Tolkien User Guide

June 2001

Table of Contents

1 Using Tolkien

1.1 Introduction

In a data mining task a lot of time is spend on the preprocessing of data. This is often a quite technical task that requires knowledge of various fields. It is a process that should become easier, faster, less technical and with more focus on the business domain that is involved.
An environment that uses a visual language to capture domain knowledge could help to make it easier to apply domain knowledge in the data mining process. Furthermore operators could be developed, that can automate (part of) the preprocessing process.

The Tolkien program[1] is a demonstrator program that takes a small step in this direction. Figure 1 shows a screen shot of the application.

The program provides the following functionality:

- import function for importing (part of) a database schema;
- creation and manipulation of data models;
- propositionalization operator for automatic propositionalization of multi relational data;
- generation and export of CDBL code.

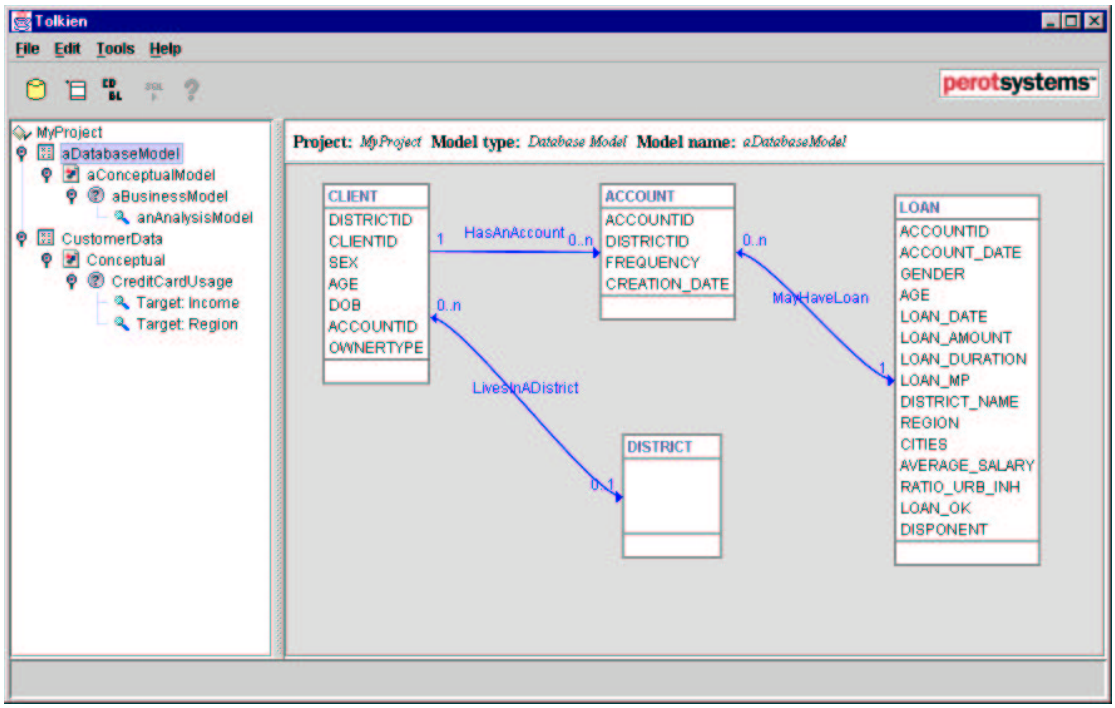

Figure 1: Screen shot of the Tolkien client application.

[1] WARNING: Tolkien is a demonstrator program and not intended to be used in a production environment. It is still in development and although the functionality that is described in this document should work, be aware that bugs may occur when using this program!

Tolkien can be used to draw different types of data models:

- Database Model: shows how the model is stored in a database
- Conceptual Model: this is a "cleaned up version" of the database model. Things in the model that are not relevant to the concept (but were for example added for data management reasons) can be left out here.
- Business Model: this is a part of the conceptual model that focuses on a certain business area.
- Analysis Model: this is a part of the business model that will be focussed on for further analysis.

These models are displayed in a tree, which makes it easy to keep an overview of the created models.

One can start by importing the database model from an existing database (Oracle) or by drawing a model yourself.

Tables, attributes and associations can be added and edited. More than one association can be drawn between two tables and associations can also be drawn to a table itself. Furthermore tables and associations can be repositioned.
To draw an association hold down the CTRL-key while having the mouse above a table. Press the mouse button and drag the association to the other table.

From every model so called CDBL code (Common Declarative Bias Language) can be generated. This is code that describes the model. It could be used for multi relational data mining using an ILP algorithm. The University of Leuven (Belgium) for example has an ILP algorithm that can read this code. Others are expected to follow.

Further a propositionalization operator is provided. This can be used to propositionalize (parts of) a database schema after which a mining application can be used to mine the created result table.

The remaining part of this chapter focuses in more detail on the different parts of the program.

1.2 Projects

1.2.1 Starting a new project

To start a new project choose "New Project…" on the "File" menu. A project name will be asked and after pressing "OK" this name will be shown as the root of a tree of models at the left side of the screen; the right side of the screen will be empty. At this point new data models can be created or a database model can be imported from an Oracle database.
A project consists of data models, which in turn consist of tables and relations. All objects are stored in a Java 2 Enterprise Edition Server. When tables, attributes or relations are created on the client, they are directly stored in this server. Therefore there is no need for a "Save" option to save a project.

1.2.2 Opening an existing project

Select the “Open Project…” menu option from the “File” menu. A list of existing projects will be shown with their names and their creation dates. Select a project and press “OK”. The data will be read from the J2EE Server (this may take some time) and the tree of models will be shown. Select a model in the tree to view the corresponding model on the right side of the screen.

1.2.3 Deleting an existing project

In order to be able to delete an entire project that project must be the current project. Select the project name in the root of the tree of models and press “DELETE” or select the “DELETE” menu item from the “Edit” menu.

1.3 Data models

1.3.1 Creating a new data model

When a project is available, new data models can be created in the tree of models. Select the base for the new data model in the tree of models. The base of a data model can be the root model or a previously created data model. Now select the “Add new Model” menu item from the “Tools” menu or right click on the base position in the tree and select the “New X model” menu item from the popup menu (the “X” here stands for the type of model that can be added).
You will be asked to provide a name for the new model and after pressing “OK” the model will be added to the tree.

1.3.2 Copying a data model

When working with a tree of data models, it is very useful to be able to copy a data model to a sub data model (or another data model somewhere in the tree of data models). One may, for example, import a database schema into a database model, copy that model to a conceptual model and then edit it until it reflects the conceptual model.
To copy a data model, right click it and choose “Copy to” from the popup menu. A dialog will appear and a target data model can be chosen. The copied tables and relations will not be related in any way to the original tables and relations.

1.3.3 Deleting a data model

To delete a data model (and its submodels) from the tree of models select the data model in the tree and press “DELETE” or select the “DELETE” menu item from the “Edit” menu. Be sure that no part of the model is selected on the right side. If a table or relation was selected these will be deleted instead.

1.4 Tables, relations and attributes

1.4.1 Drawing and editing tables

The “add table” icon adds a new table to the worksheet. A table is represented by a UML class diagram figure. One or more tables can be selected by clicking on a table or by dragging a selection box around one or more tables; the selected tables have a red border. Tables can be dragged to a different position when they are selected.

Figure 2: The "add table" icon

By right clicking on a table you get a pop up menu with a few options:

- *Edit*
  By selecting “Edit”, or by double-clicking a table you can change the properties of that table.
- *Hide/Show attributes*
  By selecting this option the attributes of the table are shown or not shown on the screen depending on the status.
- *Set target*
  “Set Target” lets you select an attribute of the table as target attribute.
- *Add relation to itself*
  By selecting this option a relation from the table to itself can be created.
- *Delete*
  “Delete” deletes the table.

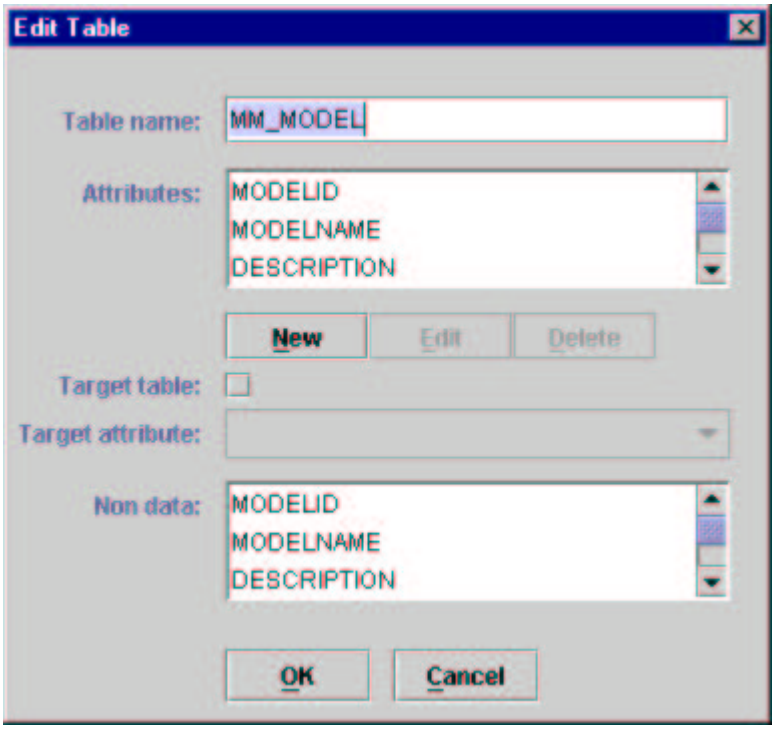

Figure 3: The "Edit Table" dialog

The Edit Table dialog
With this dialog you can change things like the table name and the attributes of the table. In the “Table name” field you can set a name for the table.
Note that actions like changing the name of a table or adding or changing attributes only change the stored data model; these actions have no direct effect on the tables or attributes that are stored in a database.

The “Attributes” list contains all attributes of the table. You can manipulate this list with the buttons at the bottom of the list. The “New” button lets you add an attribute, simply type the name and choose a type. When you select an attribute by clicking on it the “Edit” and “Delete” buttons become active. With the “Edit” button you can change the name and/or type of the attribute, with the “Delete” button you can remove an attribute from the table.

Check the “Target table” checkbox if you want this table to be the target and select a target attribute from the “Target Attribute” list.

The “Non data” list can be confusing at first. It is a list of *all* attributes of the table. Only *selected* attributes in this list are considered to be non data attributes (for example primary or foreign keys are non data attributes).
In this way the “data” or “non data” property of the attributes of a table can easily be set.
Use “Shift click” to select more adjacent attributes. Use “CTRL click” to select or deselect non-adjacent attributes.

You must provide a name for the tables before you can generate CDBL code. A table must also have attributes in order to be able to set it as a target table.

1.4.2 Drawing and editing relations

To add a relation between two tables:
- hold down the *control key*;
- click on one of the tables and press the mouse button
- drag a line to the other table while holding the control key.

To add a relation to a table itself, right-click on the table and select the “Add relation to itself” option from the popup menu.

To edit the properties of a relation double click it or right-click the relation and select "Edit" from the popup menu. An "Edit Relation" dialog appears; here you can change the name of the relation and other properties.

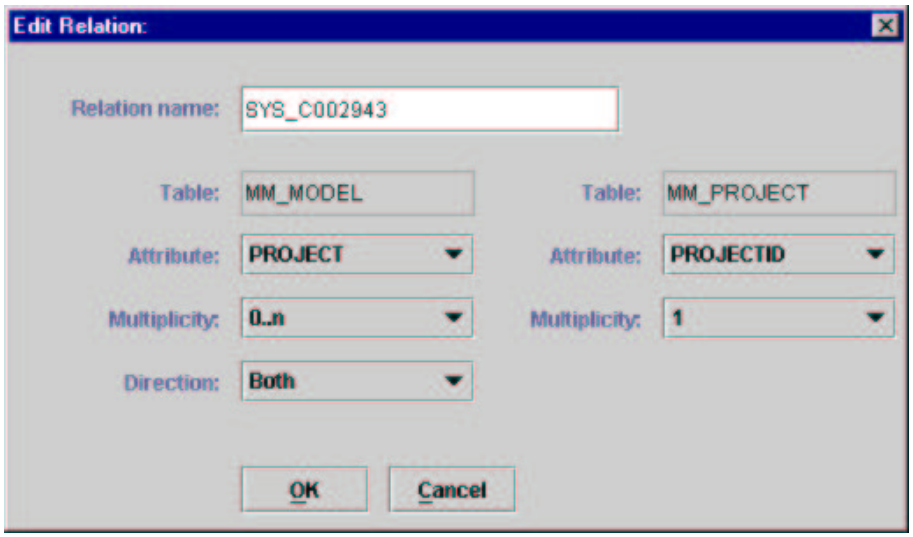

Figure 4: The "Edit Relation" dialog

The "Relation name" field contains the name of the relation (by default "NewRelation"). Use this field to change the relation name to something more meaningful.
The two "Table" fields contain the "from" and "to" table of the relation, these can never be edited.
The attributes that are linked by the relation can be altered by selecting an attribute from the "Attribute" drop-down list.
The corresponding multiplicities can be changed in the same way.
The direction in which the attribute can be used can also be changed from "Both" to "Left to right" or "Right to left". "Left to right" means from the left hand side attribute to the right hand side attribute *in the "Edit Relation" dialog*. "Left to right" and "Right to left" do *not* correspond to positions of tables on the worksheet.
You must provide a name for your relations before you can generate CDBL code.
Both tables must have at least one attribute in order to be able to draw a relation between them.

1.5 Connecting to a database to import a database schema

In stead of drawing the database schema, you can also import it from an ORACLE database (other types of databases are not supported at this time). To do this, click the "Database connect" button.

Figure 5: The "Database connect" button

You will be asked to provide a name for a new Database model which will be created in the model tree and which will be used to import the data into.
After providing a model name the Connect dialog will appear:

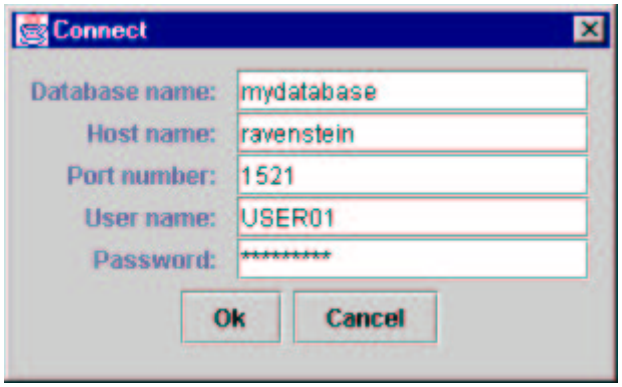

Figure 6: The "Connect"dialog

The “Connect” dialog appears with a “Database name”, “Host name”, “Port number”, “User name” and “Password” field. The standard port number to connect to an Oracle database is 1521. The input you provide is *case-sensitive*! Keep this in mind when for example entering the user name. When you have entered correct values (see the file *tnsnames.ora* in your Oracle installation for the hostname and port number of a database) all tables from the database will be displayed in a list. You can select the tables you want to import (use the SHIFT and/or CTRL keys to select or deselect one or more tables). All relations for which you import the “from” and “to” table, will also be imported.

Standard all attributes from a table are shown on the screen. These attributes can be hidden as described in the previous section.

1.6 Creating CDBL code

CDBL stands for Common Declarative Bias Language. It can be used to describe a data model. It can be used by ILP engines for data mining.
When you want to generate CDBL code for a certain model, first select that model from the tree of models. If a model is not empty, CDBL code can be generated for it.

Figure 7: The "Generate CDBL code" button

The CDBL code appears in a “Generated CDBL Code” window. You can save the generated code to a file by clicking the “Export…” button. The “OK” button lets you go back to the worksheet.

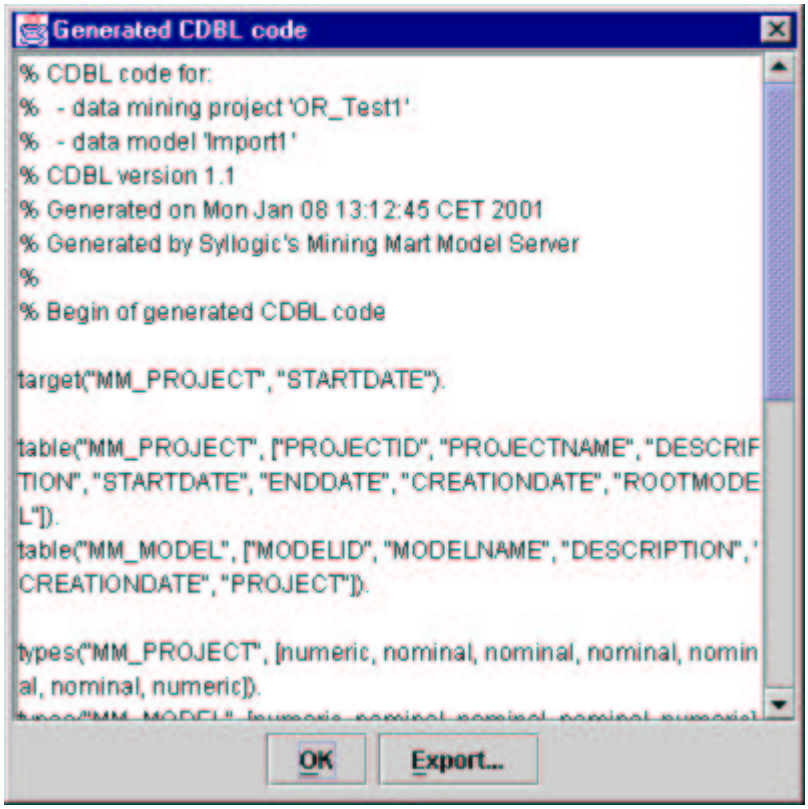

Figure 8: The "Generated CDBL code" window

You can also copy text from the “CDBL Display” window and then paste it into another application.

1.7 Propositionalization of a data model

Tolkien provides a propositionalization operator for handling multi relational data. This operator uses aggregates to capture information from various tables and merge this information into a single output table. After selecting a target table and a target attribute one can propositionalize a data model by right clicking the data model and choosing “Propositionalize data model…”.
The following dialog will appear:

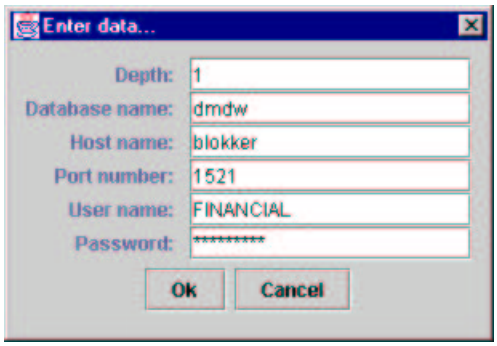

Figure 9: Dialog for entering propositionalization information

This dialog has a “Depth”, “Database name”, “Host name”, Port number”, “User name” and “Password” field.
The depth parameter tells the operator which tables to include in the propositionalization process. A depth value of one, means the operator will only include tables that have a direct connection to the target table. A depth value of two means that also tables that are connected via another table to the target table are included in the process. The meaning of the other fields will be clear. Remember that the input is case sensitive.
The operator will generate a *preprocessing tree*. A preprocessing tree is a tree of *preprocessing actions*. The actions have a primary and secondary input table, a definition and an output table. The definition is a SQL statement that describes the action.
The preprocessing tree will not be executed on the database at this point. This is an action that is left to the user and is described further on.

1.7.1 Viewing preprocessing trees

At some point of time in a project a data model may or may not have preprocessing trees associated with it. This can be viewed by right clicking a data model and selecting “List available preprocessing trees…”.
If no preprocessing tree is associated with the model a message will be shown that there are no trees. If only one preprocessing tree is available, then that one will be displayed directly. If more preprocessing trees are available, the user is presented a list to choose from.

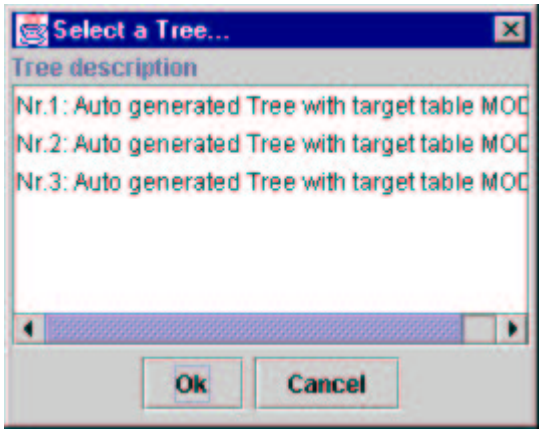

Figure 10: Dialog for choosing a preprocessing tree

After choosing a tree this tree will be shown.

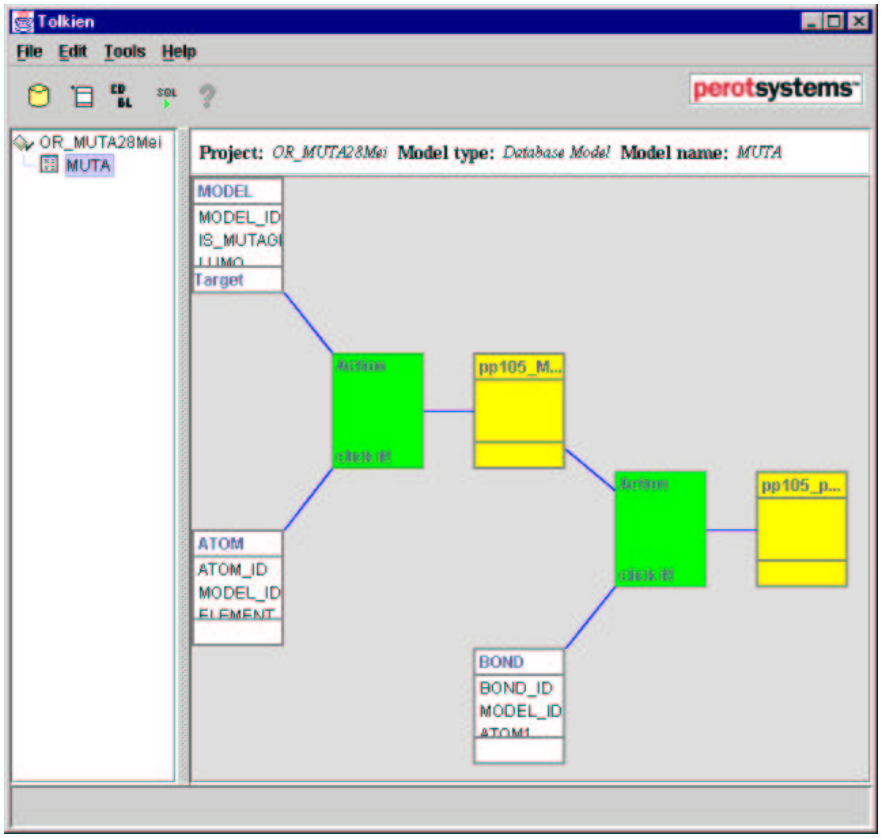

Figure 11: Example of a preprocessing tree for a model

An example of a preprocessing tree is shown in figure 11. The example shows a tree consisting of two preprocessing actions.
The properties of an action can be viewed by *double clicking* on it. An example is shown in figure 12.

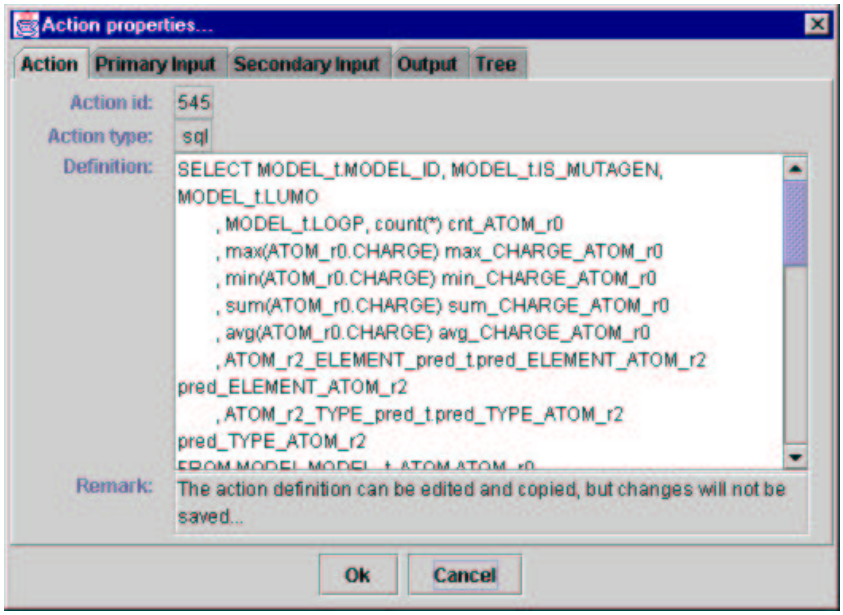

Figure 12: Dialog showing properties for a preprocessing action

1.7.2 Executing the propositionalization operator

After propositionalizing a data model a preprocessing tree is generated. The SQL code that is defined in the actions of the tree is not executed automatically on the database containing the data model. This can be done by clicking the “Execute SQL from tree” button or selecting the “Execute SQL” menu item from the “Tools” menu.

Figure 13: The execute SQL button

The button and the menu item are active only when a preprocessing tree is displayed. After clicking the button or selecting the menu item the dialog of figure 9 is shown again. The “Depth” parameter cannot be edited at this point, but the user should provide the information for making the connection to the database.
When the resulting table has been created a message will be shown with the name of the output table.

2 Questions and remarks

If you have any questions or remarks about this demonstration application, please send a message to marc.dehaas@ps.net or olaf.rem@ps.net.

3 About Syllogic Innovations

Syllogic Innovations is the innovative department of Perot Systems Nederland. Syllogic Innovations has expertise in, among other things, system integration, database management, data warehousing, machine learning, data mining and text mining. This expertise is gathered in products such as Adaptive Systems Management and solutions like Prognostics and Health Management.
Perot Systems is a world-wide supplier of information technology and business solutions. The corporation provides its clients with an integrated service offering: a combination of multiple disciplines that assist our clients in improving their business operations or in creating new business offerings.
Perot Systems delivers these integrated services through creative partnerships that are broad and unique - fostering opportunities for innovation, new levels of profit and growth and the creation of new types of business.