

The SIS Thesaurus Management System

SIS-TMS Tutorial

June 2000

Institute of Computer Science

Foundation for Research and Technology - Hellas

TABLE OF CONTENTS

1. INTRODUCTION	5
1.1 Technical Characteristics	5
1.2 Current Status	5
2. GRAPHICAL ANALYSIS INTERFACE FUNCTIONALITY	6
2.1 General Description	6
2.2 Queries	8
2.2.1 Tree Views menu	8
2.2.2 Queries menu	10
2.2.3 Retrieval menu	11
2.3 Query Result Presentation	12
2.3.1 Column Textual Display	12
2.3.2 Hierarchical Report Display	13
2.3.3 Tree Display	14
2.3.4 Star View	15
2.3.5 Object Card	16
2.4 Interaction Situations	17
2.5 Reports on Hierarchy Terms	18
3. ENTRY FORM FUNCTIONALITY	19
3.1 General Description	19
3.1.1 Task Window	19
3.1.2 Creation, Renaming, Deletion Operation Windows	20
3.1.3 Classification, Generalization Operation Window	21
3.1.4 Attribute Assignment Operation Window	22
3.1.5 Selection List	23
3.2 Selection of Task	24
3.2 Manipulation of New Facets	25
3.2.1 Create a new facet	25
3.2.2 Rename a new facet	26
3.2.3 Delete a new facet	26
3.2.4 Describe a new facet	27
3.3 Manipulation of New Hierarchies	28
3.3.1 Create a new hierarchy	28
3.3.2 Rename a new hierarchy	29
3.3.3 Delete a new hierarchy	30
3.3.4 Describe a new hierarchy	31
3.3.5 Classify a new hierarchy in a facet	32
3.4 Manipulation of New Concepts	33

3.4.1 Create a new concept	34
3.4.2 Rename a new concept	35
3.4.3 Delete a new concept	35
3.4.4 Describe a new concept	36
3.4.5 Classify a new concept	37
3.4.6 Move new concept to another hierarchy	38
3.4.7 Associate an new concept with terms from other thesauri	40
3.4.8 Disassociate a new concept with terms from other thesauri	41
3.4.9 Sort the broader term links of a new concept	42
3.5 Manipulation of Existing Facets	43
3.5.1 Rename an existing facet	44
3.5.2 Abandon an existing facet	44
3.5.3 Describe an existing facet	45
3.6 Manipulation of Existing Hierarchies	46
3.6.1 Rename an existing hierarchy	47
3.6.2 Abandon an existing hierarchy	48
3.6.3 Describe an existing hierarchy	49
3.6.4 Classify an existing hierarchy in a facet	50
3.7 Manipulation of Existing Concepts	51
3.7.1 Describe an existing concept	52
3.7.2 Abandon an existing concept	53
3.7.3 Undo Abandon an existing concept	54
3.7.4 Move descriptor to another hierarchy	55
3.7.5 Rename an existing concept	57
3.7.6.Undo Rename an existing concept	58
3.7.7 Associate an existing concept with terms from other thesauri	59
3.7.8 Disassociate an existing concept with terms from other thesauri	60
3.7.8 Sort the broader term links of an existing concept	61
3.8 Manipulation of Sources	62
3.8.1 Add a new source	63
3.8.2 Rename a source	63
3.8.3 Delete a source	64
3.8.4 Describe a source	65
3.8.5 Classify a source	66
3.9 Manipulation of Editors	67
3.9.1 Add a new editor	68
3.9.2 Delete an editor	68
3.9.3 Rename an editor	69
3.9.4 Describe an editor	70
4. UPDATE APPLICATIONS	71
4.1 Release Thesaurus	71
4.2 Create Hierarchy	72
4.3 Delete Hierarchy	73

1. Introduction

1.1 Technical Characteristics

The SIS Thesaurus Management System SIS-TMS consists of a tool to develop multilingual thesauri and a terminology server for cataloguers and for distributed access to heterogeneous electronic collections. The distinct features of the TMS are its capability to store, develop and access multiple thesauri and their interrelations under one database schema, to create any relevant view thereon and to specialize dynamically any kind of relation into new ones.

The SIS-TMS graphical user interface (also called *Graphical Analysis Interface - GAIN*) allows the unconstrained navigation within and between different thesauri and the execution of predefined queries and graphical views to identify concepts for cataloguing or database queries, to identify translations or equivalent expressions for information access in heterogeneous environments, and to control the quality and logical consistency of a system of interlined thesauri during the development.

The SIS-TMS server can be integrated in a distributed, heterogeneous environment. As a central, eventually repeated component, it can replace the cumbersome implementation and population of thesaurus management features in collection databases and library systems, due to access through its programmatic interface. It further allows automatic term expansion and translation in distributed access environment. This use requires consistency of the equivalence relations established between thesauri. The means of consistency control provided by SIS-TMS is a unique feature.

The SIS-TMS system is an application of the Semantic Index System, a general-purpose object-oriented semantic network database, product of ICS-FORTH. Its schema is based on the principles of the ISO 2788 and 5964 standards for the establishment and documentation of monolingual and multilingual thesauri. It is outcome of international co-operation with cultural organizations.

1.2 Current Status

The SIS-TMS is available on Windows '95, Windows NT, Solaris, HP-UX, and AIX.

Test versions exist with the content of the AAT (Art & Architecture Thesaurus), the thesaurus of the Royal Commission of Historical Monuments of England (RCHME) and of the French MERIMEE database and equivalence relations between the three.

A first version has been installed at the French Ministry of Culture (Inventaire General), and the Italian Ministry of Culture (ICCD).

2. Graphical Analysis Interface Functionality

2.1 General Description

The Graphical Analysis Interface (hereafter called *GAIN*) is described in detail in “*SIS-Graphical Analysis Interface, User’s Manual*”. In this tutorial we will describe in brief *GAIN*’s functionality.

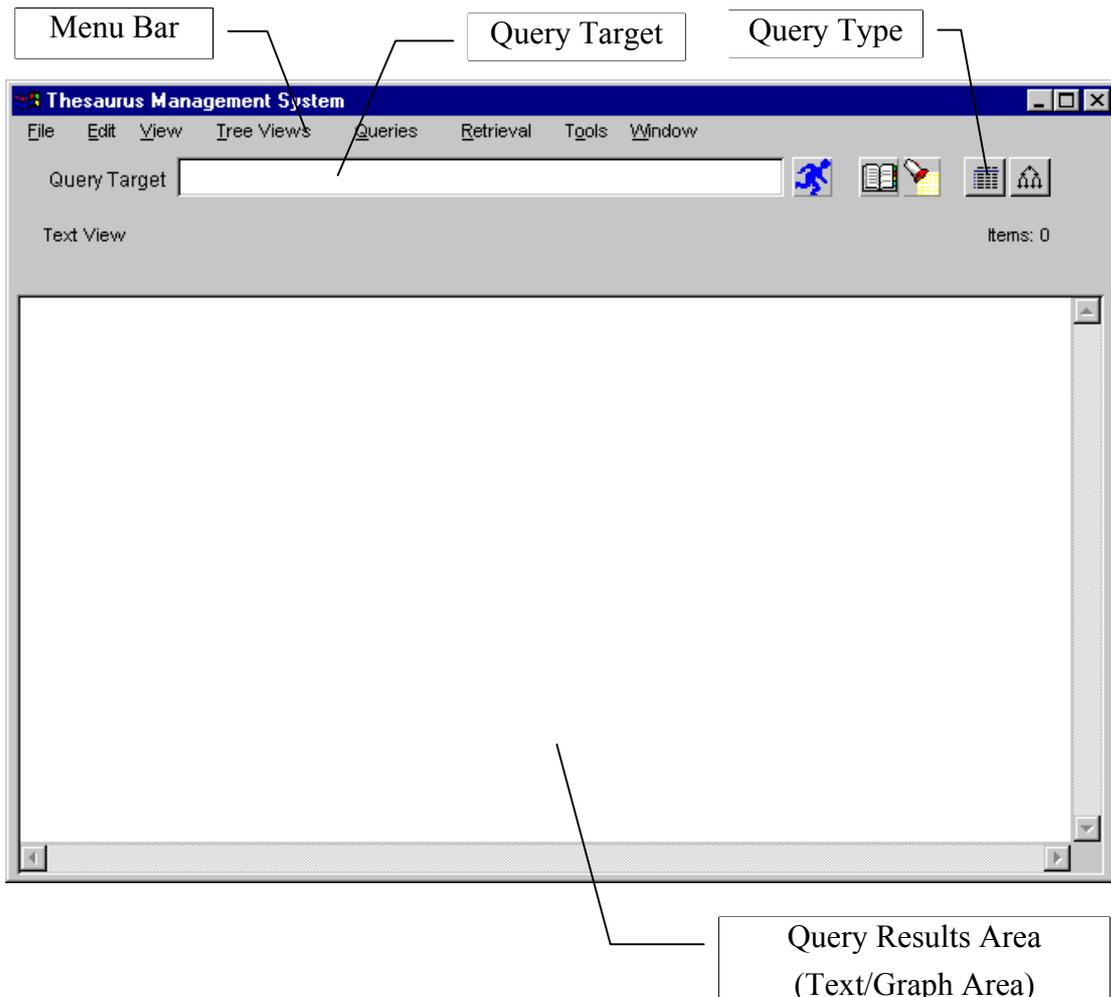


Figure 1 GAIN main window

GAIN cooperates with the query processor of the SIS base. Information can be retrieved from the SIS base, by executing one of a set of built-in queries, which are offered as a menu of choices by the interface. The query processor extracts data from the SIS base and displays the result on the screen. The result can be seen in two ways: graphically, on the window of the graphical subsystem or textually, on a text-window. There are many types of predefined queries; some of them are graphical (display the result in graphical mode), while others are textual (display the result in textual mode). The current selection of the query type is displayed by the query type field, which is always visible (see Figure 1). A query may have one parameter on which it operates or may have none. This parameter, referenced in the following as *Query Target*, must

be an object existing in the SIS base. The Query Target is always visible and can be changed in multiple ways by the user.

The GAIN main window is divided in three basic areas (see Figure 1):

- the *Menu-bar*, which provides all the built-in queries and a set of operations on the visual representation of the query results.
- the *Query Info area*, which includes the *Query Target* area and a toolbar for the most frequently used operations.
- the *Query Results area*, which displays the results (graphical or textual) of the queries to the SIS base (hereafter referred as “Text Area” and “Graph Area”)

Except of the main window a number of pop-up windows are triggered by the menubar or the toolbar selections:

- the *Object Card window*, which displays the textual description of an object.
- the *Global View window*, which displays the global view of the graph presented in the graphical window.
- the *Options window*, which enables the user to set its preferences for the fonts, colors and text messages of the user interface area.
- the *History window*, which includes a list of the last executed queries.

All the above windows and their functionality are described in detail in “*SIS-Graphical Analysis Interface, User’s Manual*”.

2.2 Queries

The SIS-TMS provides three menus of built-in queries: a) the *Tree Views*, a menu of queries whose results are displayed in graphical mode, b) the *Queries*, a menu of queries whose results are displayed in textual mode and c) the *Retrieval*, a menu of queries by classification facets, whose results are displayed in textual mode.

2.2.1 Tree Views menu

The graphical queries are performing search in depth, and they are performed on a specific target. They provide visual information about connection between objects. The query *StarView* is an equivalent graphical representation of an *Object Card*.

Figure 2 shows the *Tree View* menu. Figure 3 shows the *Global View* window, which displays the global view of the graph presented in the *Graph Area* in Figure 2.

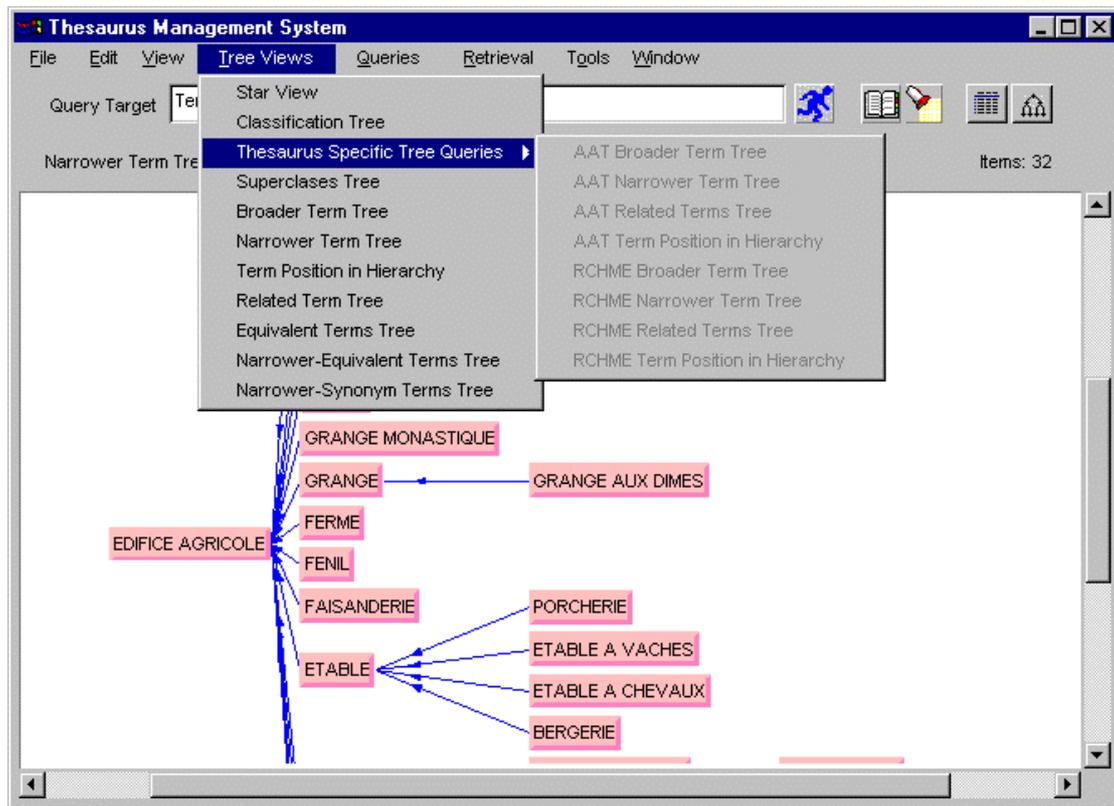


Figure 2 *Tree View* menu displays graphical query results

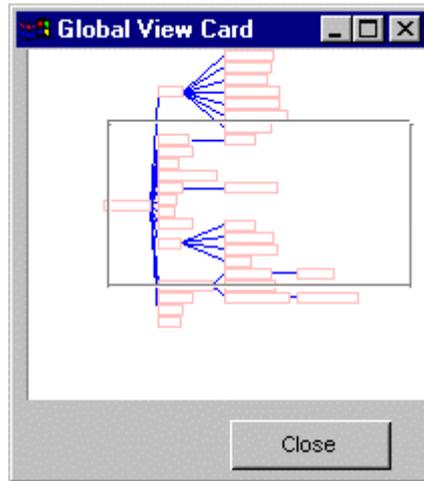


Figure 3 *Global View* window shows which part of the graph is visible.

2.2.2 Queries menu

The results of the *Queries* are displayed in the *Text area* in columns (see Figure 4). This menu provides queries about all facets, or all facets by a specific parameter (*QueryTarget*). The queries about all facets (do not apply on a specific *Query Target*) have the prefix “*List All*”.

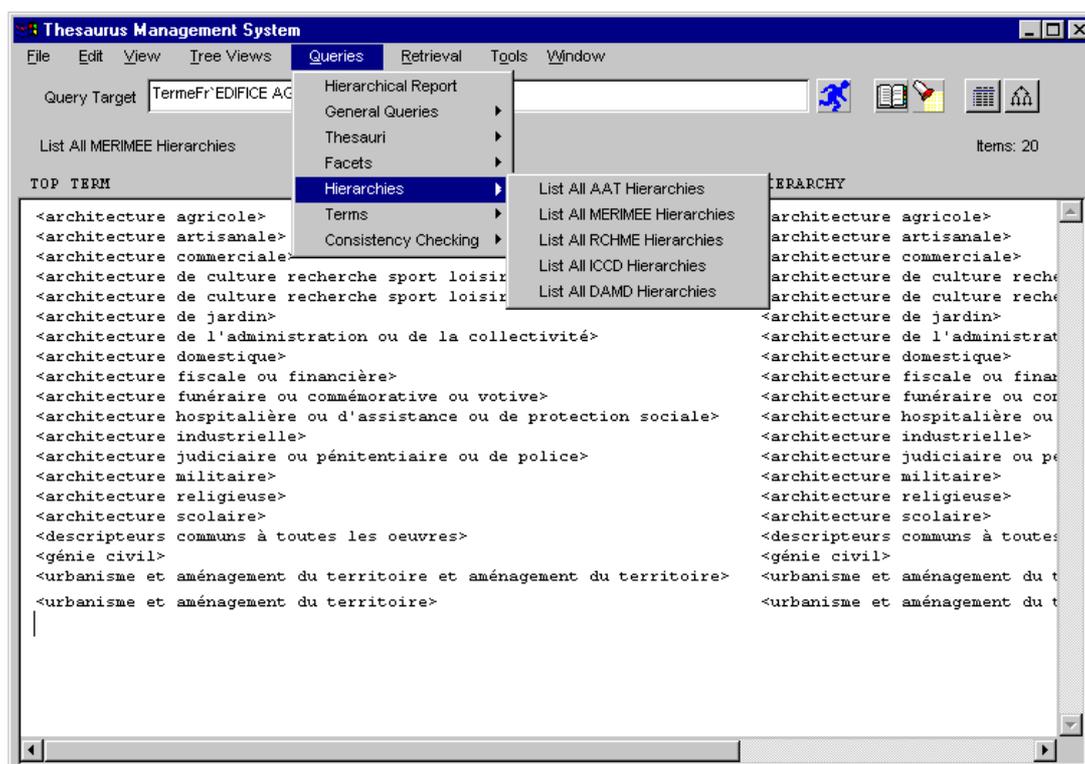


Figure 4 *Queries* menu displays textual results

2.2.3 Retrieval menu

This menu provides queries about all facets by combinations of all others. The user can "fill" the specified facet from the *QueryTarget*.

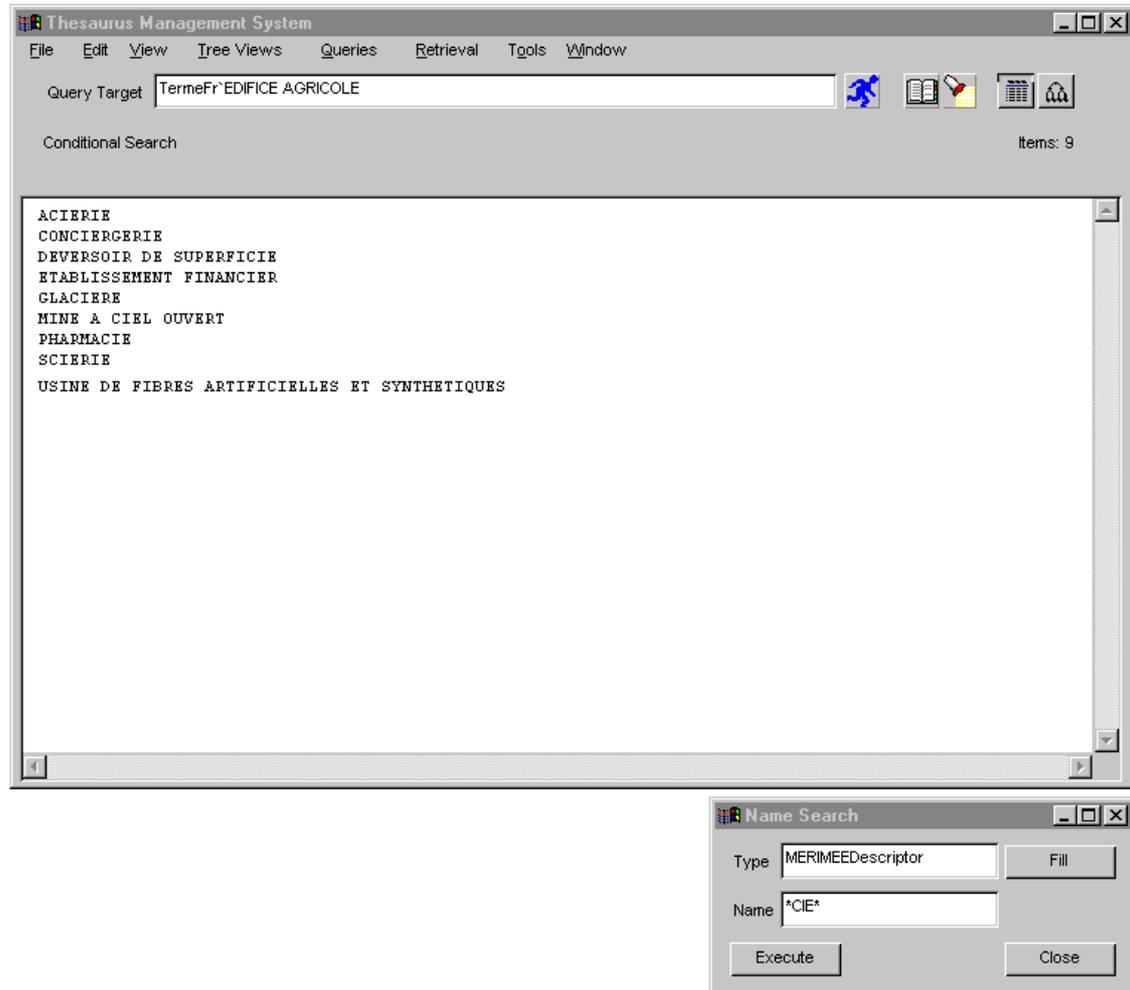


Figure 5 Retrieval menu displays textual results.

2.3 Query Result Presentation

The SIS-TMS provides various presentations for the information retrieved from the SIS base. Selecting a query from the *Queries* menu or the *Retrieval* menu the textual results are displayed in columns, while selecting a query from the *Tree View* menu the results are displayed as graphs. A specific *Tree View* query, called *Star View*, presents all the information associated with an object (designated in the *Query Target*) in the form of a tree-graph, while the same information is presented in textual form in a pop-up window, called *Object Card*.

2.3.1 Column Textual Display

The result of textual queries is the names of the objects existing in the answer set. This information may be presented in columns, as shown in Figure 6. Each column corresponds to an attribute of the objects in the answer set. The kind of each attribute appears as a label above the corresponding column.

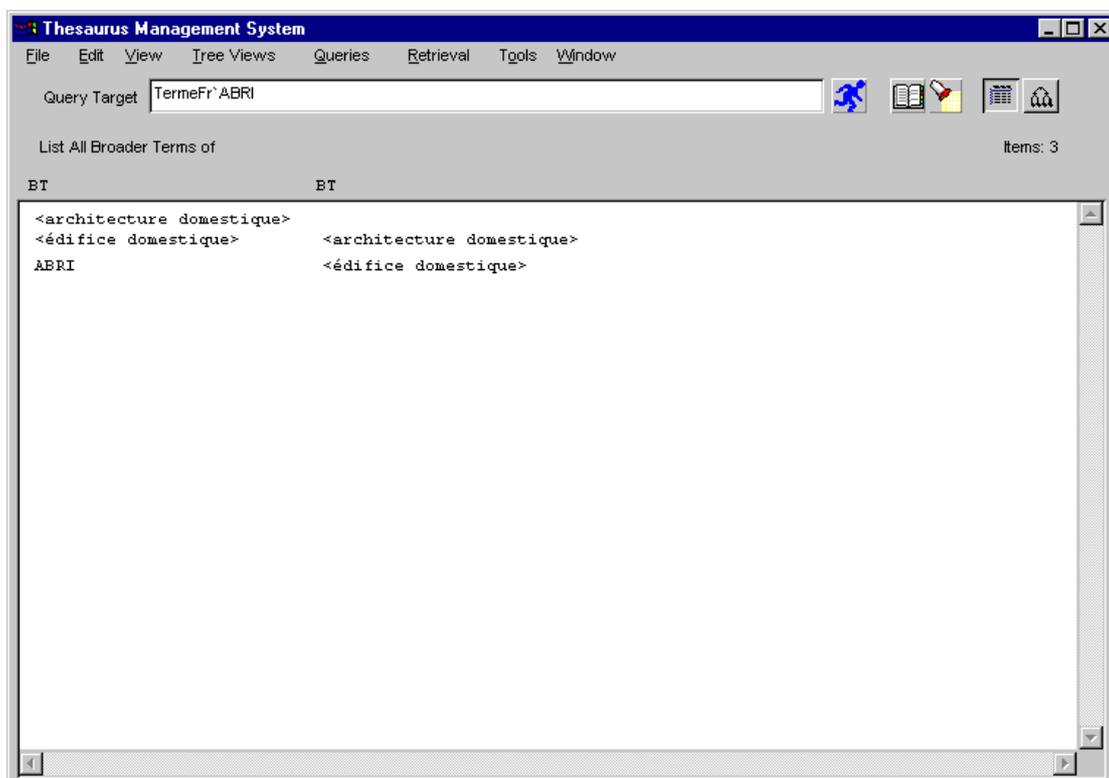


Figure 6 Text Area displays the results in columns

2.3.2 Hierarchical Report Display

The result of a hierarchical report is the textual representation of a tree graph query (e.g. Narrower Term Tree) as shown in Figure 7.

When splitting has been done for a node the symbol M is added in front of this node and the number N (declaring the repetition) is also added in front of the repeated node (see PIGEONNIER node in Figure 7).

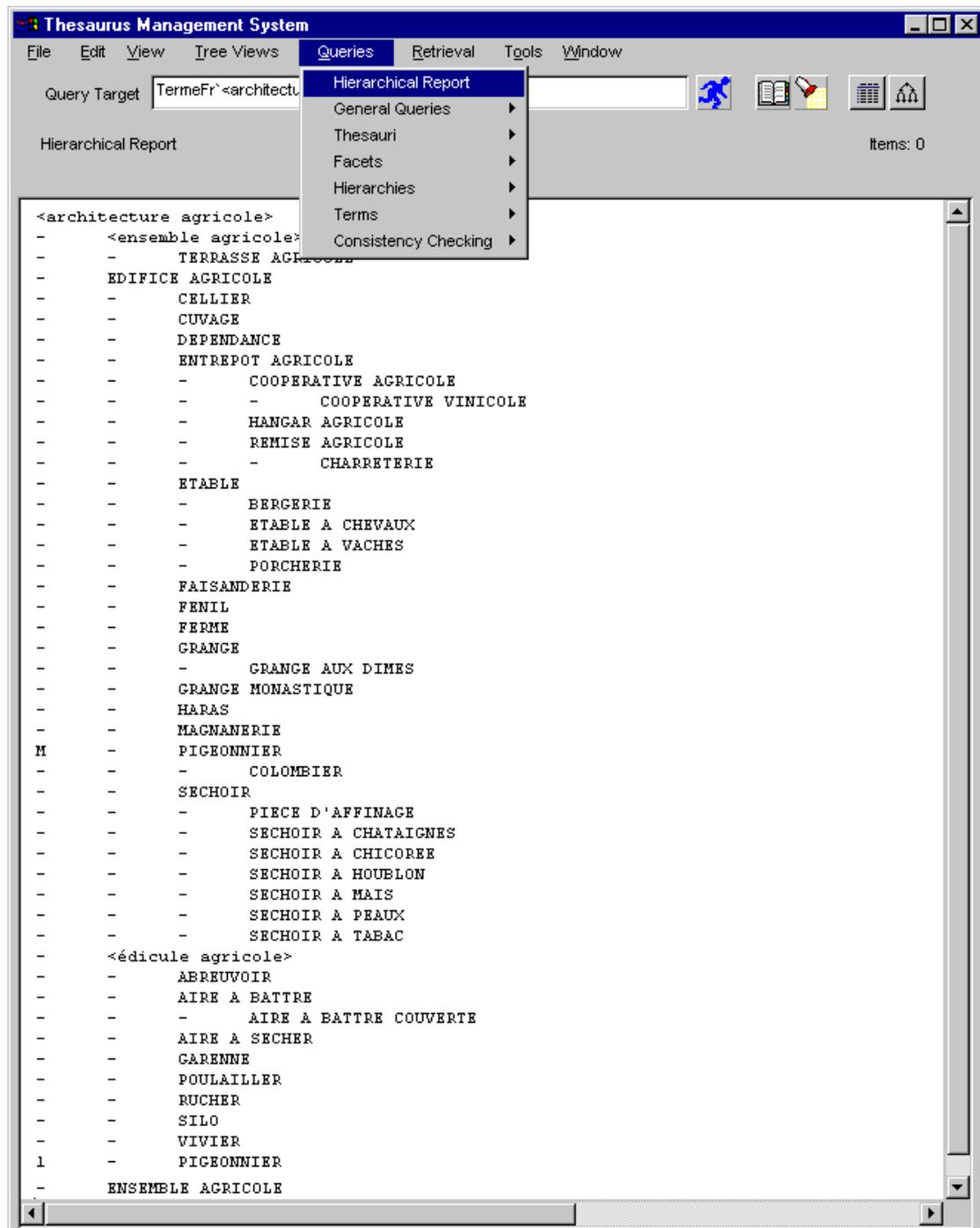


Figure 7 Text Area displays the result of the hierarchy report

2.3.3 Tree Display

The choices of the *Tree Views* menu are recursive queries, displayed as graphs (see Figure 8). Some of them require a specified kind of target. A checking is performed when the menu is mapped on screen in order to verify that the given query target is of the kind that the queries require. All queries that require a different kind of target than the one presented in the *Query Target* area automatically become inactive.

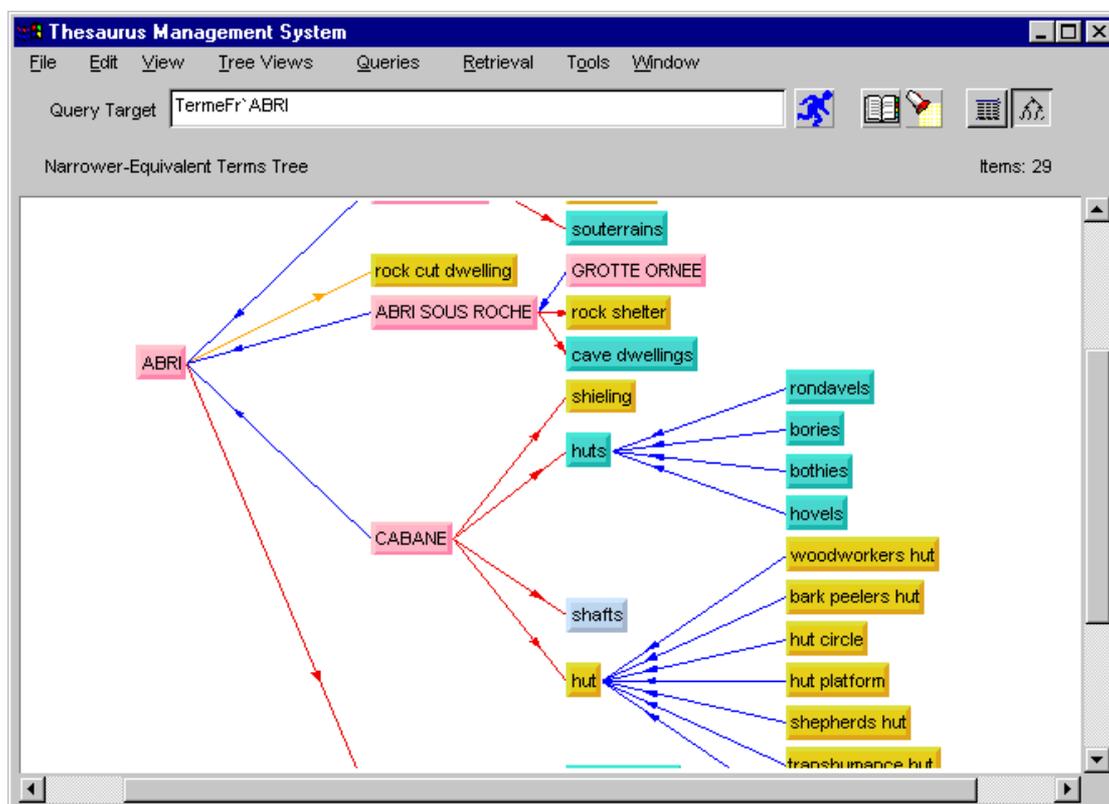


Figure 8 Tree graph results

2.3.4 Star View

By selecting *StarView* from the menu, a graphical query is executed which displays the query target as a central object (see Figure 9). The superclasses and subclasses of the central object are shown top-right and bottom-right respectively. The classes of which the central object is an instance of, are shown top-left, while if it has instances a box with the label "INSTANCES" appears bottom-left.

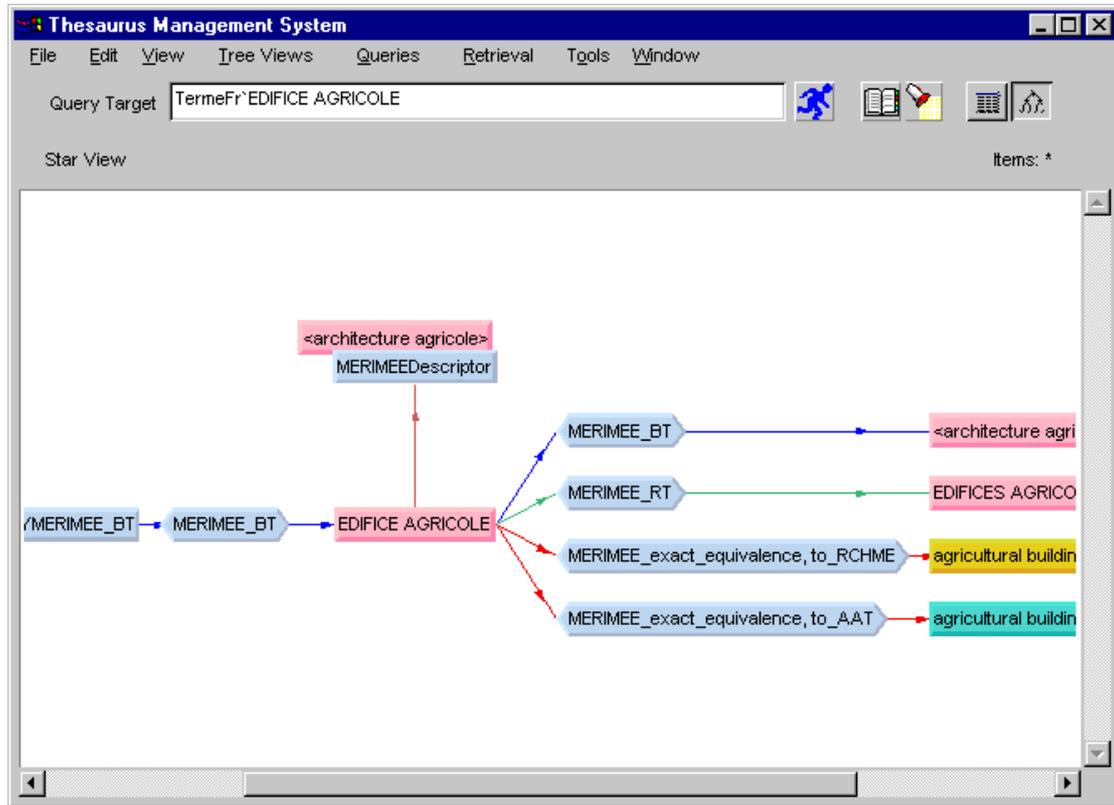


Figure 9 Star View result display

2.3.5 Object Card

The *Object Card* of an object contains the textual description for this object. The object card shows the complete information that is immediately related to this object (see Figure 10). The *Object Card* window is popped-up by clicking the right button on the object (its box in the graph or its name on the display, even on another *Object Card*).

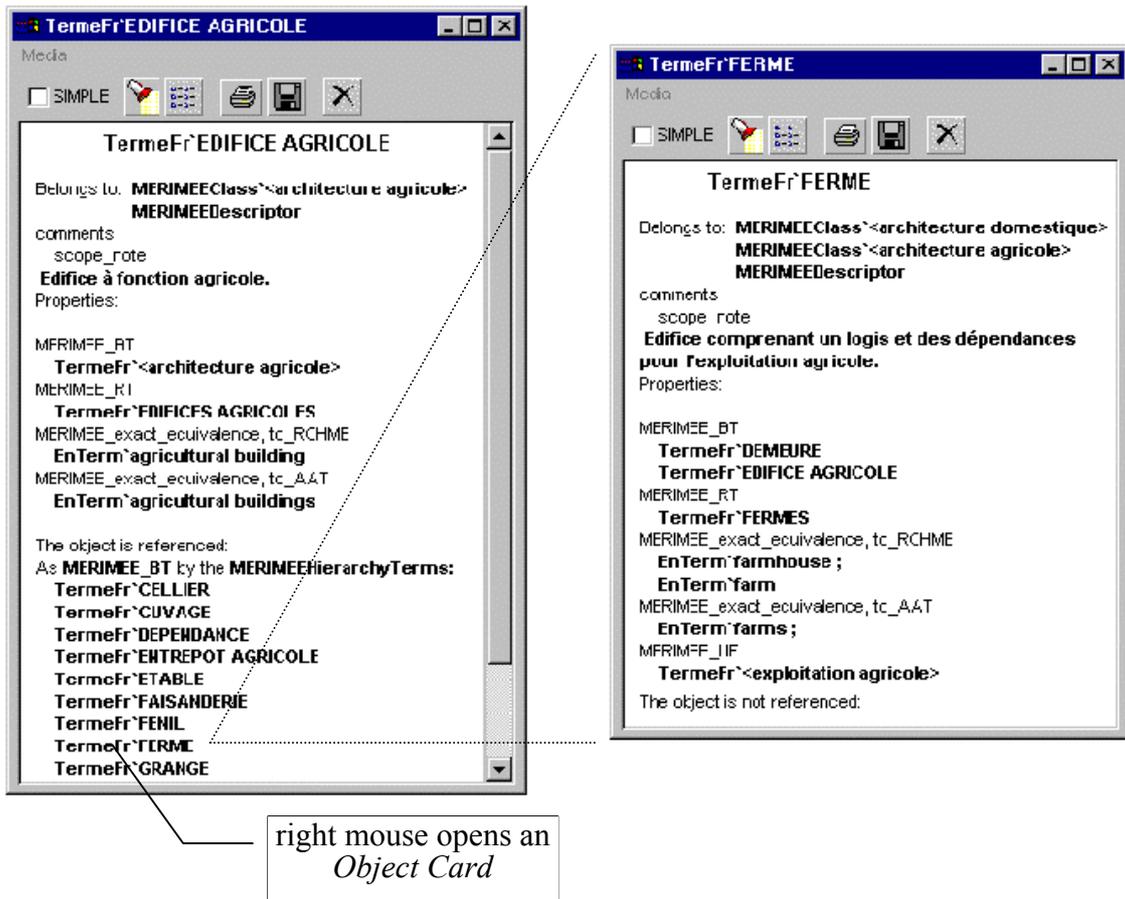


Figure 10 Object Card windows

Note: If a user tries to open an already opened object card the previous one close and the new one is built.

2.4 Interaction Situations

In this section we will describe the possible situations of interaction with the SIS-TMS Graphical Analysis Interface (GAIN).

a) Empty Screen

Initial Status	Next Status
Type a <i>Query Target</i>	<i>Query Target</i> set
Select a " <i>List All...</i> " <i>Query</i>	" <i>Text Area</i> " filled
Select a <i>Retrieval Query</i>	" <i>Text Area</i> " filled

When we first enter the system both the *Query Target* and the "*Text*"/"*Graph*" areas are empty. In this case we can either type a target on the *Query Target* area (in order to perform a query on it), or we can select from the *Queries* those that can be performed without a target (e.g. "*List All...*"). A third case is to make a retrieval based on specific criteria that we will provide to the *Retrieval* card window.

b) Query Target Typed

Initial Status	Next Status
<i>Tree Views</i> *	" <i>Graph Area</i> " filled
<i>Queries</i> *	" <i>Text Area</i> " filled

(*check: target is valid if *StarView* is selectable)

If we have set the *Query Target* we can select to perform either graphical or textual queries on it. The type of the object typed specifies the queries that are available. The queries in menu *Tree Views* will display their results in the graphical area and those in *Queries* will display their results in the textual area. Note that we can determine if an object exists in the base, if the *StarView* menu selection is available.

c) "Text", "Graph" areas filled & Object Card

Initial Status	Next Status
left mouse click	<i>Query Target</i> set
right mouse click	<i>Object Card</i>
middle mouse click	repeat last <i>Tree Views</i> query
(With two-button mouse-devices by double clicking the left button simulates the middle-button.)	(on new <i>Query Target</i>)

Toggle *Text/Graph* area

If there is information displayed on the *Text* or the *Graph* areas or an *Object Card* is open, we can set the *Query Target* by clicking on an object. The left button sets the

Query Target with the name of the selected object, the right button opens an *Object Card* for the selected object and the middle button repeats the last *Tree Views query*. The two areas can be toggled, so the user can select from both areas.

2.5 Reports on Hierarchy Terms

The SIS-TMS provides a tool for creating simple reports. This tool (see Figure 11), given a hierarchy, produces a file containing all relative information for every instance of the given hierarchy such as broader term, alternatives (synonyms), scope note and categories it belongs to.

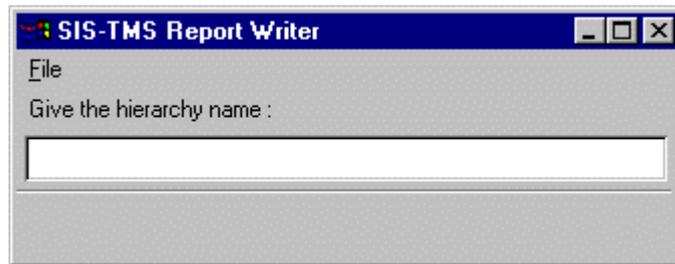


Figure 11 Report Writer window

File Menu contains three options as shown in Figure 12: *Create Report* for starting the report of the given hierarchy, *Save As...* for selecting the file the report will be saved in and *Exit* for exit terminating the program.

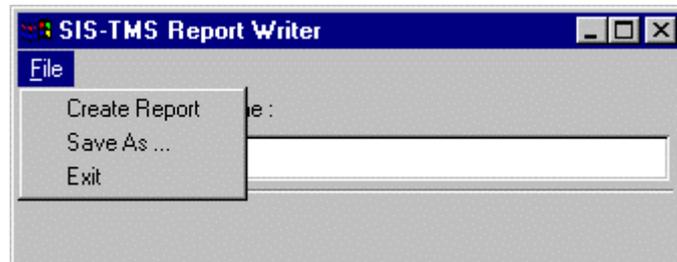


Figure 12 Report Writer menu options

3. Entry Form Functionality

3.1 General Description

The Entry Form (hereafter called *EF*) is a tool used for the interactive update of the SIS base. In this tutorial we will describe in brief *EF*'s functionality. Its functionality is described in detail in "*SIS-Entry Form, User's Manual*".

The update of the information base can be achieved with operations on objects. The available operations are creation, deletion, renaming, classification (assignment to a class), generalization (assignment to a superclass) and attribute assignment. *EF* provides the user with the capability to perform the whole set of the operations mentioned above.

The user updates the SIS base through *EF* in a task-oriented way. This means that the user can update the information base through tasks, which have been assigned to him/her. A task is defined by the objects that are allowed to be updated and the operations that can be performed on these objects. In this way, parts of the information base can be isolated and updated independently from others, using predefined operations.

EF is based on a three-activity process model:

- The first activity is the task selection: the user selects the task through which (s)he wants to update the information base. A *Task window* is popped-up which allows a set of operations to be performed on a set of objects.
- The second activity is the selection of an object and an operation: the user selects objects and operations available in the selected task. An *Operation window* is popped-up which permits the user to go on with the third activity.
- The third activity is the actual update of the information base: having selected the update operation, the user inputs data, which update the selected object. Depending on the specifications given in the process model for the selected operation, a sequence of operations that follow the selected one may be created. The operations are performed on the selected object, while the last operation of the sequence causes the update of the information base with all the changes of the operations (hereafter called *COMMIT*).

EF provides the user with a set of forms through which each activity is performed. Below we will describe in brief few basic forms or other pop-up windows of the *EF* and their functionality.

3.1.1 Task Window

The *Task window* (see Figure 13) is a window, which allows the users to select objects and perform operations on them within a specific *task*. This window keeps a list of all the objects that were selected during the specific task. Each operation triggers the display of a pop-up *Operation window*.

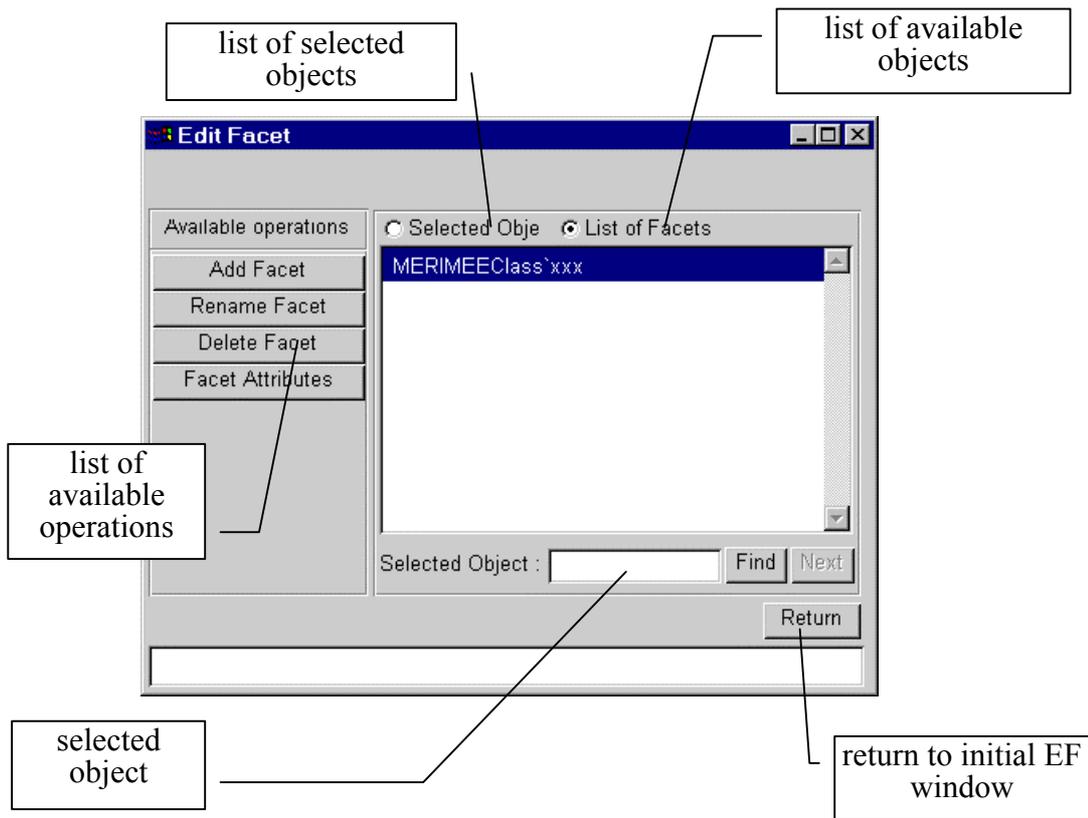


Figure 13 Task window.

3.1.2 Creation, Renaming, Deletion Operation Windows

There are three different *operation windows* for the creation, renaming and deletion operations. The operation windows have similar structure (see Figure 14): the user is asked to type (if necessary) the information needed for the update of the SIS base, and to confirm the update by selecting the "COMMIT" button. In order to return to the

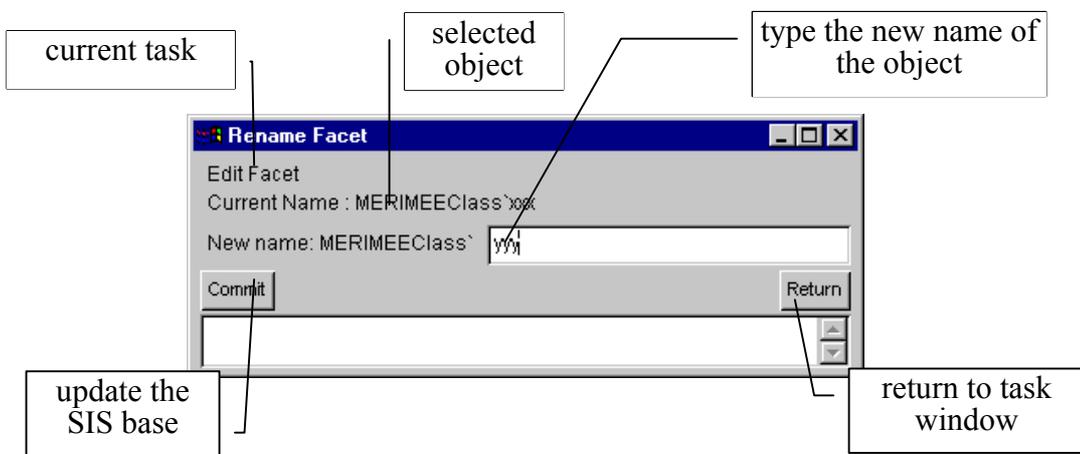


Figure 14 Rename Operation Window

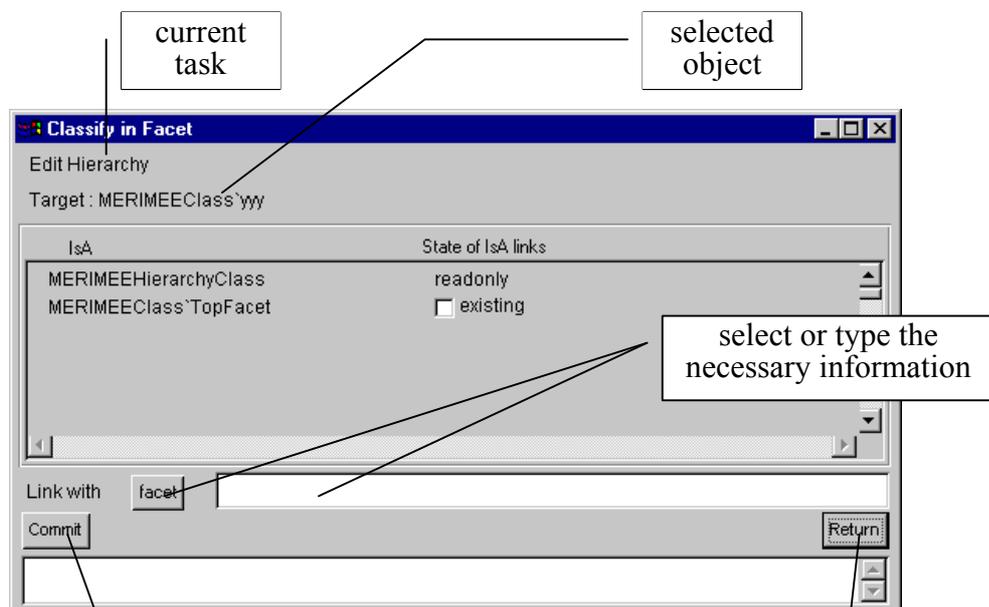
task window of the current task the “RETURN” button must be selected.

3.1.3 Classification, Generalization Operation Window

The *operation window* for the classification or the generalization of an object has similar structure (see Figure 15): the user is asked to select/type (if necessary) the information needed for the update of the SIS base, and to confirm the update by selecting the “COMMIT” button. In order to return to the *task window* of the current

Figure 15 *Operation window* for the classification of a facet.

task the “RETURN” button must be selected.



3.1.4 Attribute Assignment Operation Window

The *operation window* for the attribute assignment enables the user to add or delete attributes to the selected (see Figure 16). To confirm the update the user must select the “*COMMIT*” button. To return to the *task window* of the current task the “*RETURN*” button must be selected.

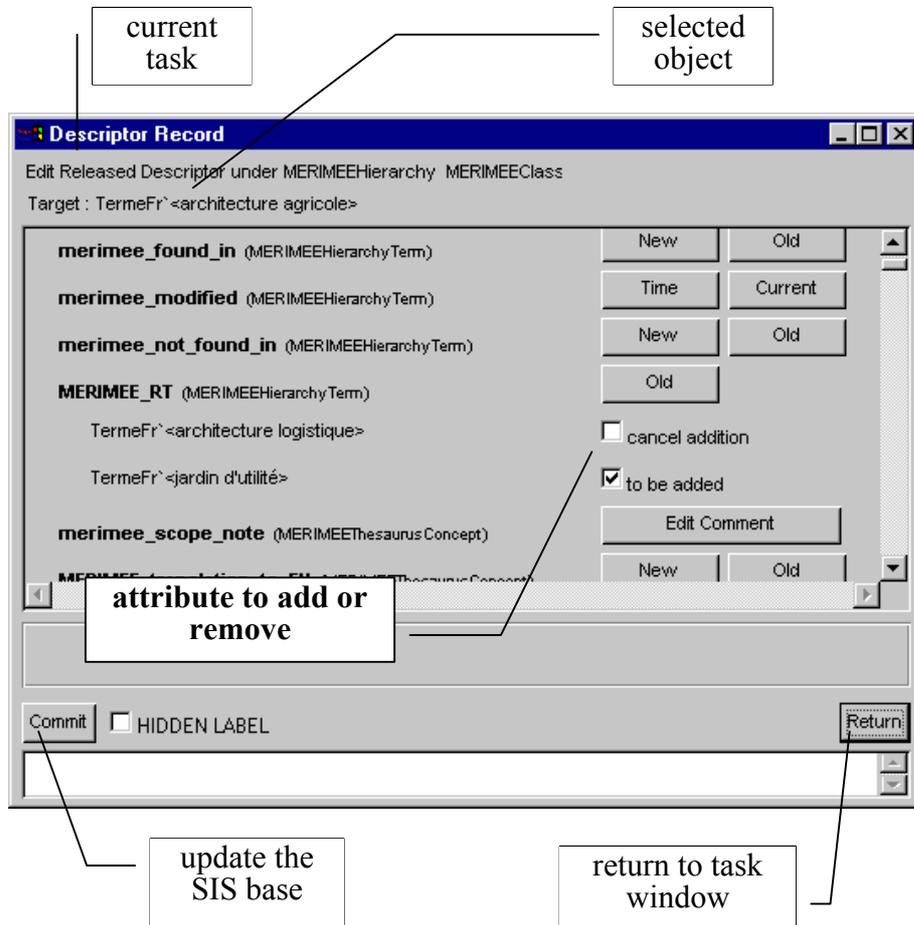


Figure 16 *Operation window* for attribute assignment for an object.

Attribute Selection Buttons

The user can “associate” the selected object with other objects of the SIS base by creating for the selected object an attribute that will point to an object in the SIS base:

by selecting “*OLD*” button, the selected object gets “associated” with an object already existing in the SIS base.

by selecting “*NEW*” button, the selected object gets “associated” with an object that the user is prompted to create.

On the *attribute selection button*, the triangle designates that the attribute to be created will point to a complex object (thus a window will pop-up for specifying additional information), while the exclamation marks designate that this attribute must exist (is obligatory).

The user can associate the selected object with a free text (e.g. scope note) which (s)he must specify by selecting , or to a *primitive value* (string, integer, real) by selecting the appropriate buttons (e.g.). The selected object may also be associated with the current time/date by selecting , or to a specific time/date value by selecting .

A detailed description of the EF windows and their functionality is given in “*SIS-Entry Form, User’s Manual*”.

3.1.5 Selection List

EF provides the user with a *Selection List window* whenever there are many objects to select from (see Figure 17).

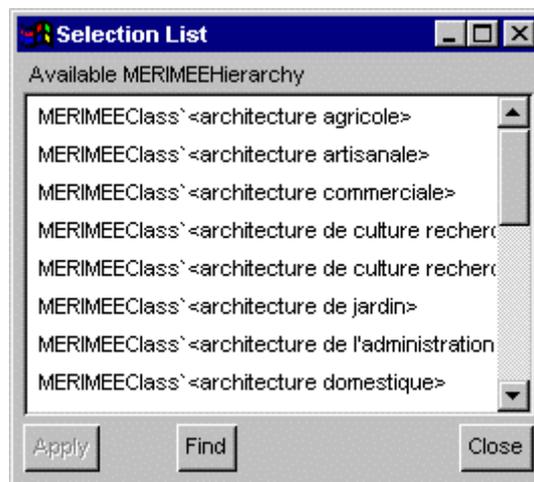


Figure 17 *Selection List window*

3.2 Selection of Task

Bellow we will describe the tasks available in SIS-TMS Entry Form tool. The EF initial window enables the user to select the task through which (s)he wants to update the information base. In Figure 18 we see the list of available tasks: the creation of new Facets, Hierarchies, Descriptors, the modification of existing(also called released) Facets, Hierarchies, Descriptors, the creation/modification of Editors, etc.

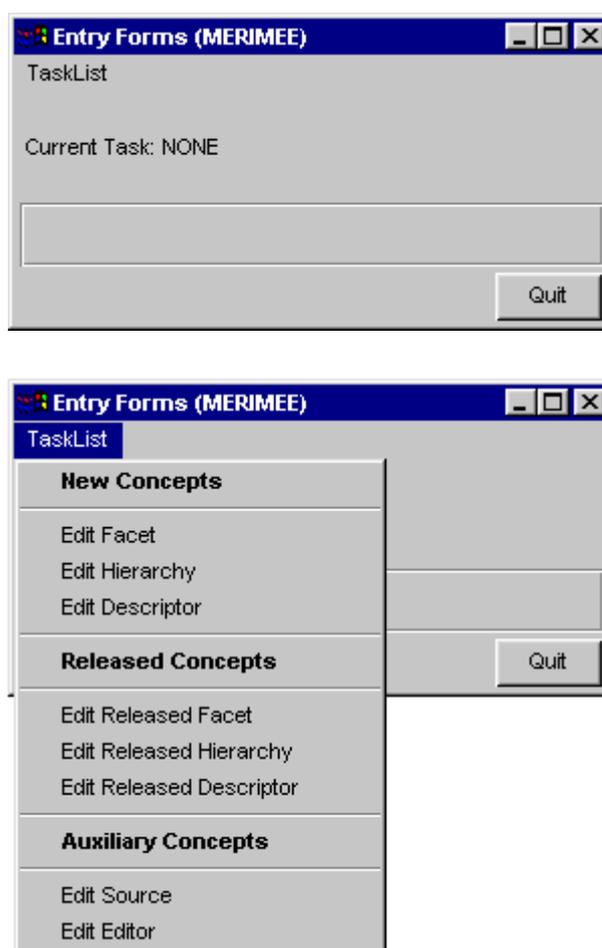


Figure 18 Selection of task from the *Task List*.

3.2 Manipulation of New Facets

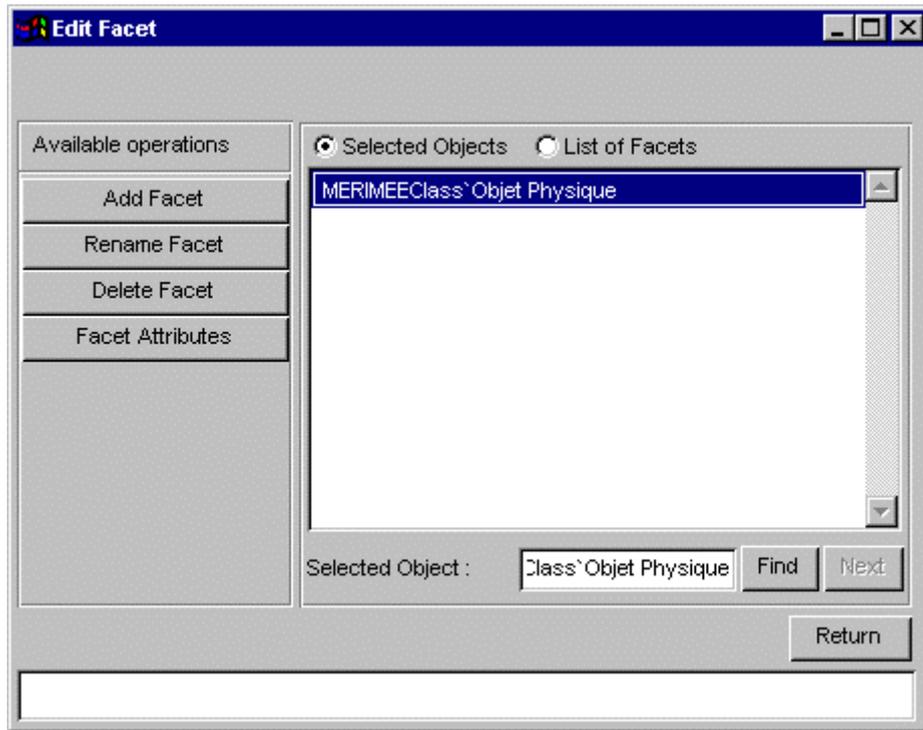


Figure 19 Task window "Edit Facet".

3.2.1 Create a new facet

Task : Edit Facet

Operation : Add Facet

Needed Input : Facet Name

Optional Input :

This operation creates a new facet. The operation fails in the following case:

- A facet with the same name already exists.

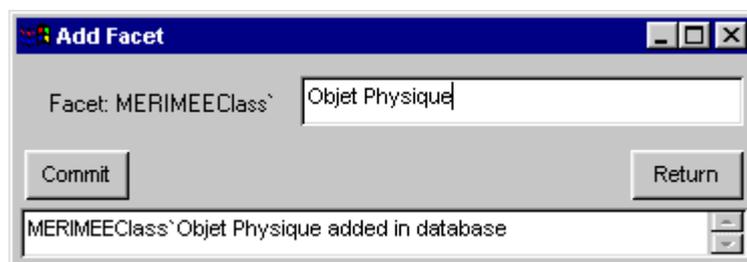


Figure 20 Operation window "Add Facet"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.2.2 Rename a new facet

- Task* : Edit Facet
Operation : Rename Facet
Needed Input : New Facet Name
Optional Input :

This operation renames a new facet.

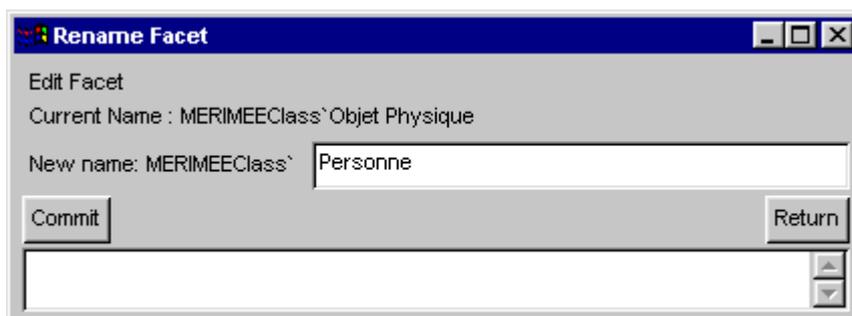


Figure 21 Operation window "Rename Facet"

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.2.3 Delete a new facet

- Task* : Edit Facet
Operation : Delete Facet
Needed Input :
Optional Input :

This operation deletes a new facet.

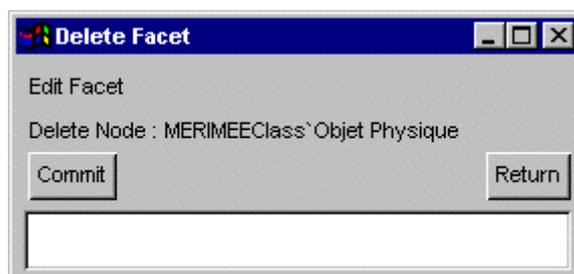


Figure 22 Operation window "Delete Facet".

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.2.4 Describe a new facet

Task : Edit Facet

Operation : Facet Attributes

Needed Input : letter_code

Optional Input :

This operation describes the relations of a new facet.

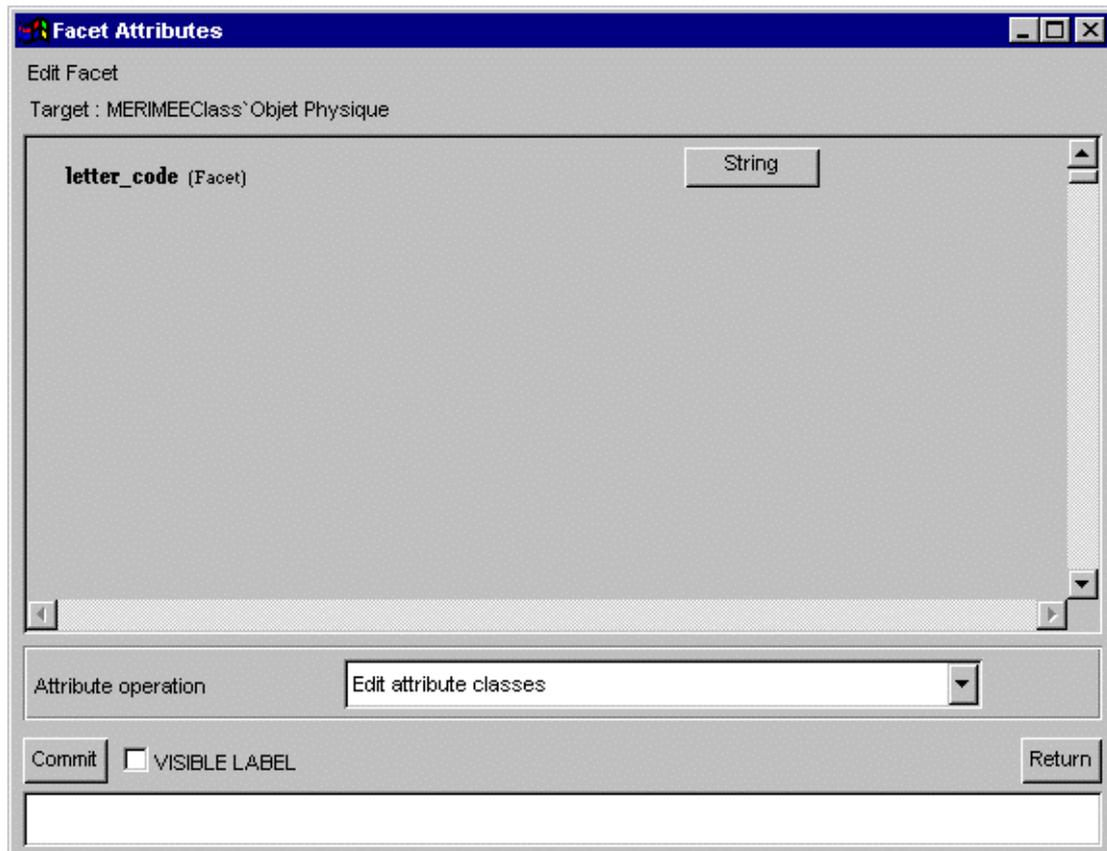


Figure 23 Operation window "Facet Attributes"

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.3 Manipulation of New Hierarchies

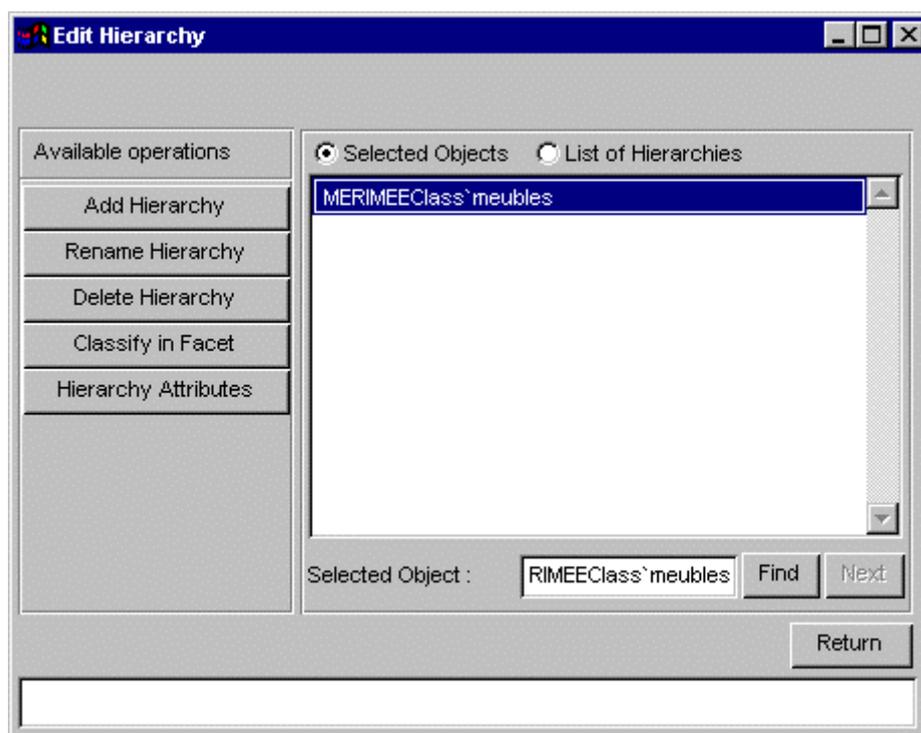


Figure 24 Task window "Edit Hierarchy".

3.3.1 Create a new hierarchy

Task : Edit Hierarchy

Operation : Add Hierarchy

Needed Input : Hierarchy Name, Facet Name

Optional Input :

The hierarchy is added in the knowledge base and classified in the specified facet. The top term of the hierarchy is created and appropriately associated with it. The operation fails in the following cases:

- A hierarchy with the same name already exists.
- The facet does not belong to the appropriate class

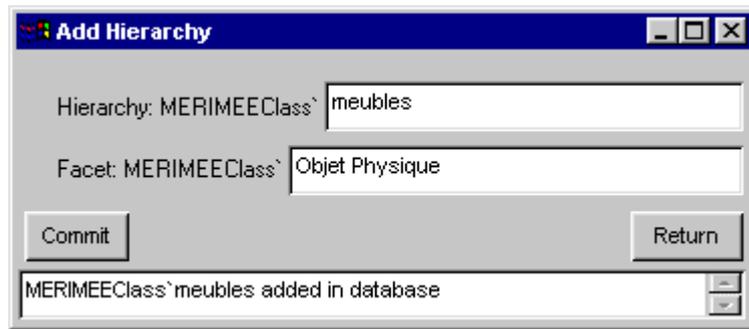


Figure 25 Operation window "Add Hierarchy"

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.3.2 Rename a new hierarchy

- Task* : Edit Hierarchy
Operation : Rename Hierarchy
Needed Input : New Hierarchy Name
Optional Input :

The hierarchy and its top term are appropriately renamed given the new hierarchy name. The operation fails in the following cases:

- A hierarchy with the same name already exists
- The top term with the same new name already exists.

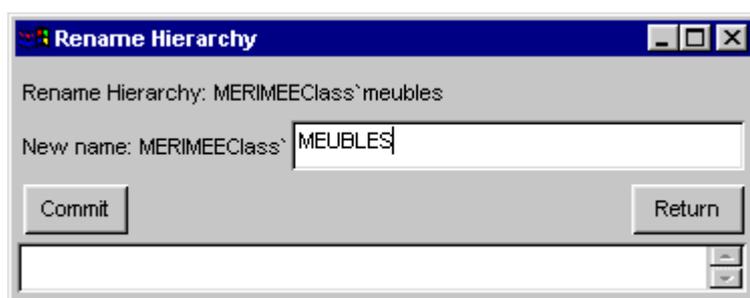


Figure 26 Operation window "Rename Hierarchy"

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.3.3 Delete a new hierarchy

Task : Edit Hierarchy

Operation : Delete Hierarchy

Needed Input :

Optional Input :

The hierarchy and its top term are deleted from the knowledge base. The operation fails in the following cases:

- One or more descriptors are classified in the specific hierarchy.
- There are links targeting to the top term of the hierarchy originating from other descriptors.

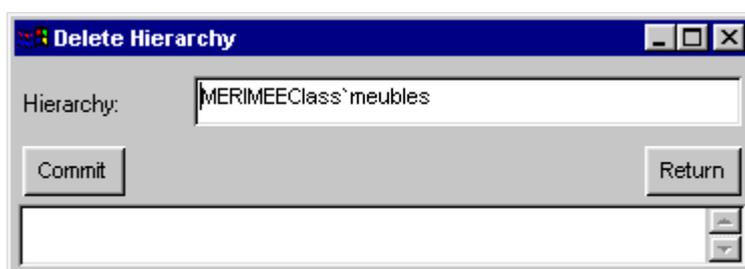


Figure 27 Operation window "Delete Hierarchy"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.3.4 Describe a new hierarchy

Task : Edit Hierarchy

Operation : Hierarchy Attributes

Needed Input : letter_code

Optional Input :

This operation describes the relations of a new hierarchy.

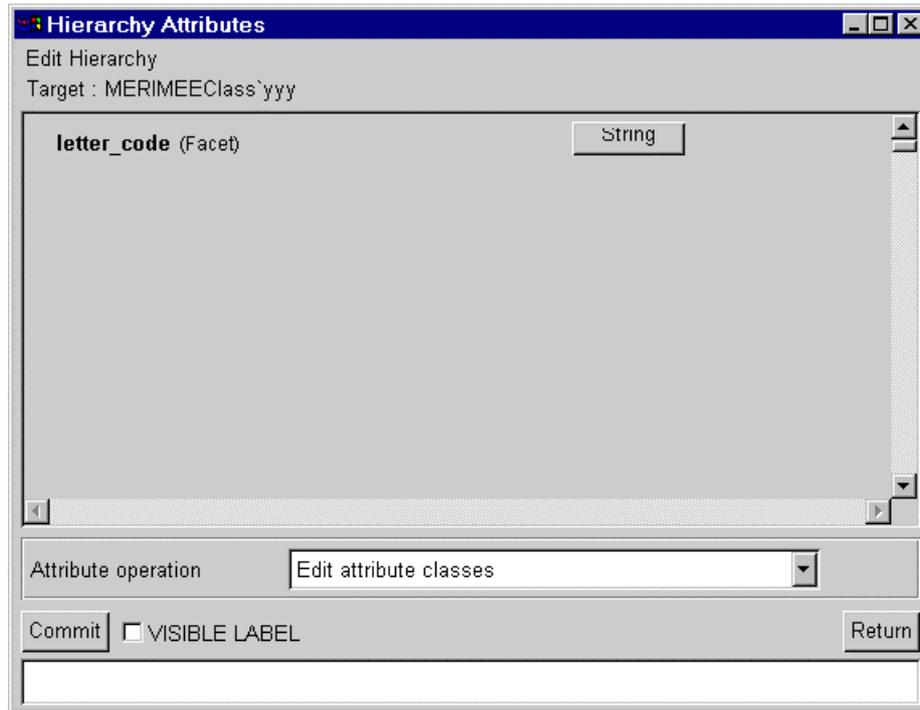


Figure 28 Operation window “Hierarchy Attributes”

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.3.5 Classify a new hierarchy in a facet

Task : Edit Hierarchy

Operation : Classify in Facet

Needed Input : Facet

Optional Input :

This operation associates a new hierarchy with a facet.

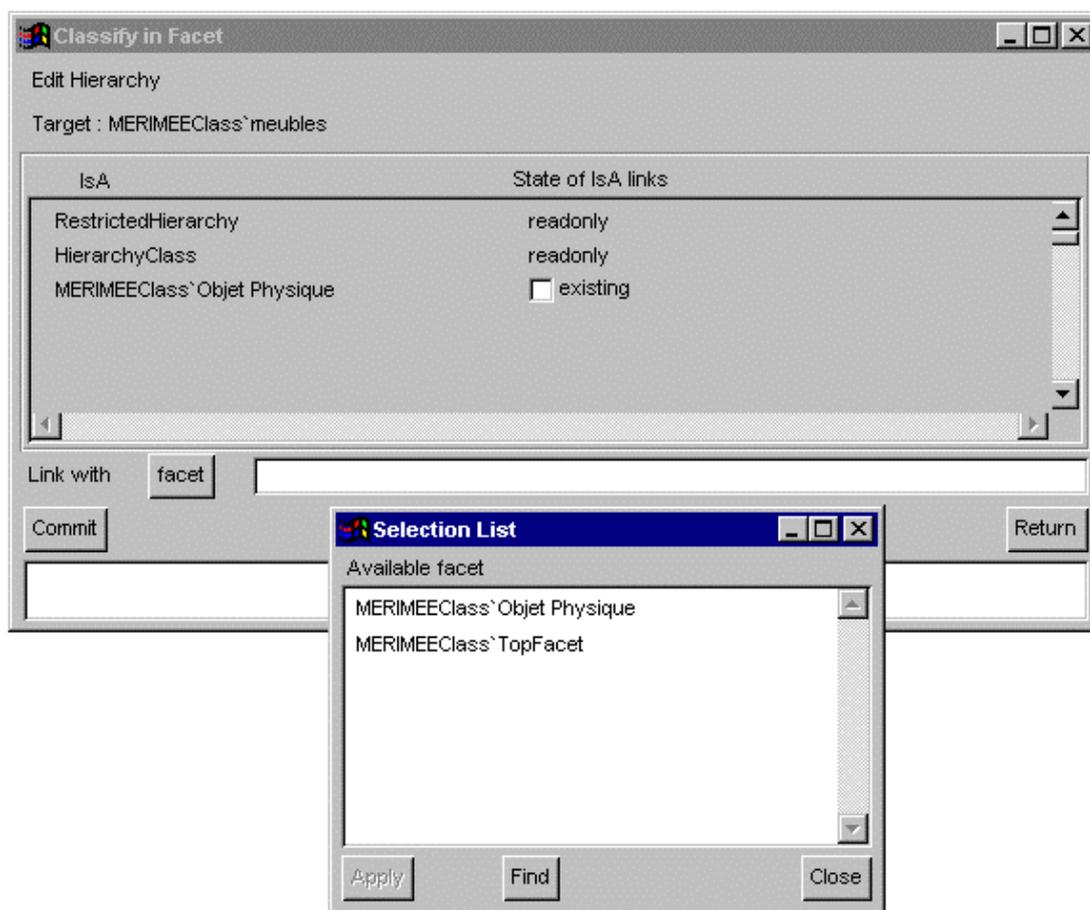


Figure 29 Operation window "Classify in Facet"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.4 Manipulation of New Concepts

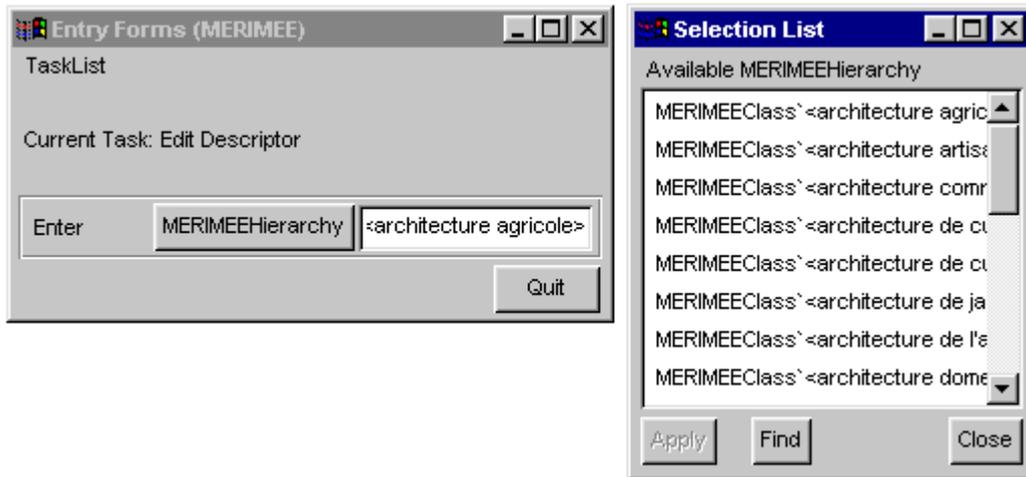


Figure 30 Selection of *task* "Edit Descriptor" requires the selection of a hierarchy.

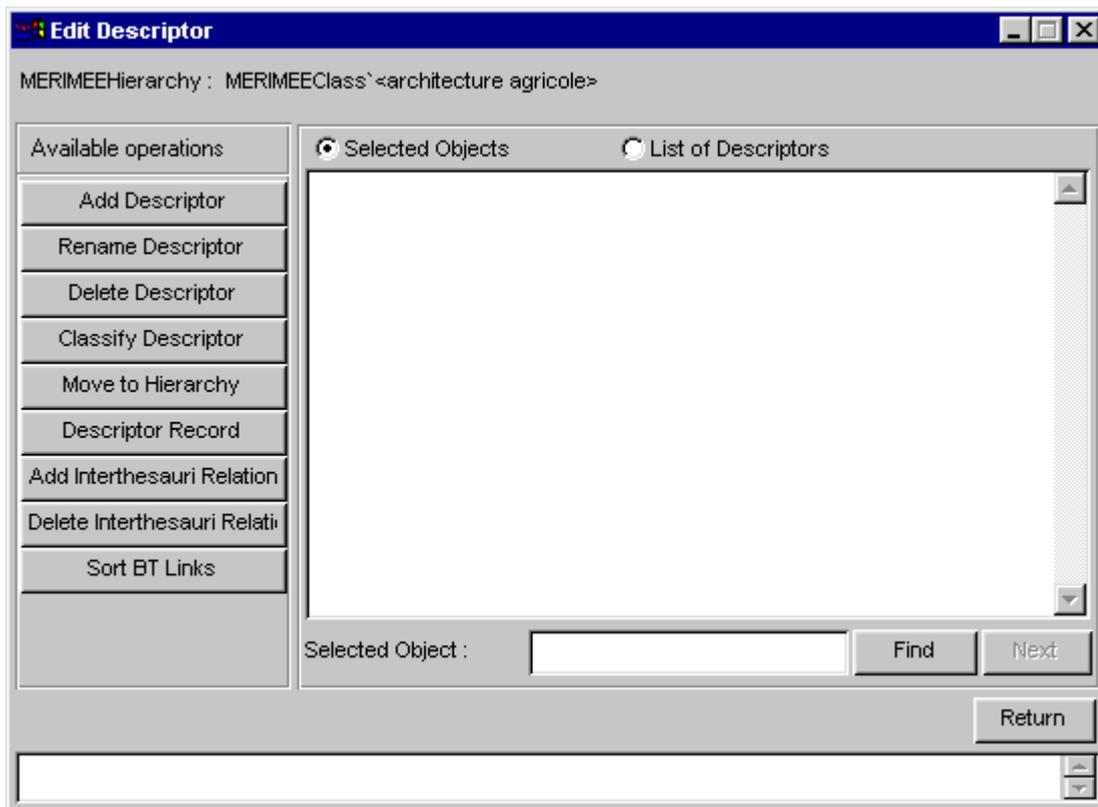


Figure 31 *Task window* "Edit Descriptor"

3.4.1 Create a new concept

Task : Edit Descriptor

Operation : Add Descriptor

Needed Input : Descriptor Name, Broader Term

Optional Input :

This operation creates a new descriptor. The descriptor is added in the knowledge base and is associated with the given broader term with a BT relation. It is also classified in its broader term hierarchies. The operation fails in the following cases:

- A descriptor with the given name already exists
- The given broader term does not exist.
- The given broader term is not a descriptor.

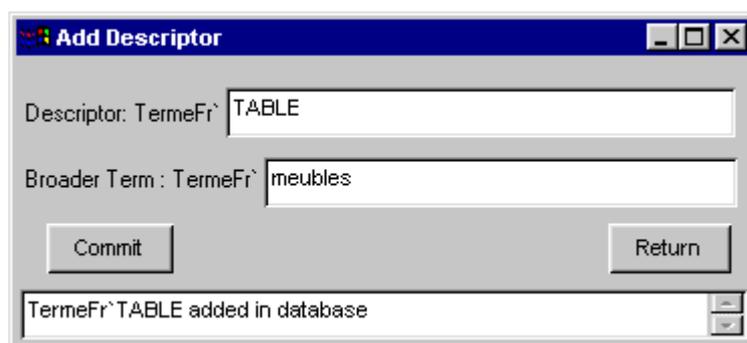


Figure 32 *Operation window "Add Descriptor"*

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.4.2 Rename a new concept

- Task* : Edit Descriptor
Operation : Rename Descriptor
Needed Input : New Descriptor Name
Optional Input :

This operation renames a new descriptor.

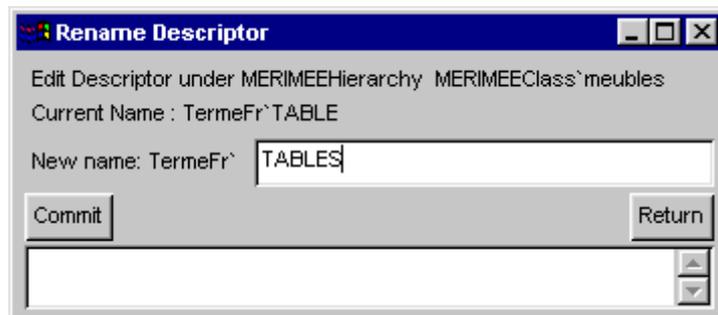


Figure 33 Operation window "Rename Descriptor"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.4.3 Delete a new concept

- Task* : Edit Descriptor
Operation : Delete Descriptor
Needed Input :
Optional Input :

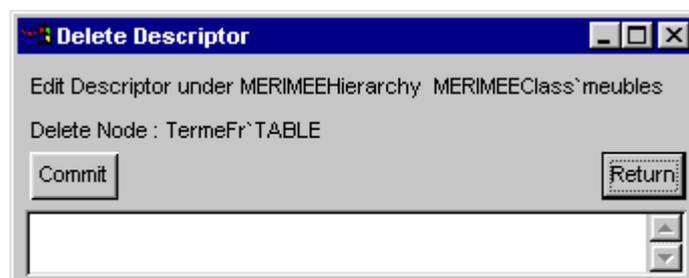


Figure 34 Operation window "Delete Descriptor"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.4.4 Describe a new concept

Task : Edit Descriptor

Operation : Descriptor Record

Needed Input :

Optional Input : created, modified, editor, found in, not found in, scope note, image, ALT, RT, UF

These operations describe the relations of a new descriptor.

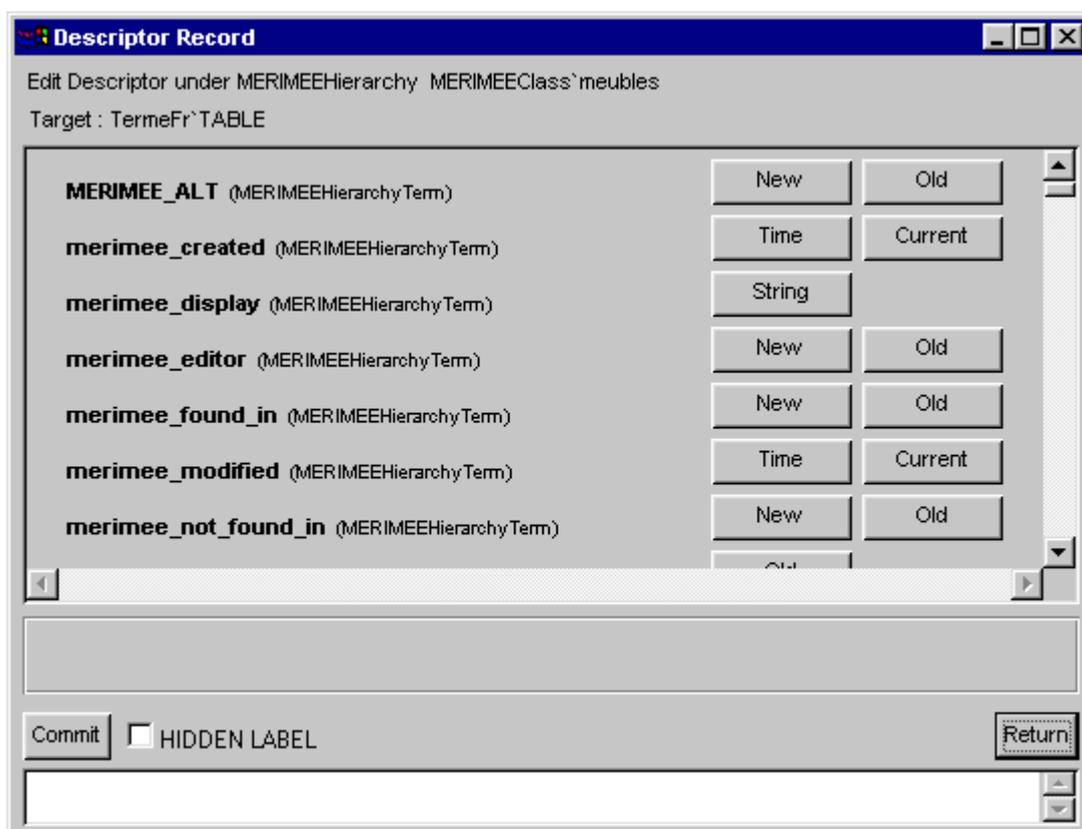


Figure 35 *Operation window "Descriptor Record"*

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.4.5 Classify a new concept

Task : Edit Descriptor

Operation : Classify Descriptor

Needed Input : Guide Term Class

Optional Input :

This operation describes a new descriptor as guide term.

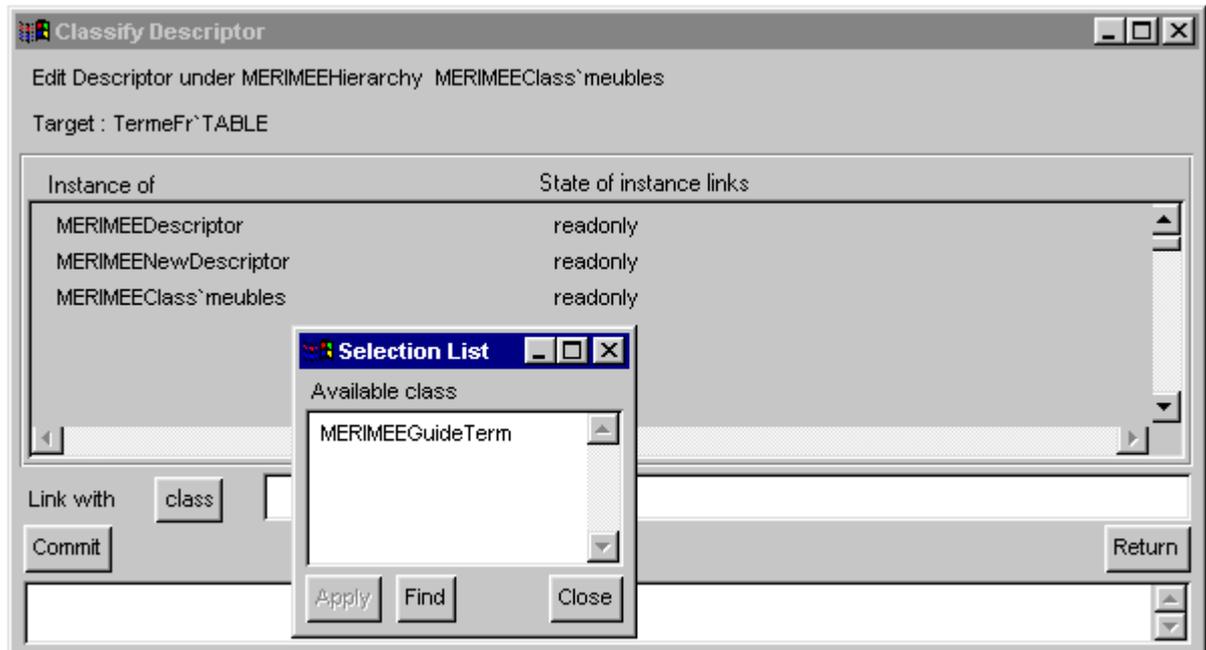


Figure 36 Operation window "Classify Descriptor"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.4.6 Move new concept to another hierarchy

Task : Edit Descriptor

Operation : Move To Hierarchy

Needed Input : New Hierarchy, New Broader Term

Optional Input :

- Move node only
In this case, the new concept is detached from the selected hierarchy (as in the case of "Abandon Descriptor" and is classified in the new hierarchy. A broader term relation is established between the new concept and the given broader term.
- Move node and subtree
In this case, the new concept and its subtree of broader term relations are detached from the selected hierarchy and are reclassified in the new hierarchy. A broader term relation is established between the new concept and the given broader term.
- Connect node and subtree
In this case, the new concept and its subtree of broader term relations are NOT detached from the selected hierarchy (as in previous case) and are multiply classified in the new hierarchy. A broader term relation is established between the new concept and the given broader term.

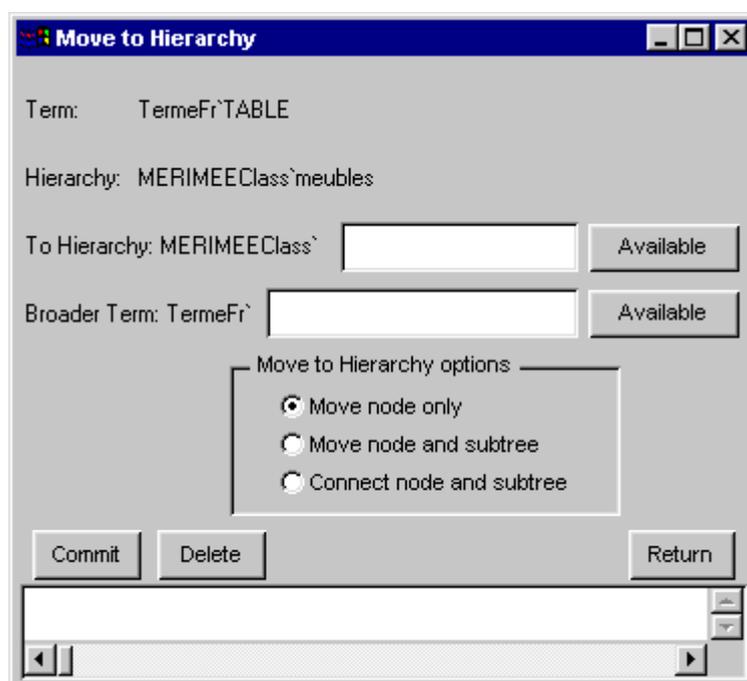


Figure 37 Operation window "Move to Hierarchy"

The operation fails in the following case:

- The broader term relation that is going to be added creates a directed cycle of broader term relations.

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

Broader term relations coming from the new concept can be deleted (except the last), by pressing the “*Delete*” button which displays a list with the existing relations. User can select a broader term relation and delete it by pressing the “*Delete*” button of this card.

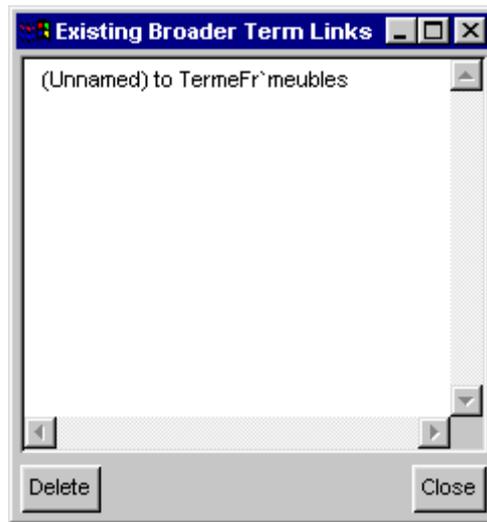


Figure 38 Operation window "Existing Broader Term Links"

3.4.7 Associate an new concept with terms from other thesauri

- Task* : Edit Descriptor
- Operation* : Add Interthesaurus Relations
- Needed Input* : From Term, Category, To Term
- Optional Input* :

This operation adds Inter thesaurus links to a new descriptor

An inter thesaurus link of type “category” is created to associate the “From Term” with the “To Term”.

The “To Term” can also be a collective concept, that is it can be a concept to express a union or an intersection of concepts of the target-thesaurus. The appropriate broader term links are constructed so as to associate the collective concept with its component concepts using the following syntax:

- concept_name1* + *concept_name2* (for union)
- concept_name1* & *concept_name2* (for intersection)

The operation fails in the following cases:

- ◇ In case the “To Term” is not a collective concept
 - The “To Term” does not exist. In this case, the operation fails because it is not legal to introduce new concepts in the target-thesaurus.
 - The “To Term” is not a descriptor.
- ◇ In case the “To Term” is a collective concept.
 - The components of the “To Term” do not exist in the target-thesaurus. In this case the operation fails because it is not legal to introduce new concepts in the target-thesaurus.
 - The components of the “To Term” are not descriptors.

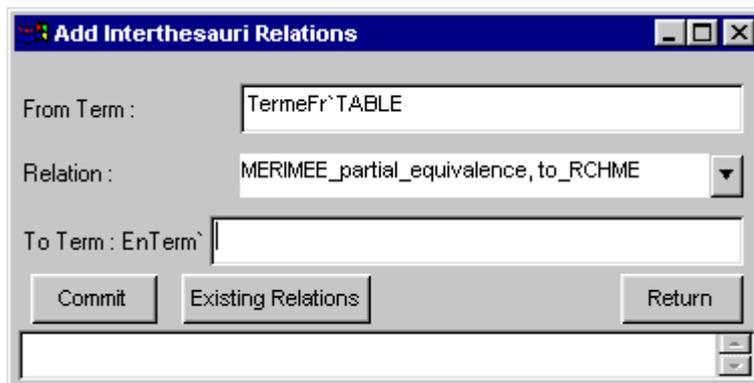


Figure 39 Operation window "Interthesaurus Relations"

To confirm the update select the “COMMIT” button. To return to the *task window* of the current task select the “RETURN” button.

3.4.8 Disassociate a new concept with terms from other thesauri

Task : Edit Descriptor

Operation : Delete Interthesaurus Relations

Needed Input : (From Term, Category, To Term)

Optional Input :

This operation deletes Inter thesaurus links to a new descriptor. The inter thesaurus link of the given category associating the “From Term” with the “To Term” is deleted”. In case the “To Term” is a collective concept, then it is deleted only when it is not associated with other descriptors. The operation fails in the following case:

- A link from “From Term” to “To Term” of type “category” does not exist.
- The “To Term” does not exist.

Figure 40 Operation window "Delete Interthesaurus Relations"

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.4.9 Sort the broader term links of a new concept

Task : Edit Descriptor

Operation : Sort BT links

Needed Input : Final order of links with drag & drop in the list of the card

Optional Input :

This operation sorts the broader term links pointing to an existing concept. This sorting determines the order of appearance of these links in SIS Graphical Analysis Interface.

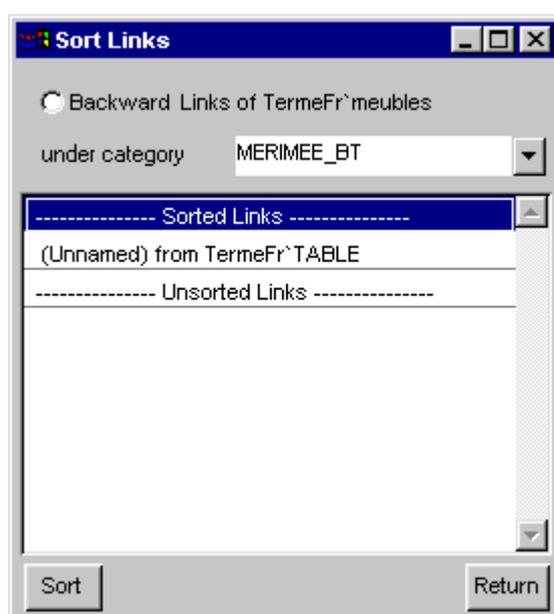


Figure 41 Operation window "Sort Links"

To confirm the sorting select the "*SORT*" button. To return to the *task window* of the current task select the "*Close*" button.

3.5 Manipulation of Existing Facets

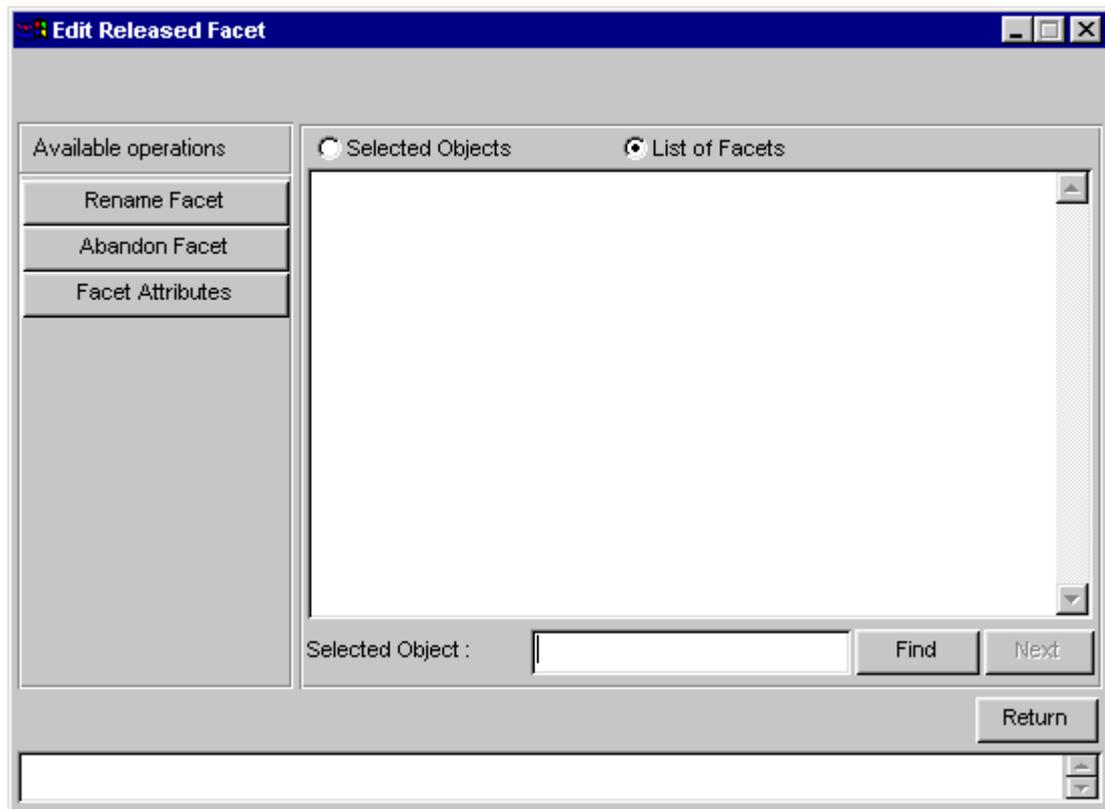


Figure 42 Task window "Edit Released Facet"

3.5.1 Rename an existing facet

Task : Edit Released Facet

Operation : Rename Facet

Needed Input : New Facet Name

Optional Input :

This operation renames an existing facet.

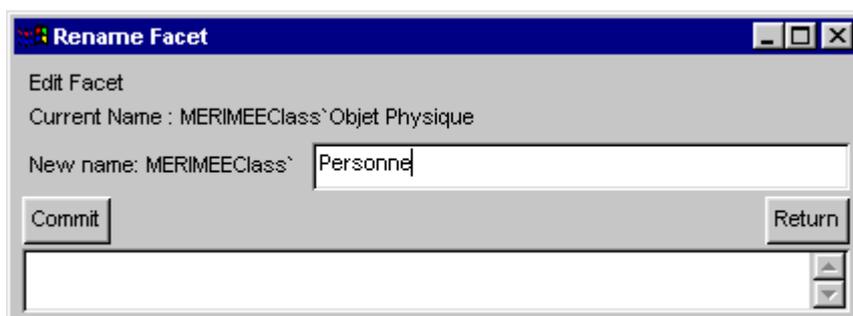


Figure 43 Operation window "Rename Facet"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.5.2 Abandon an existing facet

Task : Edit Released Facet

Operation : Abandon Facet

Needed Input : New Facet Name

Optional Input :

This operation abandons an existing facet. The facet is classified as "obsolete."

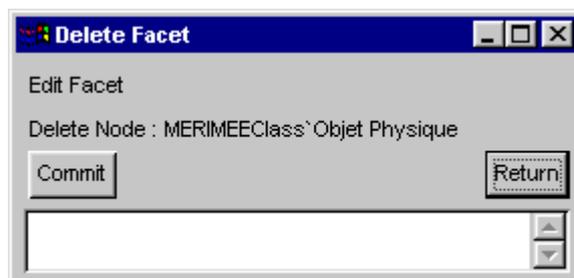


Figure 44 Operation window "Delete Facet"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.5.3 Describe an existing facet

Task : Edit Released Facet

Operation : Facet Attributes

Needed Input : letter_code

Optional Input :

This operation describes the relations of an existing facet

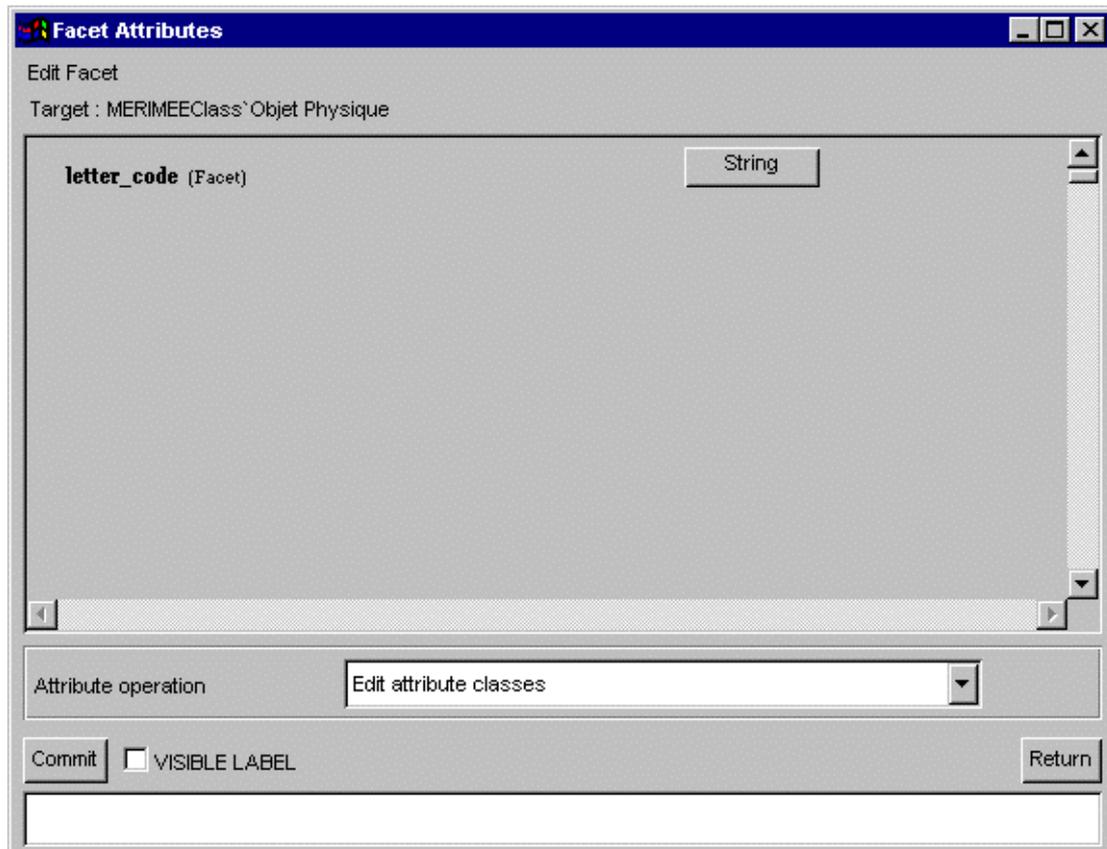


Figure 45 Operation window "Facet Attributes"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.6 Manipulation of Existing Hierarchies

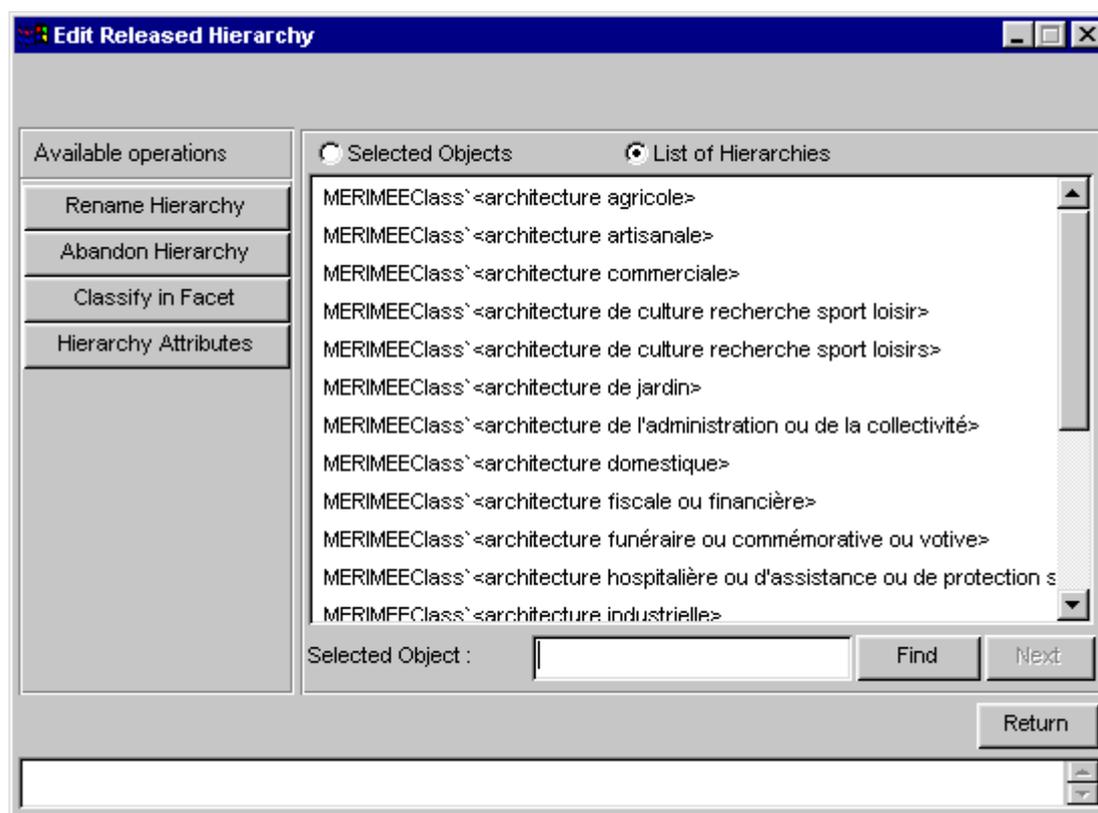


Figure 46 Task window "Edit Released Hierarchy"

3.6.1 Rename an existing hierarchy

Task : Edit Released Hierarchy

Operation : Rename Hierarchy

Needed Input : New Hierarchy Name

Optional Input :

This operation renames an existing hierarchy.

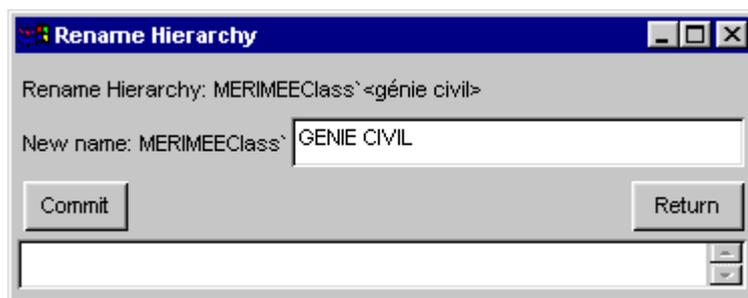


Figure 47 Operation window "Rename Hierarchy"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.6.2 Abandon an existing hierarchy

Task : Edit Released Hierarchy

Operation : Abandon Hierarchy

Needed Input : New Hierarchy Name

Optional Input :

This operation abandons an existing hierarchy. The hierarchy is classified as "obsolete".

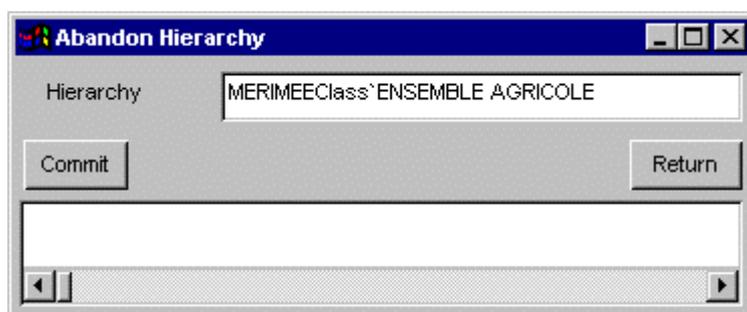


Figure 48 Operation window "Abandon Hierarchy"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.6.3 Describe an existing hierarchy

Task : Edit Released Hierarchy

Operation : Hierarchy Attributes

Needed Input : letter_code

Optional Input :

This operation describes the relations of an existing hierarchy.

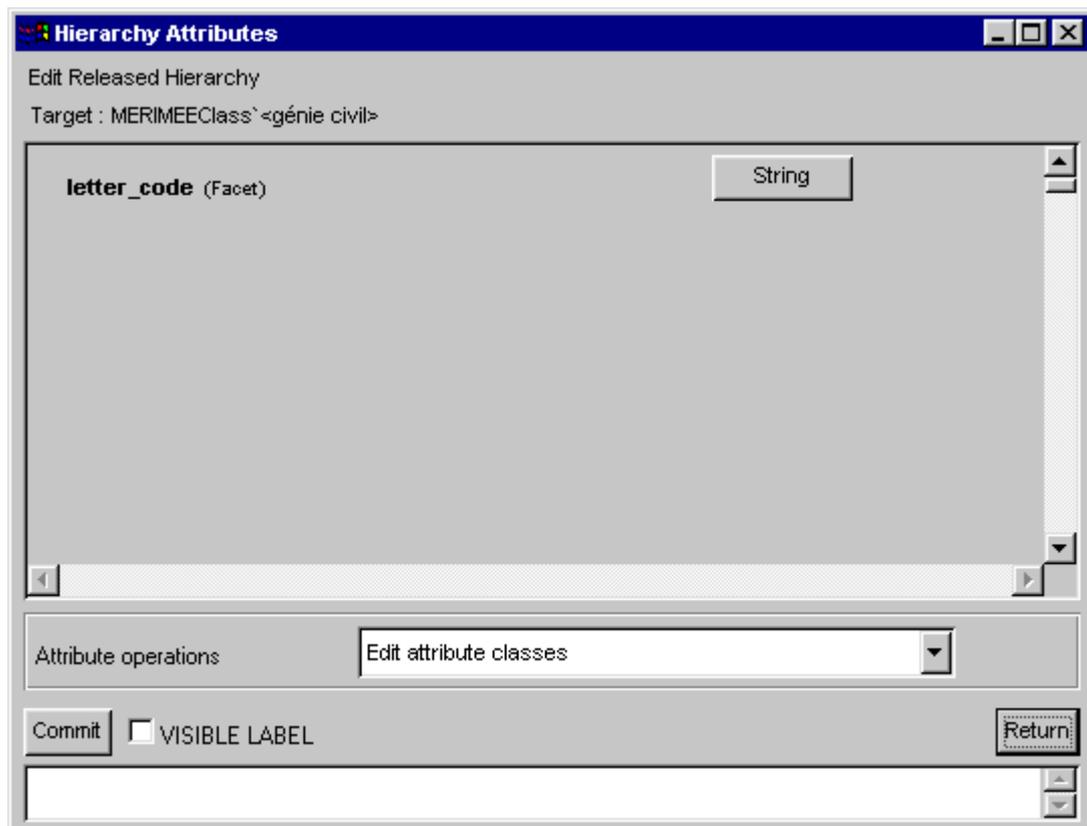


Figure 49 Operation window "Hierarchy Attributes"

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.6.4 Classify an existing hierarchy in a facet

Task : Edit Released Hierarchy

Operation : Classify in Facet

Needed Input : Facet

Optional Input :

This operation associates an existing hierarchy with an existing facet.

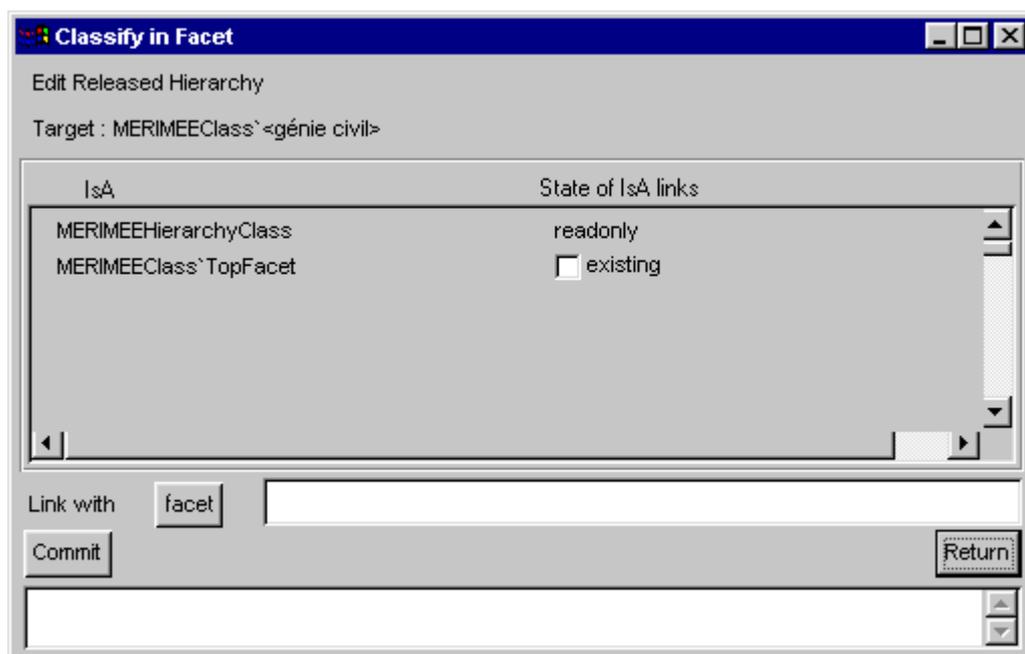


Figure 50 Operation window "Classify in Facet"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.7 Manipulation of Existing Concepts

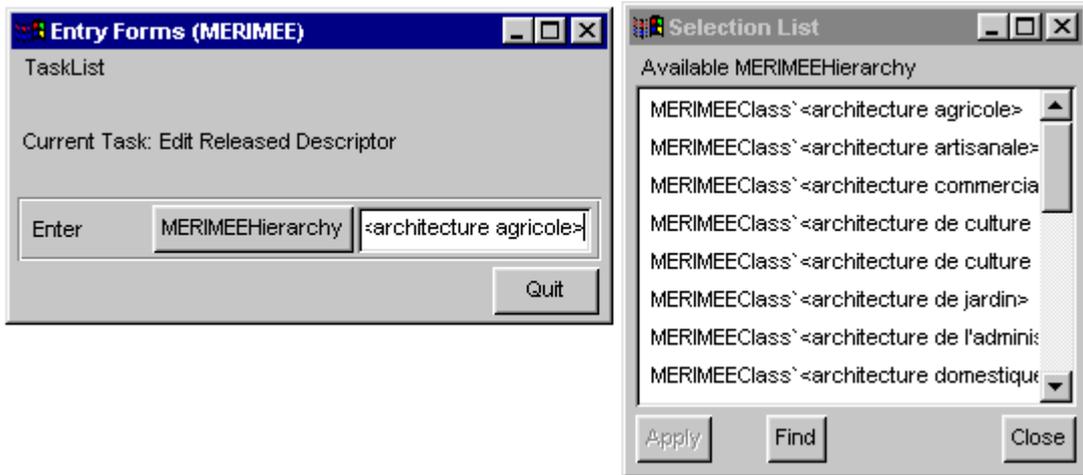


Figure 51 Selection of task "Edit Released Descriptor" requires the selection of a hierarchy.

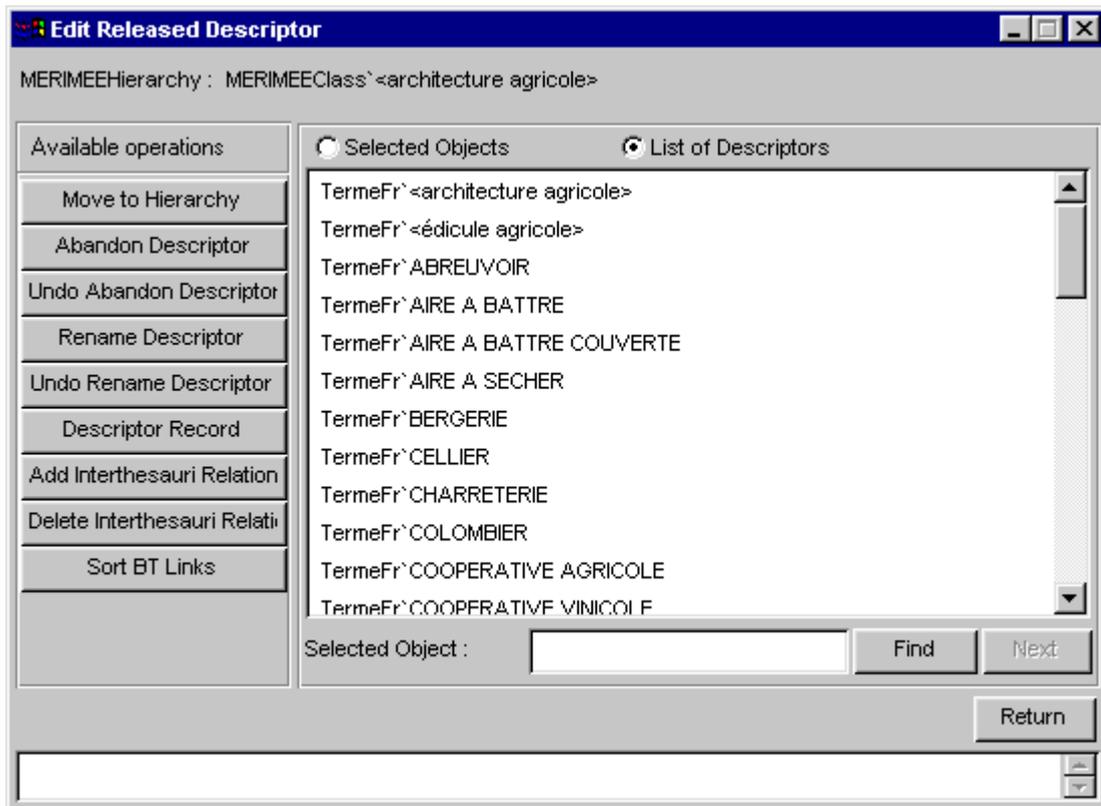


Figure 52 Task window "Edit Released Descriptor"

3.7.1 Describe an existing concept

Task : Edit Released Descriptor

Operation : Descriptor Record

Needed Input :

Optional Input : created, modified, editor, found in, not found in, scope note, image, ALT, RT, UF

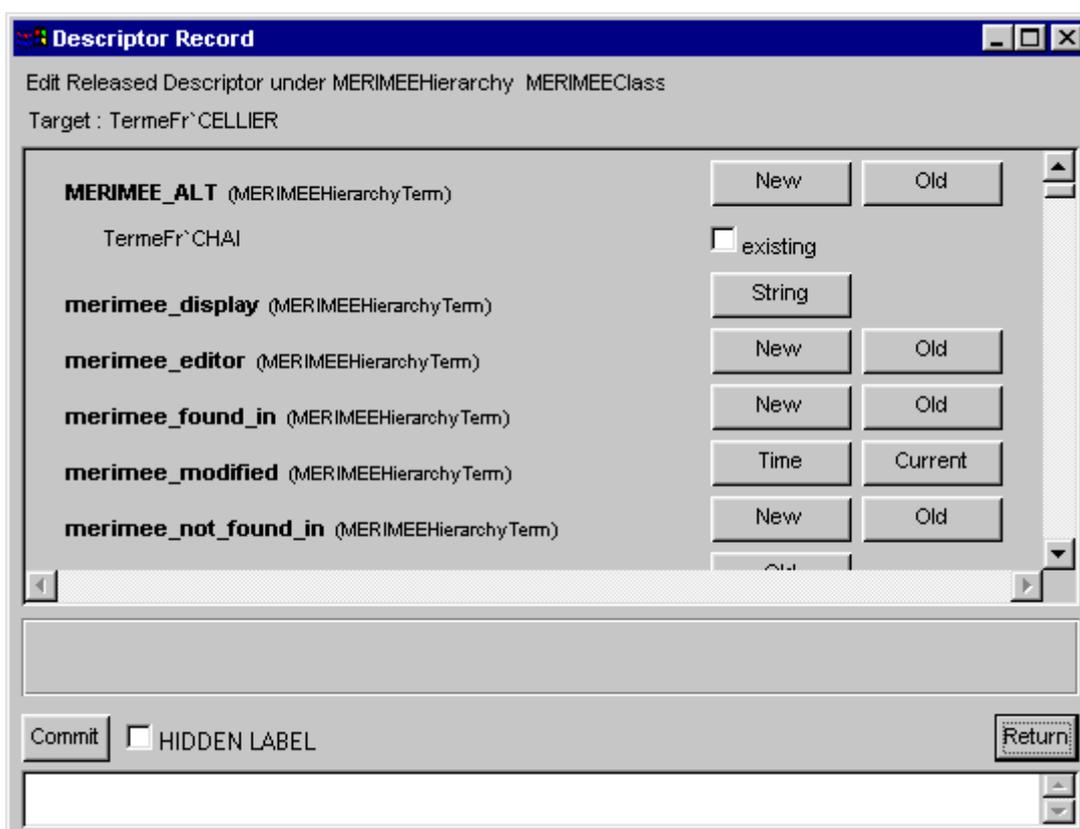


Figure 53 Operation window "Descriptor Record"

These operations describe the relations of an existing descriptor.

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.7.2 Abandon an existing concept

Task : Edit Released Descriptor

Operation : Abandon Descriptor

Needed Input :

Optional Input :

This operation abandons an existing descriptor. In this case the descriptor is classified as an "Obsolete Descriptor". It remains classified in the hierarchy but is detached from it. That is all its broader and narrower term relations are deleted and appropriate broader term relations are established between its narrower and broader terms as shown in Figure 55. The operation fails in the following cases:

- Related, Used For and Alternative term links are originating from or targeted to the descriptor.
- Inter thesauri links are originating from the descriptor.
- Descriptor is an obsolete one.

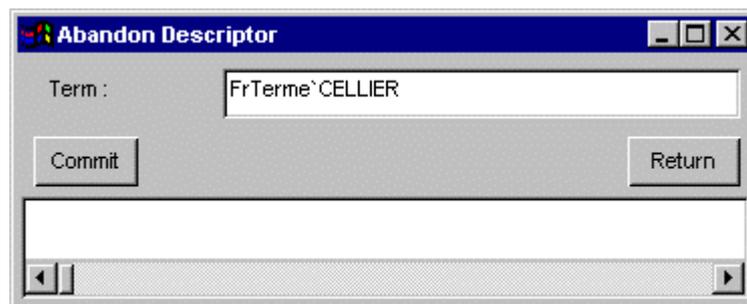
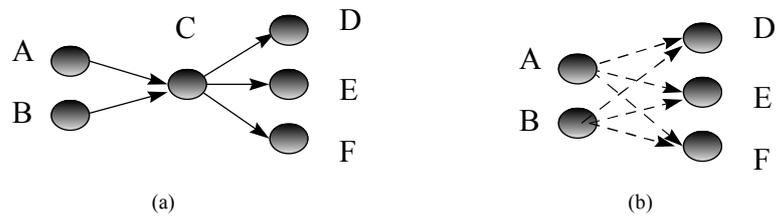


Figure 54 Operation window "Abandon Descriptor"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.



In (a) descriptor C belongs in hierarchy H and has the broader terms D, E and F and the narrower terms A and B.

In (b) descriptor C is characterized as “Obsolete Descriptor” and is detached from hierarchy H. Its broader and narrower term relations are deleted and appropriate broader term relations (dashed arrows) are established between its narrower and broader terms.

Figure 55 Schema of operation "Abandon Descriptor"

3.7.3 Undo Abandon an existing concept

Task : Edit Released Descriptor

Operation : Undo Abandon Descriptor

Needed Input : Broader Term

Optional Input :

This operation cancels the “abandon descriptor” operation. In this case the descriptor is no longer classified as an "Obsolete Descriptor", and a broader term relation is established with the given broader term. The descriptor and the given broader term must belong to the same hierarchy. The operation fails in the following cases:

- The descriptor is not an obsolete descriptor.
- The given broader term is not a descriptor.
- The given broader term is an obsolete descriptor.
- The descriptor and the broader term do not belong to the same hierarchy.

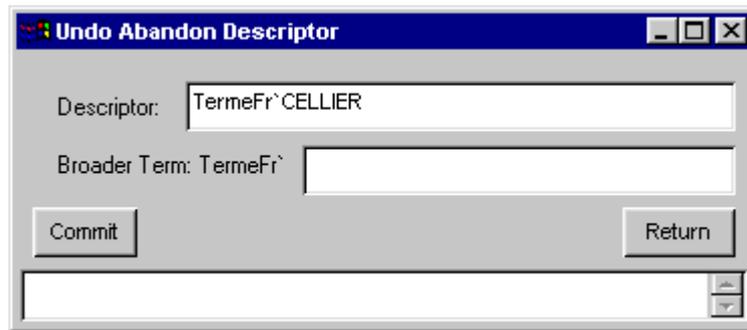


Figure 56 Operation window "Undo Abandon Descriptor"

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.7.4 Move descriptor to another hierarchy

Task : Edit Released Descriptor

Operation : Move To Hierarchy

Needed Input : New Hierarchy, New Broader Term

Optional Input :

- Move node only
In this case, the descriptor is detached from the selected hierarchy (as in the case of "Abandon Descriptor" and is classified in the new hierarchy. A broader term relation is established between the descriptor and the given broader term.
- Move node and subtree
In this case, the descriptor and its subtree of broader term relations are detached from the selected hierarchy and are reclassified in the new hierarchy. A broader term relation is established between the descriptor and the given broader term.
- Connect node and subtree
In this case, the descriptor and its subtree of broader term relations are NOT detached from the selected hierarchy (as in previous case) and are multiply classified in the new hierarchy. A broader term relation is established between the descriptor and the given broader term.

The operation fails in the following cases:

- The broader term relation that is going to be added creates a directed cycle of broader term relations.

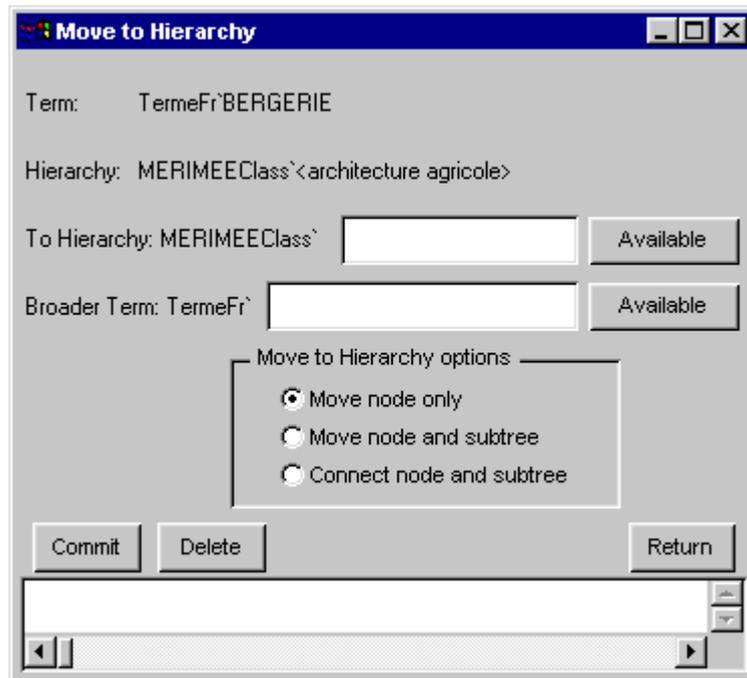


Figure 57 Operation window "Move to Hierarchy"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

Broader term relations coming from the new concept can be deleted (except the last), by pressing the "Delete" button which displays a list with the existing relations. User can select a broader term relation and delete it by pressing the "Delete" button of this card.

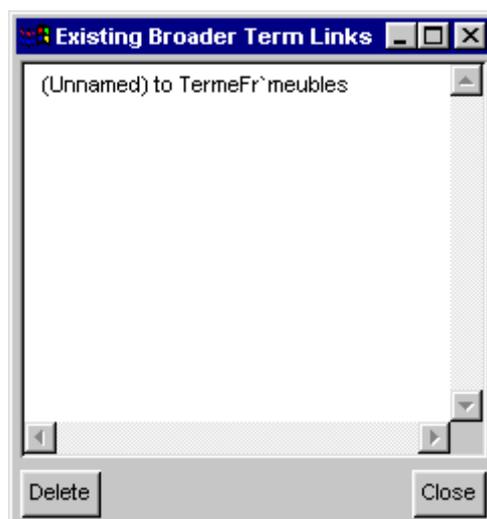


Figure 58 Operation window "Existing Broader Term Links"

3.7.5 Rename an existing concept

Task : Edit Released Descriptor

Operation : Rename Descriptor

Needed Input :

Optional Input :

This operation renames an existing descriptor. Cyclic and linear renames can be performed with the specified operation. The following cases of renaming can be performed:

- Term_A renamed to Term_B, Term_B is a not yet existing name
- Term_A renamed to Term_B. In case Term_B is an existing name not belonging in the ObsoleteTerm class of the current thesaurus, the user is asked to give a new name for Term_B. The user can give Term_A (performing a cyclic rename) or a not yet existing name. The number of renames performed can be arbitrary.

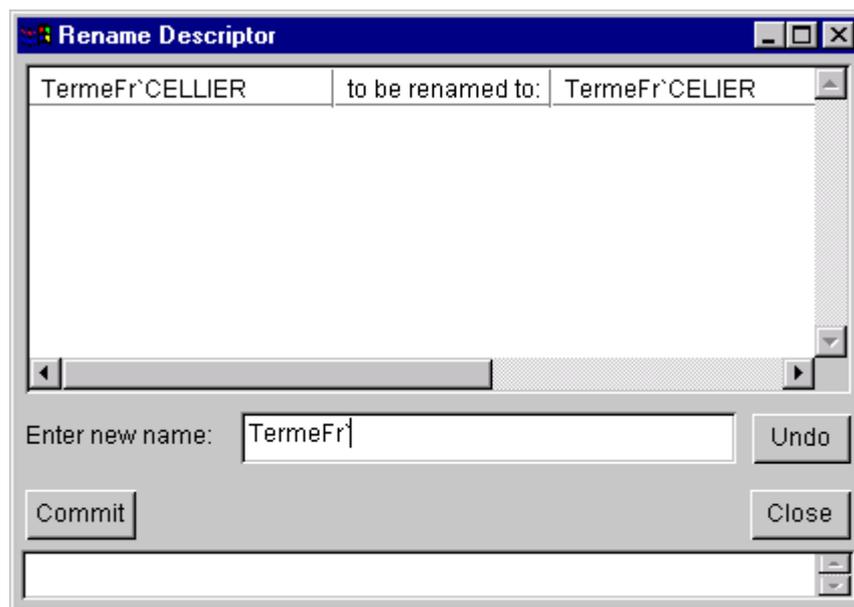


Figure 59 Operation window "Rename Descriptor"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "CLOSE" button.

3.7.6. Undo Rename an existing concept

Task : Edit Released Descriptor

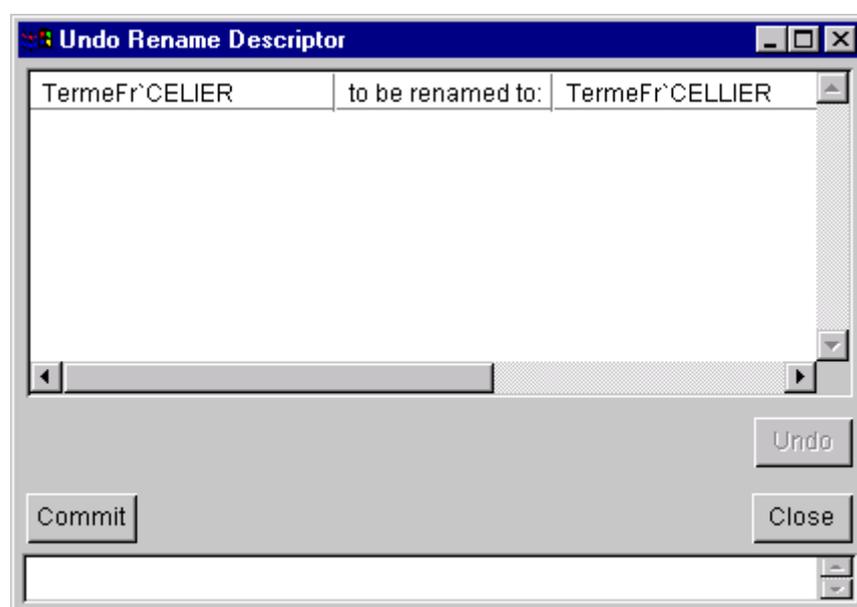
Operation : Undo Rename Descriptor

Needed Input : Descriptor's Name

Optional Input :

This operation cancels the “rename descriptor” operation. The necessary renames are performed so that the knowledge base returns to its previous state before the renames took place.). The operation fails in the following case:

- Used for links are targeting to one of the terms participating in the sequence of terms to be renamed. In this case the user is informed and the



used for links should be deleted

Figure 60 Operation window "Undo Rename Descriptor"

To confirm the update select the “COMMIT” button. To return to the *task window* of the current task select the “RETURN” button.

3.7.7 Associate an existing concept with terms from other thesauri

Task : Edit Released Descriptor

Operation : Add Interthesaurus Relations

Needed Input : From Term, Category, To Term

Optional Input :

This operation adds Inter thesaurus links to a descriptor. An inter thesaurus link of type “category” is created to associate the “From Term” with the “To Term”. The “To Term” can also be a collective concept, that is it can be a concept to express a union or an intersection of concepts of the target-thesaurus. The appropriate broader term links are constructed so as to associate the collective concept with its component concepts.

The operation fails in the following cases:

- ◇ In case the “To Term” is not a collective concept
 - The “To Term” does not exist. In this case, the operation fails because it is not legal to introduce new concepts in the target-thesaurus.
 - The “To Term” is not a descriptor.
- ◇ In case the “To Term” is a collective concept.
 - The components of the “To Term” do not exist in the target-thesaurus. In this case the operation fails because it is not legal to introduce new concepts in the target-thesaurus.
 - The components of the “To Term” are not descriptors.

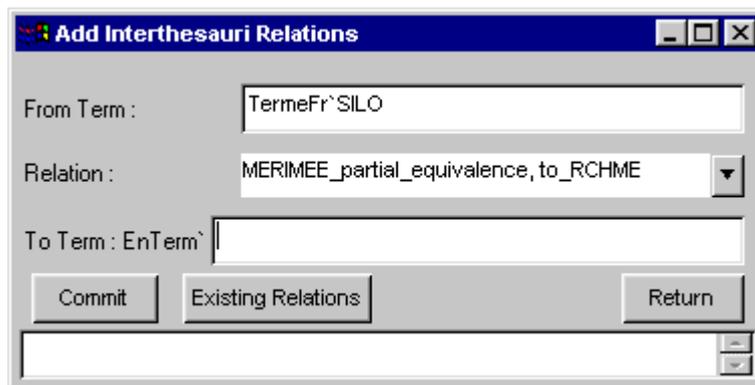


Figure 61 Operation window "Add Interthesaurus Relation"

To confirm the update select the “COMMIT” button. To return to the *task window* of the current task select the “RETURN” button.

3.7.8 Disassociate an existing concept with terms from other thesauri

Task : Edit Released Descriptor

Operation : Delete Interthesaurus Relations

Needed Input : From Term, Category, To Term

Optional Input :

This operation deletes Inter thesaurus links from an existing descriptor. The inter thesaurus link of the given category associating the “From Term” with the “To Term” is deleted”. In case the “To Term” is a collective concept, then it is deleted only when it is not associated with other descriptors. The operation fails in the following case:

- A link from “From Term” to “To Term” of type “category” does not exist.
- The “To Term” does not exist.

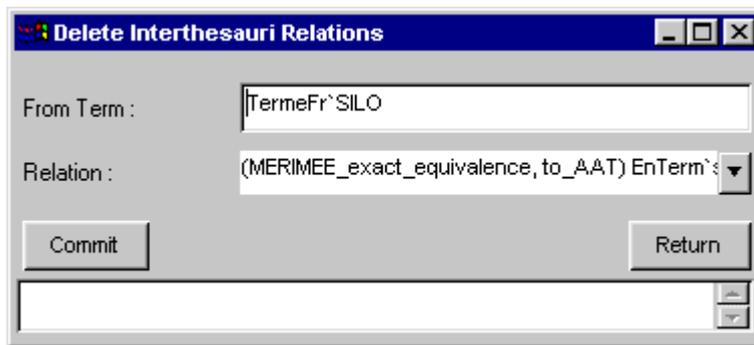


Figure 62 Operation window "Delete Interthesaurus Relations"

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

]

3.7.8 Sort the broader term links of an existing concept

Task : Edit Released Descriptor

Operation : Sort BT links

Needed Input : Final order of links with drag & drop in the list of the card

Optional Input :

This operation sorts the broader term links pointing to an existing concept. This sorting determines the order of appearance of these links in SIS Graphical Analysis Interface.

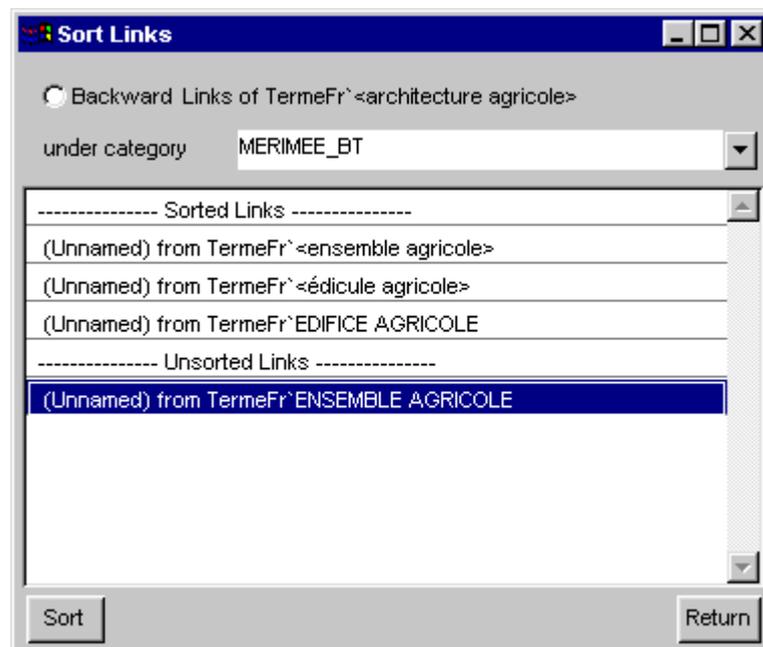


Figure 63 Operation window "Sort Links"

To confirm the sorting select the “*SORT*” button. To return to the *task window* of the current task select the “*Return*” button.

3.8 Manipulation of Sources

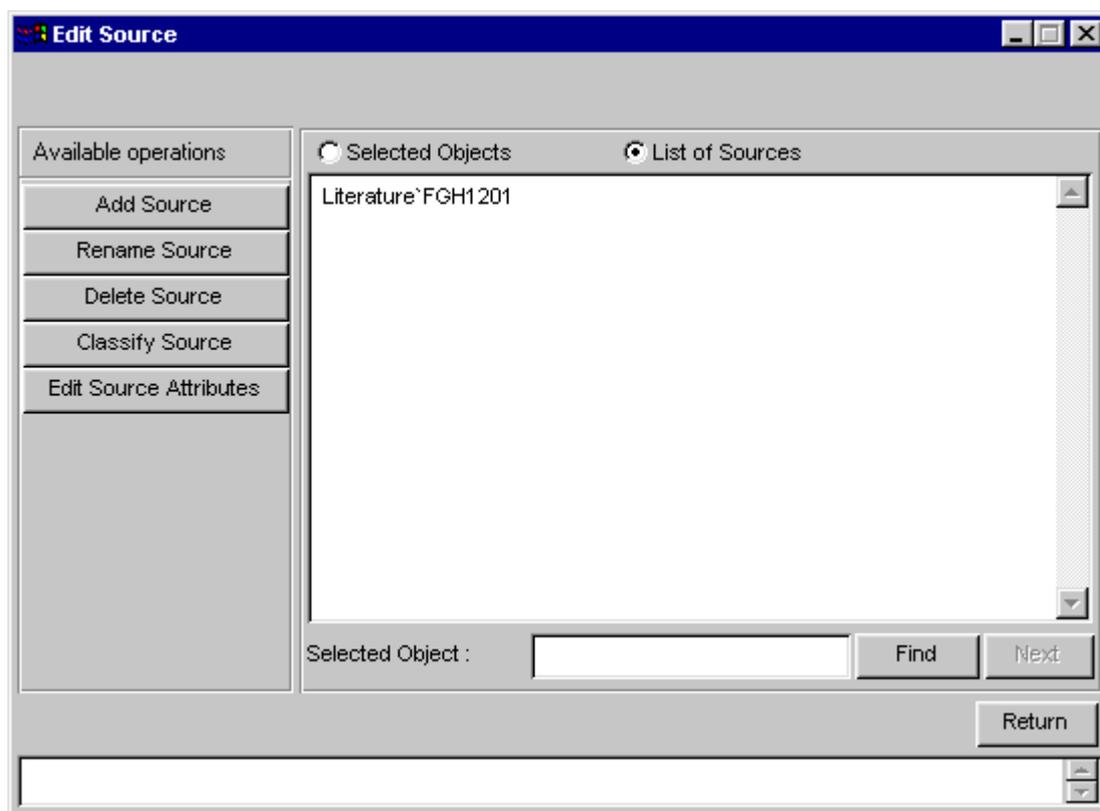


Figure 64 Task window "Edit Source"

3.8.1 Add a new source

- Task* : Edit Source
Operation : Add Source
Needed Input : Source Name
Optional Input :

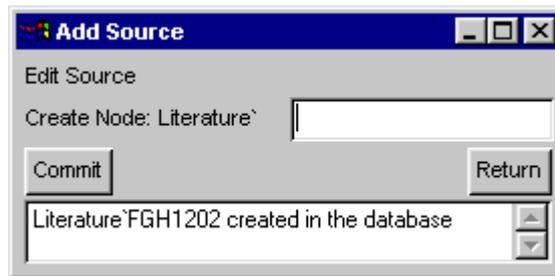


Figure 65 Operation window "Add Source"

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.8.2 Rename a source

- Task* : Rename Source
Operation : Rename Source
Needed Input :
Optional Input :

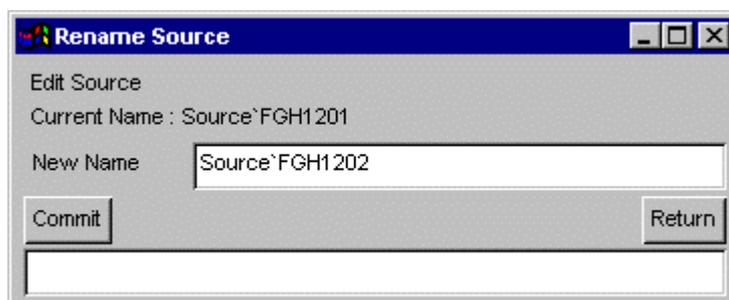


Figure 66 Operation window "Rename Source"

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.8.3 Delete a source

Task : Edit Source

Operation : Delete Source

Needed Input : The new Descriptor/Term Name

Optional Input :

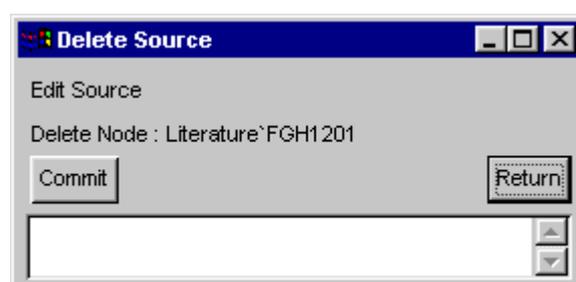


Figure 67 Operation window "Delete Source"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.8.4 Describe a source

Task : Edit Source

Operation : Edit Source Attributes

Needed Input : full_reference

Optional Input :

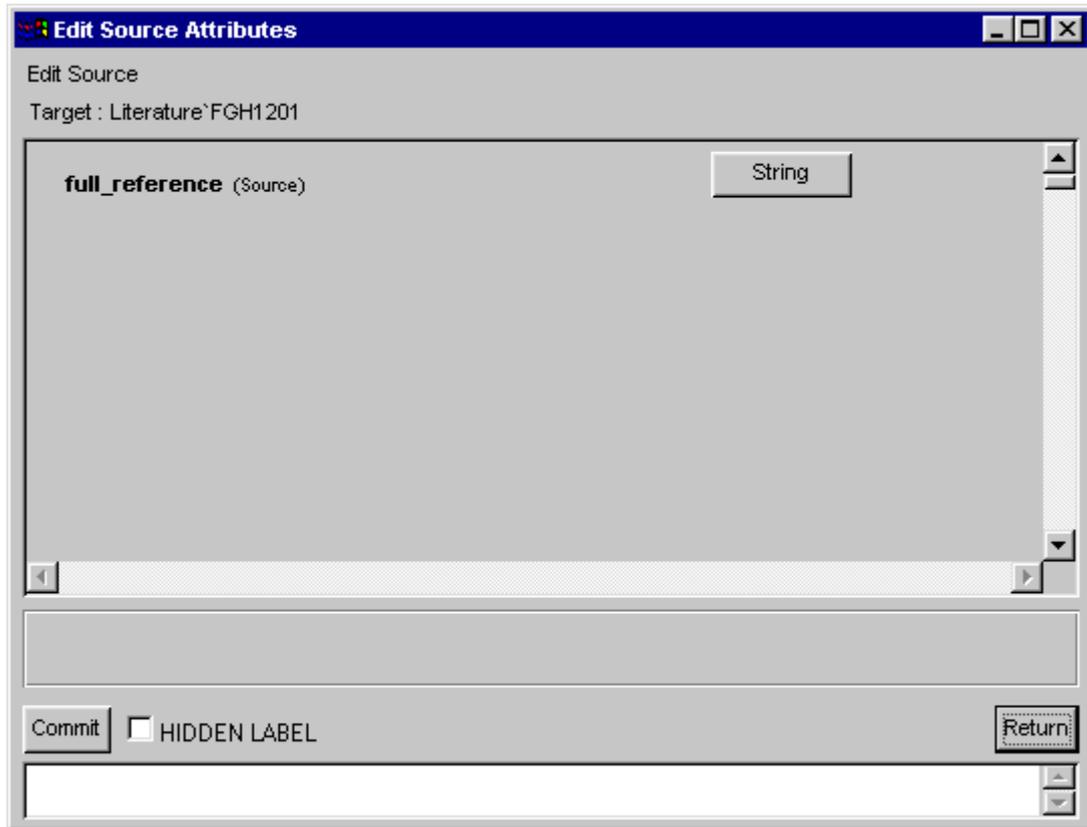


Figure 68 Operation window "Edit Source Attributes"

3.8.5 Classify a source

Task : Edit Source

Operation : Classify Source

Needed Input : Source Class

Optional Input :

This operation classifies a source under a source class.

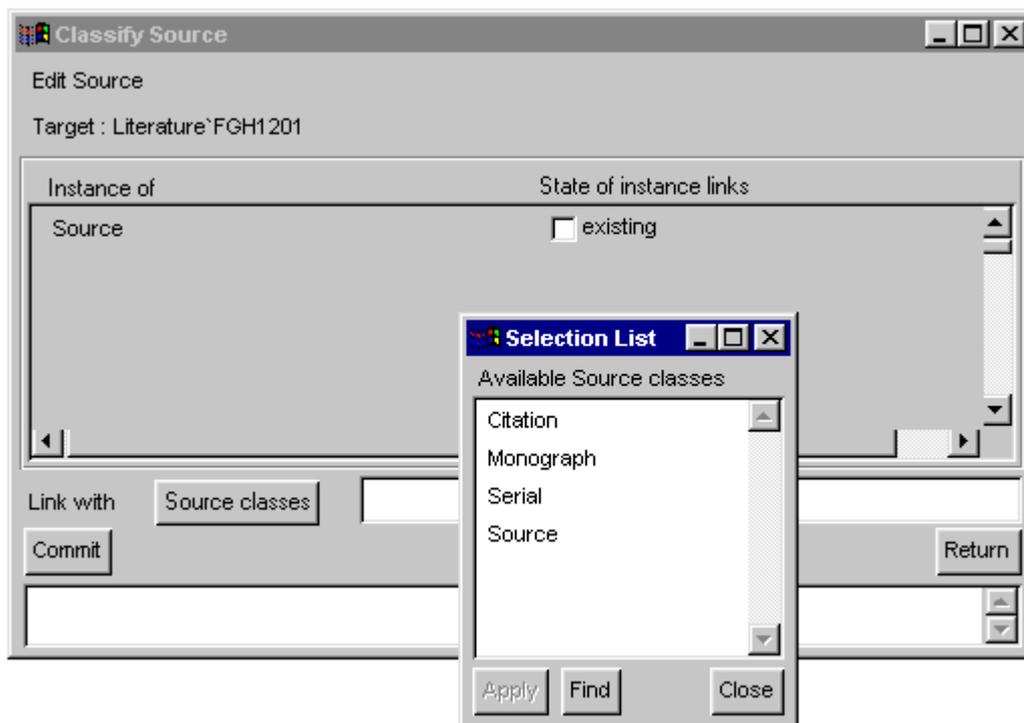


Figure 69 Operation window "Classify Source"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.9 Manipulation of Editors

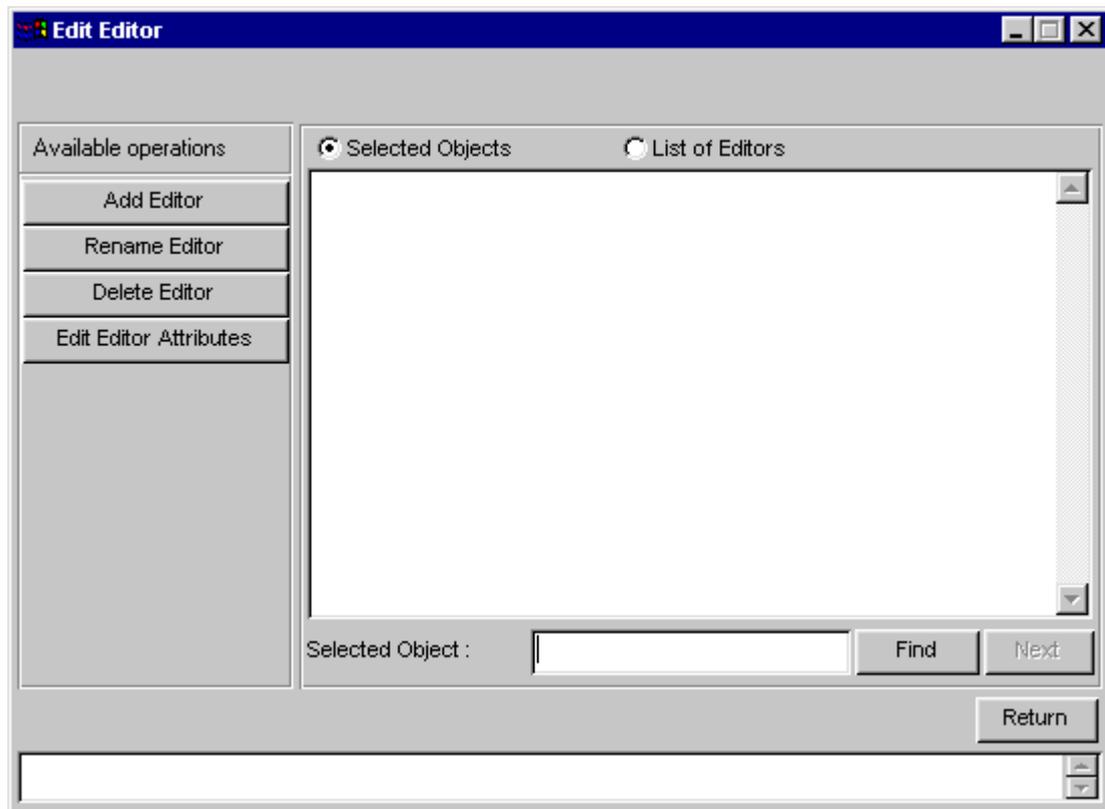


Figure 70 Task window "Edit Editor"

3.9.1 Add a new editor

- Task* : Edit Editor
Operation : Add Editor
Needed Input : Editor Name
Optional Input :

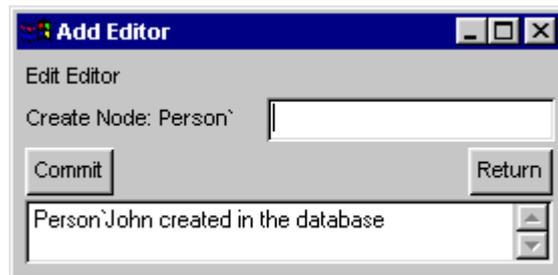


Figure 71 *Operation window "Add Editor"*

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.9.2 Delete an editor

- Task* : Edit Editor
Operation : Delete Editor
Needed Input :
Optional Input :

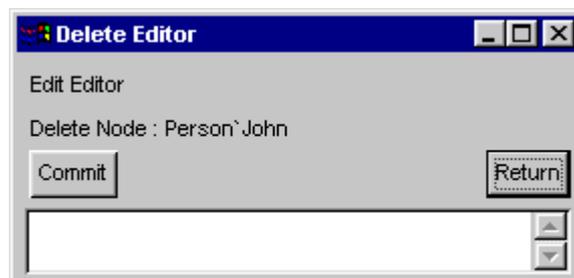


Figure 72 *Operation window "Delete Editor"*

To confirm the update select the “*COMMIT*” button. To return to the *task window* of the current task select the “*RETURN*” button.

3.9.3 Rename an editor

- Task* : Edit Editor
Operation : Rename Editor
Needed Input : The new editor Name
Optional Input :

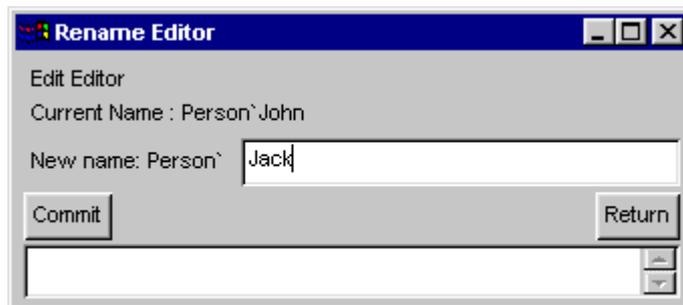


Figure 73 Operation window "Rename Editor"

To confirm the update select the "COMMIT" button. To return to the *task window* of the current task select the "RETURN" button.

3.9.4 Describe an editor

Task : Edit Editor

Operation : Edit Editor Attributes

Needed Input :

Optional Input : affiliation

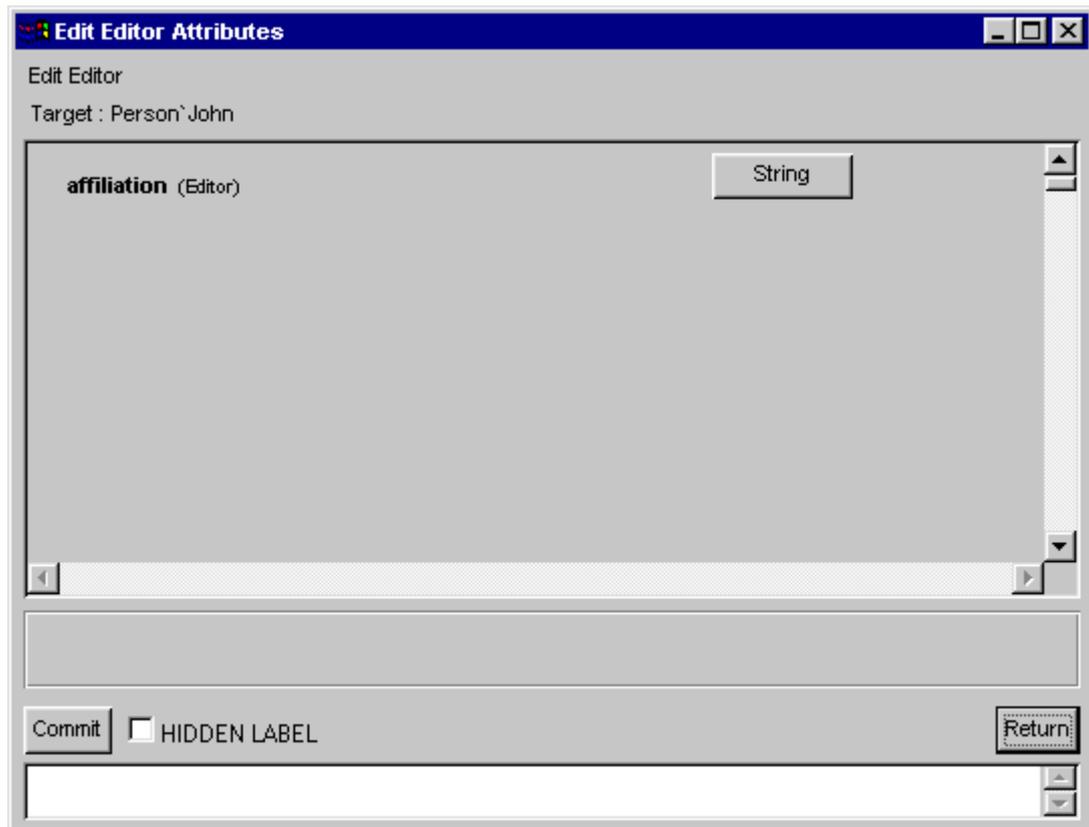


Figure 74 *Operation window "Edit Editor Attributes"*

4. Update Applications

4.1 Release Thesaurus

This functionality releases the new version of the thesaurus. Appropriate schema classes are constructed to maintain changes between versions. In this way, the thesaurus developer can identify the changes done in any version during the thesaurus development. This functionality can be performed by two steps: (a) make a backup copy of the database directory (DB_DIR), (b) using the tool.

Application : Release Thesaurus

Needed Input : The thesaurus to be released

Optional Input :

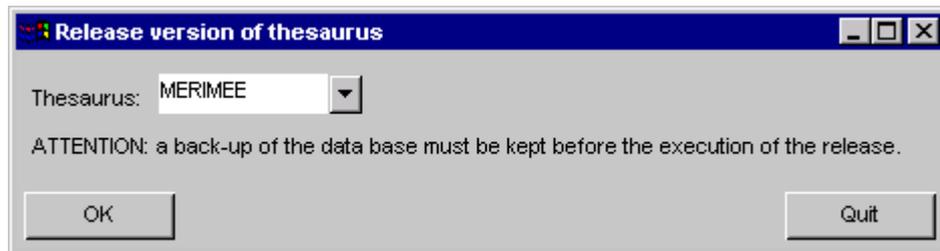


Figure 75 Application window " Release Thesaurus"

To confirm the release of the thesaurus select the "OK" button.

4.2 Create Hierarchy

This functionality adds a set of terms as instances under an existing Hierarchy from the thesaurus. When called it will:

- Read the terms from a loaded file determined by the user
- Add these terms as instances under an existing Hierarchy in a thesaurus. The thesaurus and the Hierarchy are determined by the user

Application : Create Hierarchy

Needed Input : The hierarchy to be created and the file with the terms to be added

Optional Input : The thesaurus where the hierarchy belongs

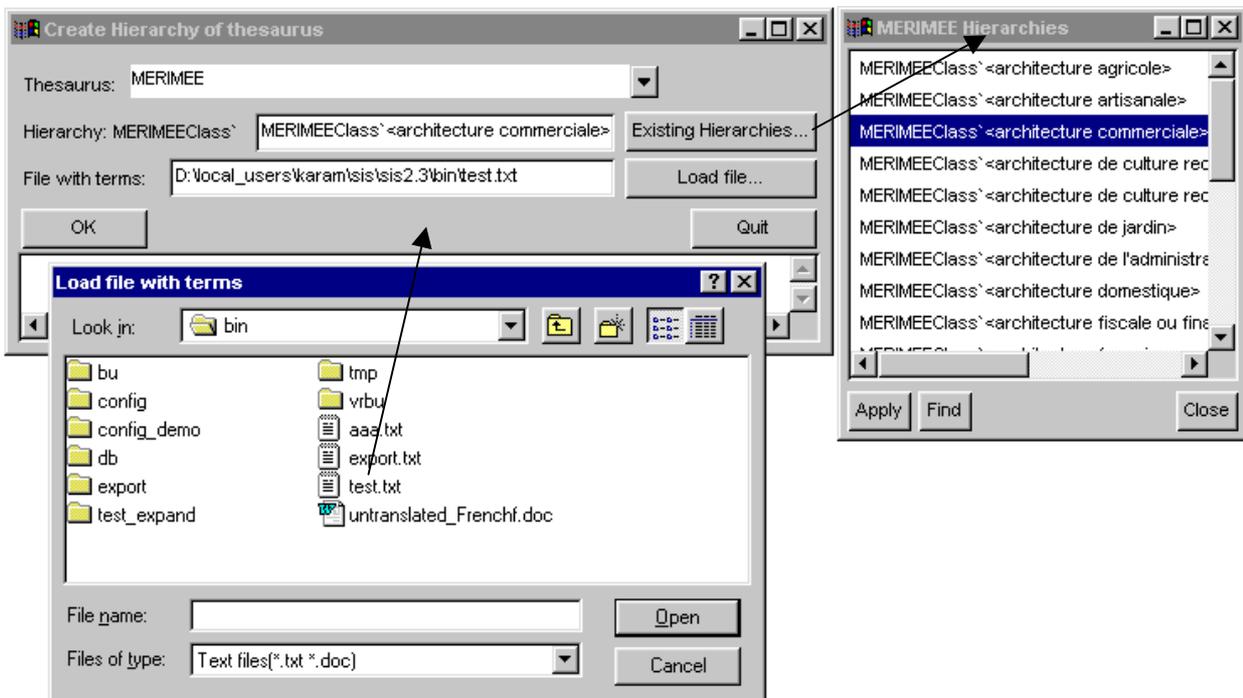


Figure 76 Application window " Create Hierarchy"

To confirm the creation of the thesaurus select the "OK" button.

4.3 Delete Hierarchy

This functionality deletes a Hierarchy from the thesaurus. When called it will:

- Delete all objects added automatically by Entry Forms when the Hierarchy is created
- Delete all objects under the hierarchy which are not under another hierarchy
- Delete any links from the objects that are being deleted and any objects pointed to by these links if they can be deleted (checked by Semantic Checker).

Application : Delete Hierarchy

Needed Input : The thesaurus to be deleted

Optional Input : The thesaurus where the hierarchy belongs

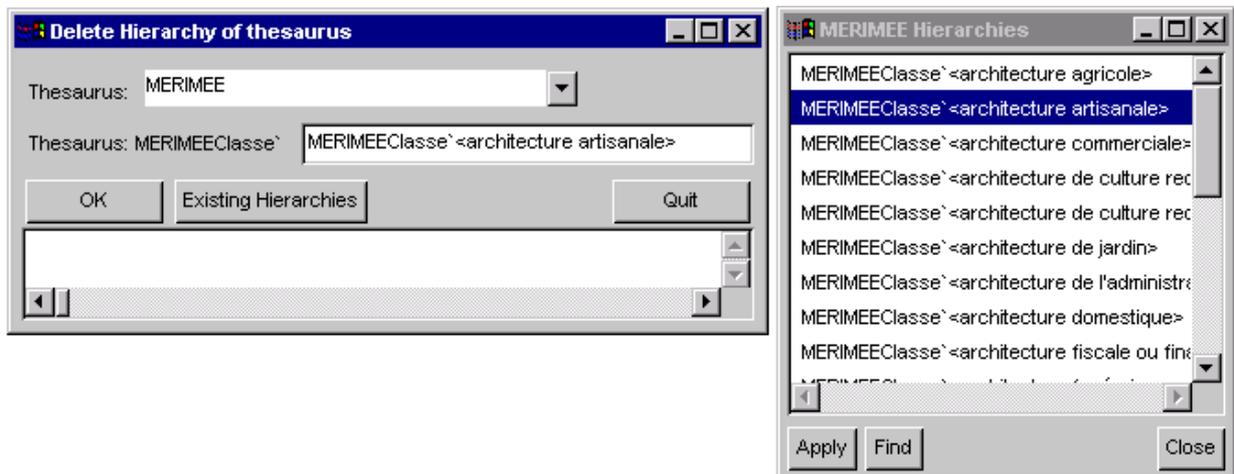


Figure 77 Application window " Delete Hierarchy"

To confirm the deletion of the thesaurus select the "OK" button.