



VRIJE UNIVERSITEIT BRUSSEL

FACULTEIT WETENSCHAPPEN

VAKGROEP INFORMATICA EN TOEGEPASTE INFORMATICA  
SYSTEMS TECHNOLOGY AND APPLICATIONS RESEARCH LAB












# STAR Lab Technical Report

## The DOGMA Modeller manual

Peter Spyns & Andriy Lisovoy

|                      |                                |
|----------------------|--------------------------------|
| affiliation:         |                                |
| keywords             |                                |
| number               | STAR-2003-11                   |
| date                 | 31/07/2003                     |
| corresponding author | Peter Spyns                    |
| status               | draft                          |
| reference            | EuroLan 2003 tutorial material |

## Content Table

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b><i>Introduction</i></b>   | <b>3</b>  |
| <b>2</b> | <b><i>A General View of DogmaModeller</i></b>  | <b>3</b>  |
| <b>3</b> | <b><i>Short overview</i></b>   | <b>5</b>  |
| 3.1      | <b>Ontology base window</b>  | <b>5</b>  |
| 3.2      | <b>Commitment modelling area</b>   | <b>6</b>  |
| 3.3      | <b>Commitments library window</b>  | <b>6</b>  |
| <b>4</b> | <b><i>The Ontology base</i></b>  | <b>7</b>  |
| 4.1      | <b>Creating an ontology base</b>   | <b>7</b>  |
| 4.2      | <b>Opening an ontology Base</b>  | <b>8</b>  |
| 4.3      | <b>Creating a context</b>  | <b>9</b>  |
| 4.4      | <b>Deleting a context</b>  | <b>10</b> |
| 4.5      | <b>Editing context properties</b>  | <b>10</b> |
| 4.6      | <b>Creating a term</b>   | <b>11</b> |
| 4.7      | <b>Deleting a term</b>   | <b>12</b> |
| 4.8      | <b>Editing term properties</b>   | <b>12</b> |
| 4.9      | <b>Creating/Deleting a lexon</b>   | <b>13</b> |
| 4.10     | <b>Editing lexon properties</b>  | <b>14</b> |
| 4.11     | <b>Searching a term in the ontology base</b>   | <b>15</b> |
| 4.12     | <b>The tree representation</b>   | <b>15</b> |
| <b>5</b> | <b><i>The Commitments</i></b>  | <b>16</b> |
| 5.1      | <b>Opening a commitment library</b>  | <b>16</b> |
| 5.2      | <b>Deleting a commitment</b>   | <b>17</b> |
| 5.3      | <b>Creating a new commitment</b>   | <b>18</b> |
| 5.4      | <b>Graphically representing a commitment</b>   | <b>19</b> |
| 5.4.1    | Subsumption relations  | 19        |
| 5.4.2    | Relationship properties  | 20        |
| 5.4.3    | Object properties  | 21        |
| 5.5      | <b>Defining semantic constraints</b>   | <b>25</b> |
| 5.5.1    | Mandatoriness   | 25        |
| 5.5.2    | Uniqueness  , Subset  , and Equality                          | 25        |
| 5.5.3    | External uniqueness  , external mandatory  , and exclusion  | 26        |
| 5.5.4    | Total-  and Exclusive -  constraints   | 26        |
| 5.5.5    | Ring constraint   | 26        |
| 5.5.6    | Frequency   | 27        |
| 5.6      | <b>Saving a Commitment</b>   | <b>27</b> |

# 1 Introduction

The DogmaModeller ontology engineering tool supports functionalities for modelling, browsing, and managing both the ontology base and the commitments. It supports modelling ontological commitments using the Object Role Modelling graphical notation, and generates the corresponding ORM Mark-up Language automatically. In addition, the DogmaModeller tool supports verbalisation of ontological commitments into pseudo natural language.

Note that knowledge about the DOGMA ontology engineering methodology as well as the ORM modelling methodology is presupposed. Important to know as well is that currently not all options of the menu bar are actually available.

# 2 A General View of DogmaModeller

Figure 1 shows a screenshot of the DogmaModeller tool displaying its main constituting parts: the ontology base (i), the commitment modelling area (ii), the commitment library window pane (iii) and an (error) message zone (iv). In the remainder of this text, we will describe these parts in more detail.

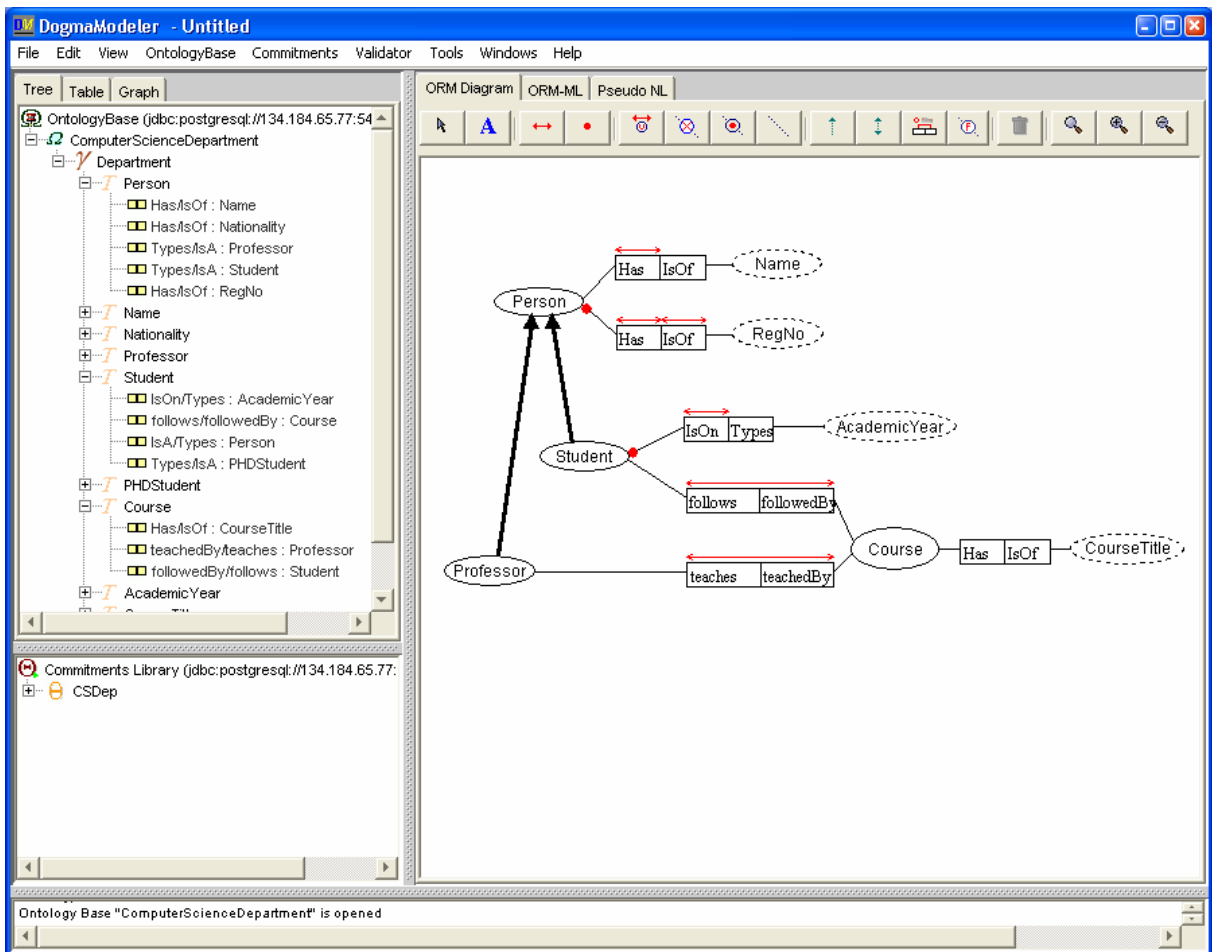
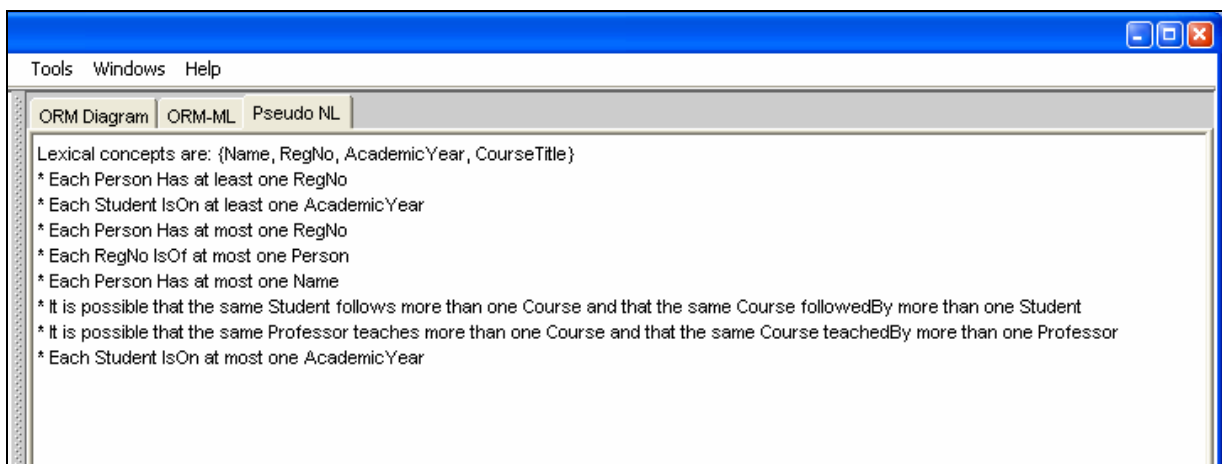


Figure 1

- 1) the *ontology base*: for editing, representing and creating an ontology base that consists of lexons. Currently, a lexon is formally defined as a quintuple:  
context label, head term label, role label, co-role label, second term label
- 2) the *commitment modelling area*: for editing, representing and creating commitments using a conceptual modelling language (here Object Role Modeling – ORM). Many details on ORM can be found in the book of Terry Halpin, entitled “*Information Modeling and Relational Databases: from conceptual analysis to logical design*” (published by Morgan Kaufmann)
- 3) the *commitment library*: represents all commitments that commit to the ontology base currently opened
- 4) the *message zone*: shows (errors) messages and logs the actions performed by user etc.

In the right window, there are three tabs:

- 1) **ORM Diagram** (see Figure 1): this is the commitment modelling area that will be explained in more details in section 5)
- 2) **ORM-ML** (ORM Mark-up Language – see Figure 3): this is an XML representation of the content of the ORM diagram window. It is a convenient way to exchange commitments across various applications. It is automatically generated by the DogmaModeller tool.that supports saving ORM-ML into text files, or uploading it into an ontology server. Currently, no more details on the ORM Mark-up Language will be provided.
- 3) **Pseudo-NL**: Figure 2 shows the corresponding Pseudo Natural language (fixed-syntax English sentences) of the ORM diagram from Figure 1. It is automatically generated by the tool by applying predefined templates to the commitments’ content. We believe that this allows non-experts to (help to) check, validate or build the commitment rules and will simplify the modeling process.



**Figure 2**

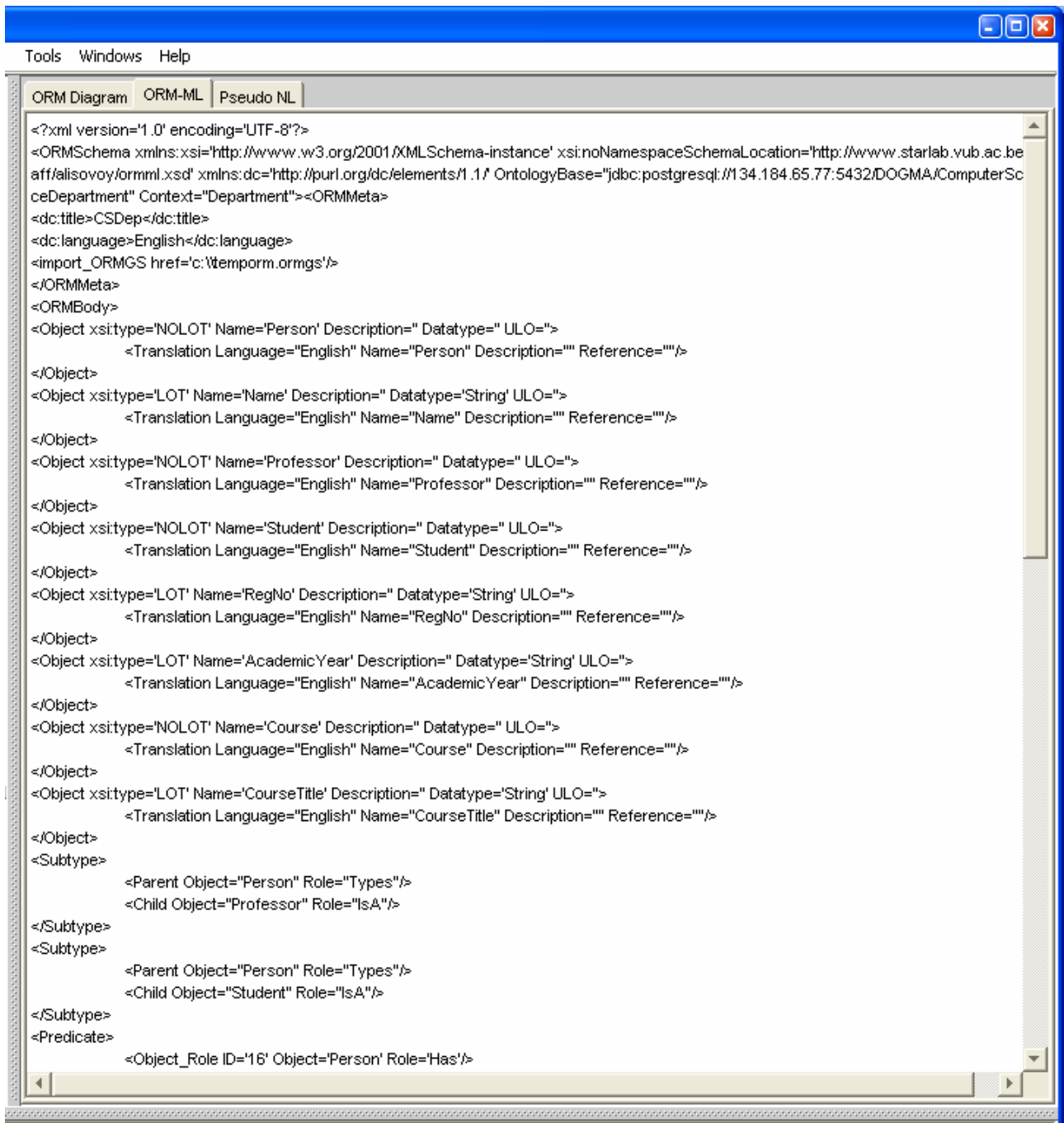


Figure 3

### 3 Short overview

#### 3.1 Ontology base window

Before building ontological commitments, ontology builders should define their lexons in the ontology base window, in case it is empty. This window (see Figure 1) presents the set of lexons  $\{< \gamma : \text{Term}_1, \text{Role}, \text{Co-Role}, \text{Term}_2 \}>$  in a tree-like structure. The first level, ( $\Omega$ ) represents ontology bases (e.g. *computer science department*). On the second level, each node ( $\gamma$ ) represents a context (e.g.

department). Within a context, each node ( $\mathcal{T}$ ), on the third level, represents a term (e.g. *person*, *course*) while nodes ( $\square$ ) on the fourth level, represent the set of role, core and second term for that head term.

Notice that level 0 in the tree represents an ontology base server, where the content of ontology bases is hosted and managed. All transactions on the ontology base (e.g. creating contexts, editing lexons, etc.) will be transmitted, verified and executed on the server side. The DogmaModeller is connected with our DogmaServer<sup>1</sup>, which stores and serves the ontology base and the commitment layer.

### 3.2 Commitment modelling area

To build an ontological commitment, ontology builders can drag and drop lexons from the ontology base window into the ORM panel (i.e. defining an ontological view or visibility rules). When doing so, lexons will be mapped automatically into ORM fact types. Then, in order to define rules on these lexons, ontology builders can use the ORM family of constraints (see Figure 59).

*Remark:* mapping lexons as intuitive domain knowledge into ORM fact types that have predefined formal semantics is done as follows:

- a term within a context is mapped directly into an Object Type in ORM
- roles within a context are also mapped directly into ORM Roles.

In the case of ORM subtype relations have specific “built-in” semantics. Here, commitment builders need to customise the “Graph settings” window in order to specify which roles should be mapped (see Figure 46). Further, DogmaModeller does not support ORM unary roles and nested fact types.

→ Notice that our approach is not restricted to ORM; the tool is designed with flexibility of adding new plug-ins in order to support modeling commitments in other languages, e.g. EER, UML, DAML, OWL etc.

### 3.3 Commitments library window

This window pane facilitates reusability, management, and organisation issues of ontological commitments. The current implementation allows ontology builders to access and browse ontological commitments stored in a library ( $\Theta$ ). Each node ( $\Theta$ ) in the first level of the tree represents a commitment. By expanding a commitment node, the set of lexons and the set of rules – belonging to this commitment – will appear on the second level. Advanced features such as e.g. indexing, modularisation, composition, versioning, etc. of ontological commitments are ongoing research issues.

In the next sections, more details will be provided on how to create an ontology base and how to populate it with lexons (see section 4), and subsequently on how to create and define commitments on an ontology base (see section 5).

---

<sup>1</sup> For more details, or to download DogmaServer, you can access: <http://www.starlab.vub.ac.be/research/dogma/OntologyServer.htm>.

## 4 The Ontology base

### 4.1 Creating an ontology base

1. In the menu “*OntologyBase*”, you need to select “Create OntologyBase” and choose “in Database” (see Figure 4).

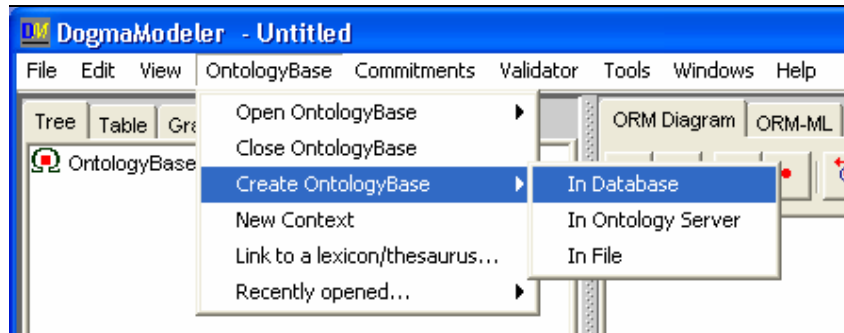


Figure 4

2. Information about the ontology base database connection will appear (see Figure 5). If you don't know exactly what information you have to enter, just press the “Default” button that restores the connection settings that most recently have been successfully used.

3. After you have entered the ontology base database connection data, click on the “Connect” button. If a connection cannot be established, it might be caused by the following reasons:

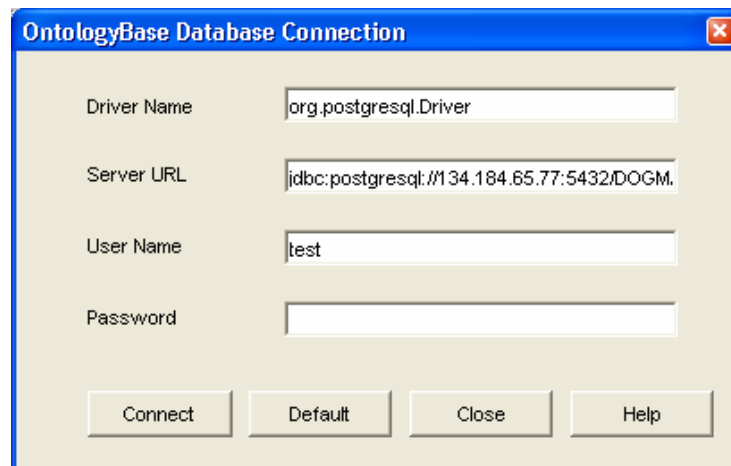
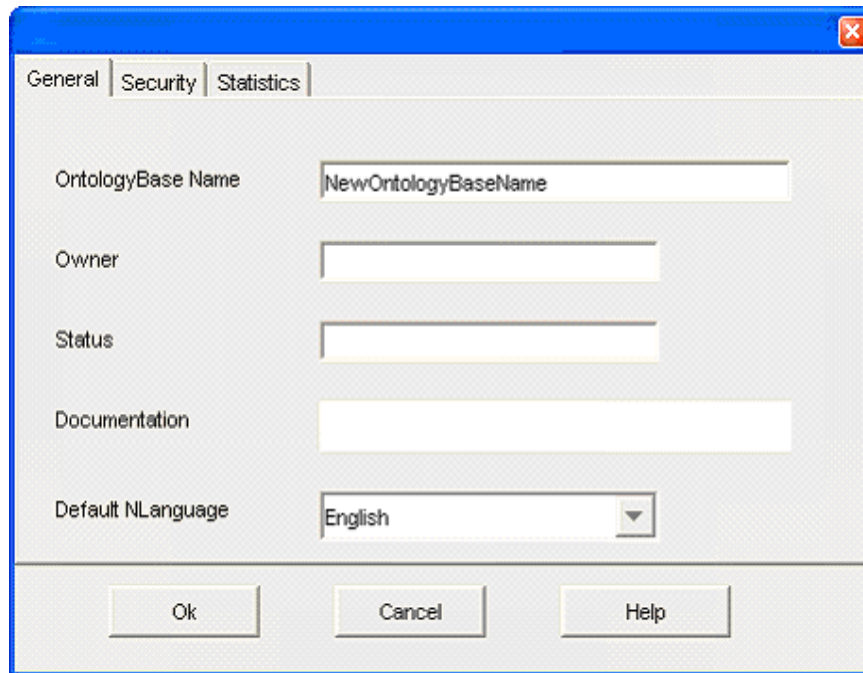


Figure 5

- You don't have connection to the Internet (and there is no local server available).
  - One of the parameter settings (driver name, server URL, user name or password) is incorrect.
- 4) Once a connection has been established, you have to specify a new “Ontology-Base Name” and some optional properties (see Figure 6). Currently, it is better to keep “English” as your default natural language (“Default NLanguage”). Enter a unique name (next to the “OntologyBase Name” label) for the ontology base you want to create. If an ontology base with the same name already exists, an error message will be displayed in the error message window pane.

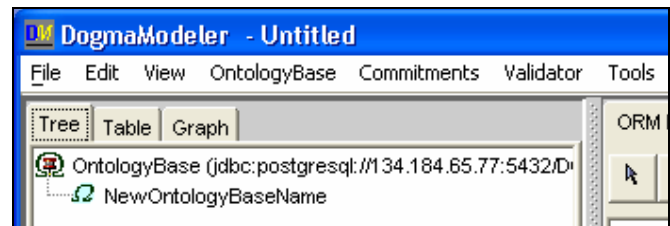


**Figure 6**

- 5) Normally, the ontology base window now displays your newly created ontology base – that, of course, is still empty for the moment (see Figure 7).

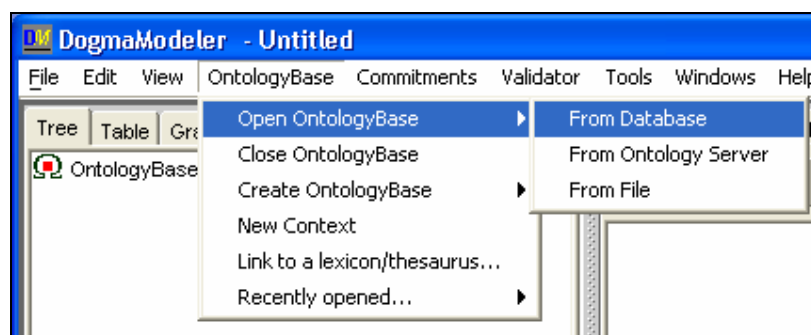
#### **4.2 Opening an ontology Base**

1. In the menu “OntologyBase”, select “Open OntologyBase” and choose “From Database” (see Figure 8).
2. Information about the ontology base data base connection will appear (see Figure 5 and item 0 of section 4.1).

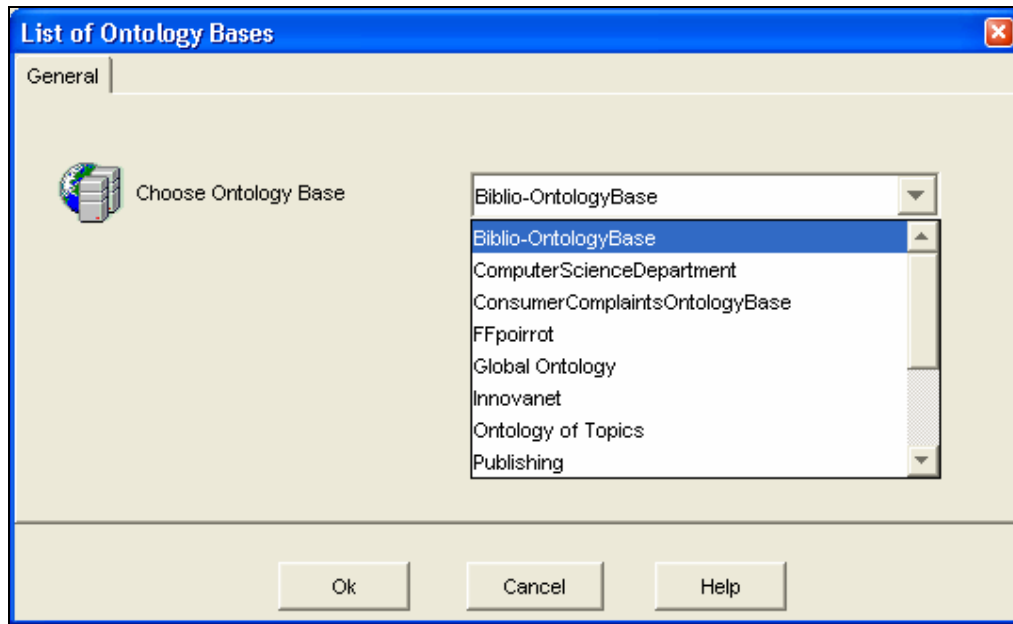


**Figure 7**

3. After you have entered the ontology base database connection data, click on the “Connect” button (see also item 0 of section 4.1).
4. If a connection is established you will need to select the ontology base you want to work on (see Figure 9):

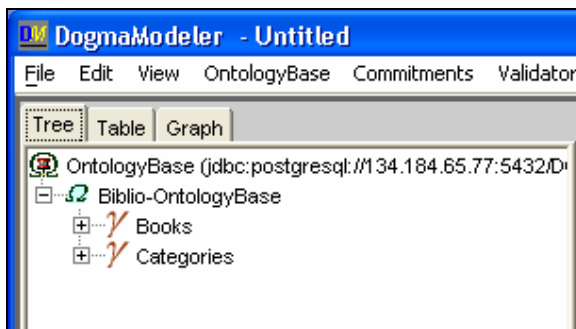


**Figure 8**

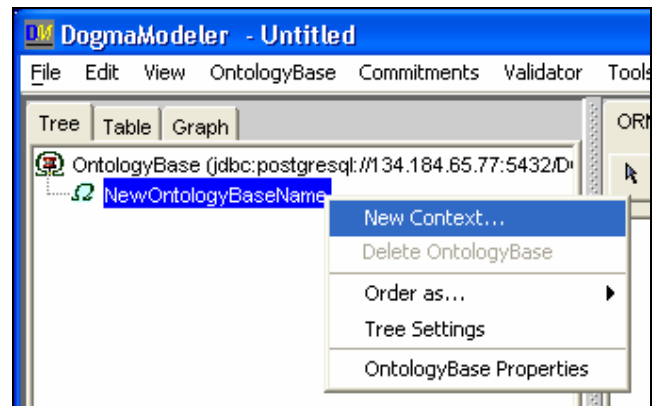


▪ **Figure 9**

5. The selected ontology base will be shown in the ontology base window pane (see Figure 11).



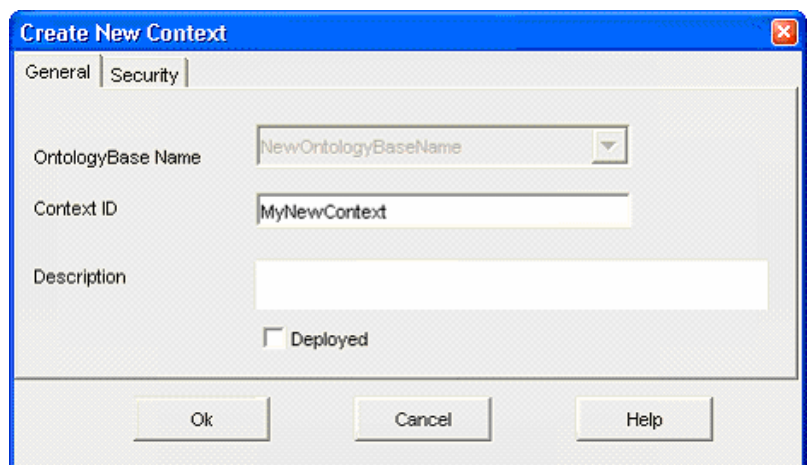
**Figure 11**



**Figure 10**

### 4.3 Creating a context

- 1) Click with the right mouse button on the ontology base node. In the popup window, select "New Context" (see Figure 10).
- 2) Enter a name (next to "Context ID") for the new context you want to create and click on "Ok". You may also want to enter a (textual) "Description" for the newly created context (see Figure 12).



**Figure 12**

- The new context will appear in the ontology base tree (see Figure 13).

#### 4.4 Deleting a context

Click with the right mouse button on the context node you want to delete. In the popup window select “Delete Context” (see Figure 14).

→ Note that deleting a context will erase all the terms and lexons that are included in that context!

#### 4.5 Editing context properties

Click with the right mouse button on the context node you want edit (see Figure 15). In the popup window, select “Context Properties” (see Figure 16). You may only change the description of the context and mark it as deployed or not (if the context is already deployed, it is not allowed to delete anything from this context).

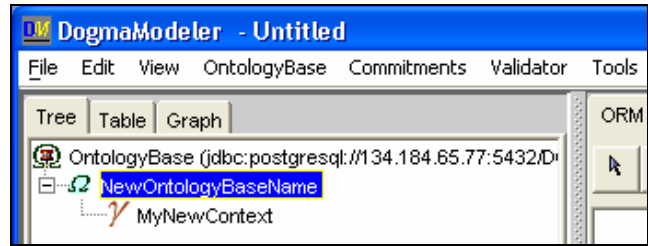


Figure 13

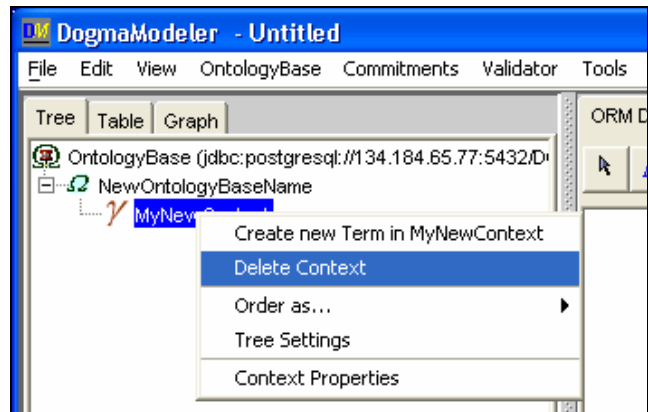


Figure 14

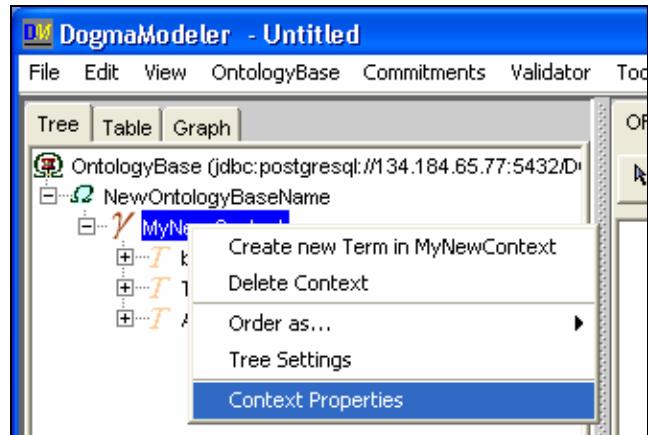


Figure 15

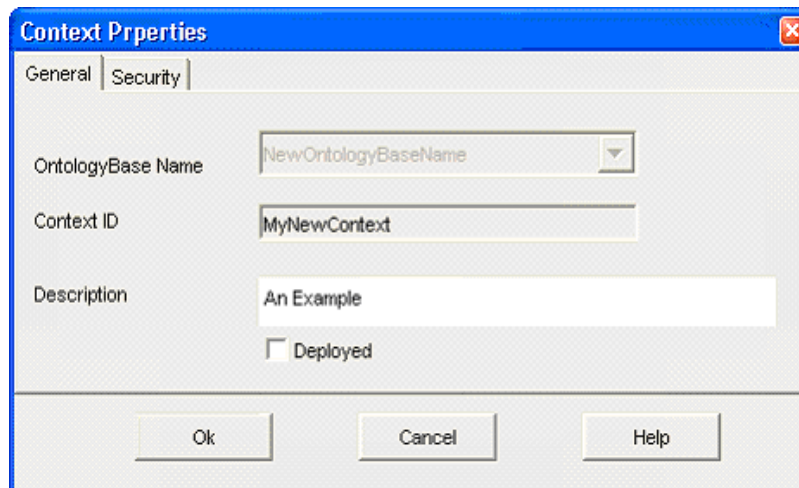


Figure 16

## 4.6 Creating a term

- 1) Click with the right mouse button on the context node to which you want to add a new term. In the popup window select “Create new term in *MyNewContext*” (see Figure 17).
- 2) Enter the name for the new term you want to add (the “term” box of the “New Term” dialogue box and press “Ok” (see Figure 18).

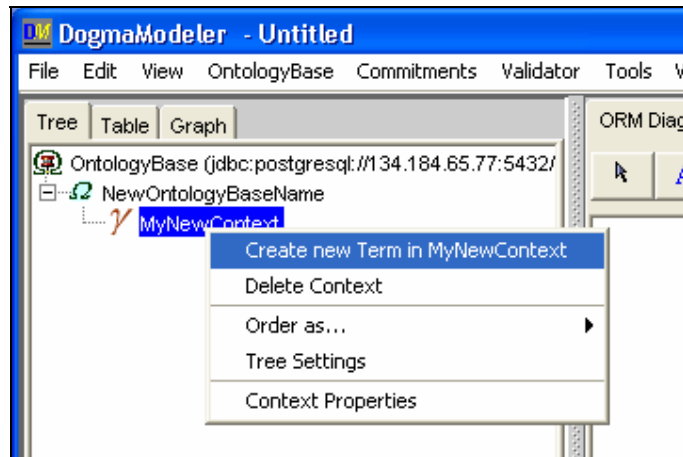


Figure 17

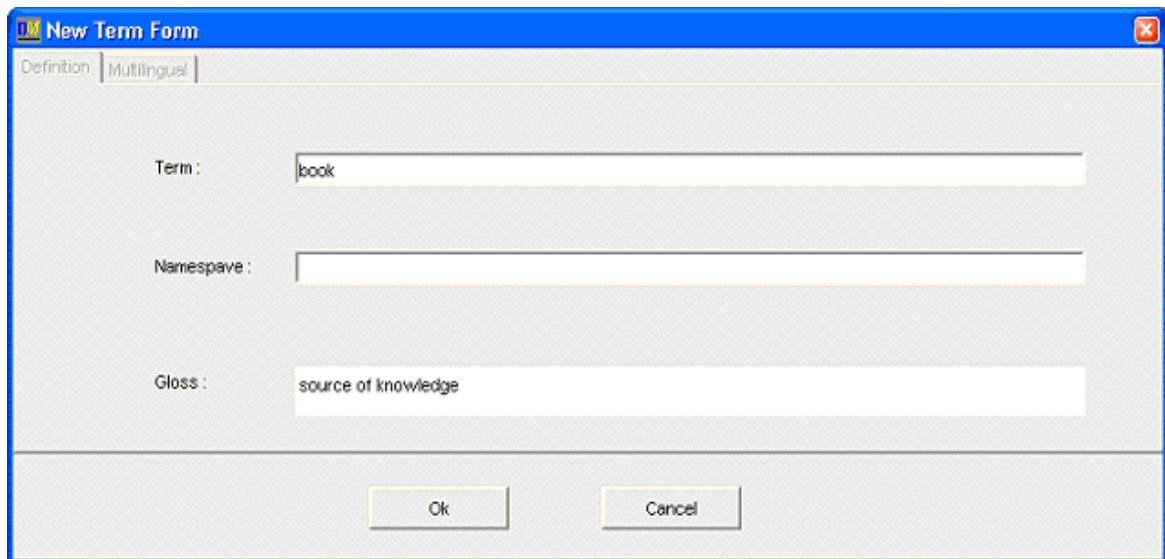


Figure 18

- 3) The newly added term (in this example “book” - see Figure 19) will appear in the context you’ve previously selected (in this example “MyNewContext” – see Figure 17).

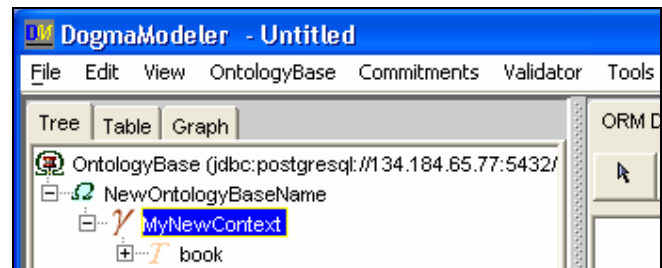


Figure 19

## 4.7 Deleting a term

Select with the right mouse button the term node you want to delete. In the popup window select “Delete Term” (see Figure 20).

→ Note, that the action “Delete Term” will delete all the lexons in which the term selected for deletion is involved either through a role or a co-role !

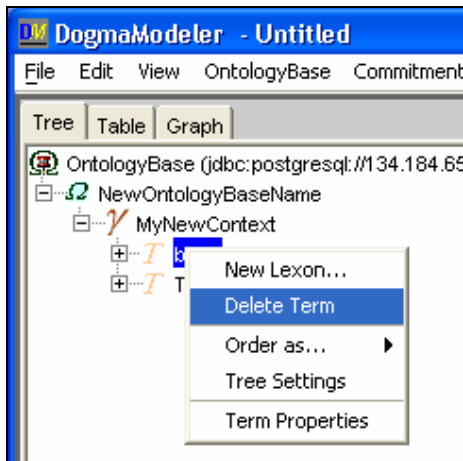


Figure 20

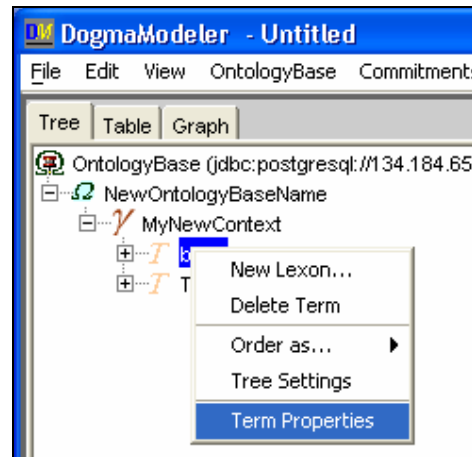


Figure 21

## 4.8 Editing term properties

- 1) Select with the right mouse button the term node you want to edit. In the popup window, select “Term Properties” (see Figure 21).
- 2) The “Term Properties” dialog box consists of two tabs, each showing different properties. Note that the default language (as it has been set for the entire ontology base – see Figure 6) determines the language variant that is shown under the definition tab.

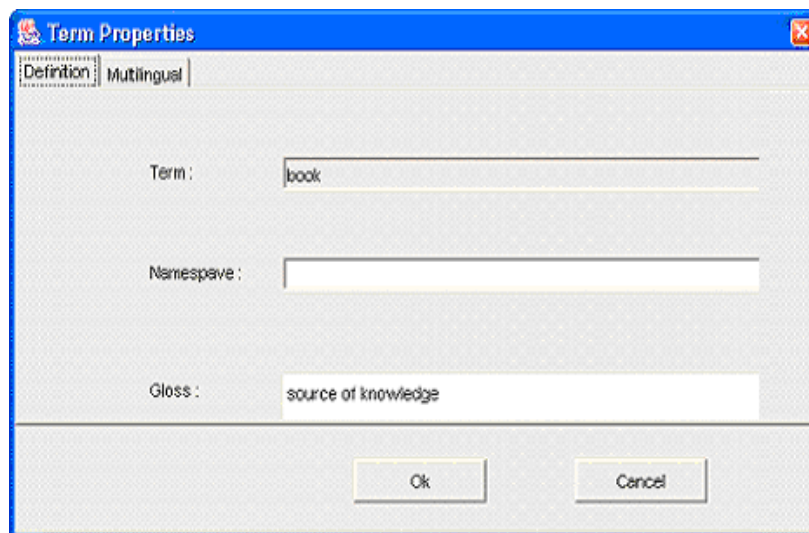


Figure 22

- a. The “Definition” tab: to edit the namespace (basically the origin or source of the gloss) and gloss of the term (see Figure 22). For the moment, the namespace is not considered.
- b. The “Multilingual” tab: to add/delete/edit translations of the term (see Figure 23) if wanted or needed. Click on “Apply” to have your changes become effective.

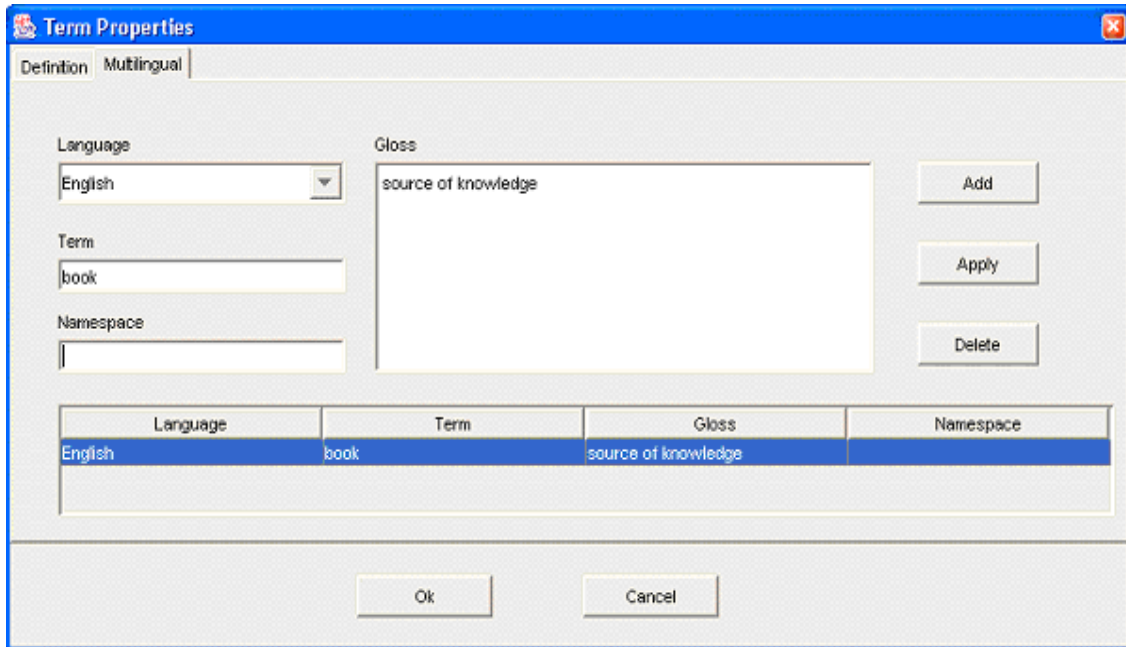


Figure 23

#### 4.9 Creating/Deleting a lexon

- 1) Select with a click of the right mouse button a term node to which you want to add a new lexon. In the popup window select “New Lexon” (see Figure 24).
- 2) Now three combo boxes appear under the selected term, allowing a user to choose from the roles and terms already present in the ontology base. If the user wants to add new ones, (s)he has to select “New Role” or “New Term” appearing as the first choice in the respective combo boxes. Figure 25 and Figure 26 show a user who has chosen to add “Has” as the role, “IsOP” as the co-role, and “title” as the second term of the new lexon.

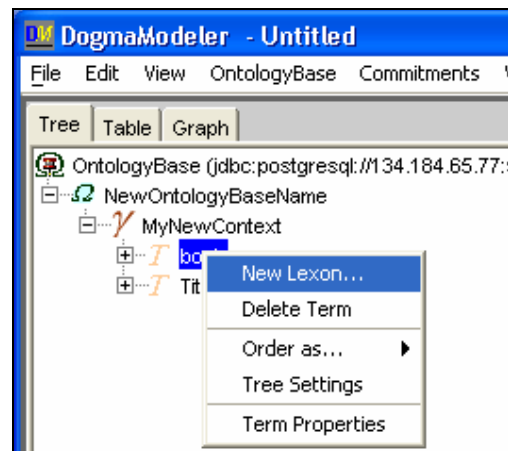


Figure 24

→ If you’ve changed your mind and want to undo the creation of a new lexon, position the mouse over the last combobox, click with the left mouse button and then click on the Escape key. Once the new lexon has been completely created (see below), use the “Delete Lexon” option (see Figure 28).



Figure 25

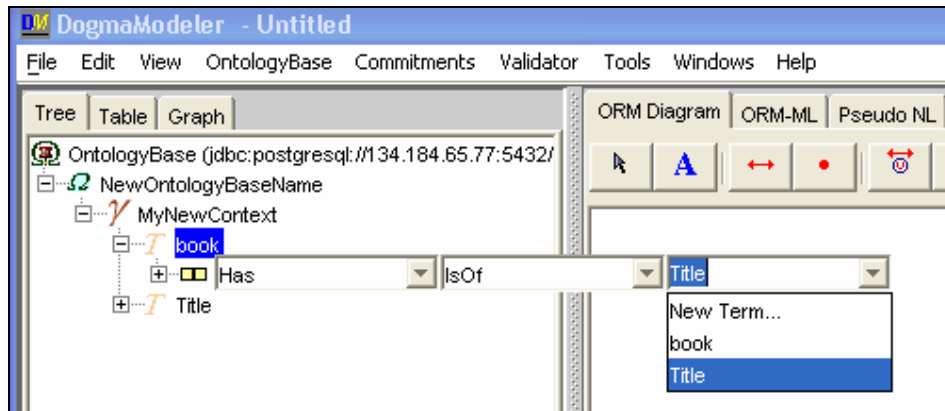


Figure 26

- 3) After a role, co-role and second term have been chosen, the user has to position the cursor above the 3<sup>rd</sup> combobox, click, and press the “Enter” button:
- 4) The new lexon will appear as linked to the chosen head term (see Figure 27). The “inverse” lexon (i.e. head term becomes second term and vice versa, role becomes co-role and vice versa) is also added to the context.

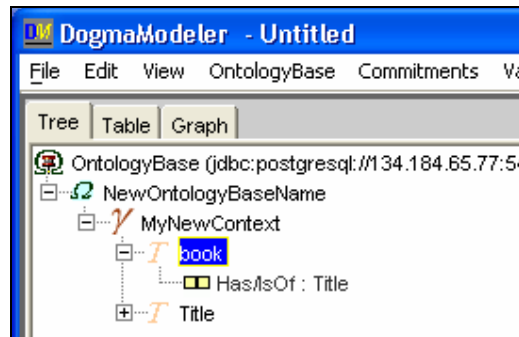


Figure 27

#### 4.10 Editing lexon properties

In order to change the name of a role (or co-role) of a lexon, a user can click with the right mouse button on the lexon node (s)he wants to edit. In the popup window, (s)he selects “Properties...” and chooses the “Role” or “CoRole” (see Figure 28).

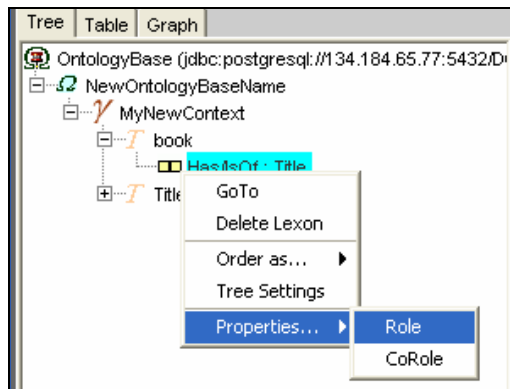


Figure 28

## 4.11 Searching a term in the ontology base

- 1) When an ontology base contains a large amount of terms in different contexts, it might be difficult to find a term by browsing the tree with the mouse. With the special “Search” box (see Figure 29), a user can easily search for the head term of the lexon (s)he needs. The buttons “<” and “>” stand for the “find previous” and “find next” commands respectively. In practice, it equals to searching in the next or previous context as all the lexons with the same head term are linked to that head term (meaning that one head term is displayed per context). A search history is kept and can be reached by opening the drop down menu next to the search entry box.

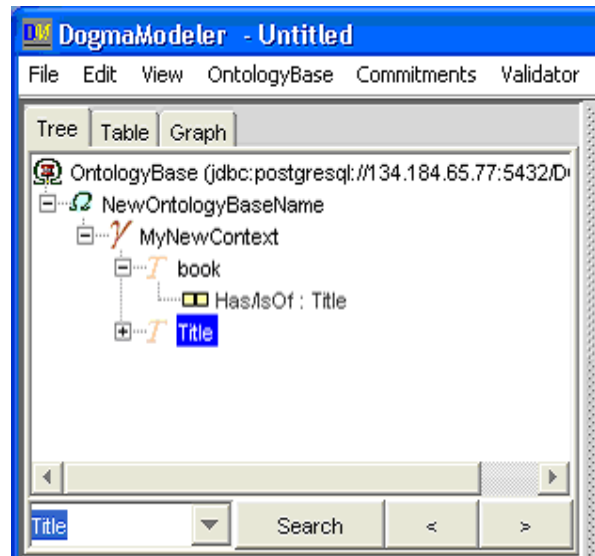


Figure 29

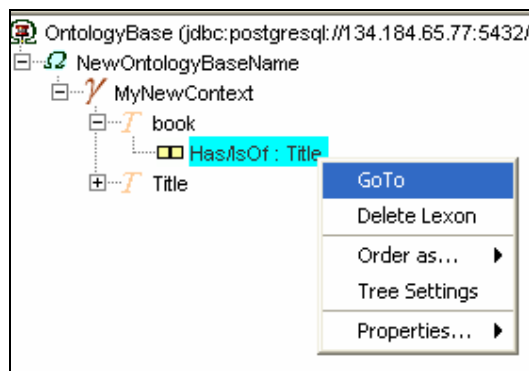


Figure 30

- 2) Select a lexon node linked to a head term, click on the right mouse button, and select “GoTo” option to move back and forth between a lexon and its “inverse” (see Figure 30).

## 4.12 The tree representation

The ontology base tree can be represented in “Ascending Order”, “Descending Order” or “Unordered” (= order of lexon creating) order. Click with right mouse button on any place in the ontology base tree window, and select in the popup window “Order as...” and choose between “Ascending”, “Descending” or “Unordered” (see Figure 31).

NOTE: all changes to an ontology base are immediately stored in the database. Therefore, there is no specific “save ontology base” command. You can simply close the database connection (“Close OntologyBase”).

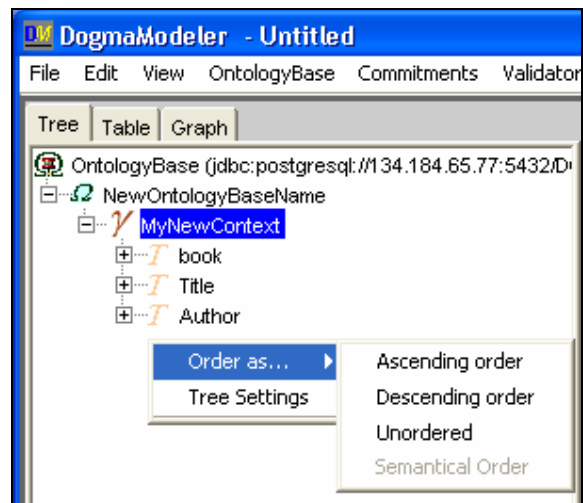


Figure 31

## 5 The Commitments

### 5.1 Opening a commitment library

→ Note that commitments will be displayed only if an ontology base has been opened previously.

- 1) In the menu “Commitments” you need to choose “Load Commitment(s)” and “From Database” (see Figure 32)

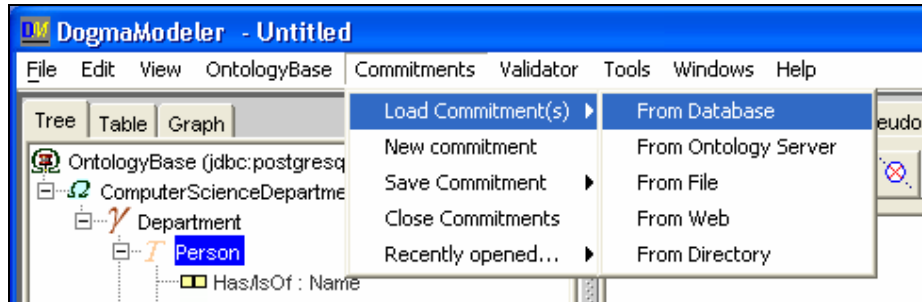


Figure 32

- 2) The Commitment database connection window will appear (if you don't know exactly what information you have to enter, just click on “Default” (i.e. your last successful connection) (see Figure 33)

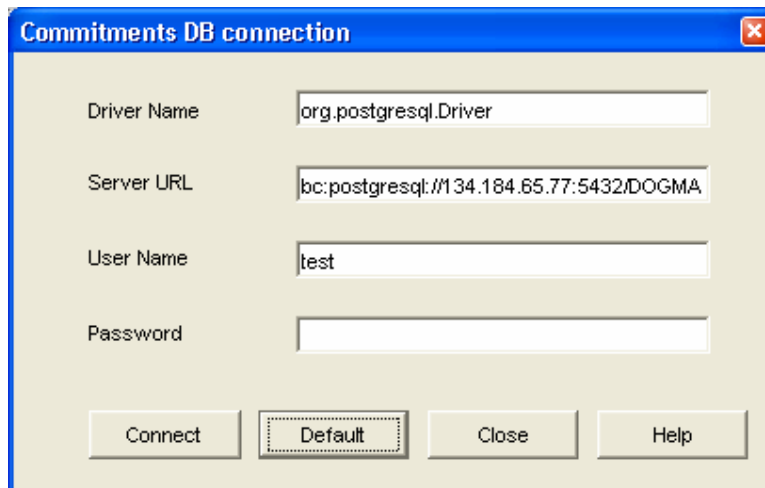


Figure 33

- 3) The commitment library window will show the list of commitment rules that commit to the ontology base just opened (see Figure 34).
- 4) When you expand a commitment node, all its lexons and verbalised constraints (see Figure 35) are displayed.

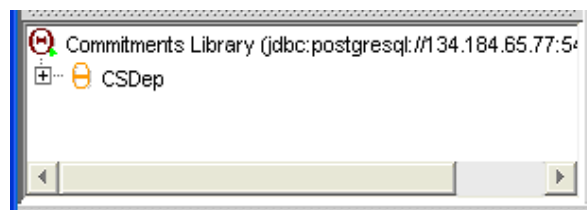


Figure 34

## 5.2 Deleting a commitment

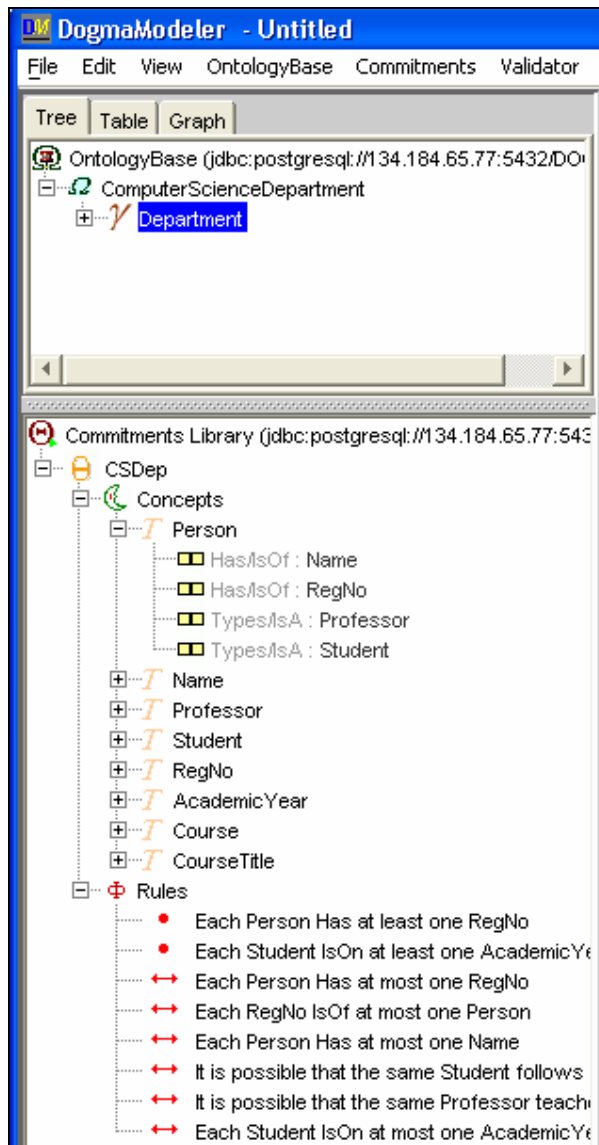


Figure 35

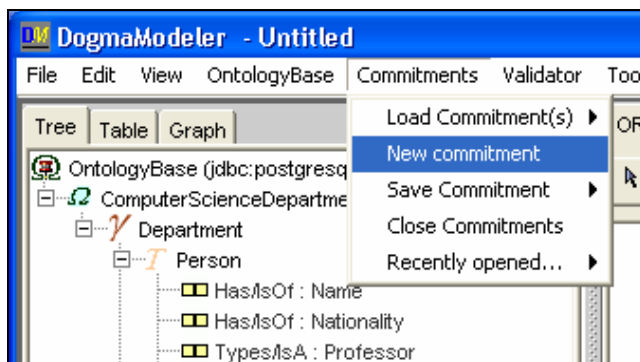


Figure 37

1) To prevent a commitment from being deleted or changed, a commitment can be locked. Click with right the mouse button on a commitment node and set the “Set commitment *Department* locked” by marking the checkbox (see Figure 38).

2) In order to delete a commitment, click with the right mouse button on the commitment node and the popup window will appear. Choose “DeleteCommitment”. Unless the commitment is locked (cf. supra), in which case one cannot delete a commitment (see Figure 36), the commitment will be deleted.

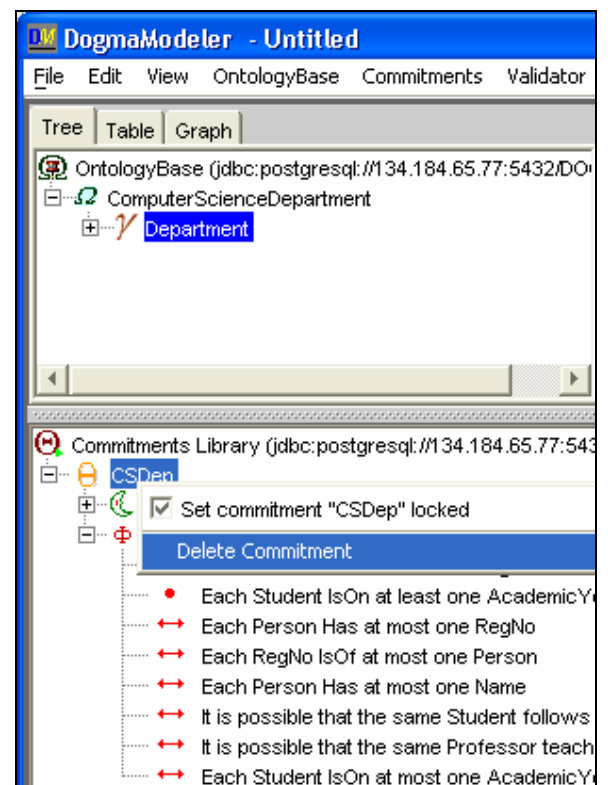


Figure 36

### 5.3 Creating a new commitment

- 1) Don't forget to open an ontology base first (see section 0). Otherwise, an error message will appear in the error message area.
- 2) From the menu select "Commitments" and then choose "New Commitment" (see figure Figure 37).

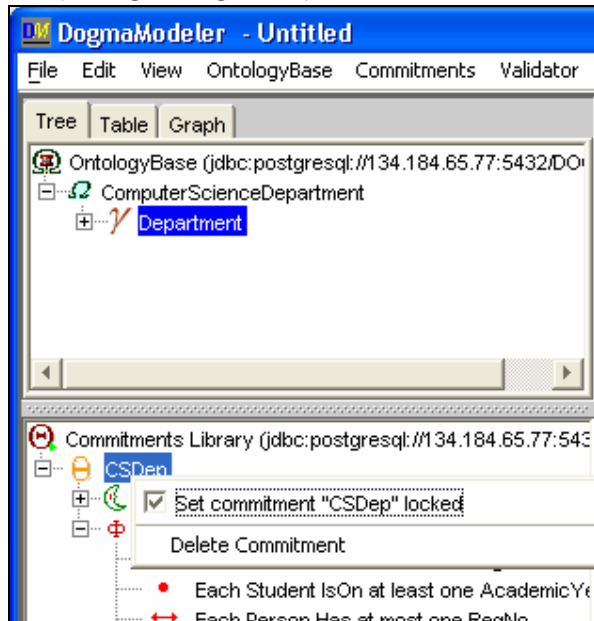


Figure 38

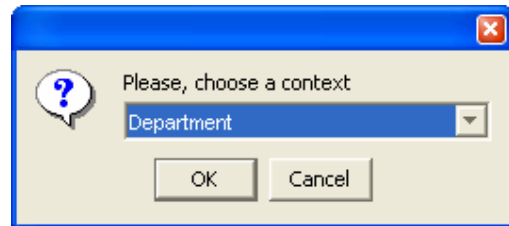


Figure 39

- 3) Select a context in which the lexons appear for which you want to define the new commitment (see Figure 39).  
→ Note that for defining that commitment, you cannot use lexons belonging to other contexts.

- 4) In the "Commitment Meta Data" form you must provide some meta information to characterise the new

commitment. The most important field are the name ("Title" – see Figure 40) and the "Language" (that has already been set via the default language option – see Figure 6).

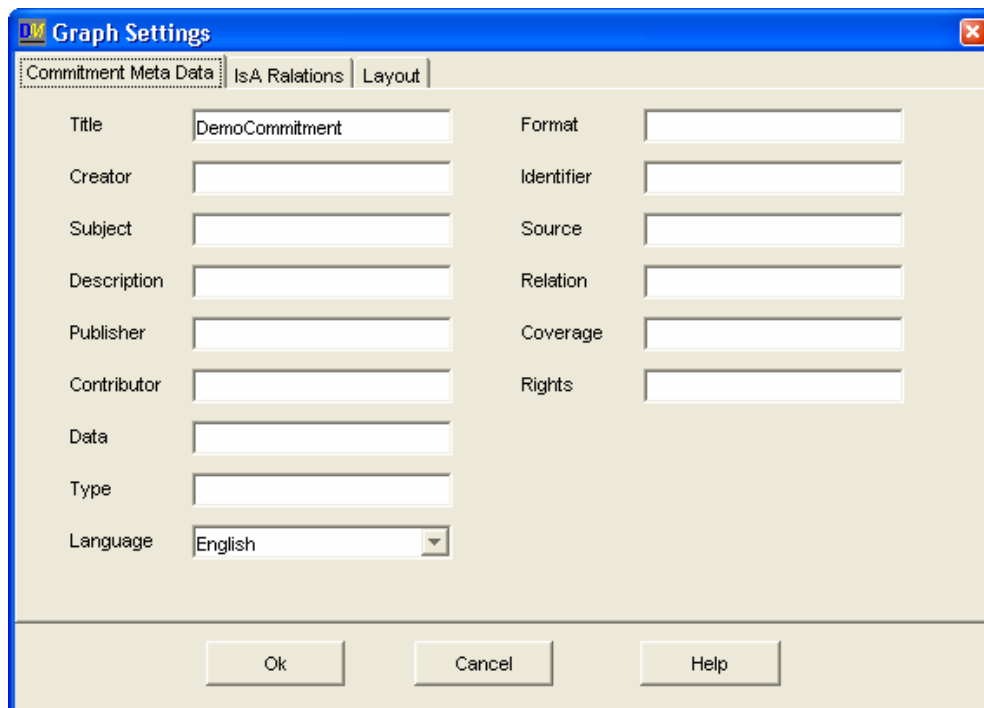


Figure 40

- The lexons (belonging to the context selected) for which you want to define a commitment can be dragged and dropped one by one from the ontology base tree to the commitment modelling area. Click with the left mouse button on a lexon node, drag it to the desired spot in the commitment modelling area and finally release the left mouse button (see Figure 41).

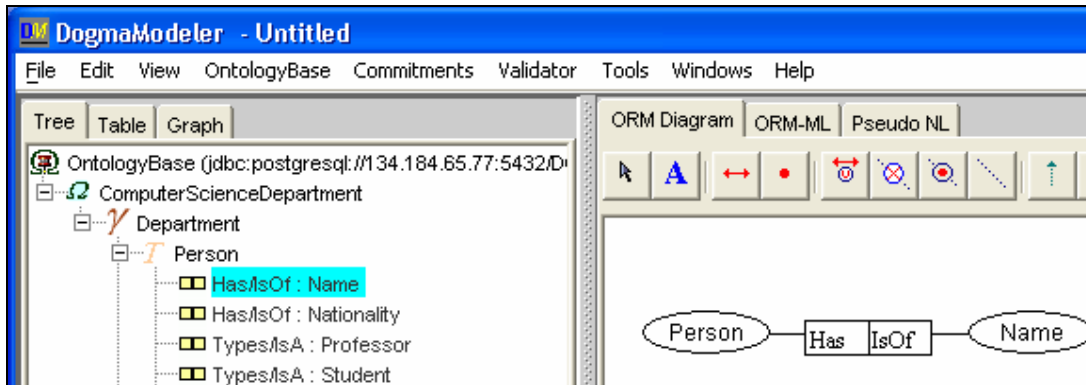


Figure 41

- By keeping the Ctrl+Shift keys down and by using the arrow keys on your keyboard you can move a graphical object pixel by pixel (see Figure 42). Of course, you have to select first the desired object or set of objects.

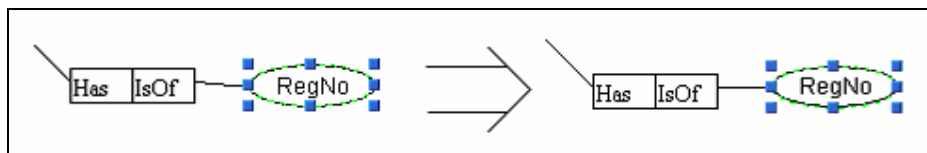


Figure 42

## 5.4 Graphically representing a commitment

Next to the specific graphical conventions that are used to represent semantic constraints (see section 5.5), the lexons selected by a commitment can be represented in some varying ways and styles. We refer to the ORM book by Halpin for a more detailed explanation.

### 5.4.1 Subsumption relations

A subsumption relation (is\_a, IsA, subclassOf etc.) is normally drawn by means of a box for the two roles (see Figure 43).

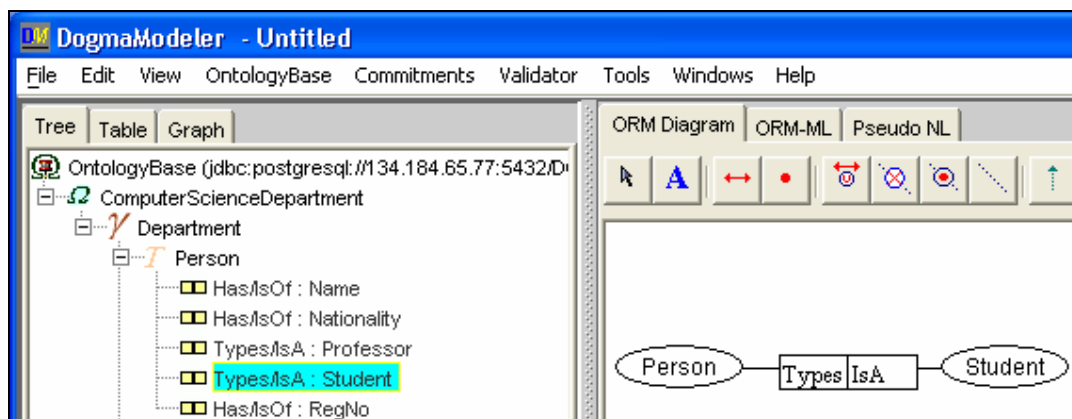


Figure 43

Alternatively, a subsumption relationship can be represented by an arrow pointing from the child to the parent object (see Figure 44). In this case, you have to specify your subsumption relationship.

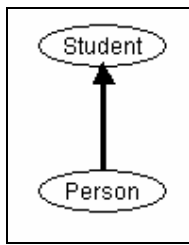


Figure 44

1) Therefore, click with right mouse button in the commitment modelling area. In the popup window that is displayed subsequently, select “Graph Settings” (see Figure 45)

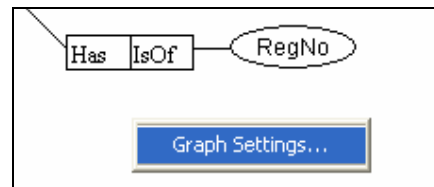


Figure 45

2) Select in the dialog window that now appears the “IsA Relations” tab (see Figure 46). Enter the role/co-role relationship labels in the text field and click on “Add”. As a result, the newly entered relationship label will be shown in the list of IsA relationships defined. When dragging lexons to the commitment modelling area, the subsumption relationship will be displayed by means of an arrow.

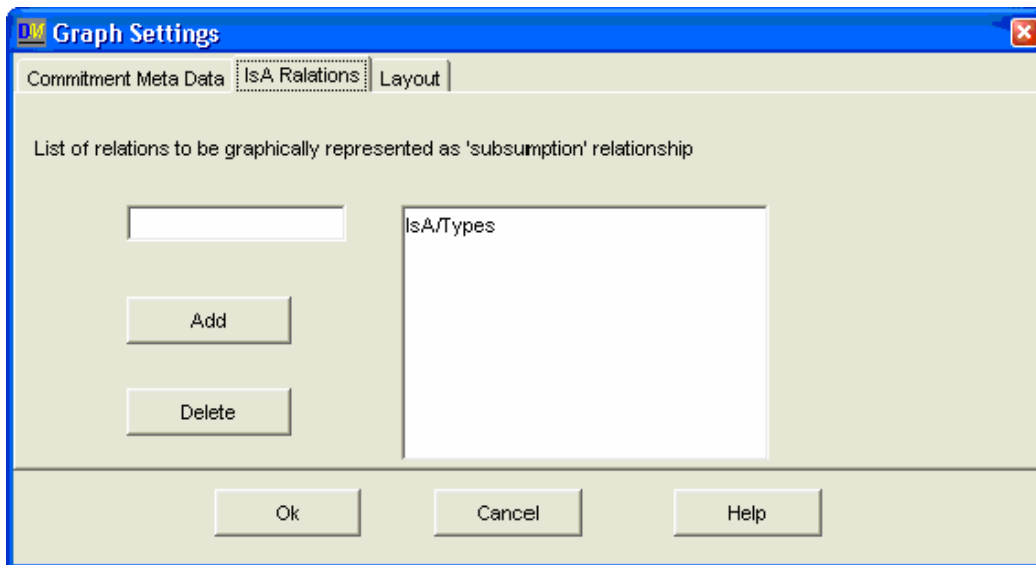


Figure 46

→ Note that the role should only be a subtyping relation (e.g., is\_a, IsA, subclassOf etc...) and the co-role must represent a supertyping relation.

### 5.4.2 Relationship properties

To reverse roles (e.g. of Figure 47), click with the right mouse button on a relationship. In the popup window, select “Reverse” (see Figure 48). The result is shown in Figure 49.

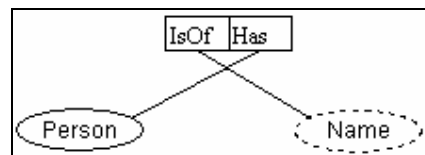


Figure 47

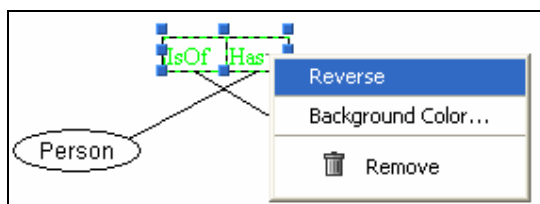


Figure 48

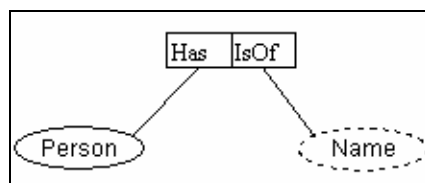


Figure 49

### 5.4.3 Object properties

An object can have specific properties and values. In order to edit them, click with the right mouse button on the object of which you want to edit/modify the properties. A pop-up window with four options is now displayed .

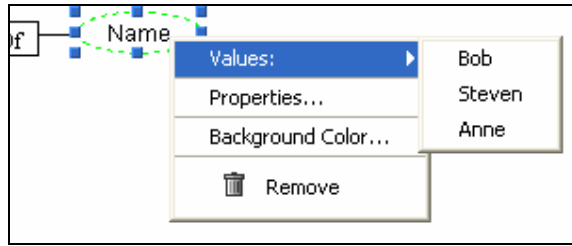


Figure 50

- 1) The trash bin to delete the object.
- 2) Select “Values:” to view the values that have been defined for the object (see Figure 50).
- 3) Select “Background Color” to change the background for an object (see Figure 51). Just click on the colour (see Figure 52) you want the selected object to have as background (see Figure 53). This also applies for relationships.

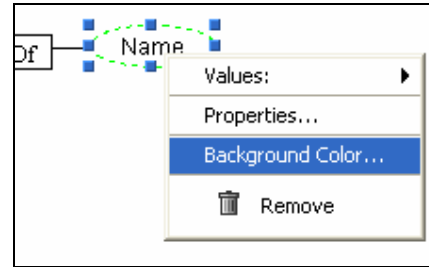


Figure 51

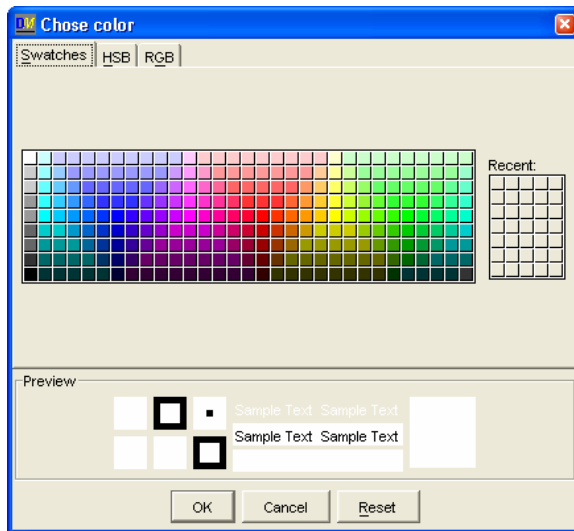


Figure 52

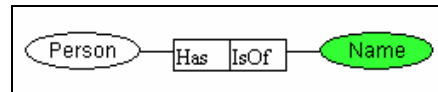


Figure 53

- 4) Select “Properties” to view/edit” object properties (see Figure 54). The “Object Type Properties” window having four tabs now appears (see Figure 55):

- i) “Definition” – you can view the definition of the concept represented by this object and you may define this object as LOT/NOLOT (“Lexical Type”). In principle, it is more correct to always define it as a NOLOT. As a consequence, the “Data Type” tab (see

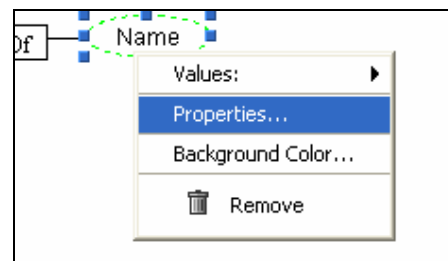


Figure 54

below) information should not be considered. A “Gloss” gives an explanation on the object, and. the “namespace” specifies the source and origin of the gloss.

The image shows a software dialog box titled "Object Type Properties". It has a blue title bar with a close button in the top right corner. Below the title bar is a tabbed interface with three tabs: "Definition", "Multilingual", and "Data Type". The "Definition" tab is currently selected. The main area of the dialog contains the following fields:

- Term :** A text input field containing the word "Name".
- Namespave :** An empty text input field.
- Gloss :** A large, empty rectangular text area for entering a gloss.
- Lexical Type :** A dropdown menu with "LOT" selected.

At the bottom of the dialog, there are two buttons: "Ok" and "Cancel".

**Figure 55**

ii) the “Multilingual” tab: to view various translations of a concept.

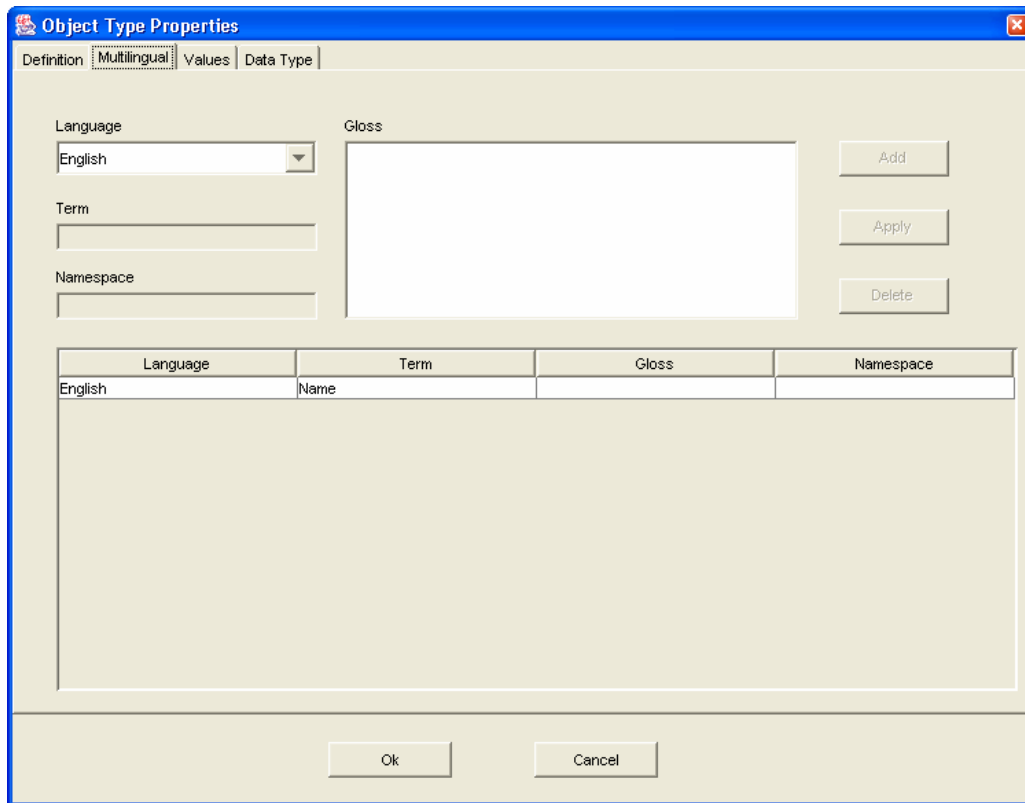


Figure 56

iii) the “Values” tab: to define values for the object (see also Figure 50).

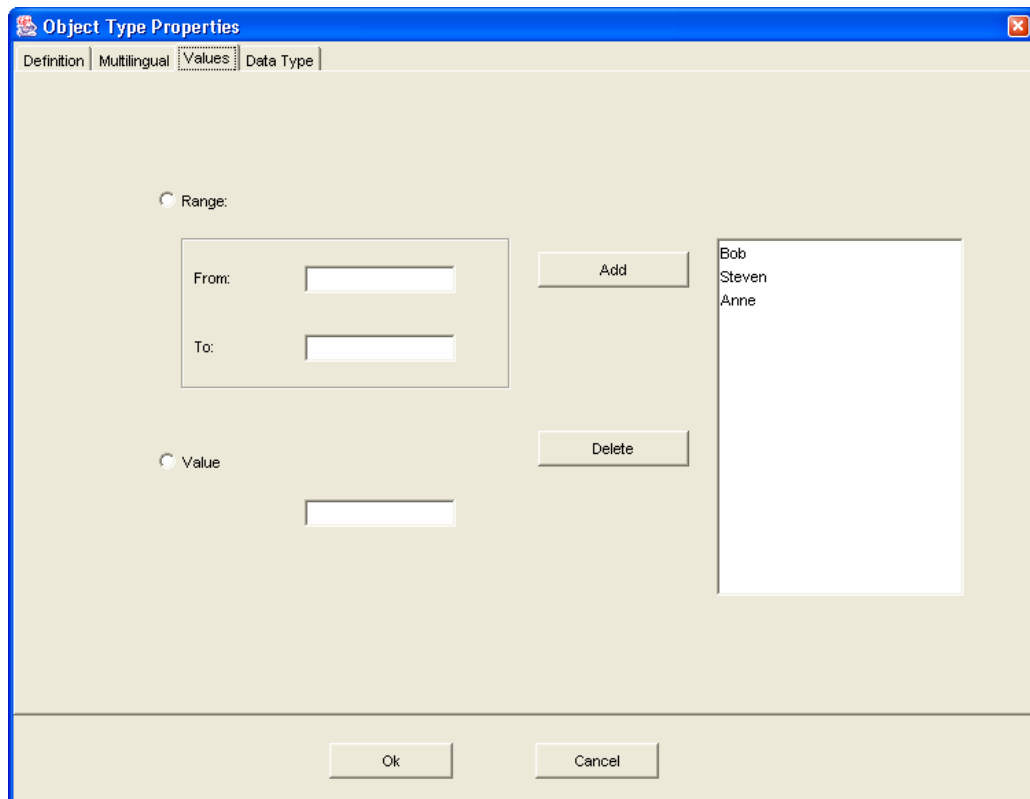
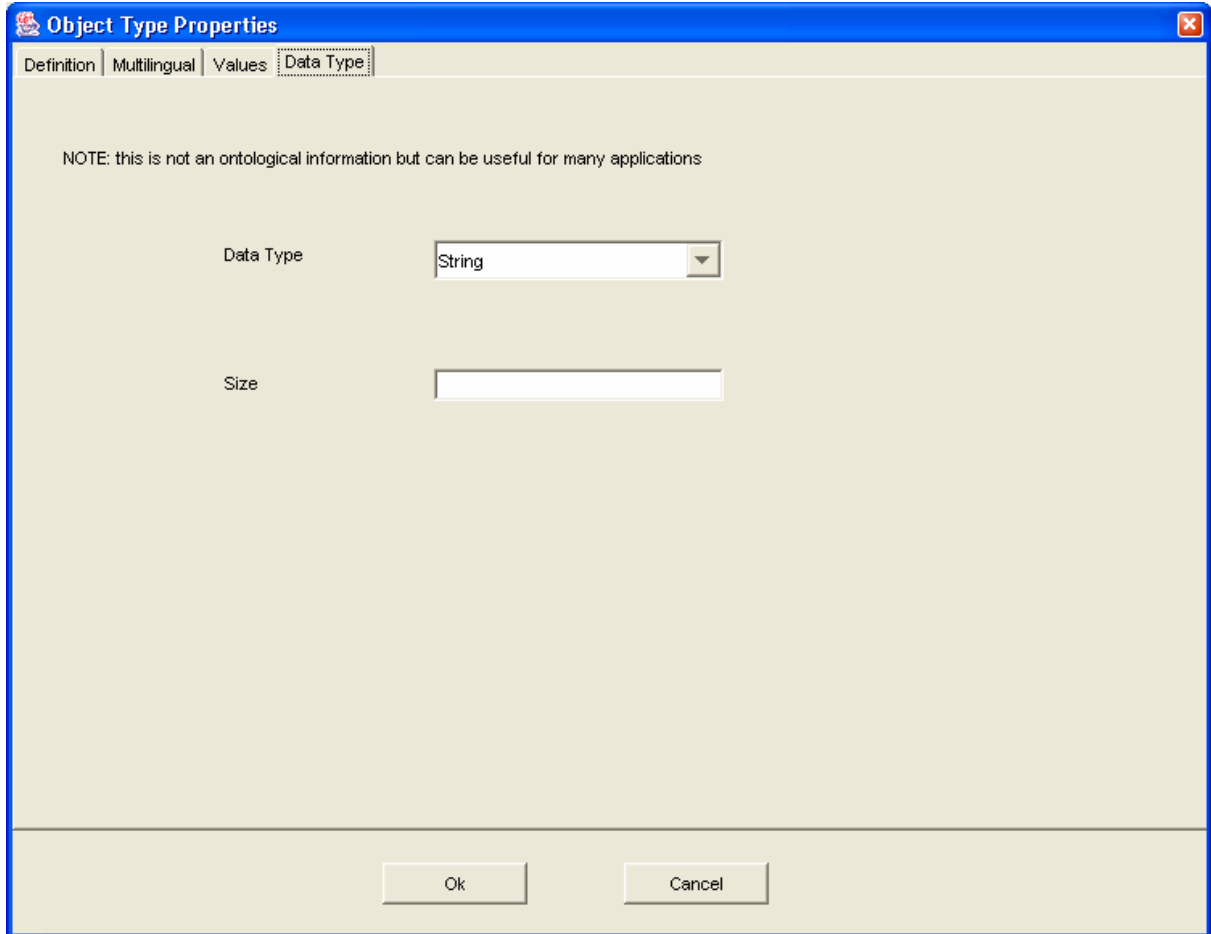


Figure 57

- iv) the “Data Type” tab: to define a data type for the selected object. As this kind of information (only relating to LOTS) falls outside of the realm of the ontology strictly speaking, we don’t add more details here for the time being.



**Figure 58**

## 5.5 Defining semantic constraints

In this section, an explanation is given on how to define commitments, i.e. adding semantic constraints on the particular lexons selected from the lexon base. The (graphical) style closely resembles the Object Role Modelling language. An overview of the most important ORM constraints that will be used for ontology engineering in “the DOGMA way” follows. We refer to the Halpin book for the exact meaning/definition of each constraint. However, keep in mind that the book on ORM explains on how to model a database schema (and not an ontology). In our case, we are clearly modelling an ontology, so some discrepancies will be unavoidable.

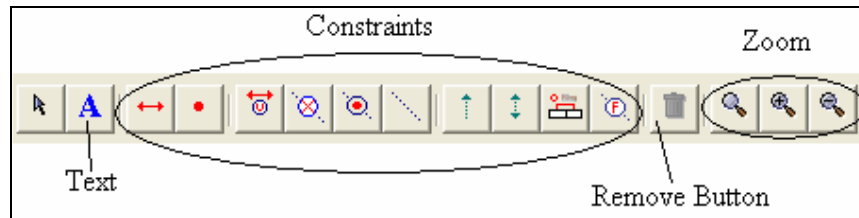


Figure 59

The toolbar (see Figure 59) of the commitment definition pane displays 16 icons of various nature:

1. general icons:
  - a) the arrow and text icons that allow to toggle between text-editing and drawing mode
  - b) the garbage bin: by clicking on it, the previously selected object is erased.  
→ Beware: an undo functionality is not yet available !
  - c) 3 magnifying glasses: to zoom in or out
2. 10 specific icons that stand for an ORM constraint (see the following subsections).

### 5.5.1 Mandatoriness

Select on the tool bar the icon that represents the mandatory constraint and click on the role on which you want to impose a mandatory character. In order to remove the mandatoriness, select the mandatory constraint icon, click on the right mouse button and select “remove”.

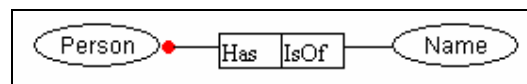


Figure 60

### 5.5.2 Uniqueness , Subset , and Equality

Select on the tool bar the icon that represents the uniqueness, subset or equality constraint.

When you want to put the constraint on two roles, position the mouse above the first role, click on the left mouse button, drag the mouse to the second role, and then release the mouse button. If, in the case of the uniqueness constraint, you only want to select one role, you only want to position the

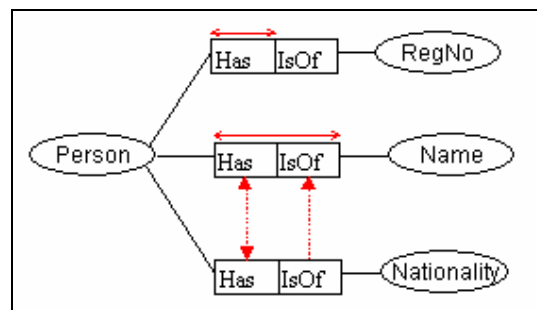


Figure 61

mouse above the (single) role concerned and click on the left mouse button.

### 5.5.3 External uniqueness , external mandatory , and exclusion

- 1) Select on the tool bar the icon that represents the external uniqueness, external mandatory or exclusion constraint. Position the mouse on a point where you want to insert the constraint icon in the graphical diagram and subsequently click on the left mouse button.

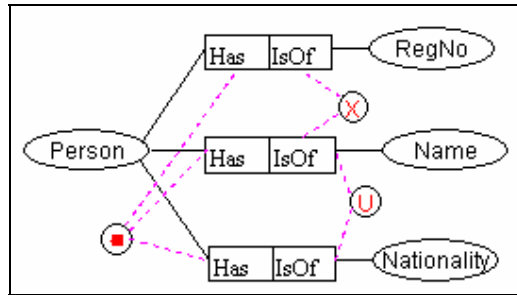



Figure 62

- 2) Select on the tool bar the “external constraint” connector (represented by ) and connect the constraint with the roles (one at a time). With subtype relations, you can select roles with the shift key in combination with left mouse button clicks. Then select the icon (the connection is made in the middle).

### 5.5.4 Total- and Exclusive - constraints

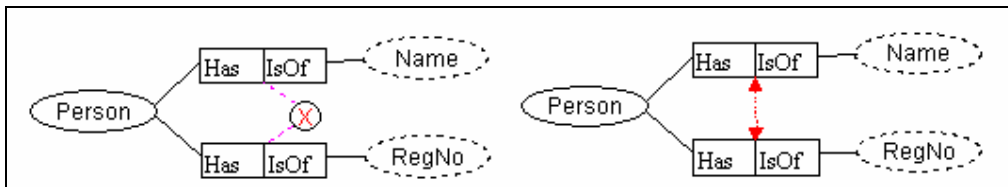




Figure 63

To define graphically an exclusive and total constraint between (two) subtype relations (see Figure 64), you need to select these relations (using Ctrl key + left mouse button) and click on  or  in the Tool Bar:

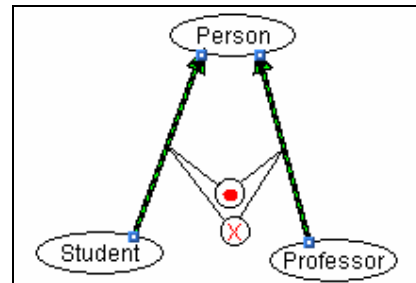


Figure 64

### 5.5.5 Ring constraint

Select from the tool bar the icon that represents the ring constraint (see Figure 65) and click on the role you want to constrain.

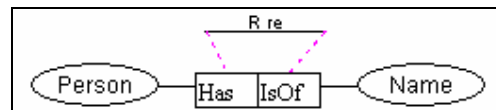


Figure 65

Click with the right mouse button on the ring constraint and in the popup window that appears you need to specify the characteristics of the ring constraint (see Figure 66).

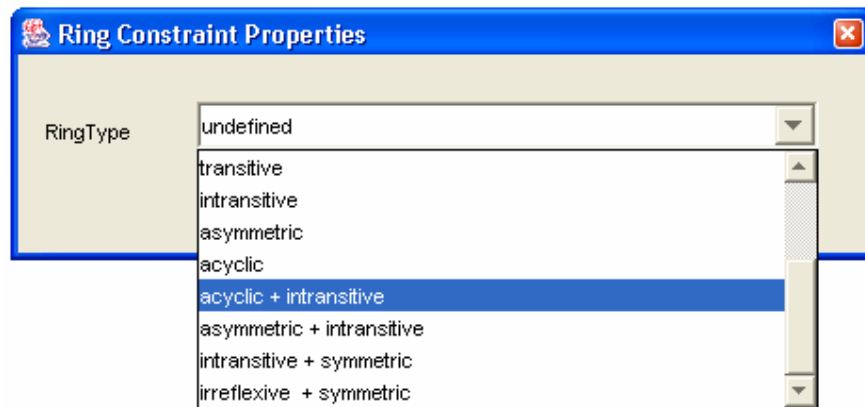


Figure 66

### 5.5.6 Frequency

1) Position the mouse on a point where you want to insert the constraint icon in the graphical diagram and subsequently click on the left mouse button.

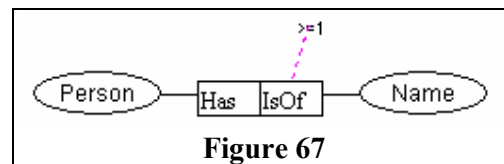



Figure 67

2) Select on the tool bar the “external constraint connector” icon (displayed as ) and connect the constraint with a role.

3) Click now with the left mouse button on the constraint and in the popup window that appears you can specify the frequency bounds that apply (see Figure 68).

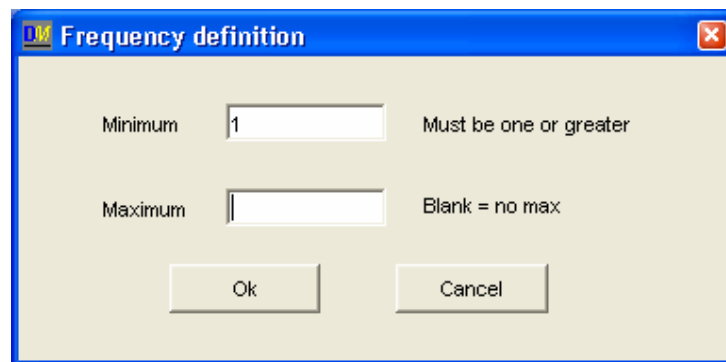


Figure 68

## 5.6 Deleting a constraint

Locate the mouse above icon representing the constraint in the commitment modelling area, click with the right mouse button and choose the "remove" option.

## 5.7 Saving a Commitment

- 1) In the menu "Commitments" select "Save Commitment":
  - If you choose "Save to file" or "Save as ... to file", you have to provide a file location where you want to store the commitment (see Figure 69).

- “Save to DB” stores the commitment into the database to which you are currently connected (see item 2) of section 5.1). Subsequently, the commitment library displays the commitment just stored (see Figure 70).



**Figure 69**



**Figure 70**

Acknowledgments: Thanks to Jan, Mustafa and Robert for lending a page. The implementation of the tool is entirely the work of Andriy with Mustafa guiding him.