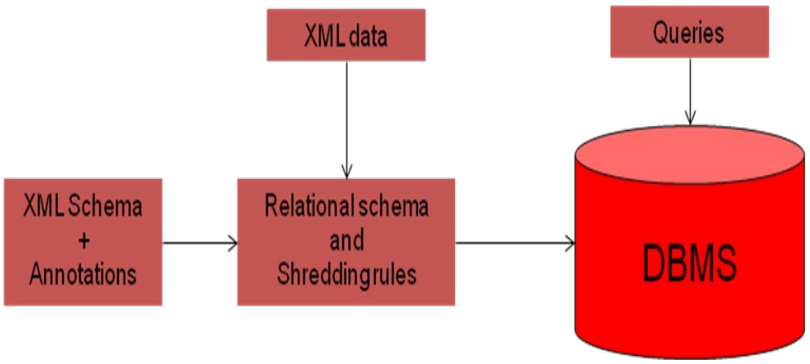
HShreX – introduction

The purpose of HShreX is to find a method of efficient storage which also provides import of new data with relatively little effort from the end user. The tools provided by the database management system often requires a step of designing the hybrid storage and data transformations which is relatively complicated and time consuming.



The main architecture and principles behind HShreX are described in the figure above. The starting point is an XML schema for the data that we need to import. HShreX analyses this schema and produces a relational schema based on its content. In addition there is a possibility for the user to change the given translation by adding annotations to the original schema.

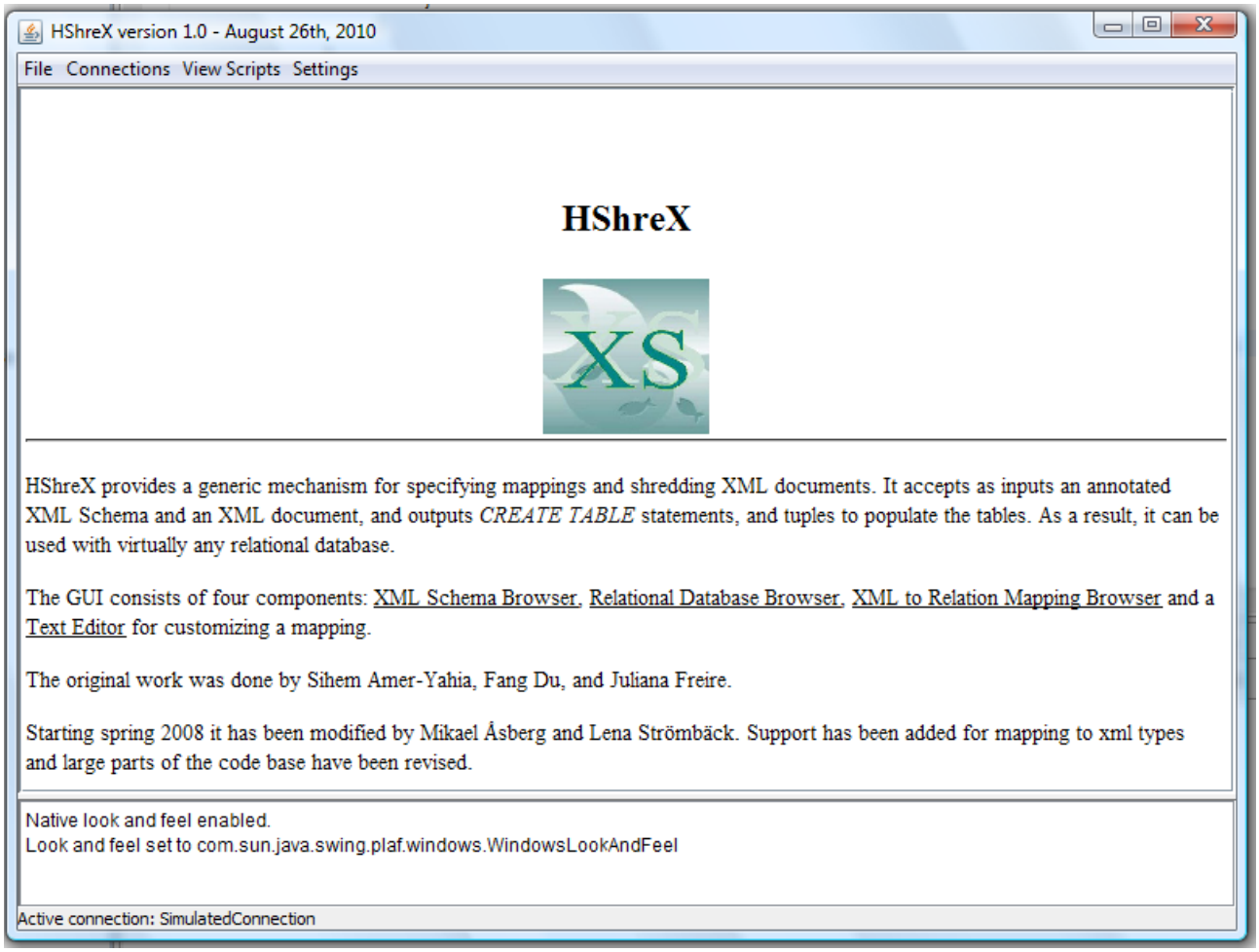
The result from the XML schema analysis is a relational schema that could be loaded into a database. In addition the analysis output a set of translation rules. With these rules any XML data file that are valid to the XML schema can be shredded and downloaded into the database. The user can then query the data by using the standard SQLXML query language provided by all data managers providing hybrid XML storage.

The shredding rules can be influenced using annotations and for a comprehensive description about the supported annotations and their effect on the mapping, please see the document ”Annotations for the HShreX System”.

This is version 0.4.2 of this document, dated September 16th 2010.

HShreX – Usage

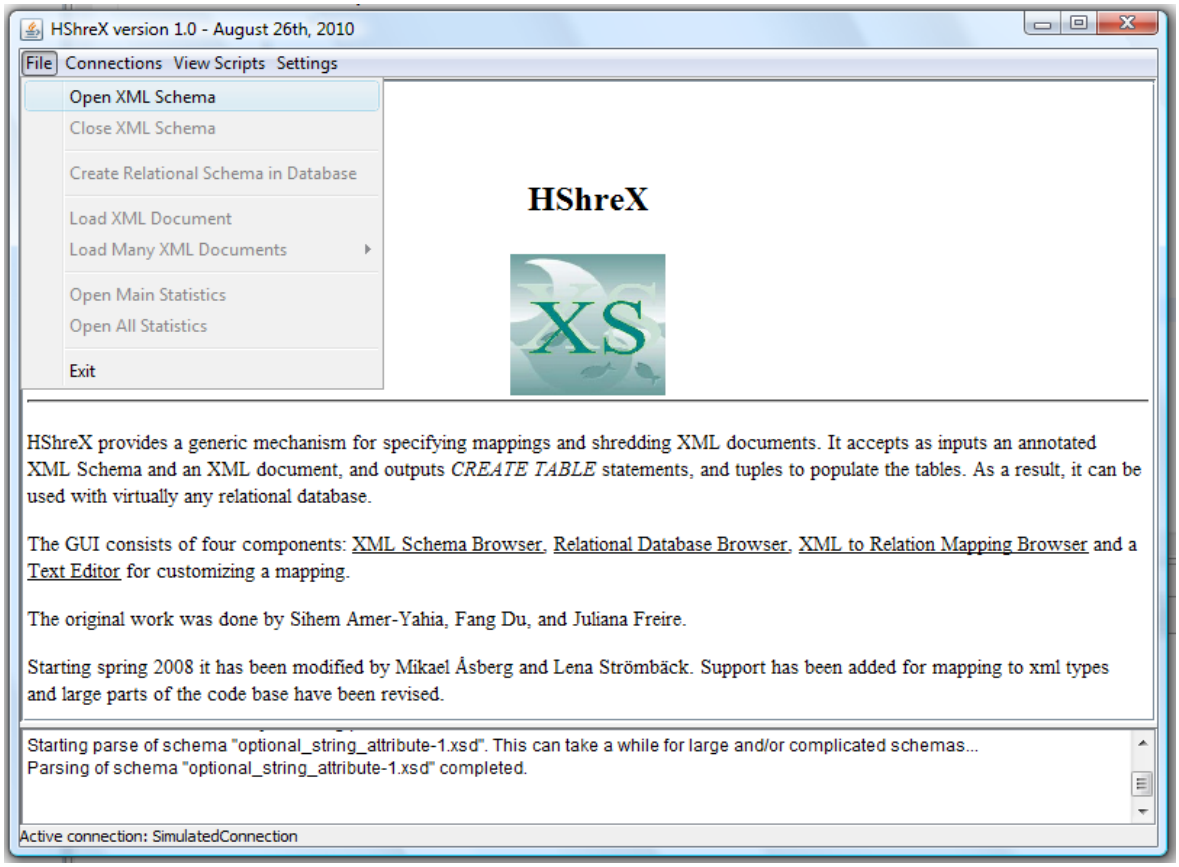
When you first start HShreX you are greeted by the following screen:



This is the welcome screen and it's what's displayed when there's no XML schema open.

Loading XML Schema

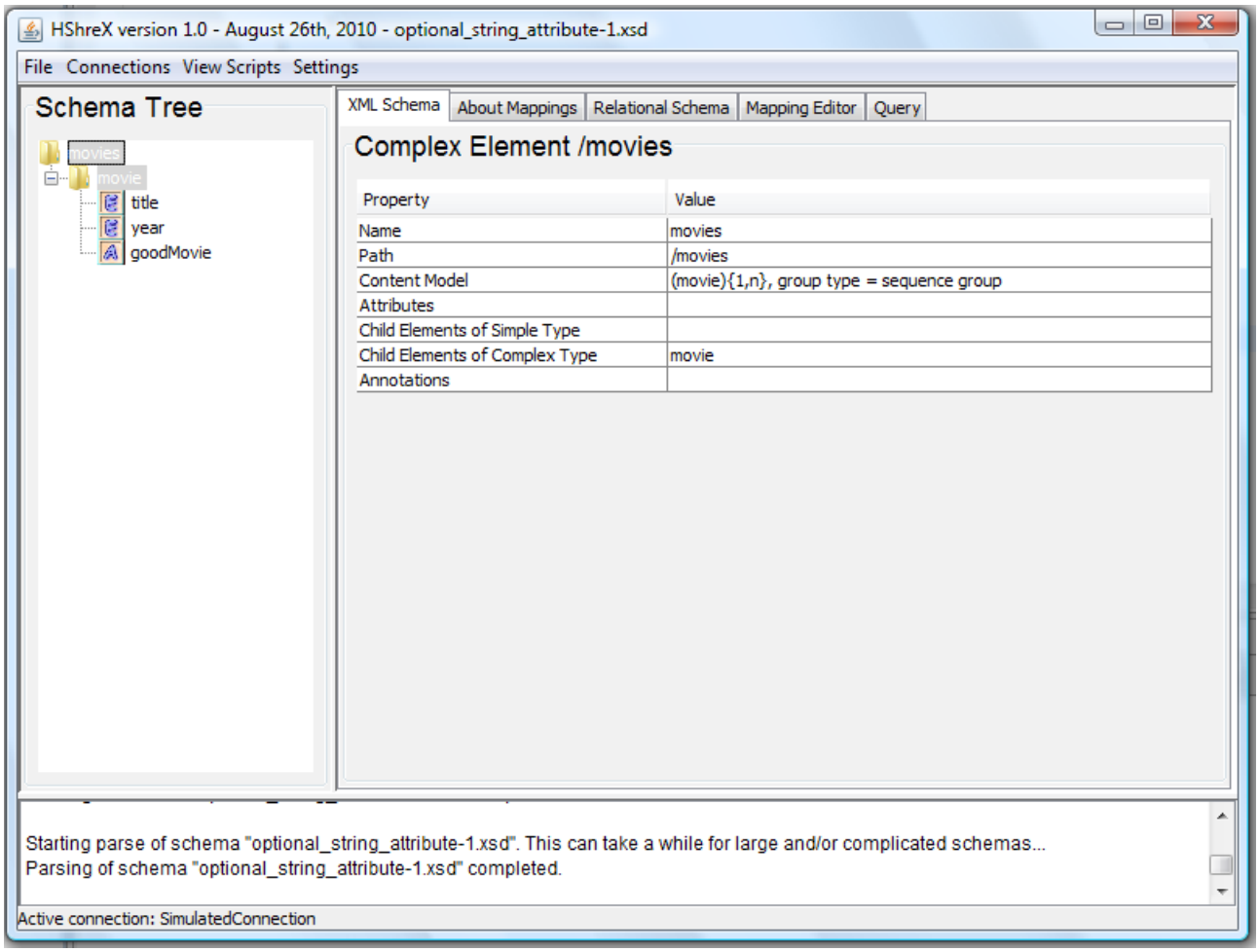
There's not much we can do until we've opened an XML schema, and we do that by selecting the menu item ”Open XML Schema” under the File menu:



When this menu item is selected a standard file selection dialog appears allowing us to select the XML schema we wish to work with. In the screenshot on the following page I will show what happens when we open a schema that has the following content (and this example schema I will use throughout the document):

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="movies">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="movie" type="movieType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="movieType">
    <xs:sequence>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="year" type="xs:gYear"/>
    </xs:sequence>
    <xs:attribute name="goodMovie" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:schema>
```

The XML Schema Tab

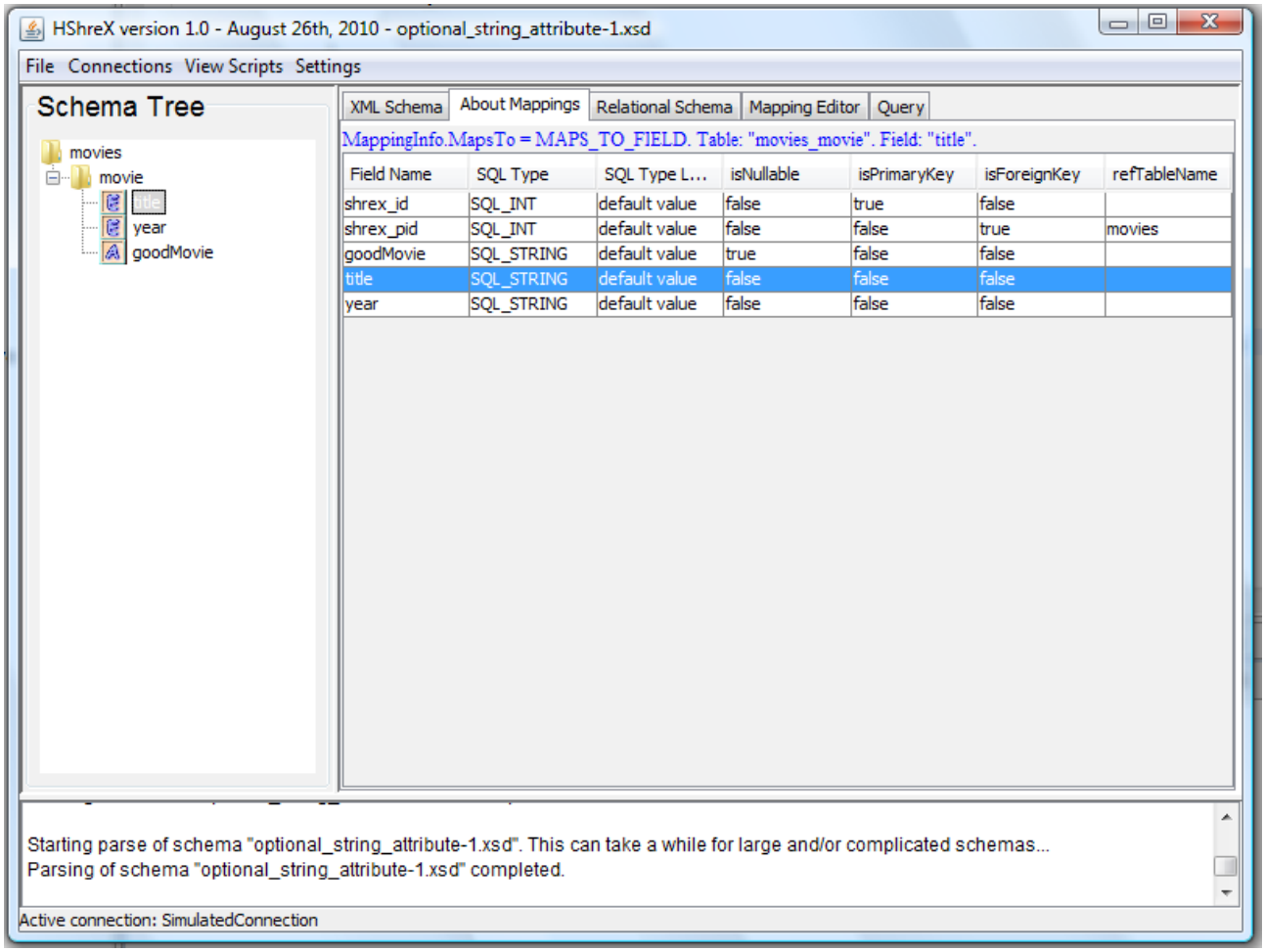


As you can see, when an XML schema is loaded, the welcome screen disappears and the HShreX window is split into two (also note how the title of the program has changed to the name of the currently opened schema). On the left-hand side we have the Schema Tree which is a simple tree representation of the XML schema where all complex elements in the schema acts as containers and all simple element and attributes acts as leafs in the tree. On the right-hand side we have five tabs. By default, the first tab, XML Schema, is selected initially and it describes information about the currently selected node in the schema tree. In the screenshot above the root node is selected in the schema tree and we can see information regarding it. As other nodes are selected in the tree the information in the XML Schema tab is updated accordingly.

One important property to notice in the XML Schema tab is “Annotations” which describes the special shredding annotations. The particular element shown in the screenshot above has no annotations, in fact, the entire XML schema has no annotations and simply means the default mapping is the desired mapping.

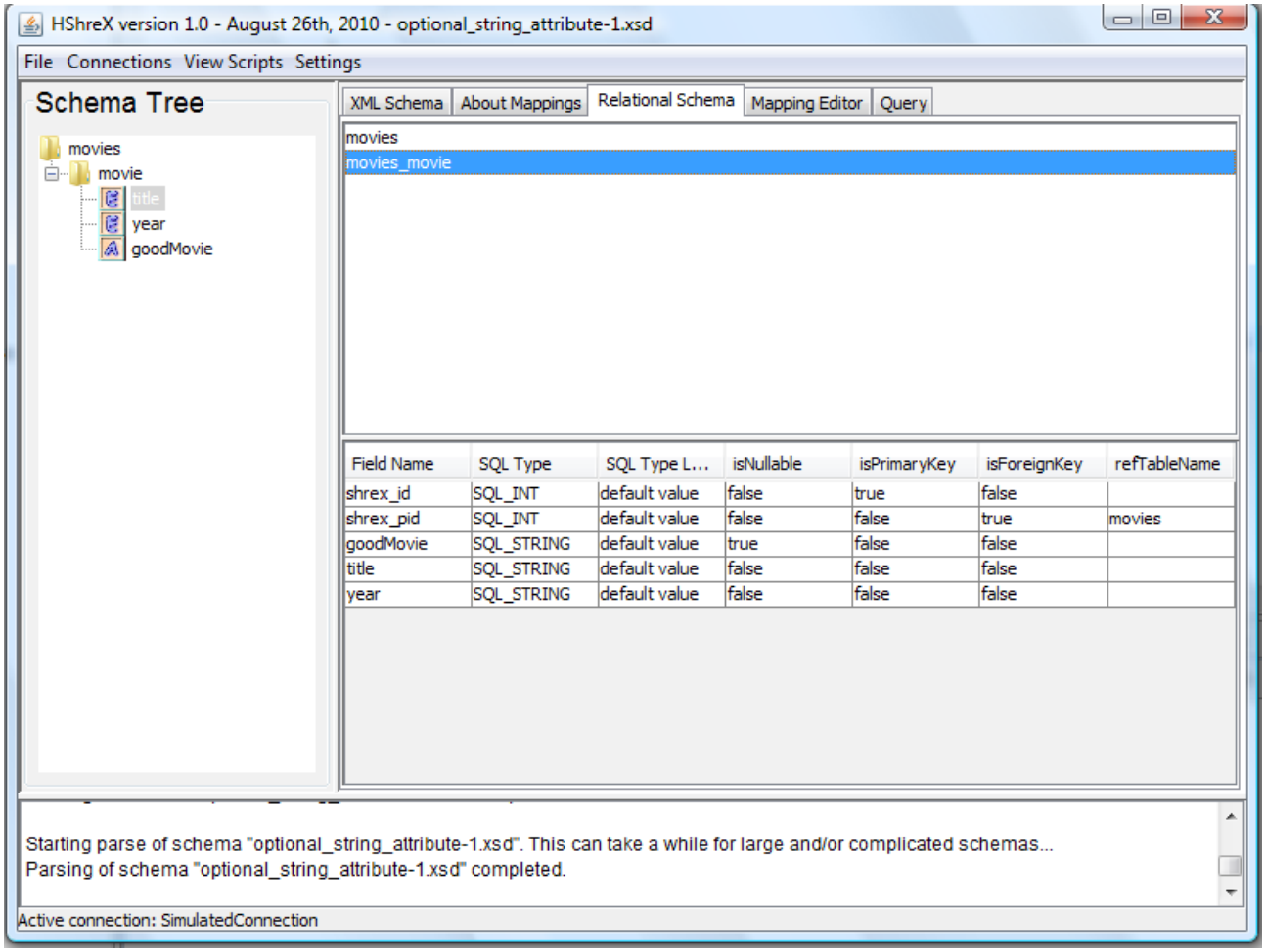
Let us now move on to the tab labeled “About Mappings” where we can see how our XML schema elements (complex and simple) and attributes have been mapped onto a relational model.

The About Mappings Tab



Here we have selected the simple element title in the complex element movie in the schema tree and on the “About Mappings” tab we can see how it's mapped. First we have a text which contains a complete path to the element or attribute (separated by underscores) and information on how it's mapped. We are also shown a table giving a nice presentation of the relational table involved in this particular mapping. As you select different nodes in the schema tree the mapping information displayed here will automatically update. Here we can only see one relational table at a time and if we're interested to get an overview of all generated tables we should go to the next tab: Relational Schema.

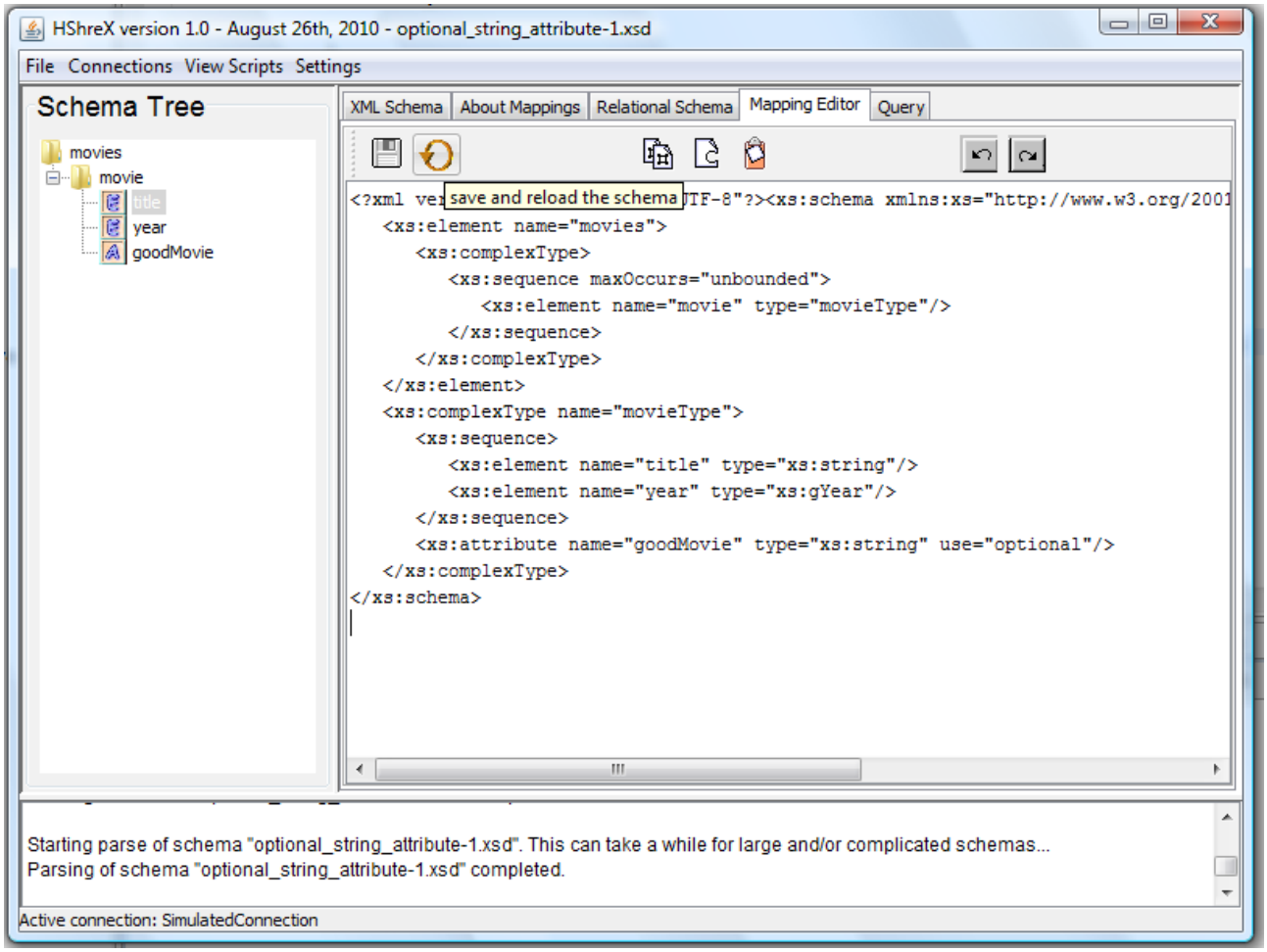
The Relational Schema Tab



The Relational Schema tab is split into two. At the top are all our generated tables and at the bottom you can see details about the currently selected table similar to the output in the About Mappings tab.

The next tab is labeled “Mapping Editor” and it's explained on the next page.

The Mapping Editor

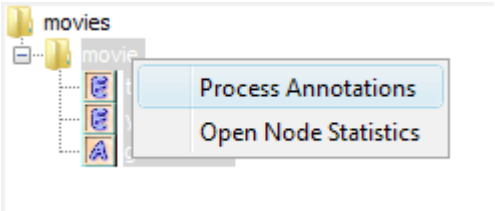


The Mapping Editor displays the schema in its text form, using its original formatting. Here the user can perform basic editing operations like cut, copy, paste, undo, and redo. The main purpose of this tab is to enable the user, from within HShreX, to add, remove, and alter annotations to get a different relational mapping. Changes can be saved (be careful, the existing schema file will be overwritten without prompting) and the schema can be reloaded using the button high-lighted in the screenshot above. Note that reloading the schema also saves it to disk, again overwriting the old file. If one attempts to reload a schema that isn't valid, the currently opened schema is kept open.

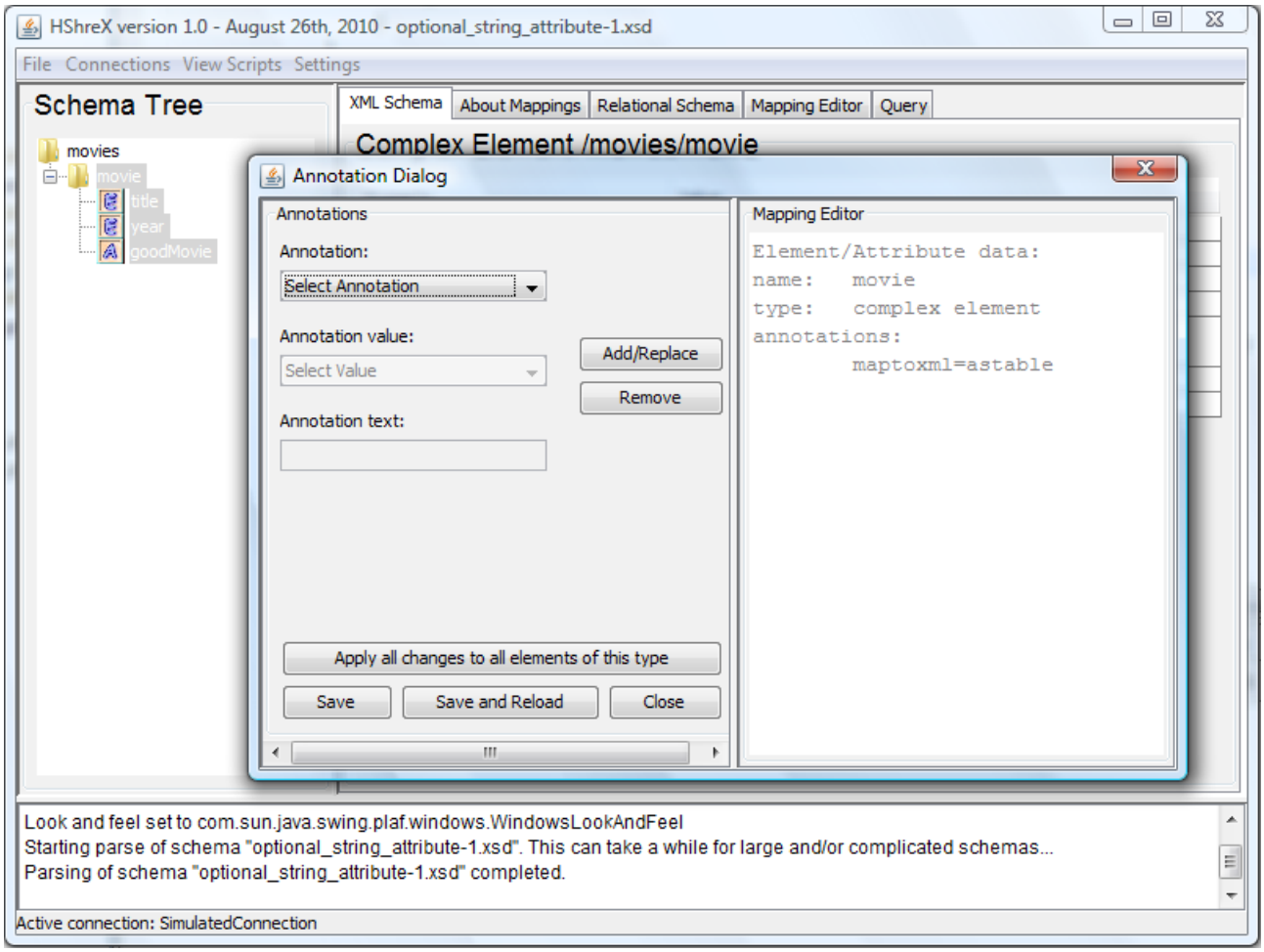
In the latest HshreX version an easier interface to add, remove and alter annotations is provided (see "Annotation Dialog" chapter). This tab is still available, however we recommend to use the special dialog for annotation processing.

The Annotation Dialog

When navigating in the schema tree we can open a dialog for the currently selected node where we can process/edit its annotations. This option is available when you use the right mouse button over a node in the schema tree and choose the “Process Annotations” menu item.



Then the “Annotation Dialog” appears (it refers to the selected tree node):



Here we have a fully functional interface for annotations editing. The user can choose an annotation among the annotations available in the “Annotation” drop down. Some of the annotations have fixed range of values while others have free text values. When the user chooses an annotation one of the fields “Annotation value” or “Annotation text” becomes available, depending of the possible values for currently selected annotation.

When the user chooses “Add/Replace” button the annotation is applied (**but not saved**) to the selected node and appears in the right side in the annotation dialog. If we have had already a value for this annotation the old value is replaced by the new one without prompting. When the user

selects an annotation and presses the “Remove” button the annotation with the same name is removed from the element if exists. If such annotation does not exist for this element nothing happens. Since some combinations of annotations, for an element or an attribute are not valid, each annotation is validated (prior to adding) regarding already available annotations. If the current combination is not valid an error message is shown to the user and the current operation ends without effect on the tree node.

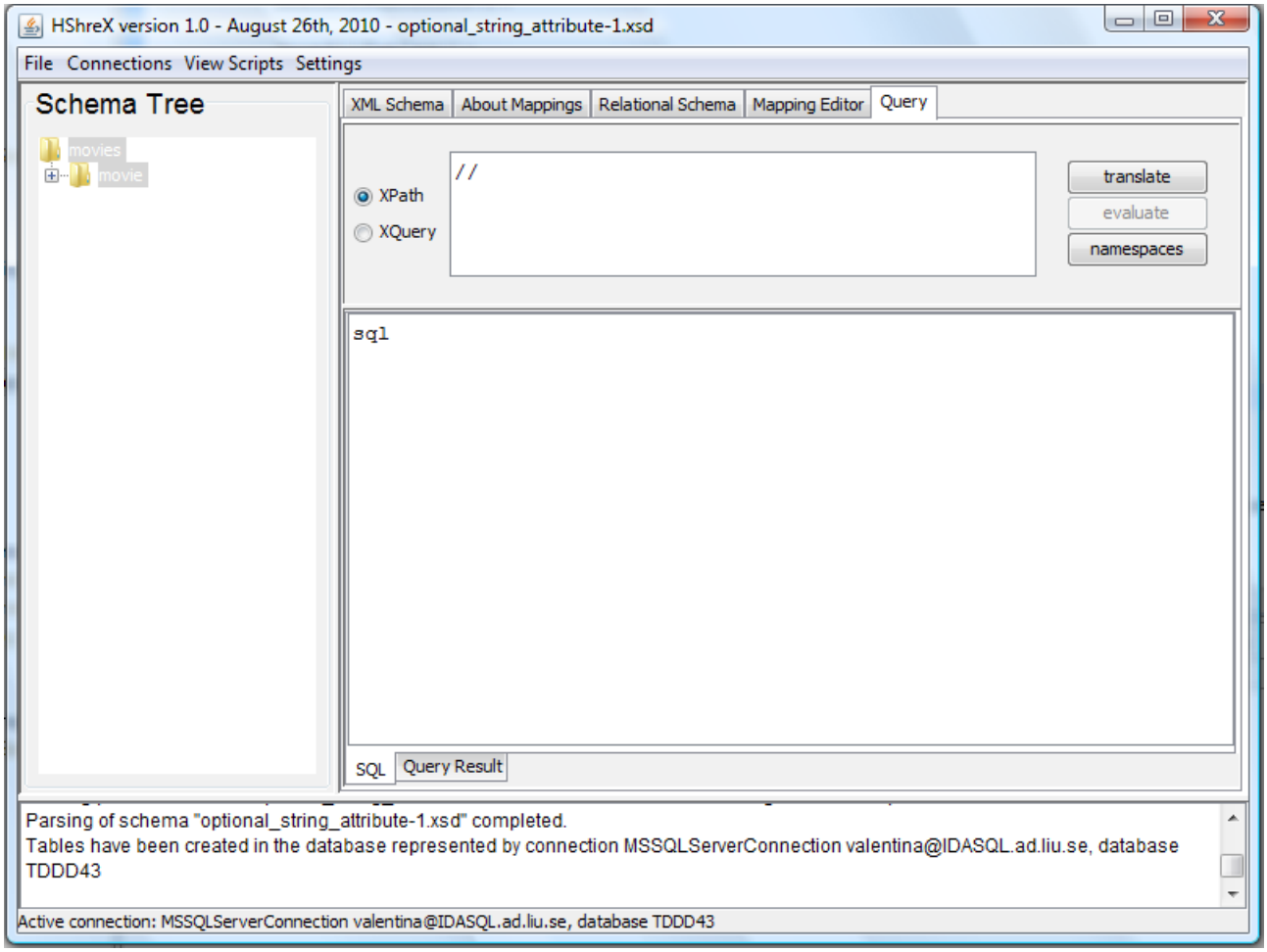
A useful feature is provided through the “Apply all changes to all elements of this type” button i.e. the currently added/removed annotations will be applied to all elements of this type in the XML schema with a single action.

Press the “Save” button in order to save the changes. This action saves the changes to the file (be careful, the existing schema file will be overwritten without prompting) without reloading it. This means that the user will be able to see the changes in the interface, however these changes will not have any effect on the relational schema mapping. The user needs to use the “Save and Reload” button to force these changes to apply on the relational schema mapping as well. Note that the “Save and Reload” button also saves the schema to the disk, again overwriting the old file.

The “Close” button closes the dialog if all changes are saved. If not reminds the user that there are unsaved changes and leaves the dialog open. The basic data and the annotations for the current node are listed in the right side of the dialog.

The Query tab

Note: The “Query” tab is under development, so it works only for simple queries. Be careful if you decide to use it at all.



Create Relational Schema in Database

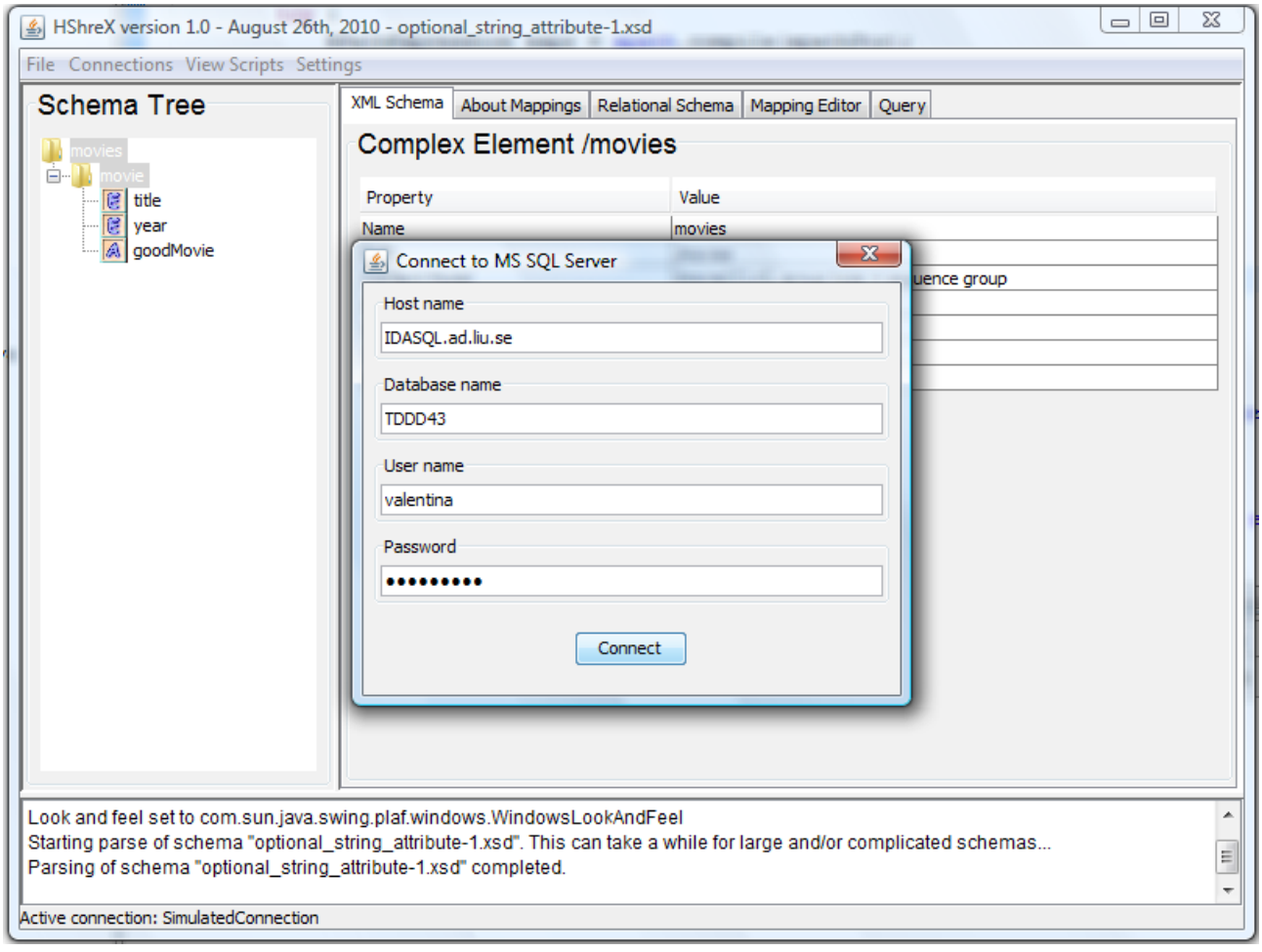
When an XML schema has been successfully opened, the menu item “Create Relational Schema in Database” under the ”File” menu becomes enabled. The user needs to choose this option to create the relational schema in the currently active connection. First thing, that we do when we create the schema in the database, is to drop the existing tables. After that we create the new tables and their foreign keys.

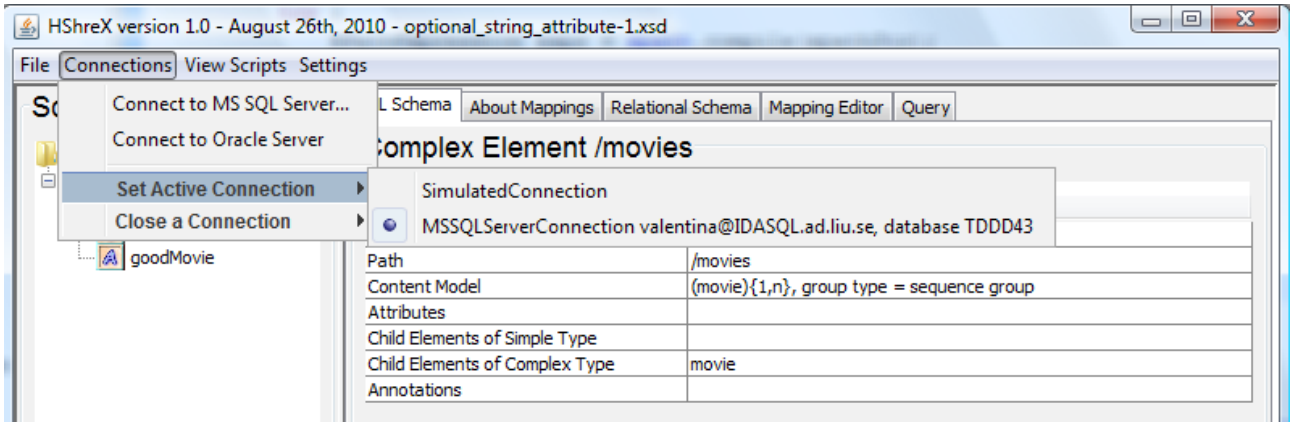
Note: The creating of the relational schema and the loading of the data may be slow. During this time the GUI “freezes” (i.e. not accept any user actions).

The user can create the relational schema using a simulated connection (SimulatedConnection) – always availble, or a database connection. Currently HShreX supports MS SQL and Oracle database servers. **We use only MS SQL server in the TDDD43 course.**

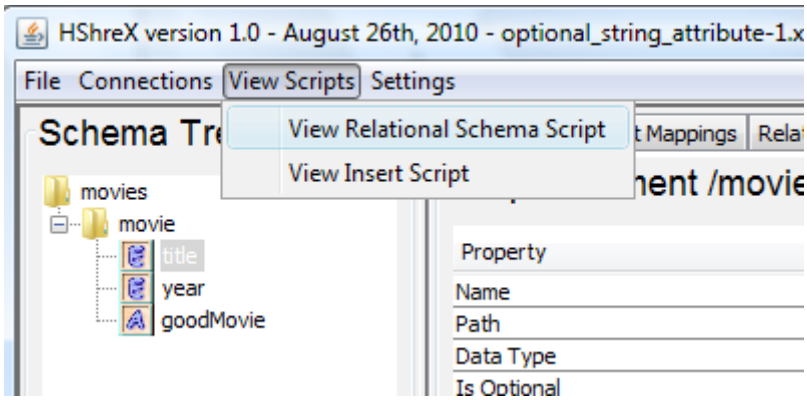
Connections can be managed under the ”Connections” menu. Here you can connect to a live database, set an active connection and close a connection (however, the simulated connection cannot be closed, wouldn't make much sense to do that).

Choose ”Connect to MS SQL server”, under the ”Connections” menu, to connect to the MS SQL. Fill the required data in the dialog and press the ”Connect” button. The user gets a message that shows the state of the connection – ”Connection successfully established with MS SQL server” when the connection is established or an error message in case of an error. The user can choose which connection to be active under the ”Connections” menu. If there are no database connections the SimulatedConnection is the active connection.





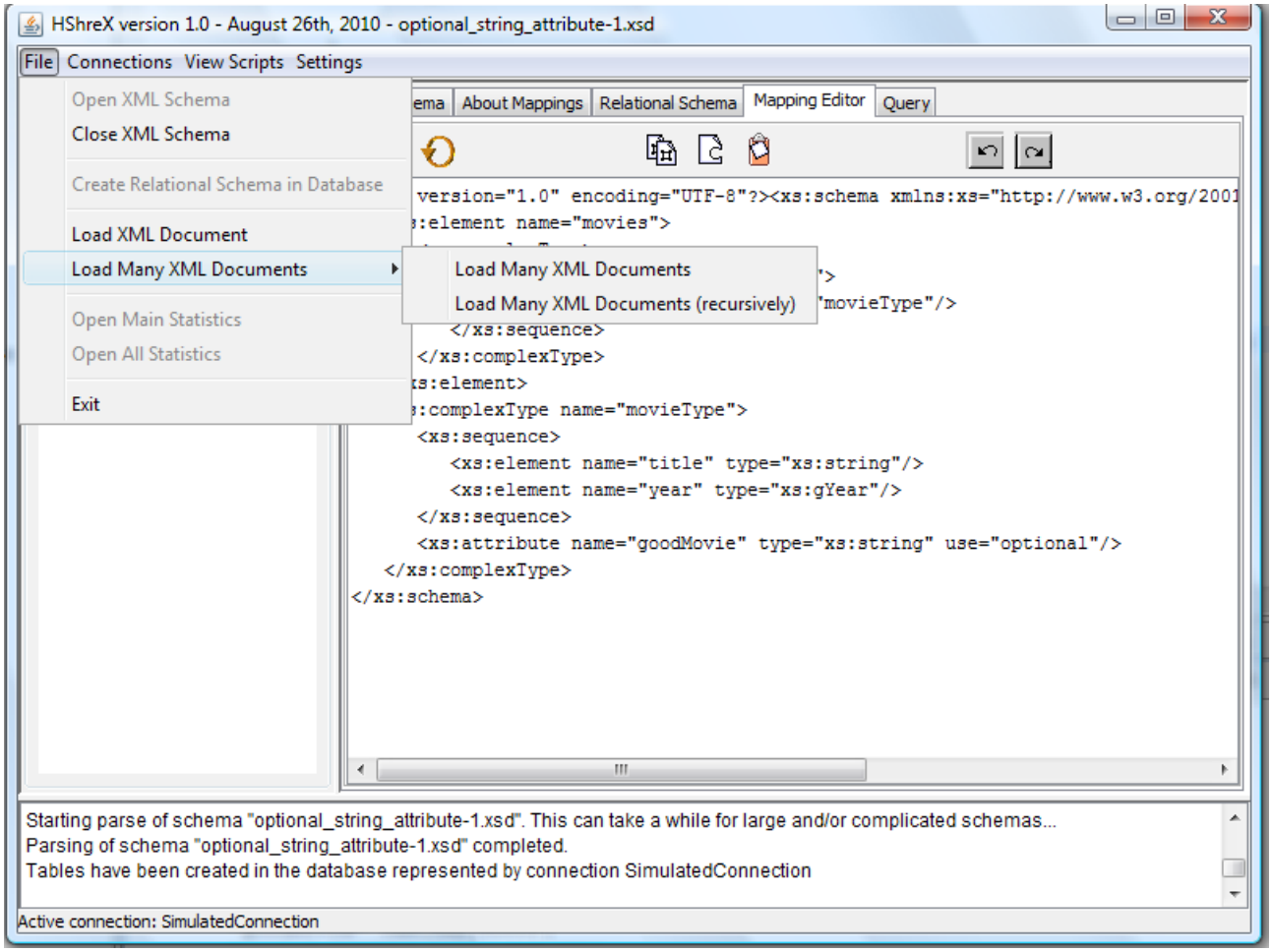
When the relational schema is created the user can see the create schema scripts under the ”View Scripts” → ”View Relational Schema Script”. The insert data scripts are available under the ”View Insert Script” menu item after an XML data file is loaded.



On the following page I will explain how to load data into your database after successfully creating the relational representation as just explained.

Loading XML Data

If the creation of the relational schema in the database is successful, you can now load data into your tables. You do that by selecting one of the menu items that have now been enabled, see picture:



The menu item “Load XML Document” enables you to select a single XML-file and load that into the database. If you want to load several files, you can choose one of the menu items under the sub-menu “Load Many XML Documents” which enables you to select a directory from which all .xml-files are loaded. The recursive one will go through the entire directory tree starting from the root directory that is selected (be careful with that one).

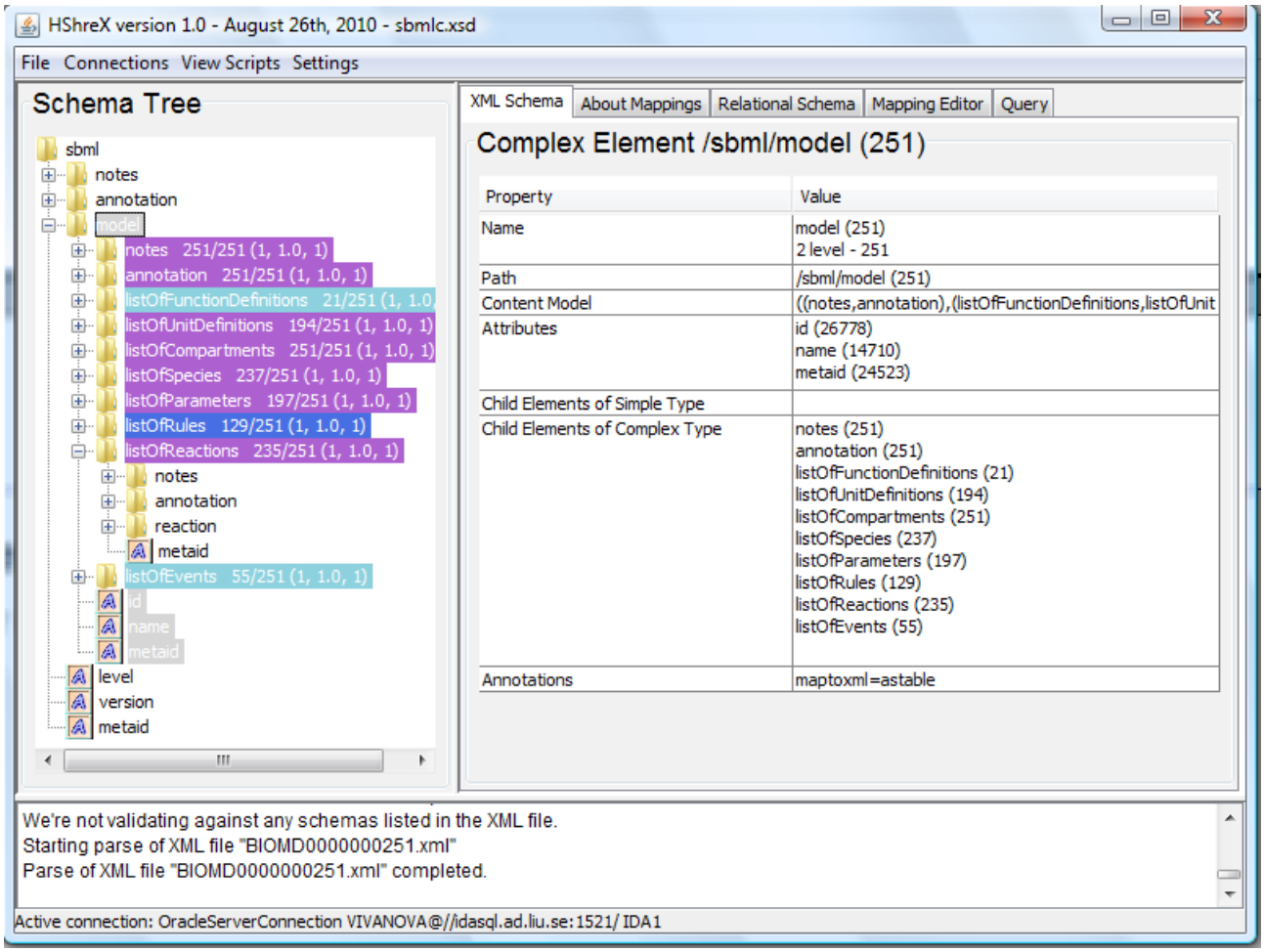
So what happens when we open a given XML file? Well, unless validation has been turned off (which you do by toggling the menu item “Session Settings”->”Validate XML Documents”), HShreX will validate the XML file against all schema references found in the XML document. Note that HShreX doesn't associate the currently opened schema with XML document, but instead checks the XML document for schema references. Also, you should only turn off schema validation if you have XML documents that have no schema reference in them, but you know validate against the schema you're working with.

If validation passes (or, if it's turned off, the XML-document passes the well-formed check) an insert script is generated and executed in the currently active database connection. The insert statements and their values are available under the ”View Scripts” → ”View Insert Script” submenu.

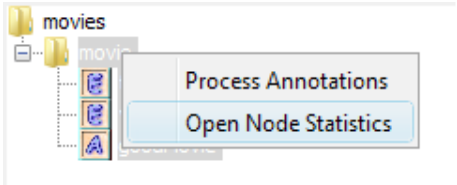
Note: The creating of the relational schema and the loading of the data may be slow. During this time the GUI “freezes” (i.e. not accept any user actions).

The statistical data in HShreX

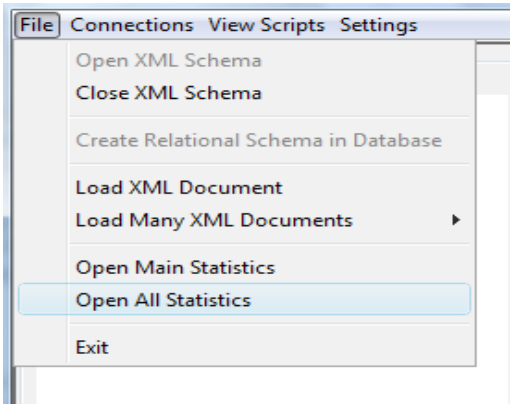
After the XML data file/s is/are loaded the HShreX schema tree looks like the following picture:



When the user selects/opens a node in the schema tree a statistical information for this node and its children becomes available. This data is presented in the interface as a number pair in the child node name where the first number shows the number of times the child element occurs under its parent and the second number shows the number of times the parent element is presented in this dataset at this level. The three numbers, in the parenthesis in the child node name, show the minimum, the maximum and the average number of times this child occurs for this parent. The children nodes are colored in three different colors to facilitate the users perception for the number of times that the children nodes appear under the current parent node. More detailed statistical information for this node is presented in the “XML schema” tab in the right side. The same information is available in the “Open Node Statistics” dialog:



Additional statistical data can be found in “Open Main Statistics” and “Open All Statistics” dialogs under the “File” menu. The data is visualized in a dialog. Statistical information for the whole loaded dataset is shown in the ”Show Main Statistics Dialog” dialog. This information includes



path/s to the loaded file/s, time for loading in the database, the total number of attributes, elements, and levels in the file/s, the average depth for the elements, number of unique paths and elements. In the ”Show All Statistics Dialog” dialog is available the information from the “XML schema” tab and the ”Show Main Statistics Dialog” dialog as well as information for the number and the structure of the unique paths and elements at different levels in the dataset.

A few things to note...

Here's a short summary of things to note when using HShreX.

- When working in the mapping editor (we recommend the ”Annotation Dialog” instead) and the annotation dialog and making changes to schema, the old schema file is overwritten. This is true for reloading the schema too which saves any changes first.
- The XML data files that are loaded are not associated with the opened schema, instead any schema references are extracted from the XML files themselves and they're validated against those. If your XML file does not contain any schema references, you should turn off validation by toggling the menu item “Session Settings” → ”Validate XML Documents”. This means that you yourself must make sure that any XML file you try to load actually validates against the schema that is loaded, or you will get errors about missing mapping information when parsing the XML data file (this isn't a really a HShreX problem, but a usage error when one is trying to match the wrong XML data file with the current schema).
- When loading many XML data files and you choose the recursive way, be careful because HShreX will then go down the entire tree of the directory you select, trying to load any .xml-files that it encounters and there's no clean way to interrupt this process if you happen to select a wrong directory. Be extra careful if you're running against a live database connection on a production machine as well.
- The user can change the size of the VARCHAR and the CLOB fields. This option is available in the “Settings” → “Change Field Sizes...” sub menu.
- When using a live database connection, HShreX does not attempt to avoid invalid column or table names. It's up to the user to detect and fix any such problems, should they occur, using the fieldname/tablename set of annotations.
- We have used version 2008 of SQL Server exclusively, HshreX might also work with version 2005 but that is completely untested. We have used HshreX with Oracle 11R1 Enterprise Edition and Oracle 11R2 Enterprise Edition as well. **We use only MS SQL server in the TDDD43 course.**
- If you start HShreX and do not have anything in the working area contact with your lab assistant.
- The creating of the relational schema and the loading of the data may be slow. During this time the GUI “freezes” (i.e. not accept any user actions).