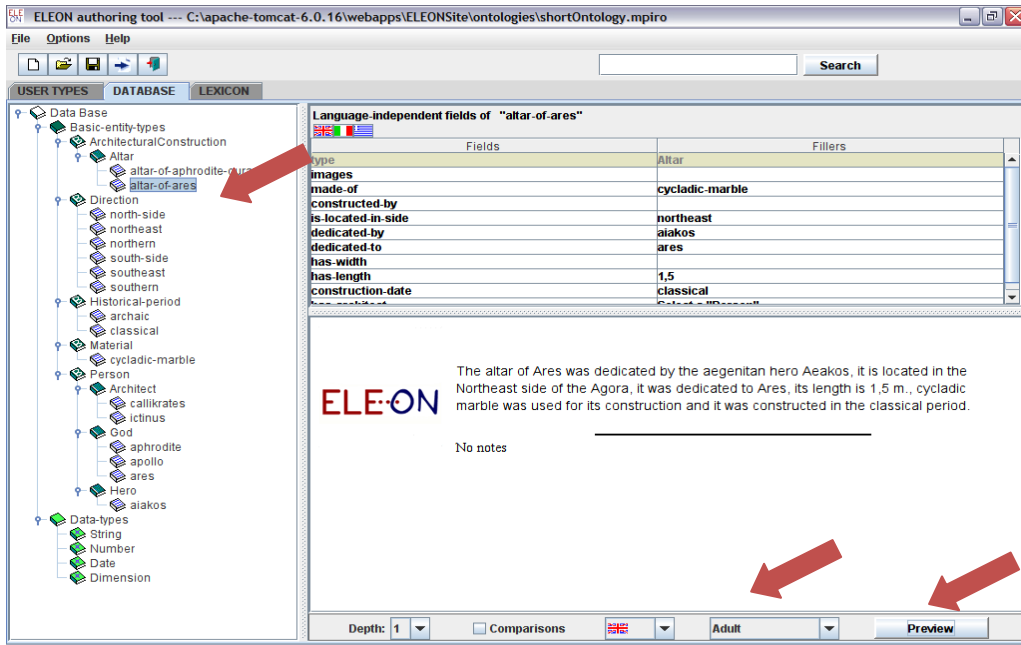

Generated Text

Once the ontology, the microplans, the lexicon and the user models are defined, ELEON can call the NLG engine by clicking on the *Preview* button and selecting the appropriate user model (child or adult in this case). Once this occurs, ELEON exports the Ontology (in OWL format), and the microplans, the lexicon and the user models (in RDF format). The four files will be used by the Natural OWL natural language generation engine to create text, which will be imported by ELEON.

For the `altar-of-ares` the following figure depicts the generated text for the child user type.
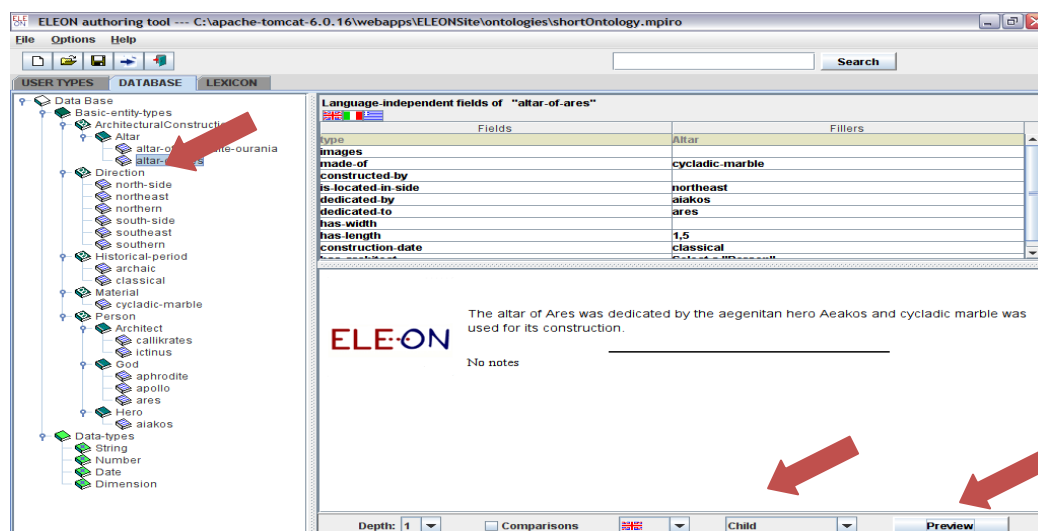
For the `altar-of-ares` the following figure depicts the generated text for the adult user type.

Quick User Guide to ELE·ON

The reader is encouraged to Preview the generate text for the `altar-of-aphrodite-ourania.`