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

 3.4.1 Create a new concept 3.4.2 Rename a new concept 3.4.3 Delete a new concept 3.4.4 Describe a new concept 3.4.5 Classify a new concept 3.4.6 Move new concept to another hierarchy 3.4.7 Associate an new concept with terms from other thesauri 3.4.8 Disassociate a new concept with terms from other thesauri 	34 35 35 36 37 38 40 41
3.4.9 Sort the broader term links of a new concept	42
3.5 Manipulation of Existing Facets 3.5.1 Rename an existing facet 3.5.2 Abandon an existing facet 3.5.3 Describe an existing facet	43 44 44 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 Kename 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 61
5.7.8 Soft the broader term miks of an existing concept	01
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 generalpurpose 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.

Menu Bar			_ Quer	ry Targe	t	Query '	Гуре –	٦	
MB Thesaurus Mana	agement Syste	m							⊐ ×
<u>Eile E</u> dit <u>V</u> iew	<u>T</u> ree Views	Queries	<u>R</u> etrieval	T <u>o</u> ols <u>W</u> ir	ndow				_
Query Target		/				3	🔟 🍾	<u>ش</u>	
Text View								ltems: 0	
									4
				١					
4								Þ	7
						0	m . D 14	a A na c	
				<u> </u>		Que	iy Kesuli	s Area	
						(Te	xt/Graph	Area)	

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

🏽 Global View Card 💦 🗖 🗙
Close

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*".

📲 Thesaurus Management Syster	n		
<u>F</u> ile <u>E</u> dit <u>∨</u> iew <u>T</u> ree Views	<u>Q</u> ueries <u>R</u> etrieval	T <u>o</u> ols <u>W</u> indow	
Query Target TermeFr`EDIFICE AG	Hierarchical Report General Queries Thesauri Facets	> >	🚮 🔝 🔪 🏢 🏠 Items: 20
TOP TERM	Hierarchies	List All AAT Hierarchie	es ERARCHY
<pre><architecture agricole=""> <architecture artisanale=""> <architecture commerciale=""> <architecture culture="" de="" n<br=""><architecture culture="" de="" n<br=""><architecture de="" l'adminis<br=""><architecture domestique=""> <architecture fiscale="" ou="" s<br=""><architecture fiscale="" ou="" s<br=""><architecture fiscale="" ou="" s<br=""><architecture funéraire="" ou<br=""><architecture funéraire="" ou<br=""><architecture industrielle<br=""><architecture industrielle<br=""><architecture industrielle<br=""><architecture nelligieuse=""> <architecture scolaire=""> <acchitecture scolaire=""> <descripteurs communs="" to<br="" à=""><génie civil=""> <urbanisme aménagement<="" et="" pre=""></urbanisme></génie></descripteurs></acchitecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></pre>	Terms Consistency Checking recherche sport lois stration ou de la co stration ou de stration ou	List AII MERIMEE Hiera List AII RCHME Hierarch List AII ICCD Hierarchi List AII DAMD Hierarch ollectivité> votive> u de protection socia. de police> ménagement du territo:	Architecture agricole> architecture artisanale> architecture commerciale> architecture de culture reche architecture de culture reche architecture de jardin> <architecture de="" jardin=""> <architecture domestique=""> <architecture con<br="" funéraire="" ou="">le> <architecture con<br="" funéraire="" ou=""><architecture industrielle=""> <architecture industrielle=""> <architecture militaire=""> <architecture religieuse=""> <architecture solaire=""> <architecture sola<="" td=""></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture>
•			

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).

📲 Thesaurus Managemen	t System				_ 🗆 ×
File Edit View Tree V	/iews Queries	Retrieval Too	s Window		
	L Kenenela				
Query Target TermeFr` <a< td=""><th>rchitectu</th><th>ical Report</th><td></td><td>3</td><td>_ Ⅲ</td></a<>	rchitectu	ical Report		3	_ Ⅲ
	General	Queries 🕨			
Hierarchical Report	Thesaur	i ▶			ltems: 0
	Facets				
	Hierarch	ies 🕨			
<pre></pre>					
- <ensemble agr<="" td=""><th>icole> Consiste</th><th>ncy Checking 🕨</th><td></td><td></td><td></td></ensemble>	icole> Consiste	ncy Checking 🕨			
TERRAS	SE AGRICONE	no, oncoming .			
- EDIFICE AGRIC	OLE				
CELLIE	R				
CUVAGE					
DEPEND	ANUE OT ACDICOLE				
ENIREP	COODEDATINE AC	DICOLE			
	- COOPERATIVE AG	TIVE VINICOLE			
	HANGAR AGRICOL	E			
	REMISE AGRICOL	E			
	- CHARRET	ERIE			
ETABLE					
	BERGERIE				
	ETABLE A CHEVA	UX ~			
	BUDCHEDIE BODCHEDIE	S			
	TADLA				
FENIL	DERIE				
FERME					
GRANGE					
	GRANGE AUX DIM	ES			
GRANGE	MONASTIQUE				
HARAS					
MAGNAN	RKTR				
	COLOMBIND				
SECHOI	R				
	PIECE D'AFFINA	GE			
	SECHOIR A CHAT	AIGNES			
	SECHOIR A CHIC	OREE			
	SECHOIR A HOUB	LON			
	SECHOIR A MAIS				
	SECHOIR A PEAU	X			
	SECHUIR A TABA	L			
ARDWIN	OIR				
AIRE A	BATTRE				
	AIRE A BATTRE	COUVERTE			
AIRE A	SECHER				
GARENN	E				
POULAI	LLER				
RUCHER					
SILO					
VIVIER	אדדס				
T - FIGRON	COLF.				
- ENSEMBLE AGRI	COLR				



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 that 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.

Elle Edit View Tree Views Queries Betrieval Tools Window Query Target TermeFr'EDIFICE AGRICOLE Star View Items: *	🐮 🖪 The	esauru	is Mar	nagement Syste	;m						_ [⊐×
Query Target TermeFr'EDIFICE AGRICOLE	<u>F</u> ile	<u>E</u> dit	⊻iew	<u>T</u> ree Views	<u>Q</u> ueries	<u>R</u> etrieval	T <u>o</u> ols	<u>Wi</u> ndow				
Star View tems: *	Que	ery Tar	get T	ermeFr`EDIFICE A	GRICOLE				3	🔟 🍾	Ш M	
<pre></pre>	Star	View									Items: *	
	/MERIN	IEE_BT	• M	<ar< td=""><td>EDIFICE AGR</td><td>ricole> scriptor</td><td>MERIM MERIM MERIM</td><td>IEE_BT IEE_RT IEE_exact_e IEE_exact_e</td><td>quivalence, to_ quivalence, to_</td><td>AAT ag</td><td>architecture ag NFICES AGRICO gricultural buildi gricultural buildi</td><td></td></ar<>	EDIFICE AGR	ricole> scriptor	MERIM MERIM MERIM	IEE_BT IEE_RT IEE_exact_e IEE_exact_e	quivalence, to_ quivalence, to_	AAT ag	architecture ag NFICES AGRICO gricultural buildi gricultural buildi	

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

<u>Note</u>: 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 Query Target	Query Target set	
Select a "List All " Query	"Text Area" filled	
Select a Retrieval Query	"Text Area" 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
Tree Views*	"Graph Area" filled
Queries*	"Text Area" 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	Object Card
middle mouse click	repeat last Tree Views 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.

8 SIS-TMS Report	Writer
<u>F</u> ile	
Create Report Save As Exit	ie :

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.

New 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 <u>Edit Comment</u>, or to a *primitive value* (string, integer, real) by selecting the appropriate buttons (e.g. <u>String</u>). The selected object may also be associated with the current time/date by selecting <u>Current</u>, or to a specific time/date value by selecting <u>Time</u>.

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).

•	🔒 Selection List
	Available MERIMEEHierarchy
	MERIMEEClass` <architecture agricole=""></architecture>
	MERIMEEClass` <architecture artisanale=""></architecture>
	MERIMEEClass` <architecture commerciale=""></architecture>
	MERIMEEClass` <architecture culture="" de="" recher<="" th=""></architecture>
	MERIMEEClass` <architecture culture="" de="" recherc<="" th=""></architecture>
	MERIMEEClass` <architecture de="" jardin=""></architecture>
	MERIMEEClass` <architecture de="" l'administration<="" th=""></architecture>
-	MERIMEEClass` <architecture domestique=""></architecture>
	Apply Find Close

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.

TaskList Current Task: NONE Quit Quit Entry Forms (MERIMEE) CaskList Released Concepts Quit Quit Quit	TaskList Current Task: NONE	
Current Task: NONE Quit Quit Entry Forms (MERIMEE) TaskList New Concepts Edit Facet Edit Hierarchy Edit Descriptor Released Concepts Quit	Current Task: NONE	
Current Task: NONE Quit Quit Entry Forms (MERIMEE) TaskList Edit Facet Edit Hierarchy Edit Descriptor Released Concepts Quit	Current Task: NONE	
Quit Quit TaskList New Concepts Edit Facet Edit Hierarchy Edit Descriptor Released Concepts Quit		
Quit Quit TaskList New Concepts Edit Facet Edit Hierarchy Edit Descriptor Released Concepts Quit		
Quit CaskList New Concepts Edit Facet Edit Hierarchy Edit Descriptor Released Concepts		
Image: Second		Quit
Entry Forms (MERIMEE) TaskList New Concepts Edit Facet Edit Hierarchy Edit Descriptor Released Concepts Quit		Gon
Entry Forms (MERIMEE) TaskList New Concepts Edit Facet Edit Hierarchy Edit Descriptor Released Concepts Quit		
TaskList New Concepts Edit Facet Edit Hierarchy Edit Descriptor Released Concepts	📲 Entry Forms (MERIMEE)	_ 🗆 ×
New Concepts Edit Facet Edit Hierarchy Edit Descriptor Released Concepts Quit	TaskList	
Edit Facet Edit Hierarchy Edit Descriptor Released Concepts Quit	New Concepts	
Edit Hierarchy Edit Descriptor Released Concepts Quit	Edit Facet	
Edit Descriptor Released Concepts Quit	Edit Hierarchy	
Released Concepts Quit	Edit Descriptor	
		Quit
Edit Released Facet	Released Concepts	
Edit Released Hierarchy	Released Concepts	
Edit Released Descriptor	Released Concepts Edit Released Facet Edit Released Hierarchy	
Auxiliary Concepts	Released Concepts Edit Released Facet Edit Released Hierarchy Edit Released Descriptor	
Edit Source	Released Concepts Edit Released Facet Edit Released Hierarchy Edit Released Descriptor Auxiliary Concepts	
Edit Editor	Released Concepts Edit Released Facet Edit Released Hierarchy Edit Released Descriptor Auxiliary Concepts Edit Source	

Figure 18 Selection of task from the Task List.

🚮 Edit Facet		_ 🗆 ×
Available operations	Selected Objects C List of Facets	
Add Facet	MERIMEEClass`Objet Physique	
Rename Facet		
Delete Facet		
Facet Attributes		
	Selected Object : Class'Objet Physique Find	Next
	R	eturn
J		

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.

👯 Add Facet	_ 🗆 ×
Facet: MERIMEEClass` Objet Physique	
Commit	Return
MERIMEEClass`Objet Physique added in database	-

Figure 20 Operation window "Add Facet"

The SIS Thesaurus Management System, Tutorial

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.

📲 Rename Facet	
Edit Facet Current Name : MERIMEEClass`Objet P	iysique
New name: MERIMEEClass' Personn	3
Commit	Return

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.

🚯 Delete Facet	_ 🗆 ×
Edit Facet	
Delete Node : MERIMEEClass'Objet Physique	
Commit	Return
J	

Figure 22 Operation window "Delete Facet".

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.

🚮 Facet Attributes			_ 🗆 ×
Edit Facet Target : MERIMEEClass`Objet Pł	iysique		
letter_code (Facet)		String	-
I			▼ ►
Attribute operation	Edit attribute classes	•	
			Return

Figure 23 Operation window "Facet Attributes"

3.3 Manipulation of New Hierarchies

🚮 Edit Hierarchy		_ 🗆 ×
Available operations	Selected Objects C List of Hierarchies	
Add Hierarchy	MERIMEEClass'meubles	
Rename Hierarchy		
Delete Hierarchy		
Classify in Facet		
Hierarchy Attributes		
		7
	Selected Object : RIMEEClass`meubles Find	Next
		Return 1
· · · · · · · · · · · · · · · · · · ·		locarri

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

👯 Add Hierarchy	
Hierarchy: MERIMEEClass` meubles	
Facet: MERIMEEClass` Objet Physique	
Commit	Return
MERIMEEClass`meubles added in database	4. V

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.

🗱 Rename Hierarchy	_ 🗆 ×
Rename Hierarchy: MERIMEEClass'meubles	
New name: MERIMEEClass' MEUBLES	
Commit	Return
	- *

Figure 26 Operation window "Rename Hierarchy"

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.

🐮 🖪 Delete Hi	erarchy	
Hierarchy:	MERIMEEClass`meubles	
Commit		Return
		4 1

Figure 27 Operation window "Delete Hierarchy"

J.J.+ Describe a	nen	merareny
Task	:	Edit Hierarchy
Operation	:	Hierarchy Attributes
Needed Input	:	letter_code
Optional Input	:	

3.3.4 Describe a new hierarchy

This operation describes the relations of a new hierarchy.

📲 Hierarchy Attributes			
Edit Hierarchy Target : MERIMEEClass`y	/Y		
letter_code (Facet)		String	-
T			▼ ▼
Attribute operation	Edit attribute classes		•
	EL		Return

Figure 28 Operation window "Hierarchy Attributes"

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.

🄀 Classify in Facet			- O ×
Edit Hierarchy			
Target : MERIMEEClass`meubles			
IsA	State of IsA links		
RestrictedHierarchy HierarchyClass MERIMEEClass`Objet Physique	readonly readonly T existing		
Link with facet			
Commit	Selection List		Return
	Available facet		
	MERIMEEClass`Objet Physique	<u> </u>	
	Apply Find	Close	

Figure 29 Operation window "Classify in Facet"

TaskList Available MERIMEEHierarchy Current Task: Edit Descriptor MERIMEEClass` <architecture agric<="" td=""> Enter MERIMEEHierarchy Quit MERIMEEClass`<architecture de="" of<="" td=""> MERIMEEClass`<architecture de="" of<="" td=""> MERIMEEClass`<architecture de="" of<="" td=""></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture>	Entry Forms (MERIMEE)	🗱 Selection List 📃 🗖 🗙
Current Task: Edit Descriptor MERIMEEClass` <architecture agri<="" td=""> Enter MERIMEEHierarchy architecture agricole> Quit MERIMEEClass`<architecture de="" g<="" td=""> MERIMEEClass`<architecture de="" g<="" td=""> MERIMEEClass`<architecture de="" g<="" td=""> Quit MERIMEEClass`<architecture de="" g<="" td=""> MERIMEEClass`<architecture de="" g<="" td=""> MERIMEEClass`<architecture de="" g<="" td=""></architecture></architecture></architecture></architecture></architecture></architecture></architecture>	TaskList	Available MERIMEEHierarchy
Enter MERIMEEHierarchy <architecture agricole=""> MERIMEEClass`<architecture de="" o<br="">MERIMEEClass`<architecture de="" o<br="">MERIMEEClass`<architecture de="" o<br="">MERIMEEClass`<architecture de="" j<br="">MERIMEEClass`<architecture de="" j<="" th=""></architecture></architecture></architecture></architecture></architecture></architecture>	Current Task: Edit Descriptor	MERIMEEClass` <architecture agric="" ▲<br="">MERIMEEClass`<architecture artis<="" th=""></architecture></architecture>
MERIMEEClass`≺architecture de j MERIMEEClass`≺architecture de j	Enter MERIMEEHierarchy sarchitecture agricole>	MERIMEEClass` <architecture comr<br="">MERIMEEClass`<architecture cu<br="" de="">MERIMEEClass`<architecture cu<="" de="" th=""></architecture></architecture></architecture>
		MERIMEEClass` <architecture de="" ja<br="">MERIMEEClass`<architecture de="" l'a<="" th=""></architecture></architecture>
MERIMEEClass` <architecture dom<="" th=""><th></th><th>MERIMEEClass`<architecture dome<="" th=""></architecture></th></architecture>		MERIMEEClass` <architecture dome<="" th=""></architecture>

3.4 Manipulation of New Concepts

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

👯 Edit Descriptor			_ 🗆 🗵
MERIMEEHierarchy: MERIME	EClass` <architecture agricole=""></architecture>		
Available operations	Selected Objects	C List of Descriptors	
		S Elst of Descriptors	
Add Descriptor			-
Rename Descriptor			
Delete Descriptor			
Classify Descriptor			
Move to Hierarchy			
Descriptor Record			
Add Interthesauri Relation			
Delete Interthesauri Relati			
Sort BT Links			
	Selected Object :	F	ind Next
			Return
			4

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.

📲 Add Descriptor	
Descriptor: TermeFr` TABLE	
Broader Term : TermeFr' meubles	
Commit	Return
TermeFr`TABLE added in database	

Figure 32 Operation window "Add Descriptor"

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.

🕷 Rename Descripto	Г	
Edit Descriptor under M Current Name : TermeF	ERIMEEHierarchy I r`TABLE	MERIMEEClass'meubles
New name: TermeFr`	TABLES	
Commit		Return

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"

The SIS Thesaurus Management System, Tutorial

		·F-
Task	:	Edit Descriptor
Operation	:	Descriptor Record
Needed Input	:	
Optional Input	:	created, modified, editor, found in, not found in, scope note, image, ALT, RT, UF

3.4.4 Describe a new concept

These operations describe the relations of a new descriptor.

R Descriptor Record			. 🗆 🗙
Edit Descriptor under MERIMEEHierarchy MERIMEEClass`meubles			
Target : TermeFr`TABLE			
MERIMEE_ALT (MERIMEEHierarchyTerm)	New	Old	
merimee_created (MERIMEEHierarchyTerm)	Time	Current	
merimee_display (MERIMEEHierarchyTerm)	String		
merimee_editor (MERIMEEHierarchyTerm)	New	Old	
merimee_found_in (MERIMEEHierarchyTerm)	New	Old	
merimee_modified (MERIMEEHierarchyTerm)	Time	Current	
merimee_not_found_in (MERIMEEHierarchyTerm)	New	Old	
त	<u>AH</u>	1	▶
Commit HIDDEN LABEL			Return
			4

Figure 35 Operation window "Descriptor Record"
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.

🔠 Classify Descriptor		_ 🗆 ×
Edit Descriptor under MERIN	IEEHierarchy MERIMEEClass`meubles	
Target : TermeFr`TABLE		
Instance of	State of instance links	
MERIMEEDescriptor	readonly	1
MERIMEENewDescriptor	readonly	
MERIMEEClass`meubles	readonly	
	Selection List	
	Available class	_
1	MERIMEEGuideTerm	
Link with class		
Commit		Return
	Apply Find Close	

Figure 36 Operation window "Classify Descriptor"

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.

💥 Move to Hierarchy	_ 🗆 ×
Term: TermeFr`TABLE	
Hierarchy: MERIMEEClass`meubles	
To Hierarchy: MERIMEEClass`	Available
Broader Term: TermeFr`	Available
Move to Hierarchy options Move node only Move node and subtree C Connect node and subtree	
Commit Delete	Return
	× V

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"

		1
Task	:	Edit Descriptor
Operation	:	Add Interthesaurus Relations
Needed Input	:	From Term, Category, To Term
Optional Input	:	

3.4.7 Associate an new concept with terms from other thesauri

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.

👯 Add Interthesa	uri Relations	_ 🗆 🗙
From Term :	TermeFr`TABLE	
Relation :	MERIMEE_partial_equivalence, to	
To Term : En Term`		
Commit Exi	isting Relations	Return

Figure 39 Operation window "Interthesaurus Relations"

		1
Task	:	Edit Descriptor
Operation	:	Delete Interthesaurus Relations
Needed Input	:	From Term, Category, To Term)
Optional Input	:	

3.4.8 Disassociate a new concept with terms from other thesauri

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.

Celete Interthesauri Relations		_ 🗆 X
From Term :	TermeFr`TABLE	
Relation :		T
Commit		Return
		+

Figure 40 Operation window "Delete Interthesaurus Relations"

Task	:	Edit Descriptor
Operation	:	Sort BT links
Needed Input	:	Final order of links with drag & drop in the list of the card

3.4.9 Sort the broader term links of a new concept

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.

🗱 Sort Links	- 🗆 🗵
C Backward Links of TermeFr`meubles	
under category MERIMEE_BT	•
Sorted Links	<u> </u>
(Unnamed) from TermeFr`TABLE	
Unsorted Links	
	7
Sort	Return

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.

a Edit Released Facet				_ 🗆 X
Available operations	C Selected Objects	List of Facets		
Rename Facet				<u> </u>
Abandon Facet				
Facet Attributes				
	Selected Object :		Find	Next
	p			11678
				Return
				4

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.

🗱 Rename Facet	
Edit Facet Current Name : MERIMEEClass`Objet Physique	
New name: MERIMEEClass' Personne	
Commit	Return
	4

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.

🗱 Delete Facet	- 🗆 ×
Edit Facet	
Delete Node : MERIMEEClass'Objet Physique	
Commit	Return
	4

Figure 44 Operation window "Delete Facet"

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

Recet Attributes			_ 🗆 ×
Edit Facet			
Target : MERIMEEClass`Objet Ph	ysique		
letter_code (Facet)		String	
T			▼
Attribute operation	Edit attribute classes		-
			Return

Figure 45 Operation window "Facet Attributes"

3.6 Manipulation of Existing Hierarchies

👷 Edit Released Hierarch	y 💶 🗆 🗙
Available operations	C Selected Objects C List of Hierarchies
Rename Hierarchy	MERIMEEClass` <architecture agricole=""></architecture>
Abandon Hierarchy	MERIMEEClass` <architecture artisanale=""></architecture>
Classify in Easet	MERIMEEClass` <architecture commerciale=""></architecture>
	MERIMEEClass` <architecture culture="" de="" loisir="" recherche="" sport=""></architecture>
Hierarchy Attributes	MERIMEEClass` <architecture culture="" de="" loisirs="" recherche="" sport=""></architecture>
	MERIMEEClass` <architecture de="" jardin=""></architecture>
	MERIMEEClass` <architecture collectivité="" de="" l'administration="" la="" ou=""></architecture>
	MERIMEEClass` <architecture domestique=""></architecture>
	MERIMEEClass` <architecture financière="" fiscale="" ou=""></architecture>
	MERIMEEClass` <architecture commémorative="" funéraire="" ou="" votive=""></architecture>
	MERIMEEClass` <architecture d'assistance="" de="" hospitalière="" ou="" protection="" s<="" td=""></architecture>
	MERIMEEClass' sarchitecture industrielle>
	Selected Object : Find Next
	Return
<u> </u>	<u>×</u>

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.

🕷 Rename Hierarchy	_ 🗆 ×
Rename Hierarchy: MERIMEEClass` <génie civil=""></génie>	
New name: MERIMEEClass' GENIE CIVIL	
Commit	Return
	4

Figure 47 Operation window "Rename Hierarchy"

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".

erarchy	_ 🗆 ×
MERIMEECIass'ENSEMBLE AGRICOLE	
	Return
	Þ
	erarchy MERIMEECIass`ENSEMBLE AGRICOLE

Figure 48 Operation window "Abandon Hierarchy"

3.6.3	Describe	an	existing	hierarchy
	Deserie		en seing	mer ar en y

Task	:	Edit Released Hierarchy
Operation	:	Hierarchy Attributes
Needed Input	:	letter_code
Optional Input	:	

This operation describes the relations of an existing hierarchy.

📲 Hierarchy Attributes			_ 🗆 🗵
Edit Released Hierarchy			
Target : MERIMEEClass` <génie ci<="" td=""><td>/il></td><td></td><td></td></génie>	/il>		
letter_code (Facet)		String	1
			_
4			Þ
Attribute operations	Edit attribute classes		
			Return
			*

Figure 49 Operation window "Hierarchy Attributes"

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.

🗱 Classify in Facet		
Edit Released Hierarchy		
Target : MERIMEEClass` <génie civil=""></génie>		
IsA	State of IsA links	
MERIMEEHierarchyClass	readonly	_
MERIMEEClass`TopFacet	🥅 existing	
		_1
•		
Link with facet		
Commit		Return
		4

Figure 50 Operation window "Classify in Facet"

🗱 Entry Forms (MERIMEE)	Selection List
TaskList	Available MERIMEEHierarchy
Current Task: Edit Released Descriptor Enter MERIMEEHierarchy carchitecture agricole>	MERIMEEClass` <architecture agricole=""> MERIMEEClass`<architecture artisanale=""> MERIMEEClass`<architecture commercia="" culture="" culture<="" de="" merimeeclass`<architecture="" th=""></architecture></architecture></architecture>
	MERIMEEClass' <architecture de="" jardin=""> MERIMEEClass' <architecture de="" l'admini:<br="">MERIMEEClass' <architecture domestique<br="">Papely Find Close</architecture></architecture></architecture>

3.7 Manipulation of Existing Concepts

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

📲 Edit Released Descript	or	_ 🗆 ×
MERIMEEHierarchy : MERIME	EClass` <architecture agricole=""></architecture>	
Available operations	C Selected Objects C List of Descriptors	
Move to Hierarchy	TermeFr` <architecture agricole=""></architecture>	<u> </u>
Abandon Descriptor	TermeFr`ABREUVOIR	
Undo Abandon Descriptor	TermeFr`AIRE A BATTRE	
Rename Descriptor	TermeFr`AIRE A BATTRE COUVERTE	
Undo Rename Descriptor	TermeFr`AIRE A SECHER	
Descriptor Record		
Add Interthesauri Relation		
Delete Interthesauri Relati		
Sort BT Links	TermeFr`COOPERATIVE AGRICOLE	
		<u> </u>
	Selected Object : Fi	ind Next
		Return
		4

Figure 52 Task window "Edit Released Descriptor"

- 1

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

Contract Con			- 🗆 ×
Edit Released Descriptor under MERIMEEHierarchy MERIMEEClass Target : TermeFr`CELLIER			
MERIMEE_ALT (MERIMEEHierarchyTerm)	New	Old	
TermeFr`CHAI	C _{existing}		
merimee_display (MERIMEEHierarchyTerm)	String]	
merimee_editor (MERIMEEHierarchyTerm)	New	Old	
merimee_found_in (MERIMEEHierarchyTerm)	New	Old	
merimee_modified (MERIMEEHierarchyTerm)	Time	Current	
merimee_not_found_in (MERIMEEHierarchyTerm)	New	Old	
र	<u></u>	1	Þ
Commit HIDDEN LABEL			Return
			4

Figure 53 Operation window "Descriptor Record"

These operations describe the relations of an existing descriptor.

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 is 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.

🙀 Abandon Descriptor		_ 🗆 🗵
Term :	FrTerme`CELLIER	
Commit		Return
•		Þ

Figure 54 Operation window "Abandon Descriptor"



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.

📲 Undo Aban	don Descriptor	
Descriptor:	TermeFr`CELLIER	
Broader Terr	n: TermeFr`	
Commit		Return
		4

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.

Move to Hierarchy	_ 🗆 ×
Term: TermeFr`BERGERIE	
Hierarchy: MERIMEEClass` <architecture agricole=""></architecture>	
To Hierarchy: MERIMEEClass`	Available
Broader Term: TermeFr`	Available
 Move to Hierarchy options Move node only Move node and subtree Connect node and subtree 	
CommitDelete	Return
	× *

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.

👷 Existing Broader Term Links	- 🗆 ×
(Unnamed) to TermeFr`meubles	A
	-
4	Þ
Delete	Close

Figure 58 Operation window "Existing Broader Term Links"

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.

👷 Rename Descriptor			_ 🗆 🗵
TermeFr'CELLIER	to be renamed to:	TermeFr`CELIEF	2
			T
Enter new name: TermeP	'r		Undo
Commit			Close
			4

Figure 59 Operation window "Rename Descriptor"

3.7.6.Undo	Rename	an existing	concept
U IIIUUU IIUU	Itemanie	un 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

📲 Undo Rename Descripto	r		
TermeFr`CELIER	to be renamed to:	TermeFr`CELLIER	2
•			
	_		
		L	Undo
Commit			
		_	01000
			~

used for links should be deleted

Figure 60 Operation window "Undo Rename Descriptor"

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

3.7.7 Associate an existing concept with terms from other thesauri

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.

👷 Add Interthesau	ri Relations	_ 🗆 ×
From Term :	TermeFr`SILO	
Relation :	MERIMEE_partial_equivalence, to_RCH	IME 💌
To Term : EnTerm`		
Commit Exis	ting Relations	Return
		-

Figure 61 Operation window "Add Interthesaurus Relation"

control Disussociate an existing concept with terms if on a			
Task	:	Edit Released Descriptor	
Operation	:	Delete Interthesaurus Relations	
Needed Input	:	From Term, Category, To Term	
Optional Input	:		

3.7.8 Disassociate an existing concept with terms from other thesauri

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.

Contractions		- 🗆 ×
From Term :	TermeFr`SILO	
Relation :	(MERIMEE_exact_equivalence, to_AAT)	EnTerm`: 🔻
Commit		Return
		4

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.

]

	onu	er verminning 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

3.7.8 Sort the broader term links of an existing concept

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.

👷 🖥 Sort Links		_ 🗆 ×			
C Backward Links	of TermeFr` <architecture agricole=""></architecture>				
under category	MERIMEE_BT	•			
Sorted	Links	A			
(Unnamed) from Te	rmeFr`≺e⊓semble agricole>				
(Unnamed) from Te	rmeFr`<édicule agricole>				
(Unnamed) from TermeFr`EDIFICE AGRICOLE					
Unsorted Links					
(Unnamed) from Te	rmeFr`ENSEMBLE AGRICOLE				
		Ŧ			
Sort		Return			

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

📲 Edit Source				_ 🗆 X
Available operations	C Selected Objects	List of Sources		
Add Source	Literature`FGH1201			A
Rename Source				
Delete Source				
Classify Source				
Edit Source Attributes				
			Eind	Nevt
	Scienced Object .			INGAL
				Return
				4

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	:	

👯 Add Source	_ 🗆 ×
Edit Source	
Create Node: Literature'	
Commit	Return
Literature'FGH1202 created in the database	4

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	:	

A Kename Source	
Edit Source Current Name : Source`FGH1201	
New Name Source'FGH1202	
Commit	Return

Figure 66 Operation window "Rename Source"

The SIS Thesaurus Management System, Tutorial

3.8.3 Delete a source

Task	:	Edit Source
Operation	:	Delete Source
Needed Input	:	The new Descriptor/Term Name
Optional Input	:	

🗱 Delete Source	_ 🗆 ×
Edit Source	
Delete Node : Literature`FGH1201	
Commit	Return
	4

Figure 67 Operation window "Delete Source"

3.8.4 Describe a source

Task	:	Edit Source
Operation	:	Edit Source Attributes
Needed Input	:	full_reference

Optional Input :

Ma Edit Source Attributes		_ 🗆 ×
Edit Source		
Target : Literature`FGH1201		
full_reference (Source)	String	1
		Return
		4

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.

Edit Source		
Target : Literature`FGH1201		
Instance of	State of instance links	
Source	existing	1
Link with Source classes	Available Source classes Citation Image: Citation Monograph Serial Source Image: Citation	Return

Figure 69 Operation window "Classify Source"

👷 Edit Editor				_ 🗆 ×
Available operations	C Selected Objects	C List of Editors		
Add Editor				
Rename Editor				
Delete Editor				
Edit Editor Attributes				
				~
	Selected Object :		Find	Next
			-	Between
				Return
				*

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	:	

Add Editor	_ 🗆 ×
Edit Editor	
Create Node: Person'	
Commit	Return
Person John created in the database	A V

Figure 71 Opeartion 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"

3.9.3 Rename an editor

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

Rename Editor	
Edit Editor Current Name : Person`John	
New name: Person' Jack	
Commit	Return
	4

Figure 73 Operation window "Rename Editor"

3.9.4 Describe an editor

Task	:	Edit Editor
Operation	:	Edit Editor Attributes
Needed Input	:	
Optional Input	:	affiliation

🗱 Edit Editor Attributes		- 🗆 🗵
Edit Editor		
Target : Person`John		
affiliation (Editor)	String	1
		_
<u>र</u>		-
		Return
		4

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

In the termination of termination o	esaurus	_ 🗆 🗵	MERIMEE Hierarchies
Thesaurus: MERIMEE		•	MERIMEEClass` <architecture agricole=""></architecture>
Hierarchy: MERIMEEClass* File with terms: D: Vocal_ OK Load file with terms Look jn: bu bu config config config_demo db	MERIMEEClass` <architecture commerciale=""> users\karam\sis\sis2.3\bin\test.txt</architecture>	Existing Hierarchies	MERIMEEClass` <architecture artisanale=""> MERIMEEClass`<architecture commerciale=""> MERIMEEClass`<architecture culture="" de="" rec<="" td=""> MERIMEEClass`<architecture culture="" de="" rec<="" td=""> MERIMEEClass`<architecture culture="" de="" rec<="" td=""> MERIMEEClass`<architecture de="" jardin=""> MERIMEEClass`<architecture de="" jardin=""> MERIMEEClass`<architecture de="" jardin=""> MERIMEEClass`<architecture de="" l'administra<="" td=""> MERIMEEClass`<architecture domestique=""> MERIMEEClass`<architecture fina<="" fiscale="" ou="" td=""> MERIMEECLASS` MERIMEECLASS`<architecture fina<="" fiscale="" ou="" td=""></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture></architecture>
export test_expand File <u>n</u> ame: Files of <u>type</u> : Text fi	i test txt i untranslated_Frenchf.doc [es(*.txt *.doc)	<u>O</u> pen Cancel	

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

🗱 Delete Hierarchy of thesaurus	MERIMEE Hierarchies
	MERIMEEClasse` <architecture agricole=""></architecture>
	MERIMEEClasse` <architecture artisanale=""></architecture>
Thesaurus: MERIMEEClasse` MERIMEEClasse` <architecture artisanale=""></architecture>	MERIMEEClasse` <architecture commerciale=""></architecture>
	MERIMEEClasse` <architecture culture="" de="" rec<="" th=""></architecture>
OKQuitQuit	MERIMEEClasse` <architecture culture="" de="" rec<="" td=""></architecture>
à.	MERIMEEClasse` <architecture de="" jardin=""></architecture>
	MERIMEEClasse` <architecture de="" l'administra<="" th=""></architecture>
	MERIMEEClasse` <architecture domestique=""></architecture>
	MERIMEEClasse` <architecture fine<="" fiscale="" ou="" th=""></architecture>
	Apply Find Close

Figure 77 Application window " Delete Hierarchy"

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