D.B.E.
Digital Business Ecosystem

Contract No: 507953

WP 15: DBE Business Modeling Language

D15.2: BML Editor 2nd Release

Project funded by the European Community under FP6

Contract Number: 507953
Project Acronym: DBE
Title: Digital Business Ecosystem

Deliverable N°: D15.2
Due date: 31/08/2005
Delivery Date: 16/09/2005

Short Description:
This document accompanies the Second Release of the BML Editor for the DBE project [1]. This editor will be used by Business Analysts on behalf of SMEs for defining business and service models according to the corresponding metamodels. The BML Editor as a visual modeling tool provides a UML-like Graphical User Interface (GUI), similar to that of well known UML editors, for supporting the modeling tasks and stores the created models in the DBE **K**nowledge **B**ase (KB).
Partners owning: TUC
Partners contributed: TUC
Made available to: All project partners and the EC

Versioning			
Version	**Date**	**Author, Organization**	**Description**
0.1	**16/09/2005**	**GEORGE ANESTIS - TUC**	Initial Document Creation

Quality check
1st Internal Reviewer: Not applicable
2nd Internal Reviewer: Not applicable

Table of Contents

1 Introduction

This document describes the Second Release of the BML Editor for the DBE project [1]. This editor will be used by business analysts on behalf of SMEs for defining business and service models according to the corresponding metamodels. The BML Editor as a visual modeling tool provides a UML-like Graphical User Interface (GUI), similar to that of well known UML editors, for supporting the modeling tasks and stores the created models in the DBE **K**nowledge **B**ase (KB). The current version of the editor supports both, the semantic description of the services offered by an SME and the business model of the particular SME. The former provides to the user the ability to create service models according to the **S**emantic **S**ervice **L**anguage metamodel [2,3] and the latter to create business models based on the **B**usiness **M**odel **L**anguage (BML) metamodel [5]. Both metamodels are described using OMG's MOF 1.4.

During the modeling process the user can take advantage from domain specific ontologies that have been described using the **O**ntology **D**efinition **M**etamodel (ODM) [2,3] and stored into the DBE Knowledge Base.

This document does not intend to describe the SSL and BML metamodels but only the usage of the BML Editor for the creation of these models. Consequently, it is highly recommended the reader of this document to read first the documents that describe both the SSL and the BML metamodels.

2 Technical Requirements

The BML Editor is a graphical, UML-Like Editor allowing for modeling business and service models. The adoption of eclipse as the main platform for the Service Factory Environment of DBE posed the technical requirement of developing the BML Editor as an eclipse plug-in that will be plugged as all other tools into the same platform (DBE Studio). As a consequence, the BML Editor is being developed as an Eclipse plug-in based on Eclipse 3.1, and exploiting of the GEF 3.1 and Draw 2D graphical frameworks that have been built for this platform. In the frame of the MDA approach that has been adopted by the DBE project, the BML Editor allows the creation of M1 models as direct instances of the M2 metamodels that have been defined in DBE under the BML umbrella with the use of MOF 1.4 (Meta-Metamodel).

The BML Editor is used to define business and service models. From an architectural point of view, the BML Editor exploits the DBE Knowledge Base for its persistency requirements. Service-Oriented architecture is followed for integrating the BML Editor with the DBE Knowledge Base based on the DBE ServENT platform (Swallow project [6]). The BML Editor connects with a DBE ServENT and looks up into the network to find an appropriate **K**nowledge **B**ase (KB) service. The BML Editor using the BML specific JMI [7] interfaces enables the creation of M1 Models and using the functionality of the KB-service is able to store, retrieve and update BML models in the KB.

3 Installation Instructions

In order to use BML Editor you must have installed Eclipse 3.1 as well as GEF 3.1. BML Editor is an

eclipse plugin that is dependent on some additional plugins. All the required stuff can be downloaded from SourceForge (http://cvs.sourceforge.net/viewcvs.py/dbestudio/dbestudio/studio-editors/bml/) or from TUC/MUSIC site at http://www.music.tuc.gr/DBE/soft/bml-editor. In particular you have to download the following plugins:

- org.dbe.studio.editors.bml_2.0.0.zip
- org.dbe.studio.tools.kb-toolkit_1.0.0.zip
- org.dbe.studio.tools.ontologyviewer_1.0.0.zip
- org.dbe.studio.tools.qf-sdt_2.0.0.zip
- org.dbe.studio.core.preferences_0.1.0
- org.dbe.studio.core.help_0.1.0 (this help)

Unzip the above files in the <ECLIPSE_HOME>/plugins folder and restart Eclipse.

4 User Guide

4.1 Starting BML Editor

A DBE BML Perspective has been created for BML Editor. So, the BML Editor can be activated selecting, from Eclipse main menu, Window->Open Perspective->Other->DBE Business Analysis. The BML Editor main window is presented in the following figure:

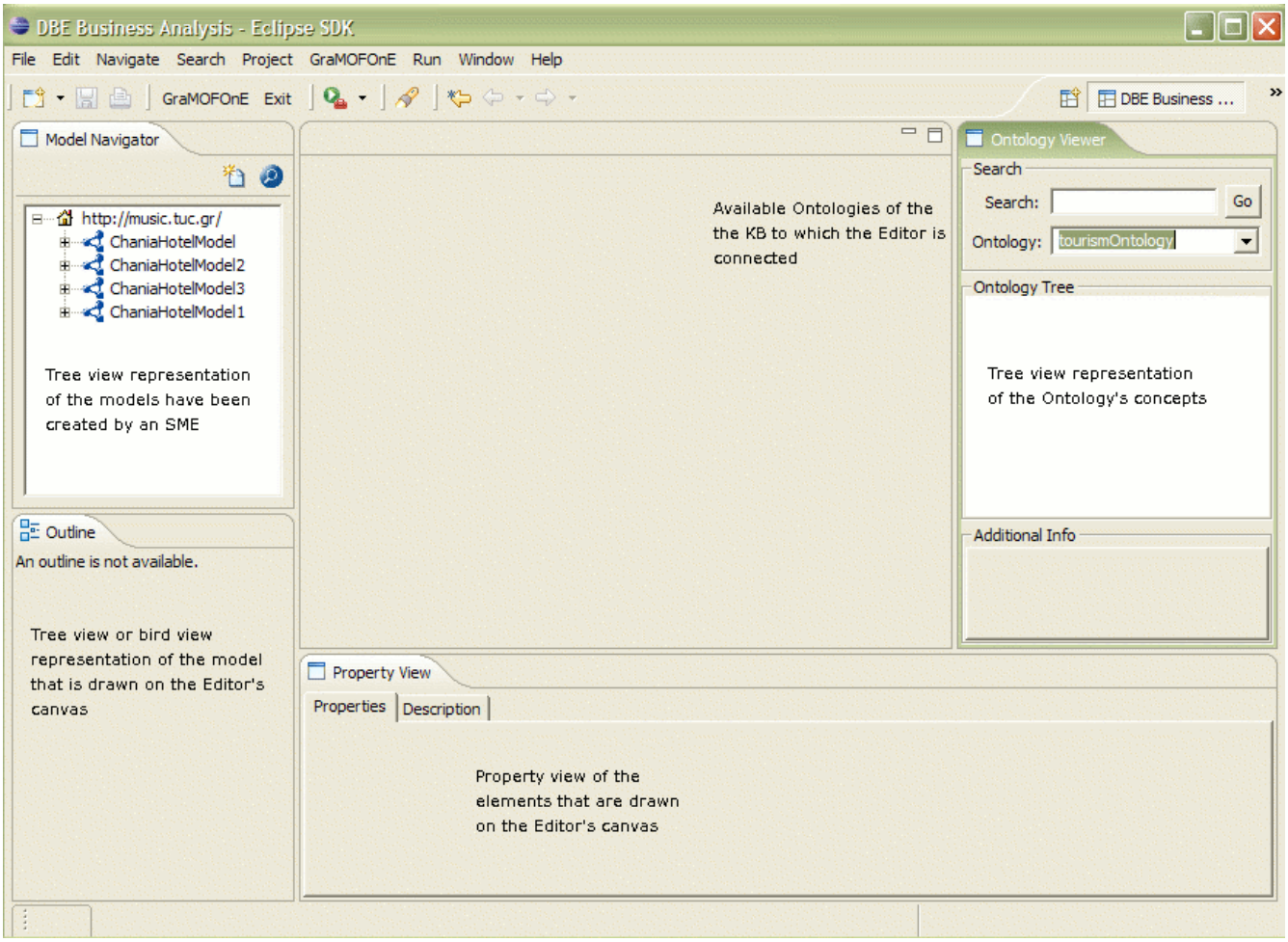

Figure 1 BML Editor Main Window

When the editor is activated it attempts to connect to a particular KB service in order to retrieve the list of models that have been created by a specific SME. An SME is identified by its SME id, a unique identifier for an SME like a URI. By default the BML Editor attempts to connect to the address http://localhost:2728 assuming that there is a local, running, instance of the KB while the SME identifier is set to http://www.mySME.com/. The user can change these values selecting from Eclipse main menu Window->Preferences->DBE Studio->BML Editor like it is shown in the following figure:

4.2 Creating a new model

In order to create a new model the user has to click on the icon in Model Navigator View. The editor window is shown in the next figure:

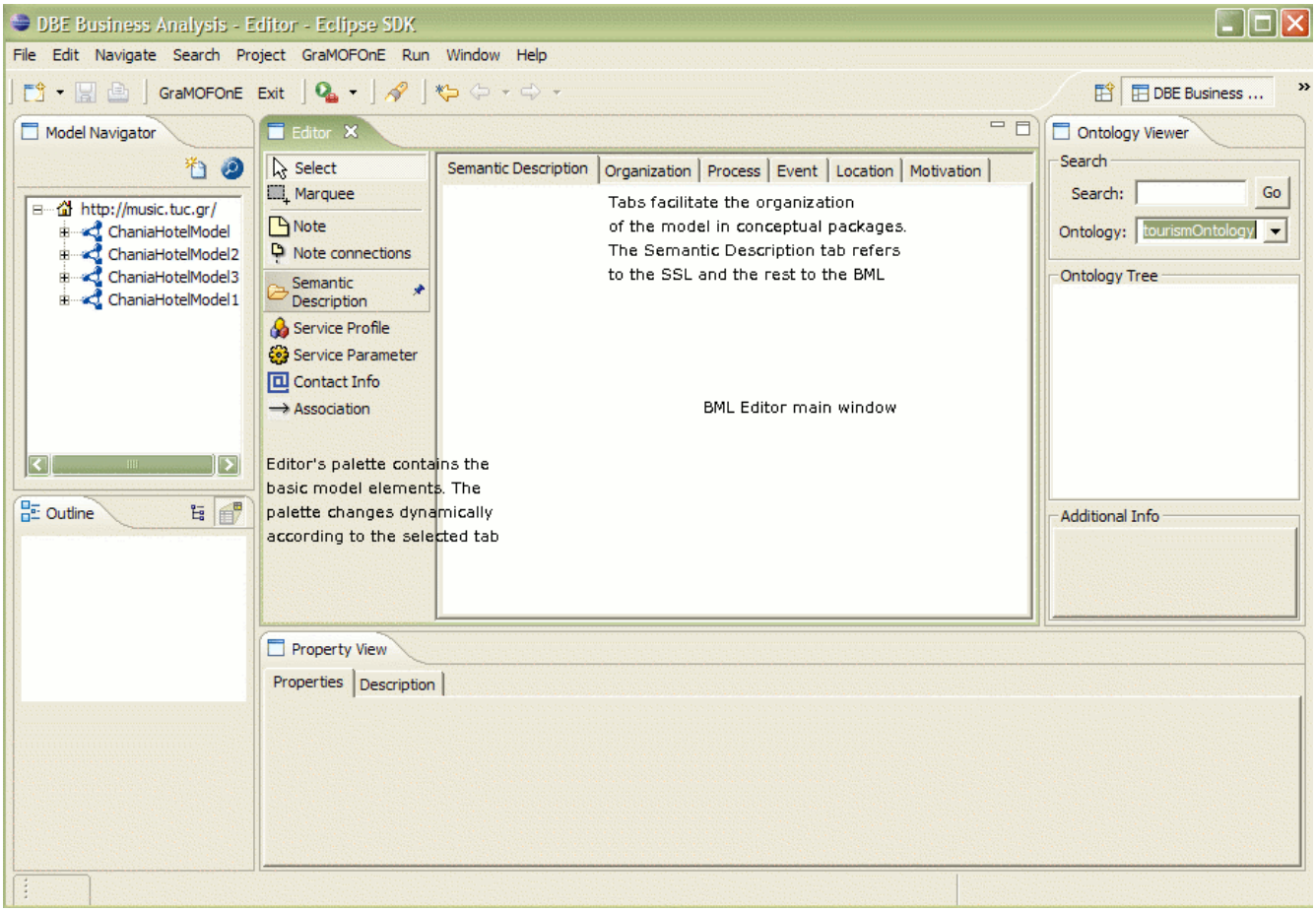

Figure 2: The BML Editor main window

4.3 Creating Semantic Descriptions of Services (SSL)

4.3.1 Creating a ServiceProfile

The first thing a user has to do in order to create the semantic description for a service is to create a Service Profile. This can be done selecting from the Editor's palette the corresponding tool and clicking on the Editor's canvas. A ServiceProfile is visually represented by a rectangle separated in tree areas. The first area (traveling top-down) contains the name of the ServiceProfile, the second one the attributes that a ServiceProfile may contains and the third the functionalities that can be defined as part of the particular profile.

Right-clicking in the area bellow the ServiceProfile name (the attributes area) a menu will appear. Selecting the "Add attribute" item one new attribute will be created. Following the same way the user can add as many attributes as he wants. Similar is the procedure for adding functionalities as shown in the next figures:

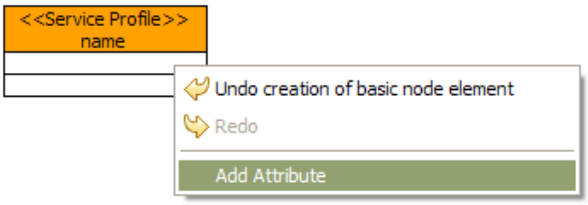

Figure 4: Adding an attribute to Service Profile

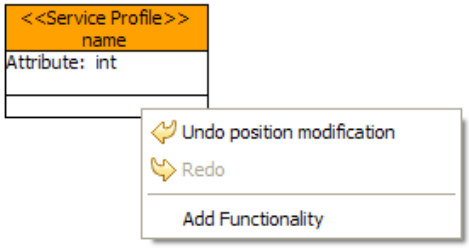

Figure 3: Adding a functionality to a Service Profile

The editing of the profile's name, attribute's name and type as well as functionalities can be done in the Property View selecting the ServiceProfile, the attribute and the functionality correspondingly as shown in the following figure:

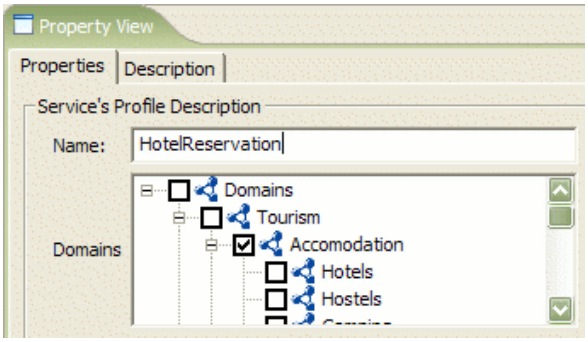

Figure 5: A Service Profile has a name and can be associated with many domains

A service functionality has name and input and output parameters. Each parameter also has name and type. The user can add input ( ) and output ( ) parameters or delete input ( ) and output ( ) ones by selecting the functionality he wants and clicking on the corresponding icons of the Property View. Double-clicking on the corresponding cells in the "*Name*" and "*Type*" columns can edit the name and the type of an input/output parameter as show in the next figure:

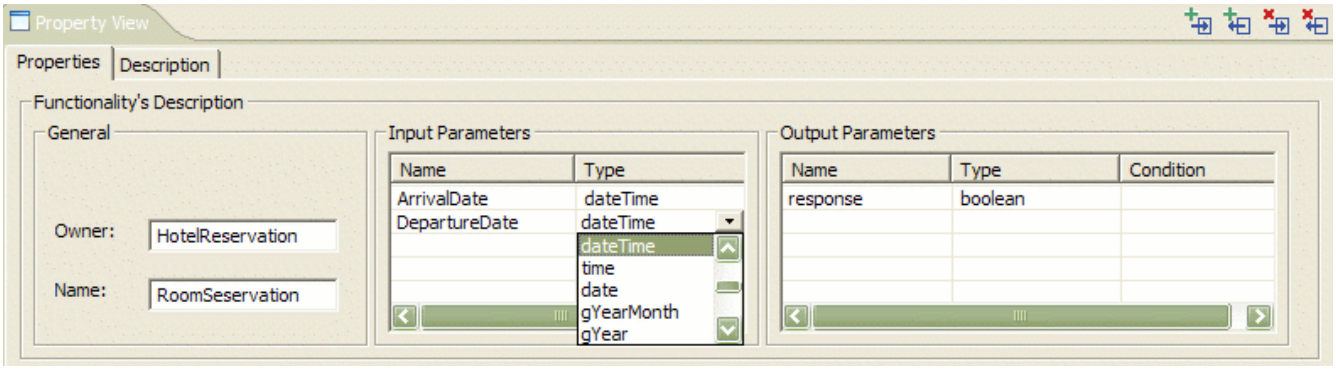

Figure 6: A ServiceFunctionality has a name and input and output parameters. Each parameter has name and type

A more complete view of the above example appears in the following figure:

4.3.2 Using Ontology Concepts as types

ServiceProfile's attributes as well as the parameters of a service functionality have name and type. The type can be primitive (int, float, string, etc.) or can be a class from an Ontology. In the previews examples we used only primitive types. In this section it will be described the usage of an Ontology's class as a type.

The usage of an Ontology's class as type can be achieved through three steps. The first step is to select from the type combo box the item "ConceptID" as shown in the next figures:

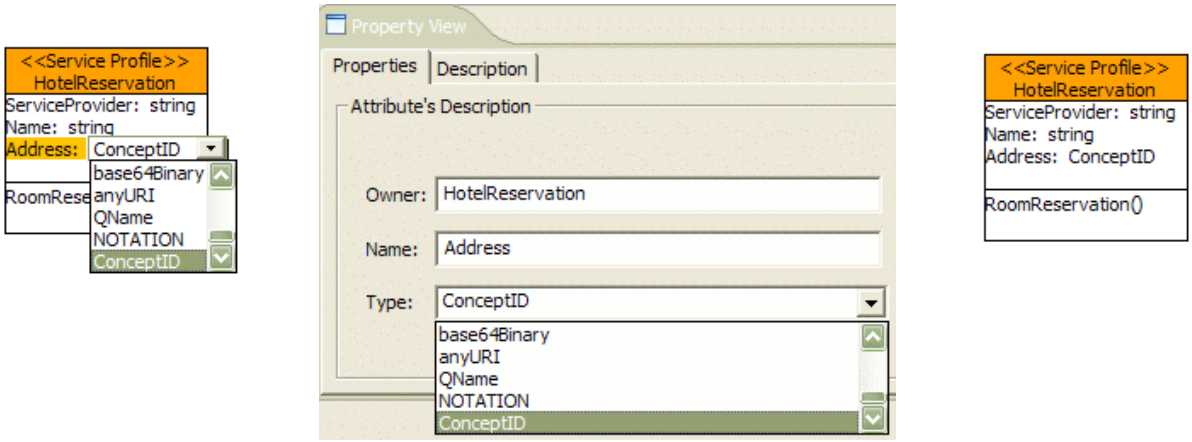

Figure 7: First step in using an Ontology's class as attribute type

The next step is to use the Ontology Viewer in order to browse an Ontology. The OntologyViewer connects to a KB node and lists the available ontologies ("Ontology" combo box). The user can select the ontology he wants and after that is able either to browse the whole ontology or to search for a particular class. If he writes nothing in the "Search" field and clicks the "Go" button the tree view of the selected ontology will be shown while if he writes something in the search field and presses "Go" he will get back the class (actually a whole part) of the Ontology that matches his request. The above are shown in the next figures:

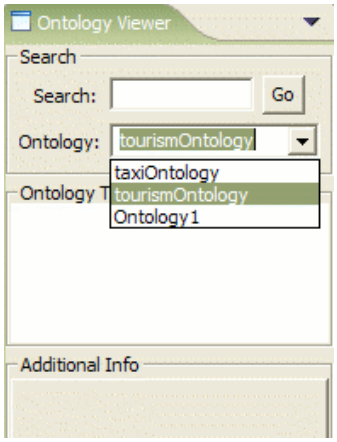

Figure 8: Select the ontology you want

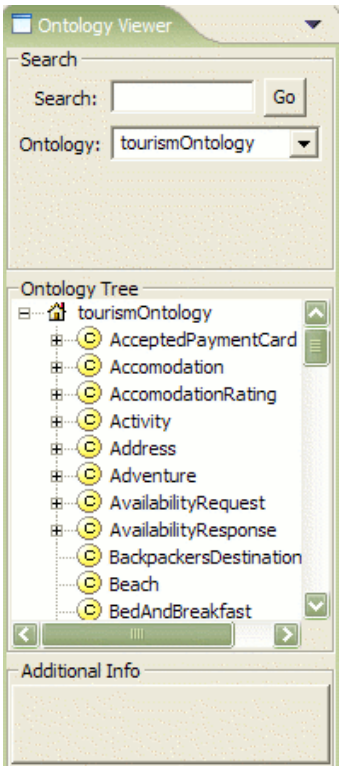

Figure 9: Select the ontology you want

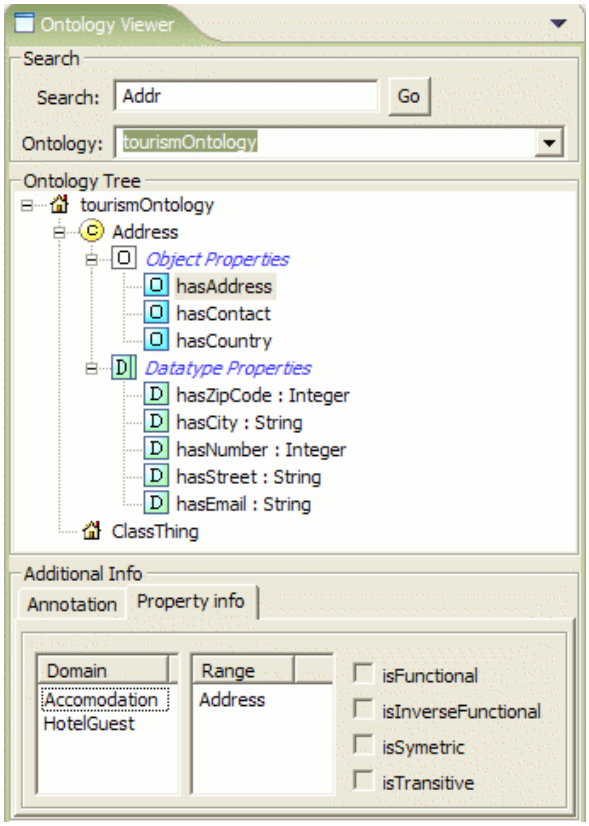

Figure 10: Search for an ontology class. Partial matching is supported

The final step is to double click on the class you wish. After that he class name replaces the type "ConceptID" of the attribute and becomes the attribute's type as shown the next figure:

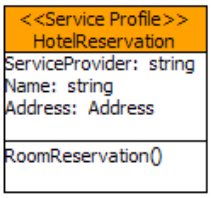

Figure 11: Using an Ontology class as type

As already mentioned, the BML Editor connects to a KB node in order to save/retrieve BML models. The Ontology Viewer is also connected to a KB node in order to get access to the available ontologies. However, it is worth to mention that these two nodes can be different, in other words, the BML Editor

can be connected to a KB node for saving its models and the Ontology Viewer can be connected to another in order to retrieve the available ontologies. Clicking on the menu button ( ) of the Ontology Viewer the user can change the KB node address to which is connected as shown in the following figures:

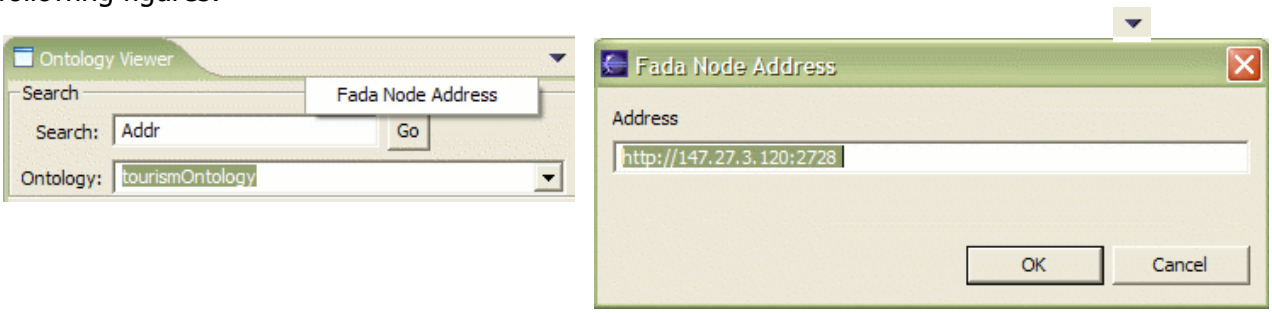

Figure 12: Changing the KB node address to which Ontology Viewer is connected

4.3.3 Creating ServiceParameter and a ContactInformation elements

The user is able to create ServiceParameter and ContactInformation elements selecting from the Editor's palette the corresponding tools and clicking on the editor's canvas. ServiceParameter and ContactInformation element have a name and a type which can be either primitive or an ontology class.

4.3.4 Creating Associations

The user can create associations between a ServiceProfile and ServiceParameter and ContactInformation elements selecting from the editor's palette the Association tool and clicking on the source and target elements. An association has a name and cardinality constraints which can be edited either in the PropertyView or by clicking twice on them. The BML Editor provide for the restrictions imposed by the SSL metamodel allowing only the creation of associations between the suitable elements. An example is shown in the figure that follows:

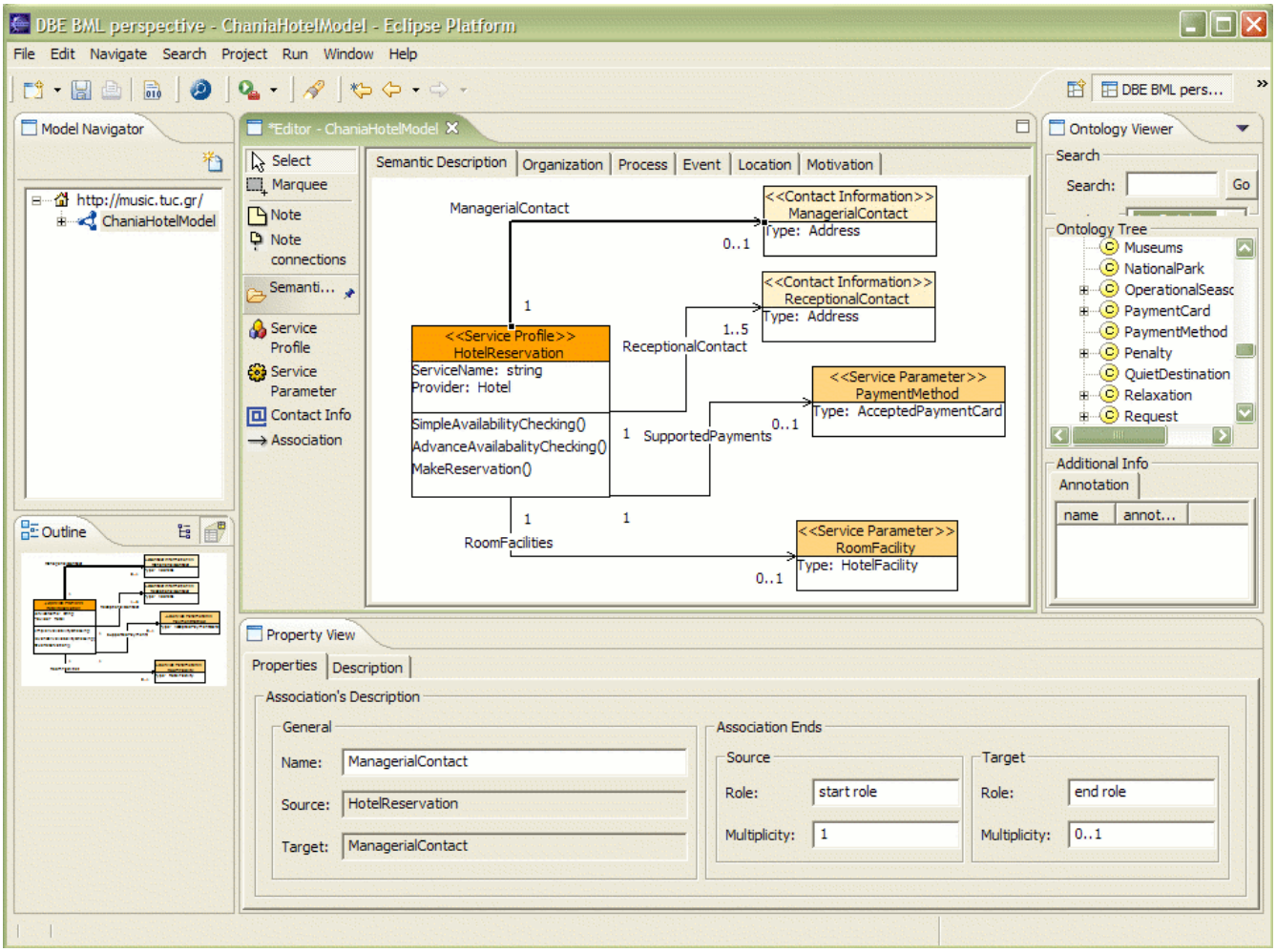

Figure 13: Creating and editing associations

4.4 Creating Business Model Descriptions (BML)

4.4.1 Creating Business Elements

A business model is conceptually divided into five parts: Organization, Process, Event, Location, Motivation and for each part there is the corresponding tab in the editor's main window. According to the active tab the editor's palette changes dynamically providing the available model elements. Selecting an element from the palette and clicking in the canvas the element's visual representation is rendered. Each business model element has a name an may contains many attributes. An attribute has name and type which can be either primitive or an Ontology's class. The steps for creating and editing attributes are the same as for a ServiceProfile.

4.4.2 Creating Business Associations

Associations among business elements are created by selecting the BML Association tool from the palette and clicking on the source and target elements. The BML metamodel provides particular associations with predefined names and cardinality constraints. The BML Editor provide for the restrictions imposed by the BML metamodel allowing only the creation of assocations between the appropriate elements. The following tables describes the allowed associations:

Organization

Source Element	*Target Element*	*Association*
BusinessEntity	Asset	owns asset
BusinessEntity	BusinessItem	provides
BusinessEntity	Product	provides
BusinessEntity	Service	provides
BusinessEntity	NetworkRole	performs
BusinessEntity	BusinessEntity	owns subunit
Network	NetworkRole	has role

Process

Source Element	*Target Element*	*Association*
Asset, BusinessItem, Product, Service (from Organization)	BusinessProcess, BusinessActivity, CollaborationActivity, Transaction	is used in, produces
Event (from Event)	BusinessProcess, BusinessActivity, CollaborationActivity, Transaction	begins, ends, generates
BusinessProcess, BusinessActivity, CollaborationActivity, Transaction	Commitment	fulfills
BusinessProcess, BusinessActivity, CollaborationActivity, Transaction	Role	involves actor
BusinessProcess, BusinessActivity, CollaborationActivity, Transaction	BusinessProcess, BusinessActivity, CollaborationActivity, Transaction	precedes
Constraint	BusinessProcess, BusinessActivity, CollaborationActivity, Transaction	before, after, during
BusinessProcess	BusinessProcess	sub process
BusinessProcess	BusinessActivity, CollaborationActivity	owns task
CollaborationActivity	Transaction	contains transaction
Asset, BusinessItem, Product, Service (from Organization)	Commitment	reserves
Contract	Commitment	establishes
Commitment	Role	involves
BusinessEntity	Role	performs
Contract, Commitment	Event (from Event)	implies

Event

Source Element	Target Element	Association
Event	Event	precedes

Location

Source Element	Target Element	Association
Path	Path	includes
Path	LocationType	is toward
LocationType	Path	is from
LocationType	BusinessEntity (from Organization)	is in

Motivation

Source Element	Target Element	Association
Means, End, Influence, Assessment	BusinessEntity (from Organization)	has
Means, End, Influence, Assessment	any Business Element	impacts on

It should be mentioned that the BML Editor protects the user from the transition to invalid states. For example it does not allow the creation of circles (or deadlocks) as shown in the next figures:

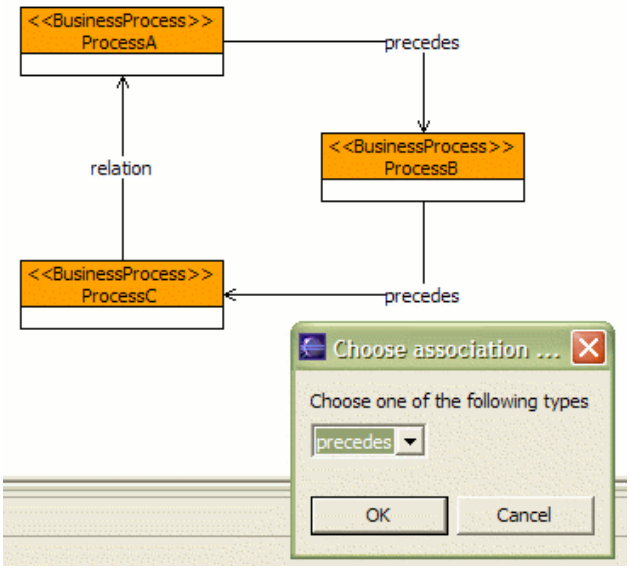

Figure 14: If you try to create a "precedes" relation between ProcessC and ProcessA will be shaped a circle

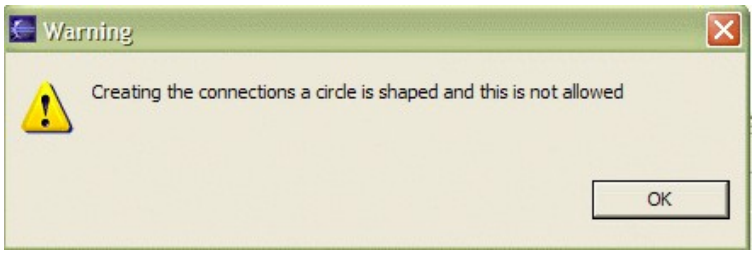

Figure 15: Warning message informing the user for the creation of a circle

4.4.3 Reusing business elements

The BML metamodel allows for the usage of elements defined in one part of the model in other parts. For instance, an Asset element, defined in the Organization part of the model, can be used (referenced) in the Process part of the Model. The BML Editor supports this need through the Outline View. The particular view provides to different looks of the model that is opened in the editor's main window, a bird view ( ) and a tree view ( ) as it is shown in the next figures:

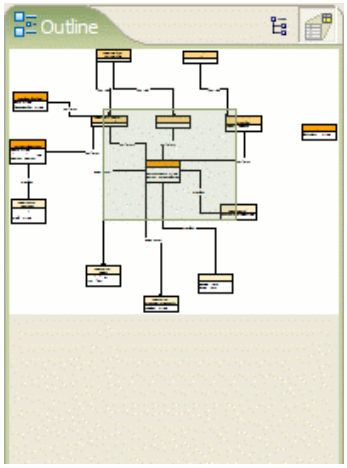

Figure 17: Bird view model repsentation that facilitates the navigation in a big model

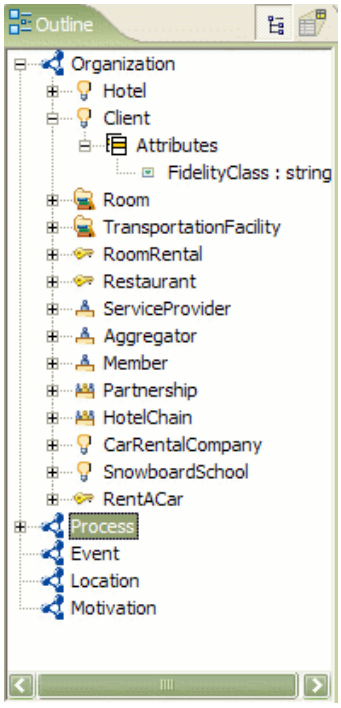

Figure 16: Tree view model representation

The user can drag and drop any element he wants from the tree view into the BML Editor main window (ensuring that does not violate the BML metamodel, actually BML Editor takes care of that).

For example, a Business Entity element is defined in the Organization part (tab) of the model but it can also be used (referenced) in the Process part. So, the user can define such an element in the Organization tab and after that can drag it from the tree in the Outline View into the Process tab in order to use it in this part of the model.

It should be noted that BML Editor takes care that the BML metamodel will not violated during this process. For example, if you drag a Network element (defined in the Organization part) into the Process part the editor will not allow this action because the BML metamodel does not support it. In addition if the user attempts to delete an element that is referenced in other parts of the model the BML Editor will not allow it and will provide a warning message.

5 Searching for existing models

The user is able to search for models through BML Editor. The search process is provided by another plugin (Query Formulator-Semantic Discovery Tool) that can be called by BML Editor clicking on the  icon in the editor's tool bar. Using this plugin the user is able to formulate queries for existing M1 models by posing constraints on the attributes of primitives that the M2 metamodels (SSL and BML) provide. The user can define soft and hard constraints. The desired models may satisfy the soft constraints, but they have to satisfy the hard ones (see D17.1 for details [5]). Finally the user is able to open, in the editor, a model that was found by right-clicking on it and choosing "Open BML Model":

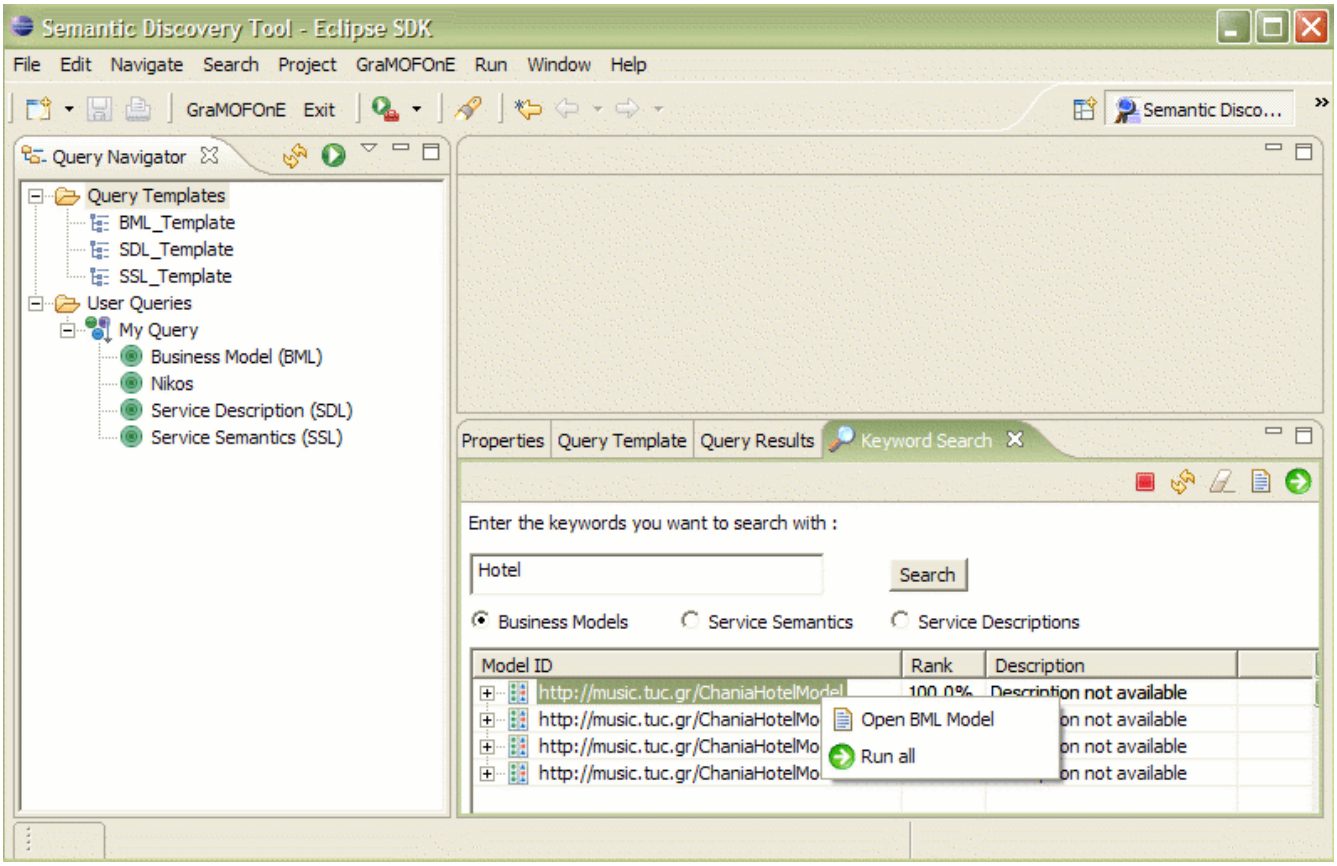

Figure 18:The result set. Double-clicking on the model or selecting the model and clicking the "Finish" button the model opens in the BML Editor

References

1. DBE: Digital Business Ecosystem, http://www.digital-ecosystem.org
2. D14.1: DBE Knowlegde Representation Model
3. D14.2: 1st Prototype Implementation of the DBE Knowlegde Representation
4. D17.1: Recommender
5. D15.1: Business Modelling Language 1.0
6. The Swallow project: http://www.sourceforge.net/projects/swallow
7. Java Metadata Interface, http://java.sun.com/products/jmi/index.jsp