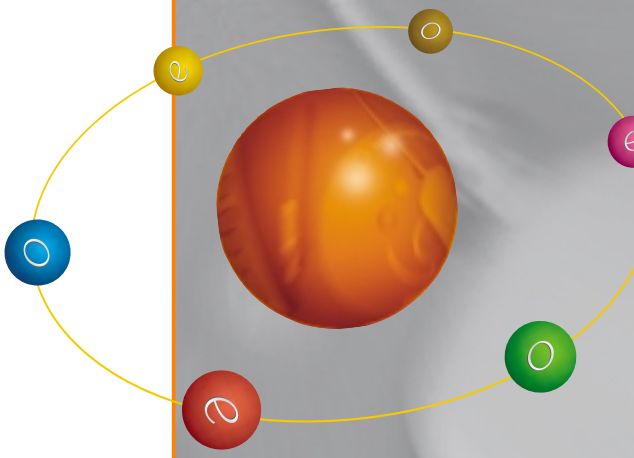
Objecteering/UML

Objecteering/UML Requirements
User Guide
Version 5.2.2



Objecteering
Software

www.objecteering.com

Taking object development one step further

Objecteering/UML version 5.2.2 - CODOBJ 001/001

Contents

Chapter 1: Introduction to Objecteering/UML Requirements

Overview of Objecteering/UML Requirements

Title

Welcome to *Objecteering/UML Requirements*!

Objecteering/UML Requirements, available in the Enterprise Edition of Objecteering/UML 5.2.2, is a sophisticated requirements management tool. The *Objecteering/UML Requirements* tool can create, modify, trace and manage a wide range of information, so as to guarantee that a project complies to specified requirements.

Objecteering/UML Requirements provides methodological support for the modeling of use cases during the needs analysis phase, and includes extensive support of traceability between requirements, dictionaries and other model elements, notably use cases.

Chapter 3 of this user guide, "*Objecteering/Requirements first steps* ", provides the user with a step by step demonstration of the workings of the *Objecteering/UML Requirements* tool. We also recommend that all users undertake the general first steps project in the *Objecteering/Introduction* user guide, in order to get to know the various general functions provided by Objecteering/UML.

Functions

Using the *Objecteering/UML Requirements* tool, it is possible to:

- create and manage requirements and their containers
- create and manage dictionaries and terms
- manage traceability
- create UML elements from spreadsheet editors
- edit elements
- check elements
- use and manage the sort feature
- use and manage filters
- generate documentation

Structure of the Objecteering/UML Requirements user guide

The *Objecteering/UML Requirements* user guide is structured as follows:

- Chapter 1 - *Introduction to Objecteering/UML Requirements*
- Chapter 2 - *Using Objecteering/UML Requirements*
- Chapter 3 - *Objecteering/UML Requirements first steps*
- Chapter 4 - *Functions of Objecteering/UML Requirements*
- Chapter 5 - *The user interface*
- Chapter 6 - *The requirements properties editor*
- Chapter 7 - *Word and Objecteering/UML Requirements*
- Chapter 8 - *Objecteering/UML Requirements annotations*
- Chapter 9 - *Parameterizing Objecteering/UML Requirements*
- Chapter 10 - *The Objecteering/UML Requirements DTD*

Glossary

- *Dictionary*: A set of terms used in a project.
- *Dictionary spreadsheet editor*: The spreadsheet editor used when working with dictionaries and terms.
- *Property*: A characteristic of a requirement (for example, its priority level).
- *Properties editor*: The Objecteering/UML properties editor, which contains a "*Requirements*" tab where the *Objecteering/UML Requirements* tool has been selected. The properties editor provides easy access to information specific to requirements, requirement containers, dictionaries and terms.
- *Property set:* A group of properties used in a project.
- *Requirement*: A function which must be included in the software or system developed, in order to satisfy a contract, specification, standard or other formally defined constraint.
- *Requirement container*: The physical grouping of a set of requirements. A requirement can only belong to one requirement container.
- *Requirements properties editor*: The window used to create, modify and delete requirement properties.
- *Requirements spreadsheet editor*: The spreadsheet editor used when working with requirements and requirement containers.
- *Term*: An item of vocabulary identified by a name, which has a clear definition in the context of the system which is to be developed.
- *Traceability*: A kind of correspondence between different elements represented. Typically, traceability links are used to express which UML model elements represent which requirements or terms.

Chapter 2: Using Objecteering/UML Requirements

Working with the Objecteering/UML Requirements tool

Introduction

Before the *Objecteering/UML Requirements* tool can be used, Objecteering/UML 5.2.2 must already have been installed, and the following steps must be carried out:

- create a working UML modeling project
- select the *Objecteering/UML Requirements* tool

Creating a working UML modeling project

For information on how to create a UML modeling project, please refer to the "*Creating or opening a UML modeling project*" section in chapter 3 of the *Objecteering/UML Modeler* user guide.

Selecting Objecteering/UML Requirements

Launch *Objecteering/UML Modeler* on your newly-created UML modeling project.

The  "*UML modeling project modules*" icon opens the window used to select the tool (as shown in Figure 2-1).

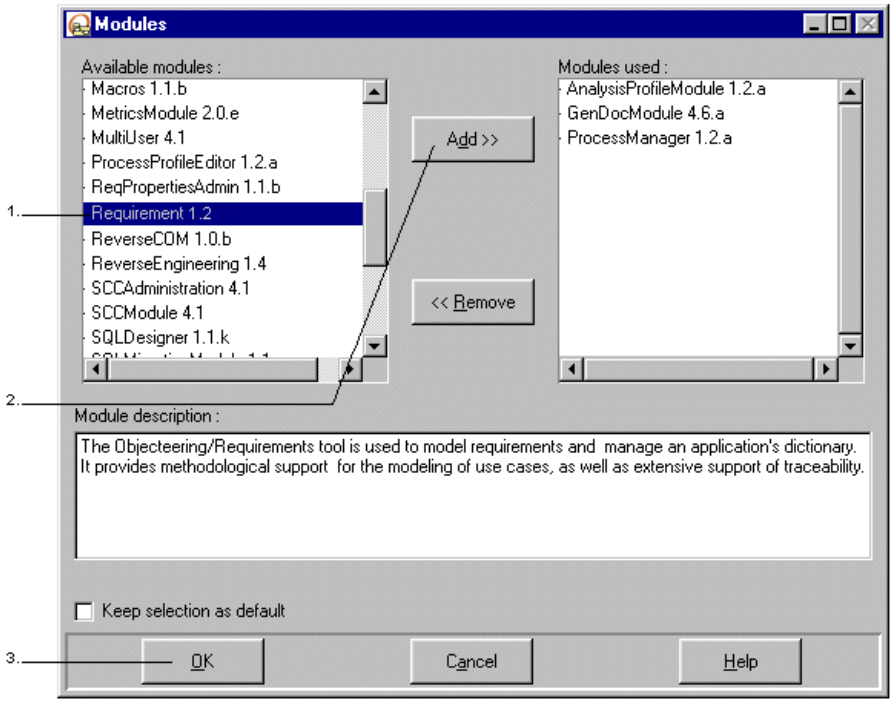


Figure 2-1. Selecting the *Objecteering/UML Requirements* tool

Steps:

1 - Select the "*Requirements*" tool from the available modules list on the left-hand side of the screen.

2 - Click on the "*Add*" button. The "*Requirements*" tool then appears in the right-hand "*Modules used*" column.

3 - Click on "*OK*" to confirm. If the "*Keep selection as default*" box is checked, the "*Requirements*" tool will automatically be available during future Objecteering/UML sessions.

Continue by selecting the "*ReqPropertiesAdmin*" tool in the same way.

For further information on tool selection, please refer to the "*Selecting modules in the current UML modeling project*" section in chapter 3 of the *Objecteering/Introduction* user guide.

Once you have selected the *Objecteering/UML Requirements* tool, the following changes can be seen (Figure 2-2).

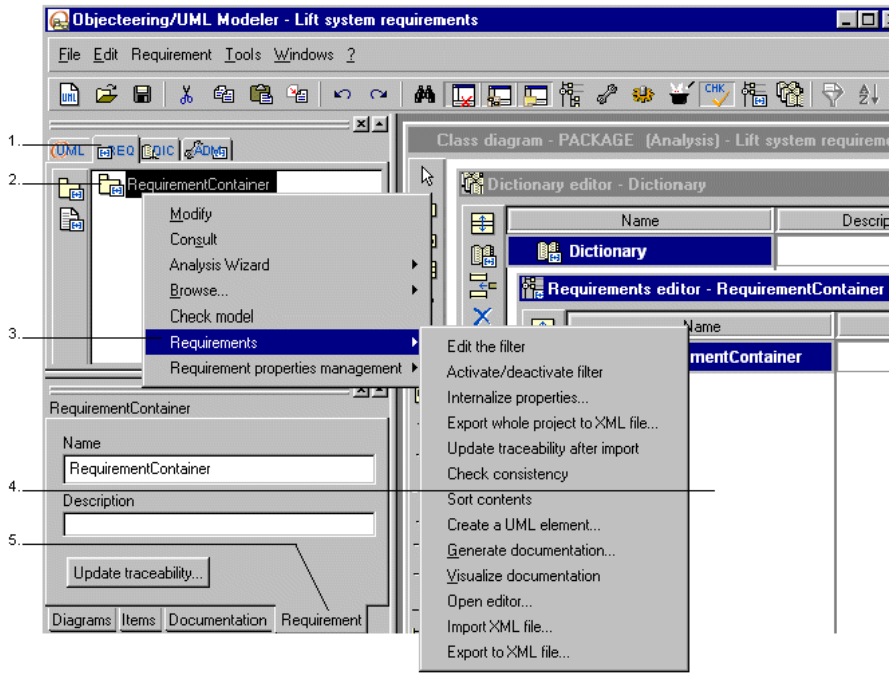


Figure 2-2. Objecteering/UML after selecting the *Objecteering/UML Requirements* tool

Key:

1 - A set of tabs used to browse between the classic UML model element explorer, the Requirements explorer, the Dictionary explorer and the Requirement Properties explorer.

2 - The main explorer shows the "*RequirementContainer*" root requirement container.

3 - Two additional context menus, "*Requirement*" and "*Requirement properties management*" are now available. The "*Requirement*" context menu is available on requirement containers, dictionaries, requirements, terms, properties and property sets, whilst the "*Requirement properties editor*" context menu is available on all elements.

4 - A class diagram, dictionary spreadsheet editor and requirements spreadsheet editor appear. Separate explorers also appear (a classic UML model element explorer, a Requirements explorer, a Dictionary explorer and a Requirement Properties explorer).

5 - A "*Requirements*" tab appears in the properties editor.

The properties editor for the Objecteering/UML Requirements tool

The properties editor is essentially a window designed to aid the user in his modeling, by providing rapid access to various information and services he may need to use.

The properties editor contains a number of tabs, including a "*Requirements*" tab when the *Objecteering/UML Requirements* tool has been selected for the current UML modeling project. This tab is used to:

- enter or modify certain information relevant to requirements on the element selected (in the explorer, the requirements spreadsheet editor, the dictionary spreadsheet editor or the Objecteering/UML graphic editors), for example, requirement properties
- manage the traceability of the element selected

For further general information on the properties editor, please refer to the "*The properties editor*" section in chapter 3 of the *Objecteering/UML Modeler* user guide.

For more specific information on the "*Requirements*" tab of the properties editor, please refer to the "*The Objecteering/UML properties editor*" section in chapter 5 of this user guide.

Chapter 3: Objecteering/UML
Requirements first steps

Overview of the Objecteering/UML Requirements first steps

Introduction

Welcome to the *Objecteering/UML Requirements* first steps, designed to help you discover the various different features of the *Objecteering/UML Requirements* tool.

By following the clearly laid-out steps, you will discover how to:

- create dictionaries
- add terms to the dictionary of your requirements project
- create requirement containers
- create requirements
- create model elements from spreadsheet editors
- create and work with use case diagrams
- create traceability links
- generate requirements documentation

These first steps last, on average, 20 minutes for an inexperienced user.

Note: The *Objecteering/UML Requirements* first steps should not be seen as a real example of requirement modeling, but rather as a means of discovering the various features provided by the tool.

Sources

For the purposes of these first steps, we are going to create a new UML modeling project named "*Lift system requirements*" (for further information on how to create a new UML modeling project, please refer to the "*Creating a new UML modeling project*" section in chapter 3 of the *Objecteering/Introduction* user guide).

Creating a dictionary

After creating our new UML modeling project, "*Lift system requirements*", and selecting the Objecteering/UML Requirements tool for this UML modeling project, we are going to start these first steps by structuring our dictionary, where all the terms which will be used in the requirements project are going to be defined.

In order to build a dictionary which is easy to use, we recommend that you first break down your dictionary into smaller dictionaries, in other words, kinds of sections, in which terms are grouped together according to the main themes.

Dictionaries, which are structuring units, can be created either in the DIC dictionary tab of the explorer or in the dictionary spreadsheet editor. To create a new dictionary, simply carry out the steps shown in Figure 3-1 below.

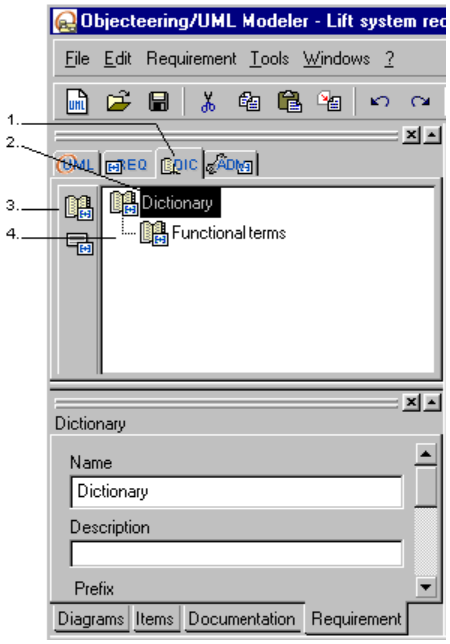


Figure 3-1. Creating a new dictionary

Steps:

1 - Click on the  tab of the explorer. The dictionary explorer is then displayed.

2 - Click on the root dictionary.

3 - Click on the  "*Create a dictionary*" icon.

4 - Enter a name for your new dictionary, for example, "*Functional terms*". Continue by creating the "*Technical terms*" dictionary under the dictionary root, and the "*Environmental terms*" dictionary in the "*Functional terms*" dictionary.

The result of these operations in the dictionary spreadsheet editor (which is opened by clicking on the  "*Open the dictionary editor*" icon) is shown in Figure 3-2.

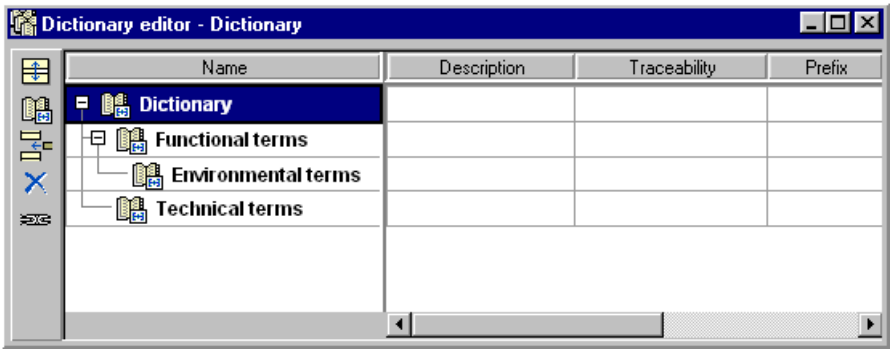


Figure 3-2. The dictionary editor containing the newly-created dictionaries

Note: Dictionaries appear in **bold** in the dictionary spreadsheet editor.

The next step is to enter a definition for each of these dictionaries. This can be done in several different ways:

- by directly entering definitions in the dictionary spreadsheet editor
- by double-clicking on the dictionary in the DIC tab of the explorer and entering definitions in the "*Modify*" window which then appears
- by right-clicking on the dictionary and choosing "*Modify*" from the context menu
- through the "*Requirements*" tab of the properties editor

In our example, we are going to enter definitions directly in the dictionary spreadsheet editor (as shown in Figure 3-3).

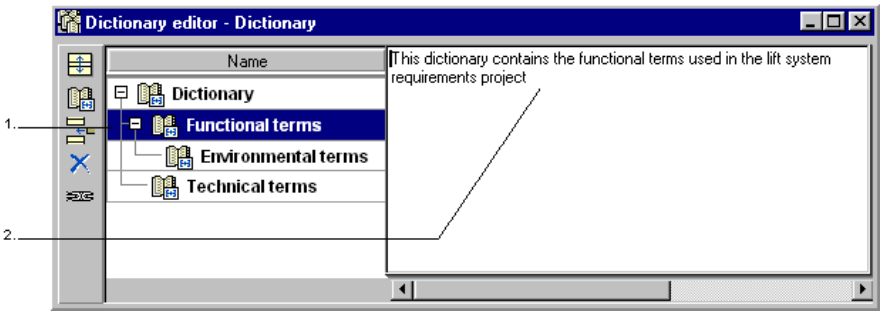


Figure 3-3. Entering a description for the "*Functional terms*" dictionary

Steps:

1 - Click on the "*Functional terms*" dictionary in the dictionary spreadsheet editor.

2 - In the "*Description*" column, enter a description of your new dictionary, for example, "*This dictionary contains the functional terms used in the lift system requirements project*".

Continue by entering descriptions for the "*Technical terms*" and "*Environmental terms*" dictionaries.

The result of these operations is shown in Figure 3-4 below.

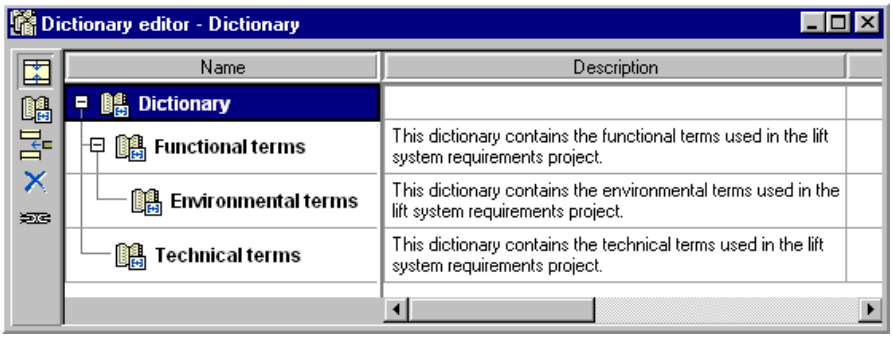


Figure 3-4. The dictionaries you have just created

Note: In Figure 3-4, the descriptions of the different dictionaries are seen in their entirety. This is because the dictionary spreadsheet editor is in "*expand*" mode. In "*compact*" mode, only part of the description is visible. To change from compact mode to expand mode or vice versa, simply click on the "*Change spreadsheet editor mode to expand*" or "*Change spreadsheet editor mode to compact*" icons in the spreadsheet editor palette.

Don't forget that if you wish to work only in your dictionary spreadsheet editor, and have it as the sole window open, you can close the main explorer, properties editor and console, simply by clicking on the "*Show or hide the console*", "*Show or hide the main explorer*" or "*Show or hide the properties editor*" icons.

Creating a term in the dictionary

After creating the different dictionaries which will be used to structure your terms, you can now proceed with the creation of the actual dictionary terms themselves (as shown in Figure 3-5 below).

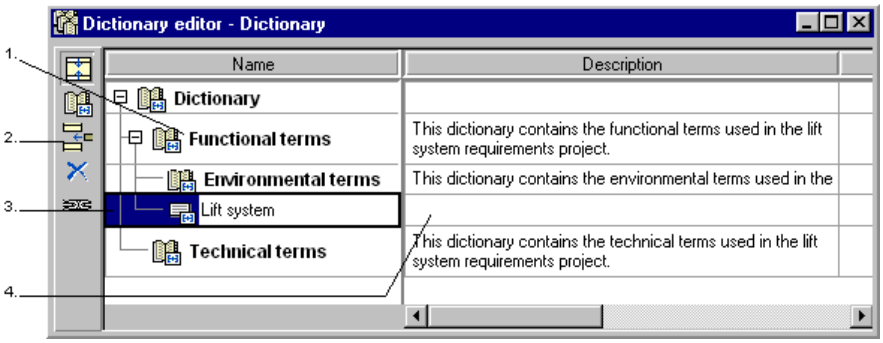


Figure 3-5. Creating a term

Steps:

1 - In the dictionary spreadsheet editor, select the "*Functional terms*" dictionary.

2 - Click on the  "*Insert an element*" icon.

3 - Enter a name for your term (in our example, "*Lift system*").

4 - Enter a description for your term (in our example, "*The lift system comprises the lift shaft and its engine, the cabin, its doors and its engine, the floors and their doors, and the different associated buttons and detectors*").

Note: Terms can also be created in the explorer, by clicking on the  "*Create a term*" icon.

Continue by creating the following terms in the "*Functional terms*" dictionary:

- "*Cabin*": Mobile part of the lift, which carries users to different floors of the building.
- "*Button*": Means through which the user can call the lift, indicate his desired destination or stop the lift.
- "*Control panel*": The set of buttons available to the user, both inside the cabin and on the landing.
- "*Detector*": An element which detects events leading to a change in the state of lift system elements.
- "*Door*": Means through which the user enters and exits the cabin (cabin doors and landing doors).
- "*User*": Person wishing to move up or down to a different floor within a building by means of the lift.
- "*Lift system supervisor*": Person in charge of managing lift system operations.

In the "*Environmental terms*" dictionary, create the following terms:

- "*Building*": Construction containing several floors, which can contain a lift system.
- "*Floor*": Level in a building, to which the lift can transport users.
- "*Landing*": Part of a floor, from which a user has access to the cabin.

In the "*Technical terms*" dictionary, create the following terms:

- "*Lift shaft*": Structure inside which the lift system functions.
- "*Engine*": Engine allowing the cabin to go up or down within the lift shaft.

The result of these operations is shown in Figure 3-6 below.

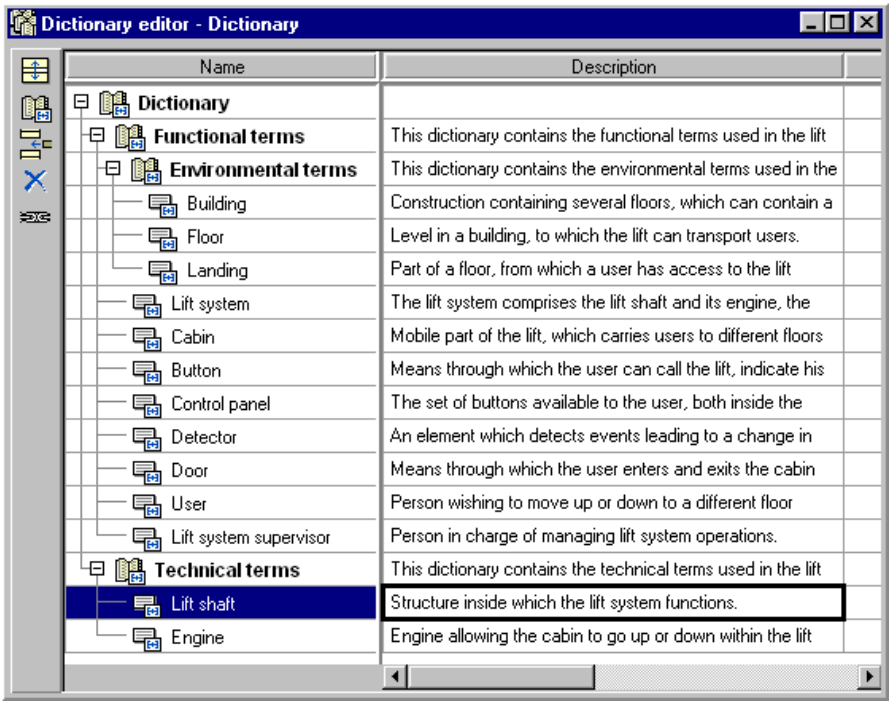


Figure 3-6. The dictionary spreadsheet editor containing several terms in three dictionaries

Note: Figure 3-6 shows the dictionary spreadsheet editor in compact mode.

Creating a requirement container

Creating a requirement container

After defining our terms in the dictionary, we are now going to create some requirements for our lift system.

Like dictionaries and terms, requirements are structured into requirement containers, used to improve visibility for the user.

Requirement containers, which are structuring units, can be created either in the requirements spreadsheet editor or in  requirements tab of the explorer. To create a new requirement container, simply carry out the steps shown in Figure 3-7 below.

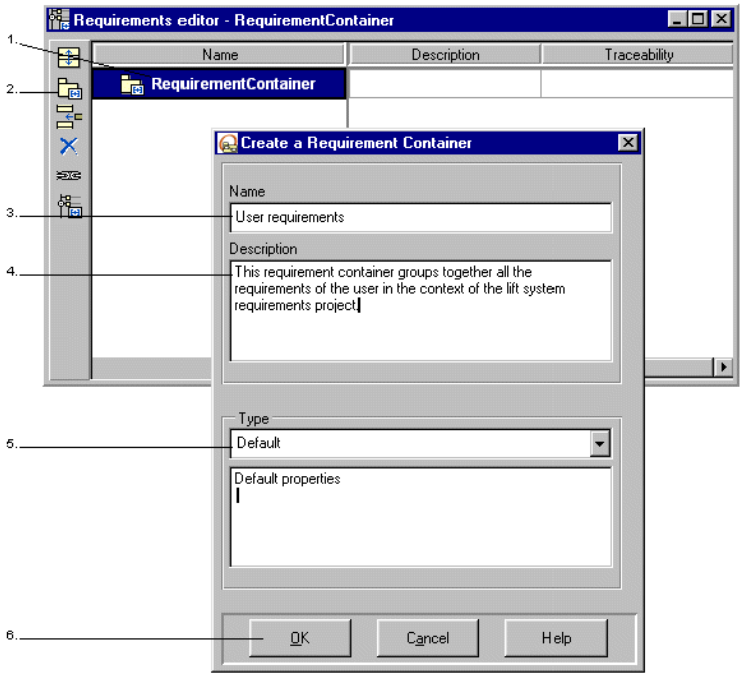


Figure 3-7. Creating a requirement container

Steps:

1 - In the requirements spreadsheet editor, click on the root requirement container.

2 - Click on the  "*Add a requirement container*" icon.

3 - Enter a name for your new requirement container, for example, "*User requirements*".

4 - Enter a description for the new requirements container (in our example, "*This requirement container groups together all the requirements of the user in the context of the lift system requirements project*").

5 - Select a type from the scrolling list. The types which appear in this list are the types provided by default, along with any other types you may have defined. We recommend that you always type new requirement containers during their creation, so as to be able to define your requirement containers and requirements in greater detail.

6 - Click on "*OK*" to confirm.

Note: A type is a set of properties which will be applied to the requirement container and all the requirements it contains. This means that for each property contained within the selected type, there will appear a column in the requirements spreadsheet editor and a corresponding field in the Objecteering/UML properties editor.

For further information on the creation of property sets, please refer to the "*Creating property sets and properties*" section in chapter 6 of this user guide.

The result of this operation in the requirements spreadsheet editor (which is opened by clicking on the  "*Open the requirements editor*" icon) is shown in Figure 3-8 below.

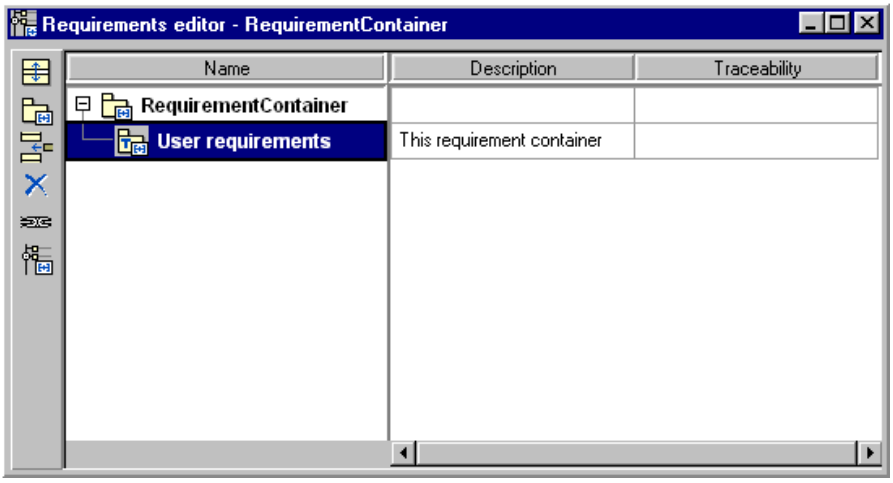


Figure 3-8. The requirements spreadsheet editor after the creation and typing of the "*User requirements*" requirement container

As you can see, the new requirement container which you typed appears in the requirements spreadsheet editor with the letter T over the requirement container icon.

Note: Requirement containers appear in **bold** in the requirements spreadsheet editor.

Continue by creating, in the requirements spreadsheet editor, the "*Lift system requirements*" requirement container, which has the following description: "*This requirement container groups together all the requirements of the lift system itself in the context of the lift system requirements project*". Select "*Functionality*" as its type.

The result of these operations is shown in Figure 3-9 below.

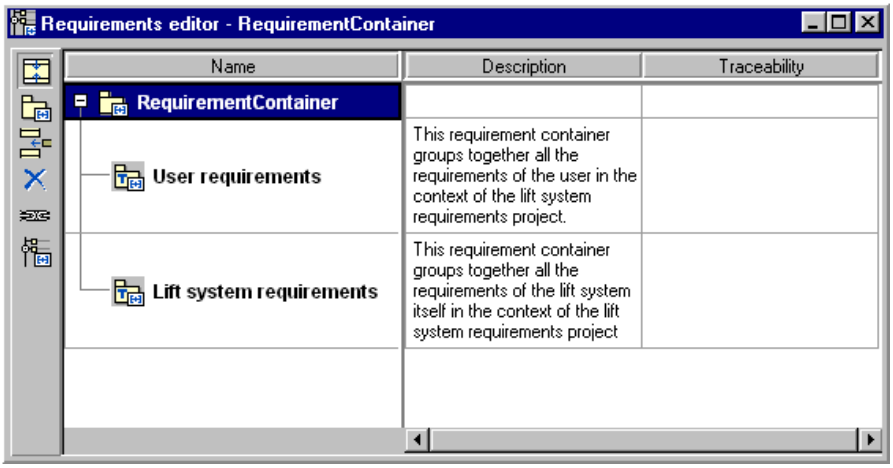


Figure 3-9. The requirement containers you have just created

Note: In Figure 3-9, the descriptions of the different requirement containers are seen in their entirety. This is because the requirements spreadsheet editor is in "*expand*" mode. In "*compact*" mode, only part of the description is visible. To change from compact mode to expand mode or vice versa, simply click on the "*Change spreadsheet editor mode to expand*" or "*Change spreadsheet editor mode to compact*" icons in the spreadsheet editor palette.

Don't forget that if you wish to work only in your dictionary spreadsheet editor, and have it as the sole window open, you can close the main explorer, properties editor and console, simply by clicking on the "*Show or hide the console*", "*Show or hide the main explorer*" or "*Show or hide the properties editor*" icons.

Defining requirement containers not created in the requirements spreadsheet editor

Had you decided to create a requirement container through the  tab of the explorer, further information, such as a description, could be defined for it in the following ways:

- directly in the requirements spreadsheet editor
- by double-clicking on the requirement container in the  tab of the explorer and entering definitions in the "*Modify*" window which then appears
- by right-clicking on the requirement container and choosing "*Modify*" from the context menu, which opens the requirement container window
- through the "*Requirements*" tab of the properties editor

In our example, we are going to enter information on our new requirement container through the "*Requirements*" tab of the properties editor (as shown in Figure 3-10).

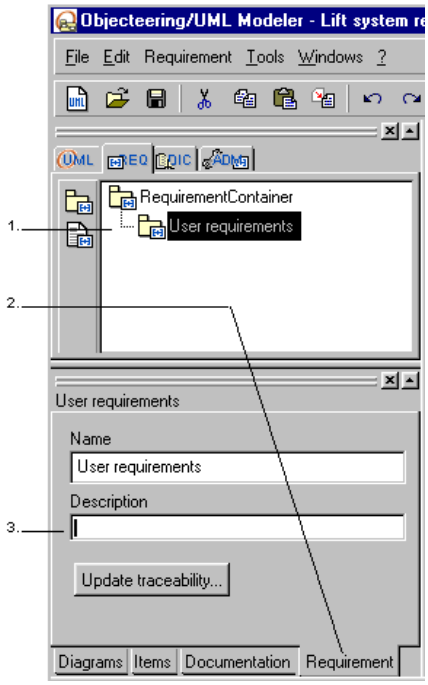


Figure 3-10. Defining a newly-created requirement container

Steps:

1 - Select the new requirement container in the REQ tab of the explorer.

2 - Select the "*Requirements*" tab in the properties editor.

3 - Enter a description of your new requirement container, for example, "*This requirement container groups together all the requirements of the user in the context of the lift system requirements project*".

Creating a requirement

Creating a requirement

We are now going to create our first requirements inside the requirement containers you have just created.

To create a requirement, simply carry out the steps shown in Figure 3-11 below.

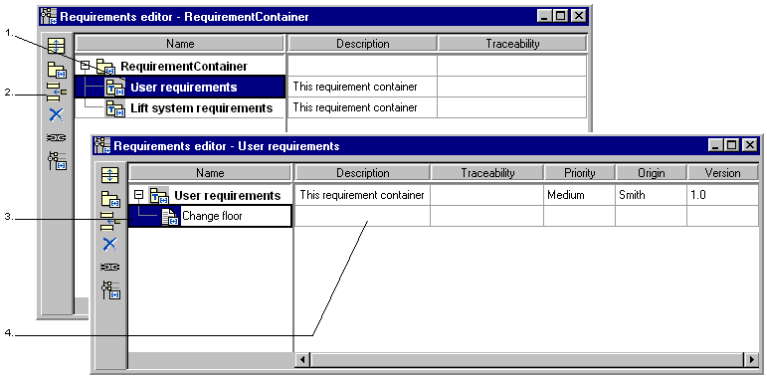


Figure 3-11. Creating a requirement

Steps:

1 - Click on the "*User requirements*" requirement container in the requirements spreadsheet editor.

2 - Click on the  "*Insert an element*" icon. A new requirements spreadsheet editor appears, containing various columns representing the properties defined in the property set with which you typed your requirement container.

3 - Enter a name for your requirement (in our example, "*Change floor*").

4 - Enter a description for your requirement (in our example, "*The user must be able to go up or down to another floor*").

Note: Requirements can also be created in the  tab of the explorer, by clicking on the  "*Create a requirement*" icon.

Continue by creating the following requirements in the "*User requirements*" requirement container:

- "*Be informed*": The user must be able to find out information on the current state of the lift system.
- "*Give orders*": The user must be able to give orders to the lift system.

In the "*Lift system requirements*" requirement container, create the following requirements:

- "*Transport user*": The lift system must be able to take the user where he wishes to go.
- "*Inform user*": The lift system must be able to communicate certain information to the user.
- "*Manage problems*": The lift system must be able to manage problems (for example, excess weight in the cabin or a power cut to the system).
- "*Manage risks*": The lift system must be able to manage situations which are potentially of risk to the user (for example, cabin doors open during transport).

The result of these operations is shown in Figure 3-12.

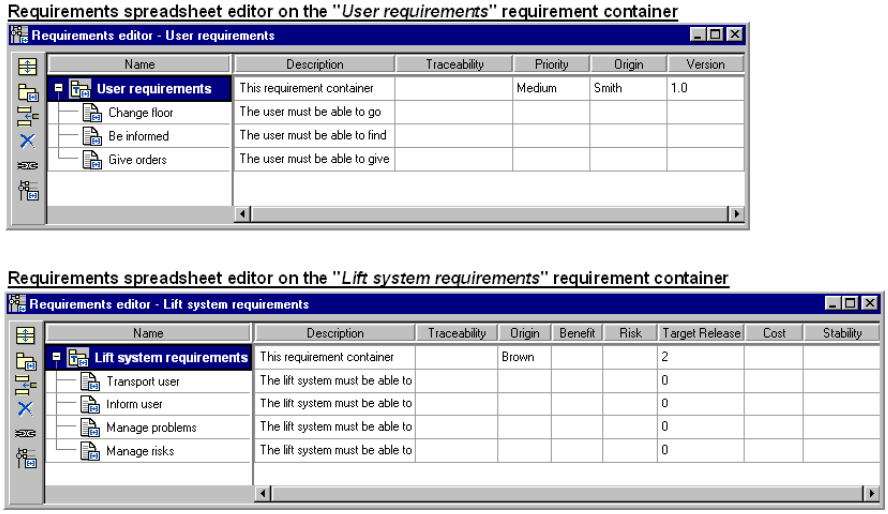


Figure 3-12. The requirements spreadsheet editors on two requirement containers typed with different property sets

As you can see, the columns appearing in the spreadsheet editor on the "*User requirements*" requirement container are different from those appearing in the spreadsheet editor on the "*Lift system requirements*" requirement container. This is because these two requirement containers were typed using different property sets (for further information on typing requirement containers, please refer to the "*Implementing property sets*" section in chapter 6 of this user guide).

Modeling the requirements project

Creating packages

We are now going to start building the model of the system which is to be developed.

As always, it is important to structure your model into coherent and logical groups. To this end, we are going to start by creating a package, "*Requirement model*", inside the "*Lift system requirements*" UML model root, which will include two new packages, "*Preliminary model*" and "*Use cases*", (as shown in Figure 3-13).

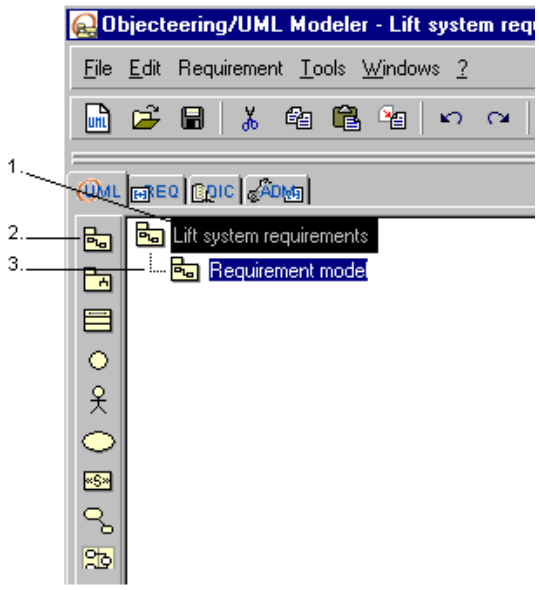


Figure 3-13. Creating the first package

Steps:

1 - In the  tab of the explorer, select the "*Lift system requirements*" root package.

2 - Click on the  "*Create a package*" icon.

3 - Enter a name for the newly created package, in our example, "*Requirement model*". If you press return, another package will be automatically created, through the continuous entry creation mode.

Continue by creating two packages inside the newly created "*Requirement model*" package, "*Preliminary model*" and "*Use cases*". The "*Preliminary model*" package will contain the actual UML model of the system to be developed, whilst the "*Use cases*" package will be used to contain the use cases developed in this context.

Creating use cases

Next, we are going to create a use case, used to express a use scenario related to the lift system which is to be developed in the context of our requirements project.

To create a use case, simply carry out the steps shown in Figure 3-14 below.

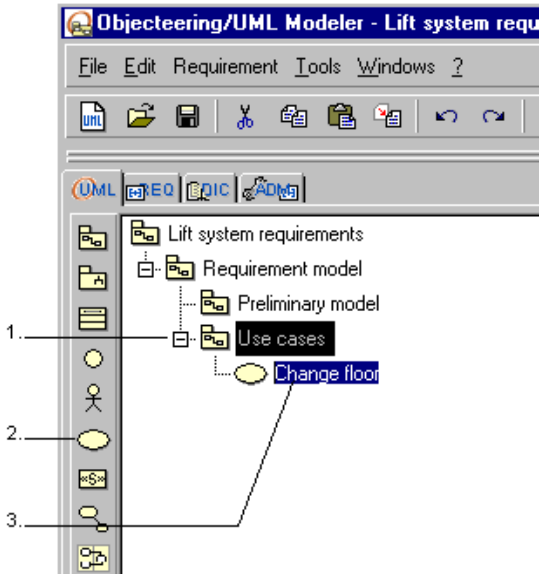


Figure 3-14. Creating a use case

Steps:

1 - In the UML tab of the explorer, click on the "*Use cases*" package.

2 - Click on the "*Create a use case*" icon.

3 - Give your use case a name, in our example, "*Change floor*".

Continue by creating the "*Manage the lift system*" use case in the "*Use cases*" package.

The result of these operations is shown in Figure 3-15 below.

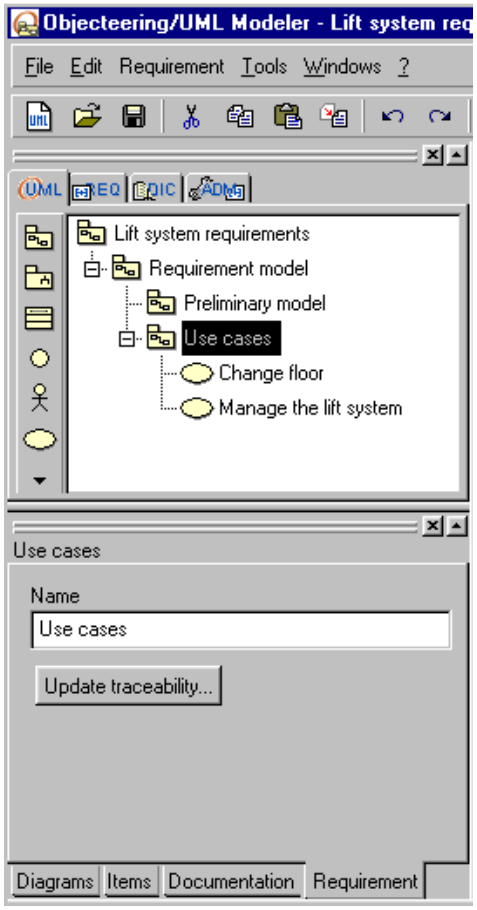


Figure 3-15. The newly created use cases

Creating model elements from spreadsheet editors

We are now going to create some model elements directly from the spreadsheet editors.

The first element we are going to create is the "*User*" actor, which will interact with the "*Change floor*" use case we have just created (Figure 3-16).

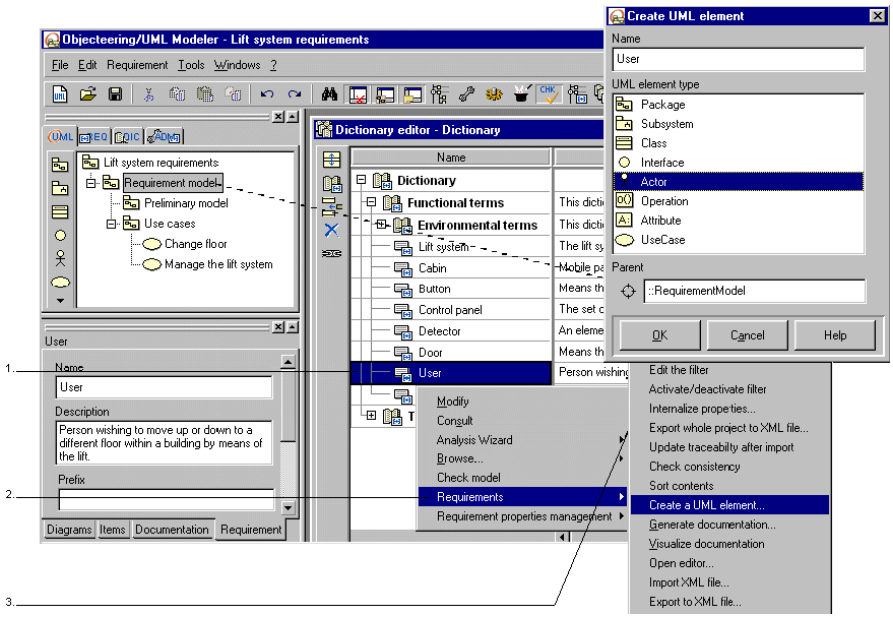


Figure 3-16. Creating the "*User*" actor from the dictionary spreadsheet editor

Steps:

1 - Right-click on the "*User*" term in the dictionary spreadsheet editor, to open the context menu.

2 - Run the "*Requirements/Create a UML element...*" command. The "*Create UML element*" window then appears.

3 - In the "*Create UML element*" window, enter a name ("*User*") and select a type ("*Actor*") for the element you wish to create. By default, the name of the original element in the spreadsheet editor is proposed (in this case, "*User*"). Finally, select the "*Requirement model*" package in the 

tab of the explorer and drag it into the "*Parent*" drop zone. Confirm by clicking on "*OK*".

The result of this operation is shown in Figure 3-17 below.

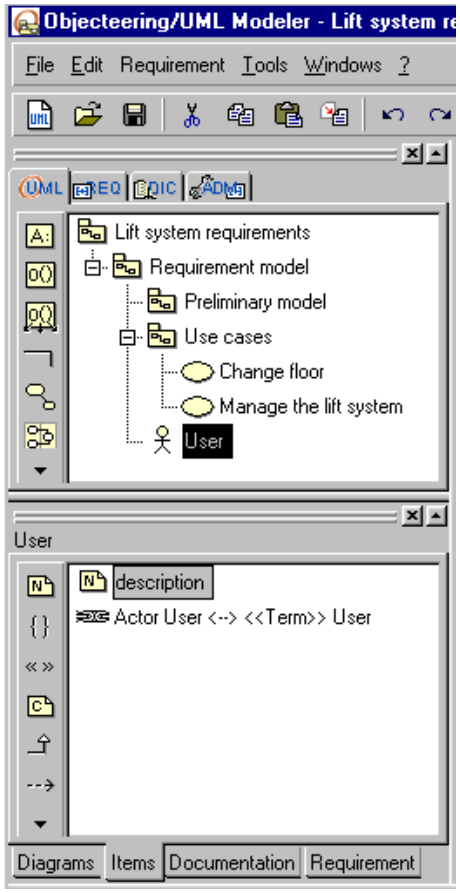


Figure 3-17. The newly created "*User*" actor in the UML tab of the explorer

In the "*Items*" tab, this new actor already has a description note and a traceability link to the "*User*" term in the dictionary, from which it was created. The content of the description note is the description entered for the "*User*" term in the dictionary spreadsheet editor.

Continue by creating the following UML elements from the dictionary spreadsheet editor:

- the "*Lift system supervisor*" actor from the "*Lift system supervisor*" term with "*Requirement model*" as its parent
- the "*Lift system*" package from the " *Lift system*" term with "*Preliminary model*" as its parent
- the "*Cabin*" class from the "*Cabin*" term with "*Lift system*" as its parent
- the "*Door*" class from the "*Door*" term with "*Lift system*" as its parent
- the "*Button*" class from the "*Button*" term with "*Lift system*" as its parent

---

Creating and working with a use case diagram

We are now going to create a use case diagram for the "*Requirement model*" package, by carrying out the steps illustrated in Figure 3-18 below.

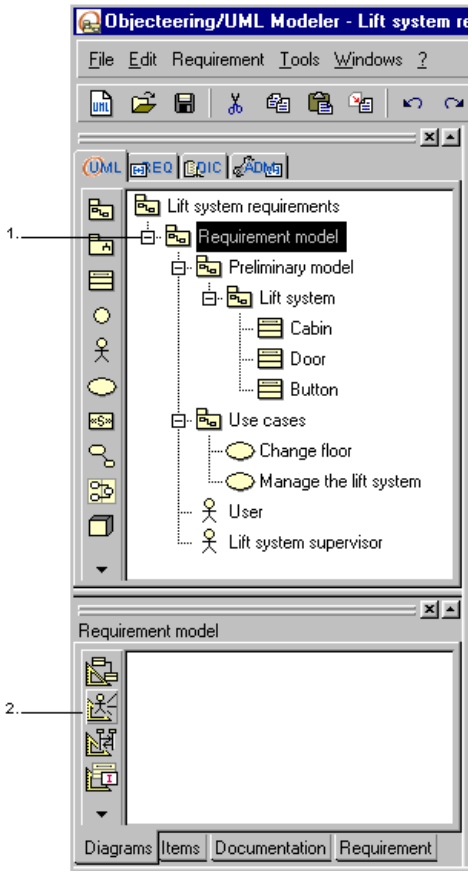


Figure 3-18. Creating a use case diagram

Steps:

1 - Select the "*Requirement model*" package in the UML tab of the explorer.

2 - In the "*Diagrams*" tab of the properties editor, click on the "*Create a use case diagram*" icon.

The new use case diagram then opens. In the use case diagram editor, we are now going to unmask the use cases and the actors which we created in the previous steps, before creating a communication link (Figure 3-19).

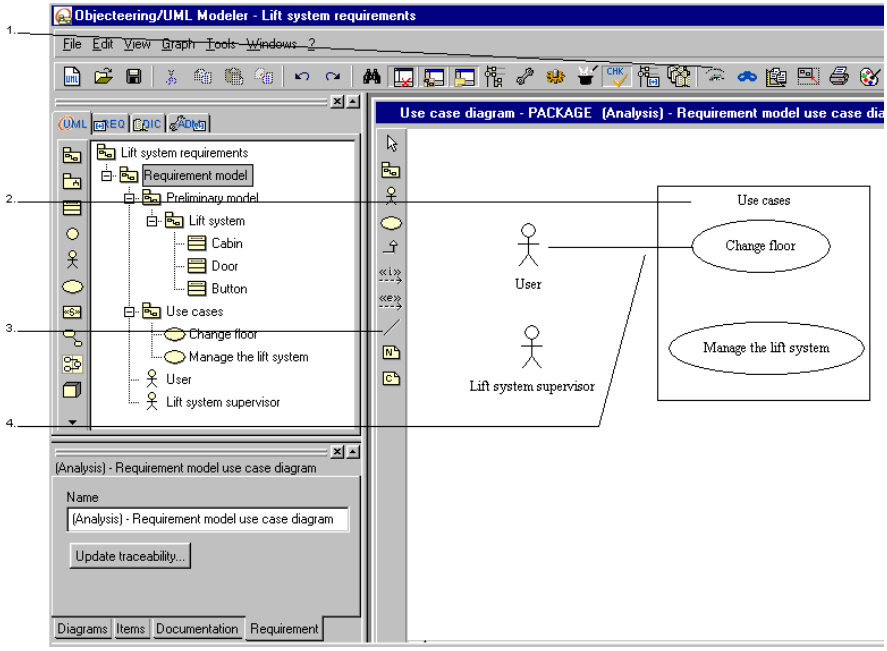


Figure 3-19. Working in the use case diagram graphic editor

Steps:

1 - In the use case diagram, click on the  "*Show contents*" icon. The "*Use cases*" requirement model, the "*User*" actor and the "*Lift system supervisor*" then appear (the "*Preliminary model*" requirement model and a description note also appear, but for the purposes of our example, we recommend that you remask them using the  "*Mask*" icon, so as not to clutter up the diagram).

2 - Click on the "*Use cases*" requirement model, and then once again on the  "*Show contents*" icon. The two requirement use cases you created during the previous steps then appear.

3 - Click on the  "*Create a communication link*" icon.

4 - Click on the "*User*" actor and then on the "*Change floor*" requirement use case. The new communication link then appears. Continue by creating another communication link between the "*Lift system supervisor*" actor and the "*Manage the lift system*" requirement use case.

The result of these operations is shown in Figure 3-20.

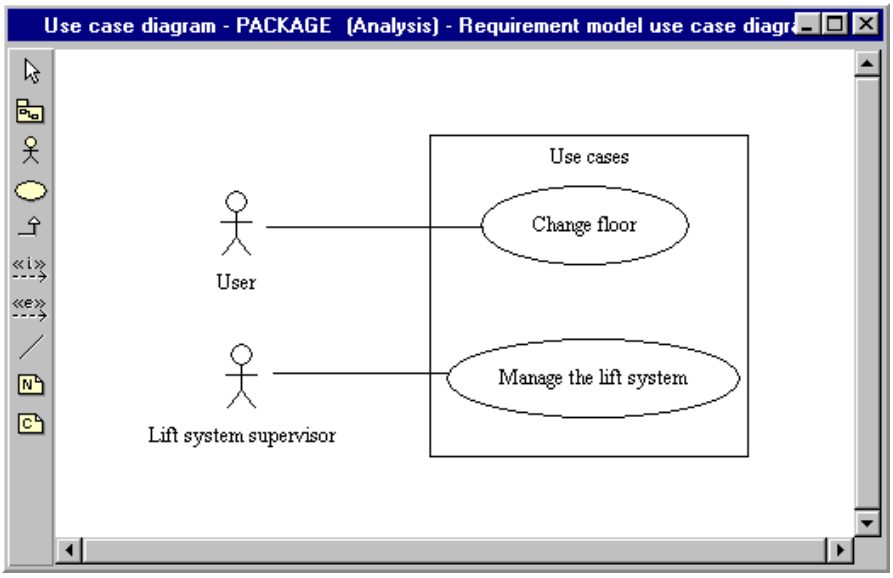


Figure 3-20. The use case diagram you have just developed

Creating a traceability link

We are now going to create a traceability link from one of the requirements we created in the requirements spreadsheet editor to one of the requirement use cases we created in the explorer (Figure 3-21).

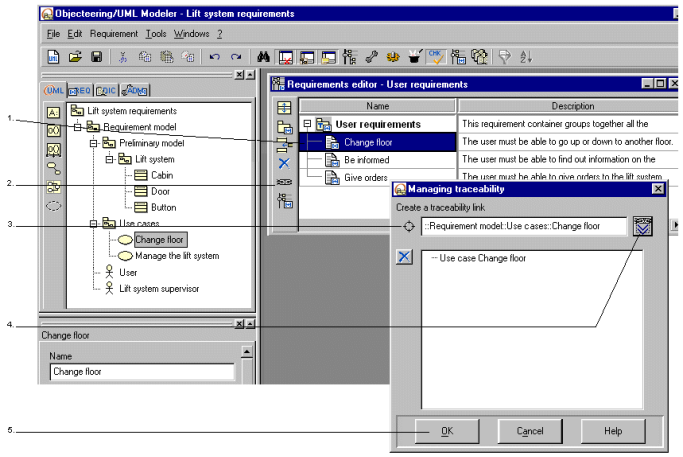


Figure 3-21. Creating a traceability link

Steps:

1 - In the requirements editor, click on the "*Change floor*" requirement.

2 - Click on the  "*Create a traceability link*" icon. The "*Managing traceability*" dialog box then appears.

3 - In the explorer, click on the "*Change floor*" requirement use case, and drag it into the drop zone of the "*Managing traceability*" window.

4 - Click on the  icon on the right-hand side of the window to confirm the creation of your new traceability link.

5 - Click on "*OK*" to confirm.

The result of this operation can be seen in:

- the "*Items*" tab of the properties editor
- the "*Requirements*" tab of the properties editor
- the requirements spreadsheet editor

Figure 3-22 shows the result of this operation in the "*Items*" tab of the properties editor.

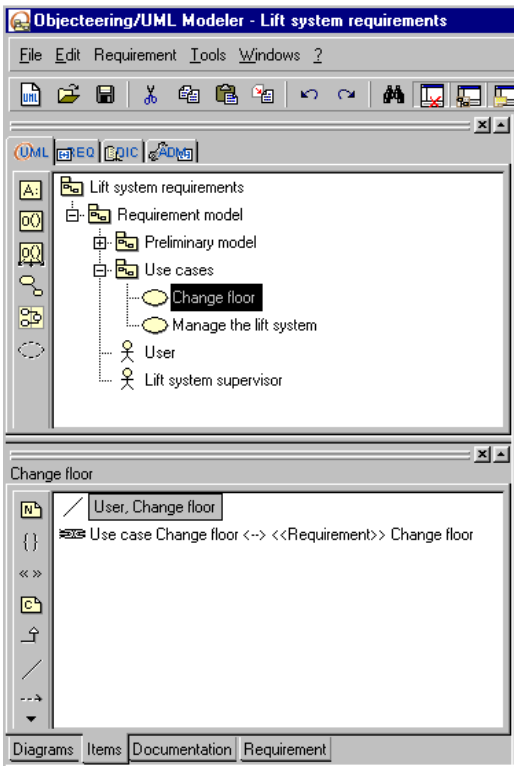


Figure 3-22. The traceability link in the "*Items*" tab of the properties editor on the "*Change floor*" use case

Figure 3-23 shows the result of this operation in the "*Requirements*" tab of the properties editor.

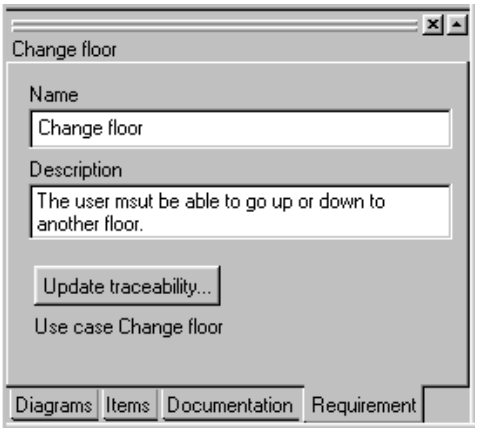


Figure 3-23. The traceability link in the "*Requirements*" tab of the properties editor on the "*Change floor*" requirement

Continue by creating the following traceability links:

- from the "*Manage problems*" requirement to the "*Manage the lift system*" use case
- from the "*Manage risks*" requirement to the "*Manage the lift system*" use case

Generating documentation

Generating requirements documentation

The final step in our first steps project is the generation of documentation on the requirements project you have just built.

To generate requirements documentation, simply carry out the steps shown in Figure 3-24 below.

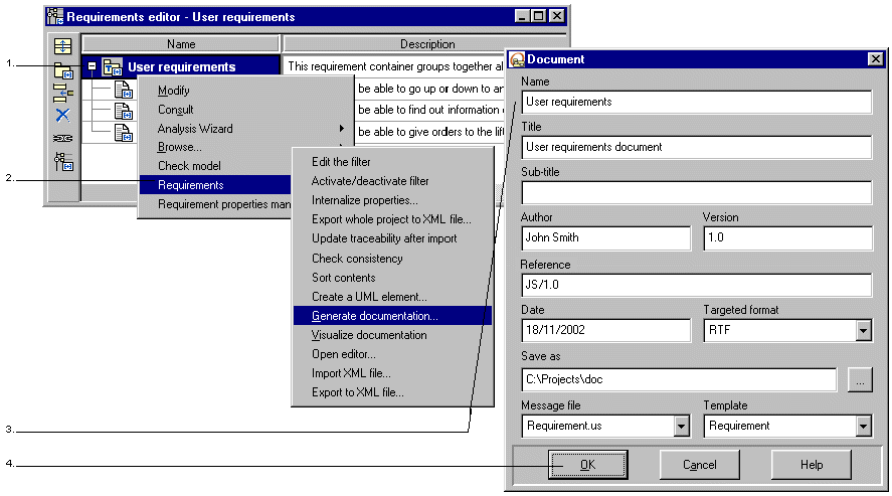


Figure 3-24. Generating requirements documentation

Steps:

1 - Right-click on the element for which you wish to generate documentation, in order to display the context menu.

2 - Run the "*Requirements/Generate documentation*" command. The "*Document*" window then appears.

3 - In the "*Document*" window, enter the necessary information (the name, title and subtitle of your document, the name of the author, the version number, the date, the desired format, the location where the file will be saved, the message file and the document template to be used). Remember to choose the "*Requirements*" document template if you are generating documentation for requirements and the "*Dictionary*" document template if you are generating documentation for a dictionary.

4 - Click on "*OK*" to confirm and to launch generation.

When you click on "*OK*" to confirm the information entered, generation is launched, and a document work product appears in the "*Items*" tab of the properties editor.

Note: Documentation can be generated in HTML or RTF format.

Chapter 4: Functions of Objecteering/UML Requirements

Overview of Objecteering/UML Requirements functions

Objecteering/UML Requirements provides a full range of services used to manage requirements, dictionaries, traceability and documentation, including services used to:

- create requirements and requirement containers
- create dictionaries and terms
- create traceability links
- create UML elements from the spreadsheet editors
- edit elements
- check elements
- sort elements
- use and manage filters
- generate requirements documentation
- generate dictionary documentation

For details on the specific commands provided, please see the "*Objecteering/UML Requirements commands*" section in the current chapter of this user guide.

Objecteering/UML Requirements commands

The "Requirements" context menu

The "*Requirements*" context menu, available by right-clicking on a requirement, a requirement container, a dictionary or a term in either an explorer, a requirements spreadsheet editor or a dictionary spreadsheet editor contains a number of commands used to run operations provided by the *Objecteering/UML Requirements* tool.

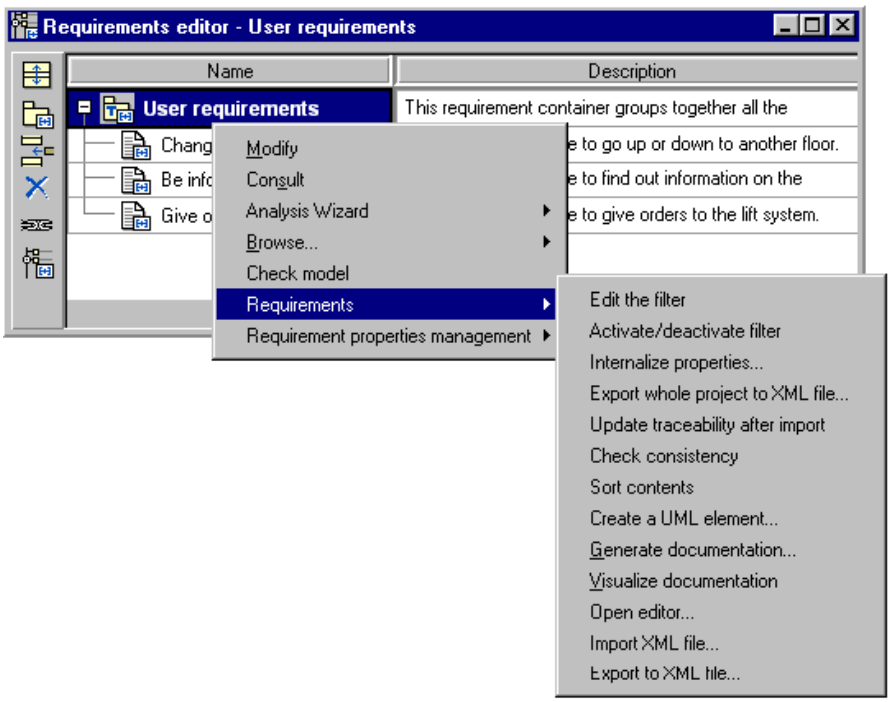


Figure 4-1. The "*Requirements*" context menu

Commands available

The ... command	is used to ...
Edit the filter	define filters for use in spreadsheet editors.
Activate/deactivate filter	activate or deactivate a defined filter.
Internalize properties	internalize default properties.
Export whole project to XML file	export the whole requirements project to an XML file.
Update traceability after import	update traceability links after successive import operations. For example, a part of a model containing traceability links is imported. Next, those missing elements involved in the traceability links are imported. This command can then be run to update the links previously broken.
Check consistency	launch consistency checks on the requirement, term, requirement container or dictionary in question (structuring units and uniqueness of names checked).
Sort contents	sort the requirements contained within a requirement container or the terms contained within a dictionary alphabetically.
Create a UML element	create a UML model element from the requirement, term, requirement container or dictionary in question. The element is created with a traceability link to the requirement, term, requirement container or dictionary, and has the same description note. You can define what kind of element you wish to create, and use the drag and drop function to designate its parent element.
Generate documentation	open the "*Document*" window, in which you can define the document you are going to create. Once you have defined all the necessary information, clicking on the "*OK*" button will launch the documentation generation operation.
Visualize documentation	visualize the documentation you have just generated using the "*Generate documentation*" command.

The ... command	is used to ...
Open editor	open the "*Requirement*" or "*Term*" dialog box when run on a requirement or a term, or open a new requirements or dictionary spreadsheet editor when run on a requirement container or dictionary (with the requirement container or dictionary in question as its root element).
Import XML file	import an XML file into your requirements project from an exterior source, for example, Microsoft Word. For further details on using Word with Objecteering/UML Requirements, please refer to chapter 7 of this user guide.
Export to XML file	export a particular element to an XML file.

For further information on these commands, please refer to the relevant sections in the current chapter of this user guide.

The DTD allowing the import and export of requirements is described in the "*DTD used in the import/export of requirements*" section of chapter 10 of this user guide.

Objecteering/UML Requirement properties management commands

The "Requirement properties management" context menu

The "*Requirement properties management*" context menu, available by right-clicking on an element in an explorer, contains a number of commands used to run operations provided by the *Objecteering/ReqPropertiesAdmin* tool.

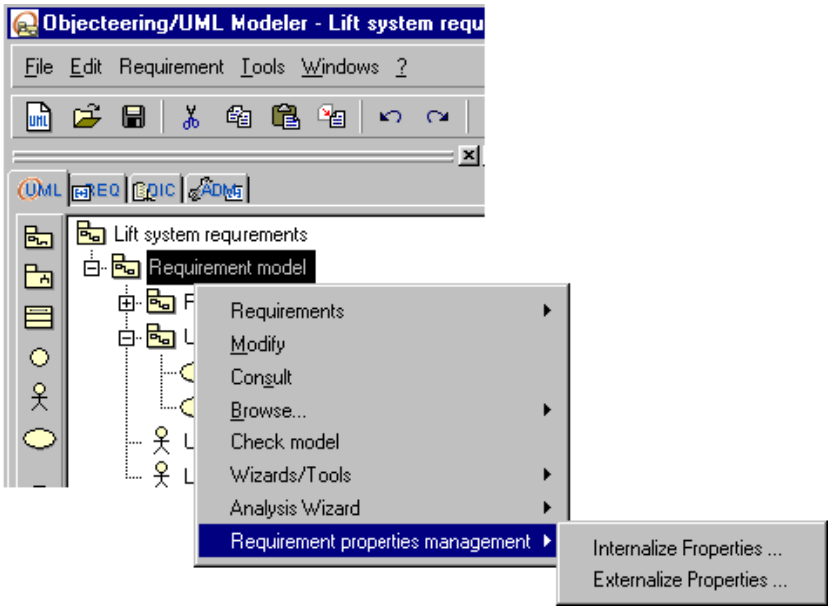


Figure 4-2. The "*Requirement properties management*" context menu

Commands available

The command ...	is used to ...
Externalize properties	externalize requirement properties you have developed, so as to make them available to other users.
Internalize properties	internalize requirement properties developed by a third party, so as to be able to use them in your project.

For further information on the commands provided in this context menu, please refer to chapter 6, "*Managing requirement properties*", of this user guide.

Creating requirement containers and requirements

Creating a new requirement container

Requirement containers, which are structuring units used to organize requirements into logical groups, can only be created inside other requirement containers.

Requirement containers can be created in the REQ tab of the explorer or in the requirements spreadsheet editor, using the "*Add a requirement container*" icon (Figure 4-3).

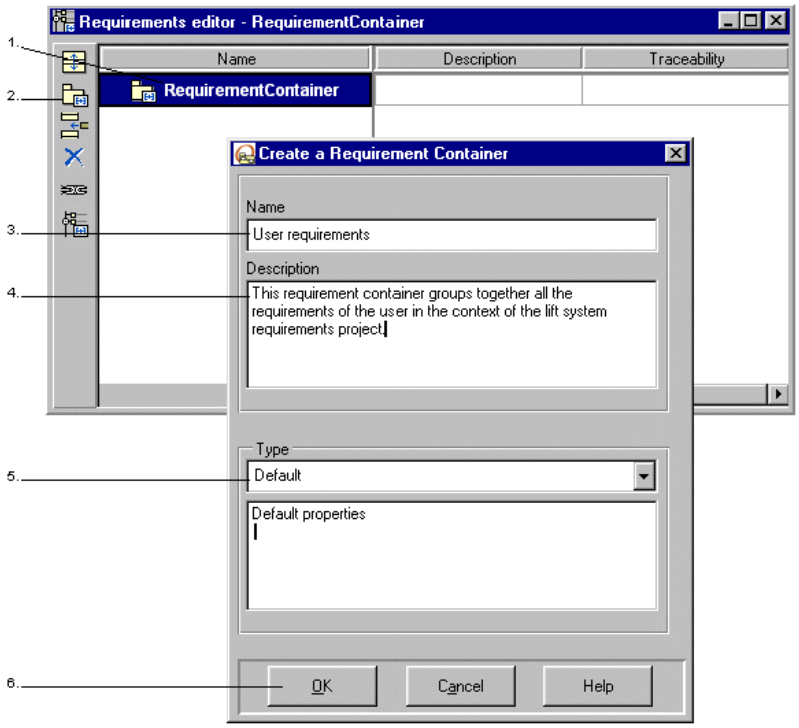


Figure 4-3. Creating a requirement container

Steps:

1 - In the requirements spreadsheet editor, click on the root requirement container.

2 - Click on the  "*Add a requirement container*" icon.

3 - Enter a name for your new requirement container.

4 - Enter a description for the new requirements container.

5 - Select a type from the scrolling list. The types which appear in this list are the types provided by default, along with any other types you may have defined. We recommend that you always type new requirement containers during their creation, so as to be able to define your requirement containers and requirements in greater detail.

6 - Click on "*OK*" to confirm.

Note: A type is a set of properties which will be applied to the requirement container and all the requirements it contains. This means that for each property contained within the selected type, there will appear a column in the requirements spreadsheet editor and a corresponding field in the Objecteering/UML properties editor.

For further information on the creation of property sets, please refer to the "*Creating property sets and properties*" section in chapter 6 of this user guide.

Note: Requirement containers appear in **bold** in the requirements spreadsheet editor, for greater visibility.

Creating a new requirement in the explorer

Requirements are created inside requirement containers, either in an explorer or in a requirements spreadsheet editor, using the  "*Create a requirement*" icon in the  tab of the explorer (Figure 4-4) or the  "*Insert an element*" icon in the requirements spreadsheet editor (Figure 4-5).

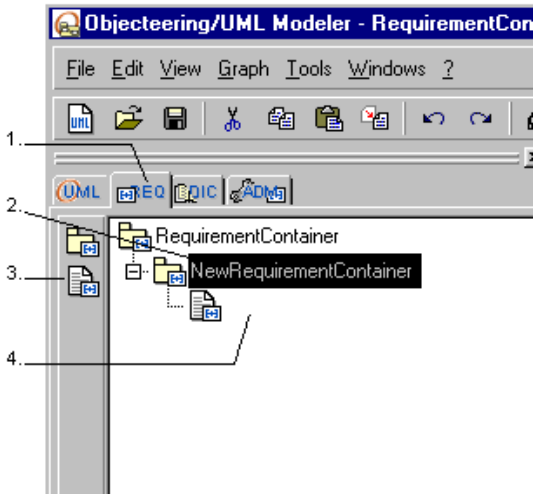


Figure 4-4. Creating a requirement in the explorer

Steps:

1 - Click on the  tab of the explorer. The requirements editor is then displayed.

2 - Select the requirement container inside which you wish to create the new requirement.

3 - Click on the  "*Create a requirement*" icon.

4 - Enter a name for the new requirement.

Creating a new requirement in the spreadsheet editor

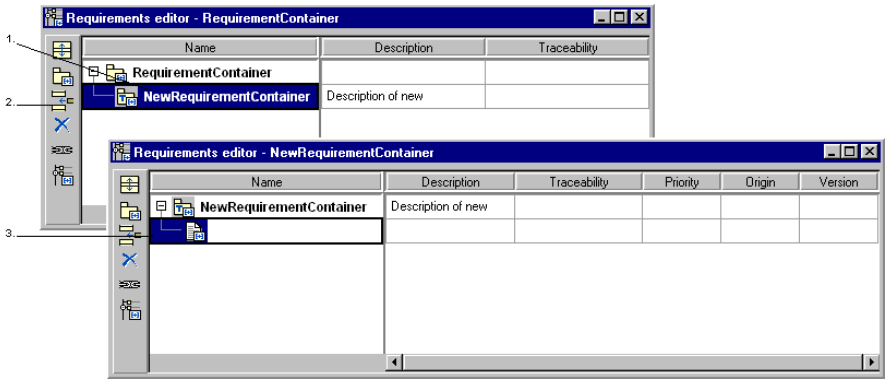


Figure 4-5. Creating a requirement in the requirements spreadsheet editor

Steps:

1 - Click on the requirement container inside which you wish to create the new requirement.

2 - Click on the  "*Insert an element*" icon. A new requirements spreadsheet editor is then opened on the requirement container in question.

3 - Enter a name for the new requirement.

Note: The requirements editor opened on the "*NewRequirementContainer*" requirement container contains various columns corresponding to the different properties contained in the property set with which the requirement container was typed.

Creating dictionaries and terms

Creating a new dictionary

Dictionaries, which are structuring units used to organize terms into logical groups, can only be created inside other dictionaries.

Dictionaries can be created in the DIC tab of the explorer or in the dictionary spreadsheet editor, using the "*Create a dictionary*" icon (Figure 4-6).

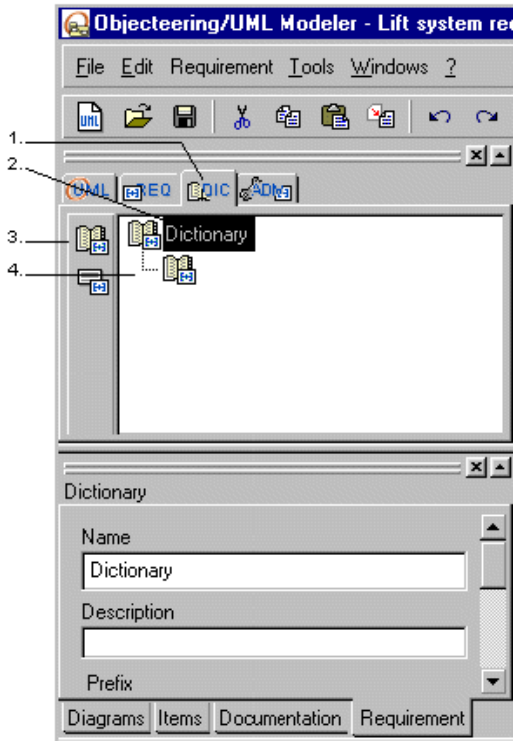


Figure 4-6. Creating a dictionary

Steps:

1 - Click on the  tab of the explorer. The dictionary explorer is then displayed.

2 - Select the root dictionary.

3 - Click on the  "*Create a dictionary*" icon.

4 - Enter a name for the new dictionary.

Creating a new term in the explorer

Terms are created either in an explorer or in a dictionary spreadsheet editor, using the "*Create a term*" icon in the tab of the explorer (Figure 4-7), or the "*Insert an element*" icon in the dictionary spreadsheet editor (Figure 4-8).

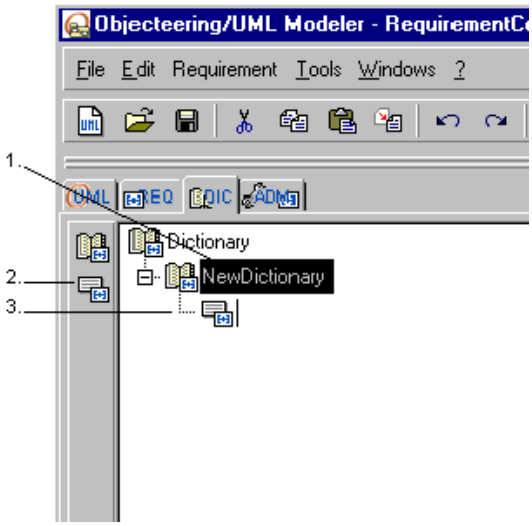


Figure 4-7. Creating a term in the explorer

Steps:

1 - In the tab of the explorer, select the dictionary inside which you wish to create a term.

2 - Click on the "*Create a term*".

3 - Enter a name for the new term.

Creating a new term in the spreadsheet editor

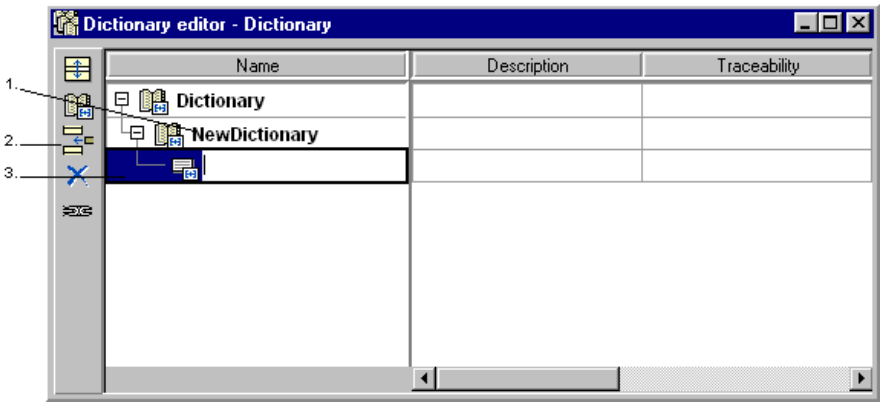


Figure 4-8. Creating a term in the dictionary spreadsheet editor

Steps:

1 - Click on the dictionary inside which you wish to create the new term.

2 - Click on the  "*Insert an element*" icon.

3 - Enter a name for the new term.

Creating traceability links

Introduction

Traceability links, which are used to track user needs throughout a project lifecycle, can be created on most model elements.

Traceability links can be created in the dictionary spreadsheet editor and therequirements spreadsheet editor, using the  "*Traceability*" icon (as shown in Figure 4-9).

Creating a traceability link from a spreadsheet editor

The procedure for creating traceability links from the dictionary or requirement spreadsheet editors is demonstrated in Figure 4-9 below.

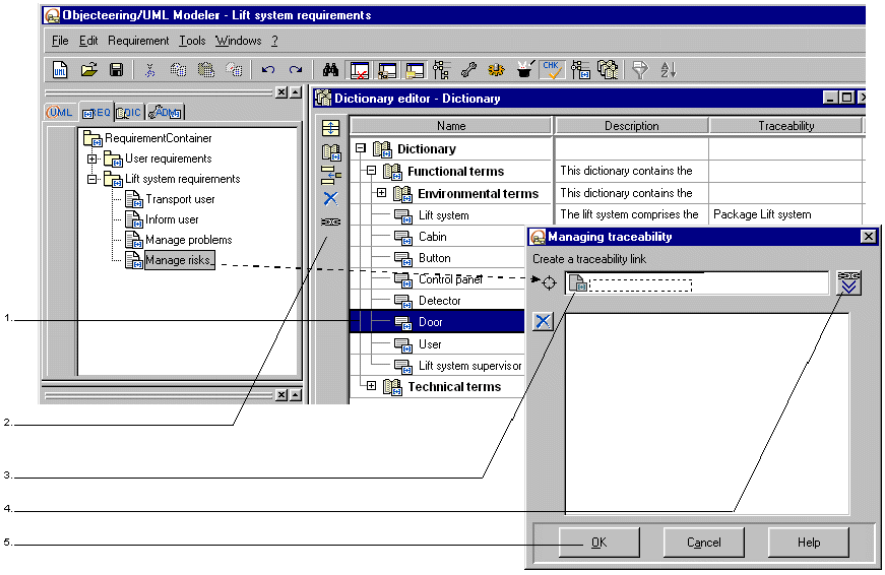


Figure 4-9. Creating a traceability link from the dictionary spreadsheet editor

Steps:

1 - In the dictionary spreadsheet editor, select the element for which you wish to create a traceability link. In our example, based on the *Objecteering/UML Requirements* first steps, we are going to create a traceability link from the "*Door*" term to the "*Manage risks*" requirement.

2 - Click on the  "*Traceability*" icon. The "*Managing traceability*" dialog box then appears.

3 - Select the "*Manage risks*" requirement in the  tab of the explorer by clicking on it, drag it to the "*Managing traceability dialog box*" and then drop it in the drop zone.

4 - Click on the  "*Add new traceability link*" icon to confirm the creation of the new link.

5 - Click on the "*OK*" button to confirm.

Creating UML elements from the spreadsheet editors

The "Create a UML element" command

During the course of his work, a developer may wish to create model elements directly from his requirements and terms. To this end, the *Objecteering/UML Requirements* tool provides the "*Create a UML element*" command.

The UML elements created using this command automatically bear a traceability link to the base element, as well as the same description note.

To create a UML element from a requirement, requirement container, dictionary or term in a requirements spreadsheet editor or a dictionary spreadsheet editor, simply carry out the steps indicated in Figure 4-10 below.

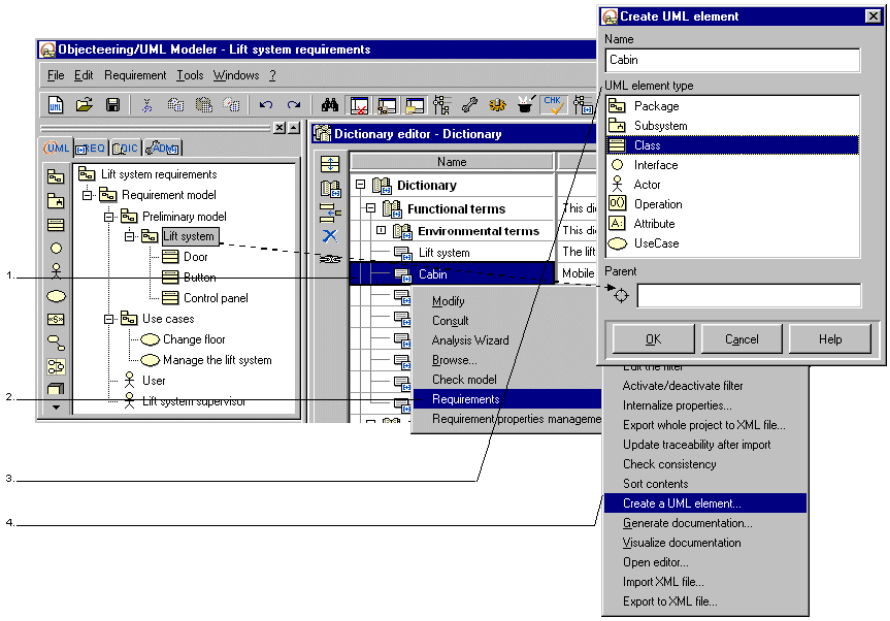


Figure 4-10. Creating a UML element from a term

Steps:

1 - In the dictionary spreadsheet editor, right-click on the dictionary or term from which you wish to create the new UML element. The context menu then appears. In our example, we are going to create a class called "*Cabin*" inside the "*Lift system*" package from the "*Cabin*" term.

2 - Run the "*Requirements/Create a UML element*" command. The "*Create UML element*" window then appears.

3 - In the "*Create UML element*" window, enter a name for the element which is being created. By default, the name of the base element (in this case, "*Cabin*") is proposed. Next, select the type of element you wish to create (in our example, a class). Finally, drag the parent element (in our case, the "*Lift system*" package) from the explorer and drop it in the "*Parent*" drop zone of the "*Create UML element*" window.

4 - Confirm by clicking on "*OK*".

The result of this operation can be seen in Figure 4-11 below.

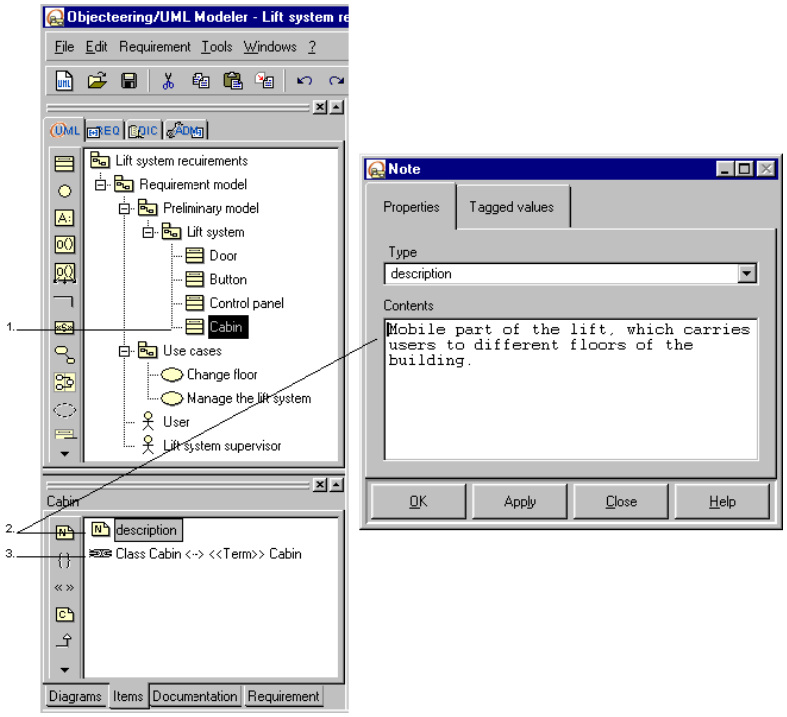


Figure 4-11. The newly created UML element

Key:

1 - The newly created class inside the "*Lift system*" package in the explorer.

2 - The description note and its contents, which are the same as those of the base element.

3 - The automatically created traceability link to the base element.

Editing elements

The "Open editor" command on a requirement or a term

If a user wishes to modify a requirement or a term, he can do so by running the "*Requirements/Open editor*" command.

Note: The  "*Open editor*" icon in the spreadsheet editor runs the same operation.

In the case of a requirement (as shown in Figure 4-12), this command opens the "*Requirement*" window, whilst for a term, the "*Term*" window appears.

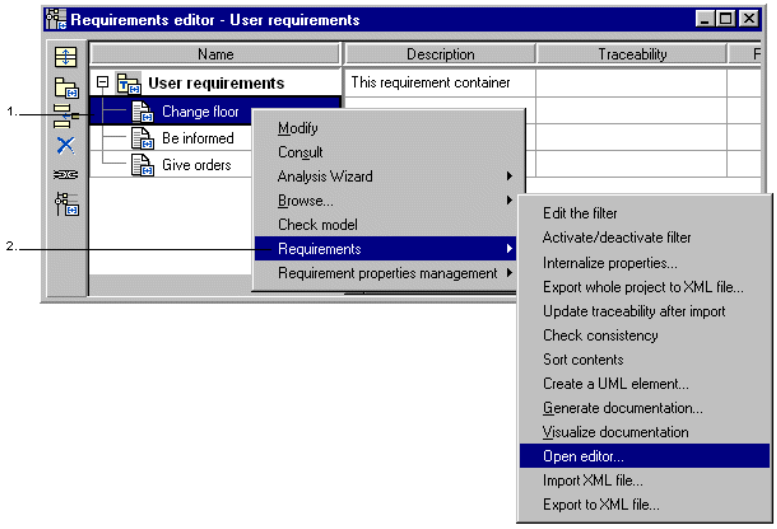


Figure 4-12. Running the "*Open editor*" command on a requirement

Steps:

1 - In the requirements spreadsheet editor, right-click on the requirement you wish to edit, in order to display the context menu.

2 - Run the "*Requirements/Open editor*" command.

The result of this operation on a requirement is shown in Figure 4-13.

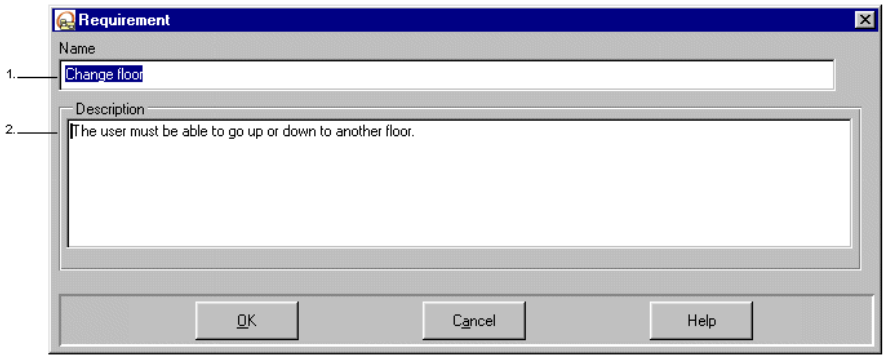


Figure 4-13. The "*Requirement*" window

Key:

1 - The name of the requirement.

2 - The description of the requirement.

Note: The fields which figure in this window vary according to the property set used to type the parent requirement container.

The user should simply make the necessary modifications, and confirm by clicking "*OK*".

When the "*Open editor*" command is run on a term (in exactly the same way as for a requirement), the "*Term*" window (Figure 4-14) appears.

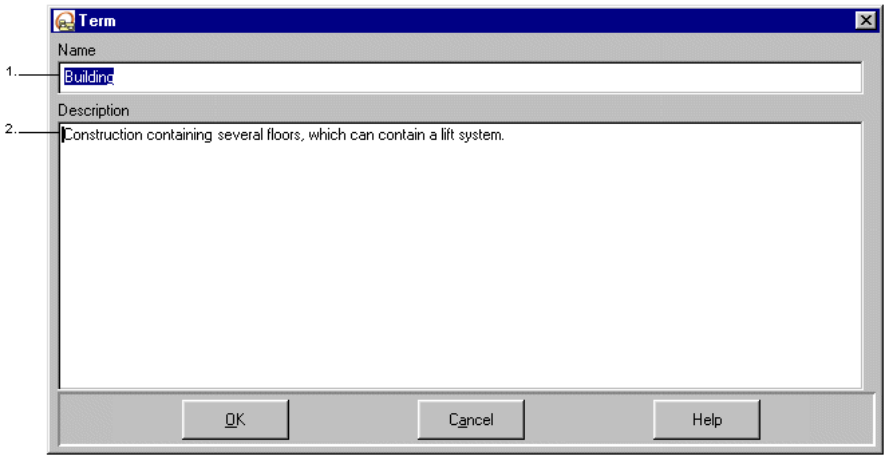


Figure 4-14. The "*Term*" window

Key:

1 - The name of the term.

2 - The description of the term.

The "Open editor" command on a requirement container or a dictionary

When the "*Requirements/Open editor*" command is run on a requirement container or a dictionary, a new requirements spreadsheet editor or dictionary spreadsheet editor is opened on the requirement container or dictionary in question (Figure 4-15).

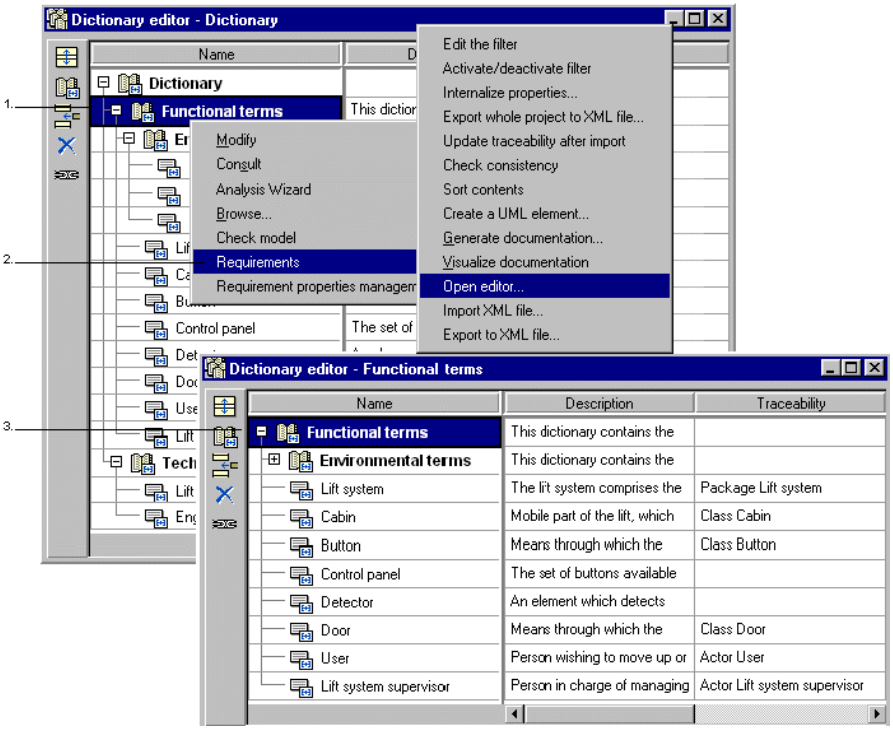


Figure 4-15. Running the "*Open editor*" command on a dictionary

Steps:

1 - In the dictionary spreadsheet editor, right-click on the dictionary you wish to edit, in order to display the context menu.

2 - Run the "*Requirements/Open editor*" command.

3 - A new dictionary editor is opened, with the element you selected for edition as its root.

Note: When this operation is run on a requirement container, a new requirements spreadsheet editor is opened, with the selected requirement container as its root.

Checking elements

The "Check consistency" command

The "*Requirements/Check consistency*" command is used to run checks on the selected requirement container, requirement, dictionary or term.

This command checks that element names are unique, and that elements are contained within the correct type of structuring unit (requirement containers for requirements, and dictionaries for terms).

To run checks, simply carry out the steps shown in Figure 4-16.

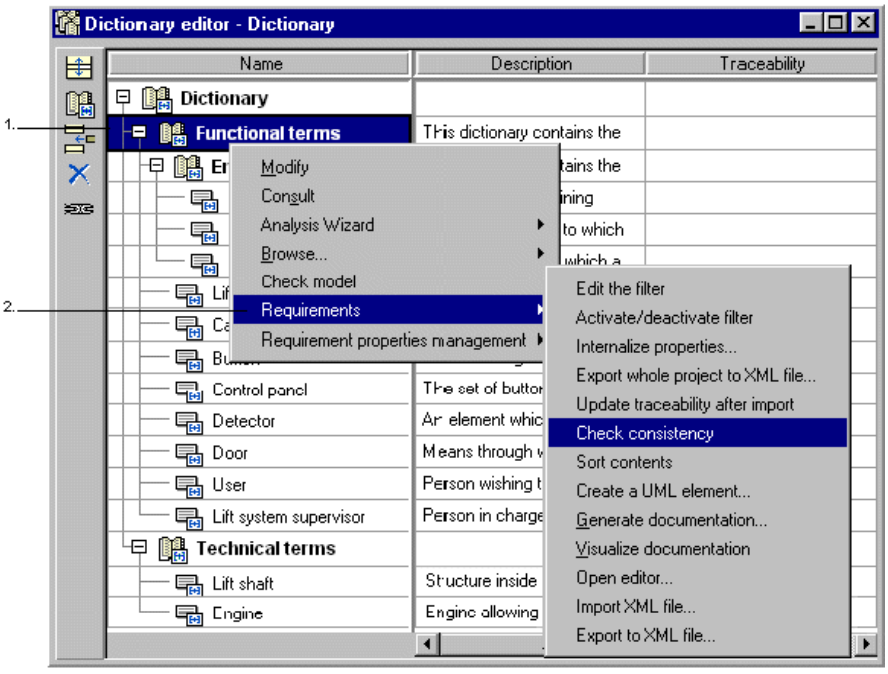


Figure 4-16. Running the "*Check consistency*" command on a dictionary

Steps:

1 - Right-click on the element you wish to check in the relevant spreadsheet editor, in order to display the context menu.

2 - Run the "*Requirements/Check consistency*" command.

If errors are detected, the "*Consistency check report*" window appears, providing details of the problem.

In our example, before running checks, we created two terms within the "*Functional terms*" dictionary, both called "*Cabin*". When the "*Requirements/Check consistency*" command was run, this fact was detected and the user informed (as shown in Figure 4-17).

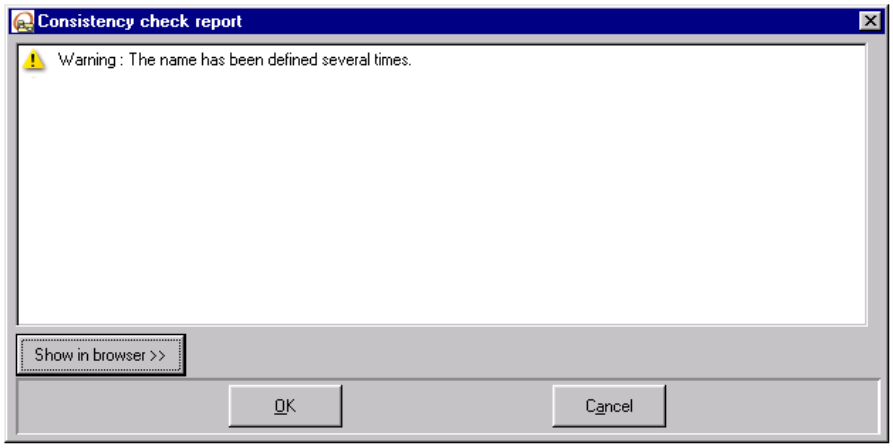


Figure 4-17. The "*Consistency check report*" window, showing errors detected after running the "*Requirements/Check consistency*" command

The "*Show in browser*" field is used to jump to the element concerned in the explorer.

Sorting elements

The "Sort contents" command

The "*Requirements/Sort contents*" command is used to sort the requirements or terms contained within a requirement container or a dictionary alphabetically (Figure 4-18).

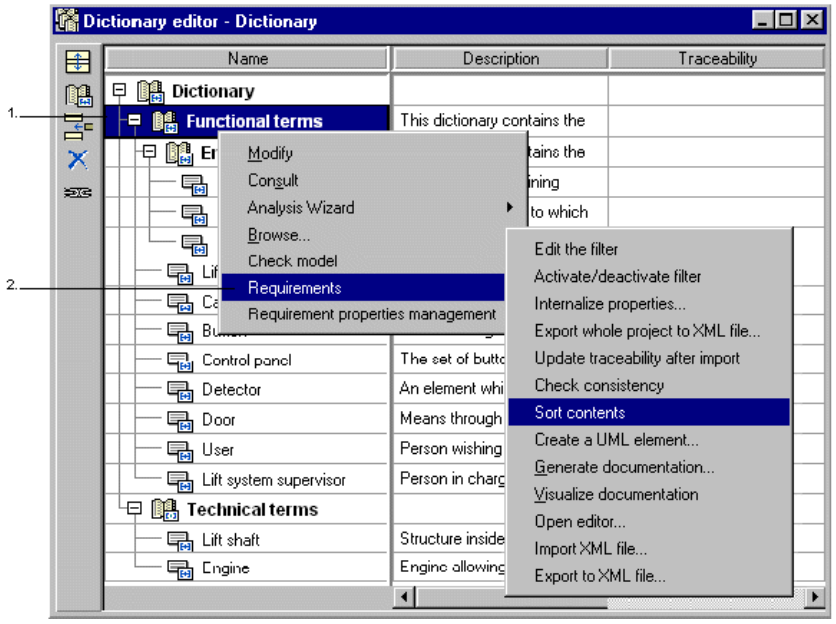


Figure 4-18. Running the "*Sort contents*" command

Steps:

1 - Right-click on the requirement container or dictionary whose contents you wish to sort alphabetically (in our example, we have selected the "*Functional terms*" dictionary), in order to display the context menu.

2 - Run the "*Requirements/Sort contents*" command.

The result of this operation is shown in Figure 4-19.

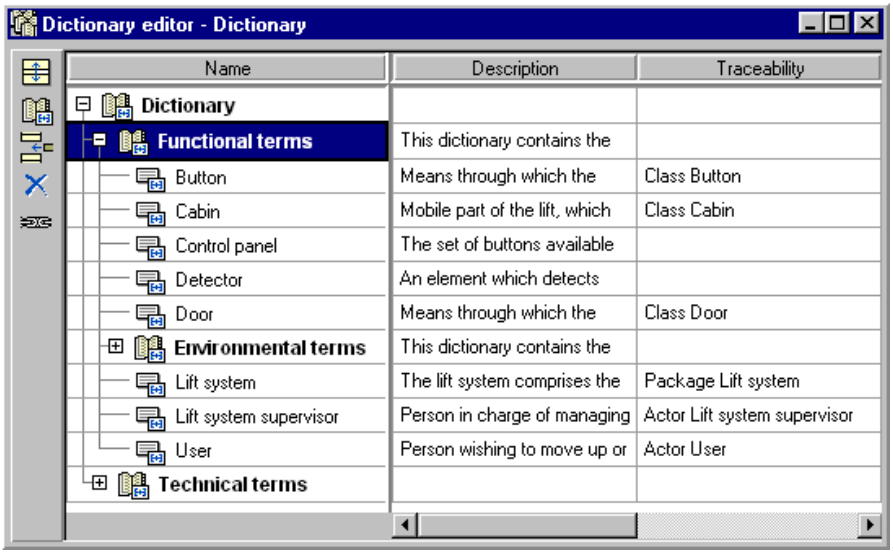


Figure 4-19. The terms in the dictionary have been sorted alphabetically

Filtering elements

Overview of the filter function

The "*Requirement/Filter...*" and "*Edit the filter*" commands can both be used to filter elements present in a spreadsheet editor according to an existing property, thereby improving visibility.

Opening the "Edit the filter" window

To open the window in which you can create and edit filters, there are two possibilities:

- by clicking on the  "*Activate/Deactivate filter*" icon in the tool bar
- by clicking on the "*Requirement/Filter...*" menu in the menu bar
- by clicking on the "*Requirement/Edit the filter*" command in the context menu available in a dictionary spreadsheet editor or a requirement spreadsheet editor
- by clicking on the "*Requirement/Activate/deactivate filter*" command in the context menu available in a dictionary spreadsheet editor or a requirement spreadsheet editor

Applying a filter

In the example shown in Figure 4-20, we are going to create a filter run on the names of dictionaries and terms in the dictionary spreadsheet editor. We are going to specify that we only wish to see dictionaries and terms whose names contain "*lift*".

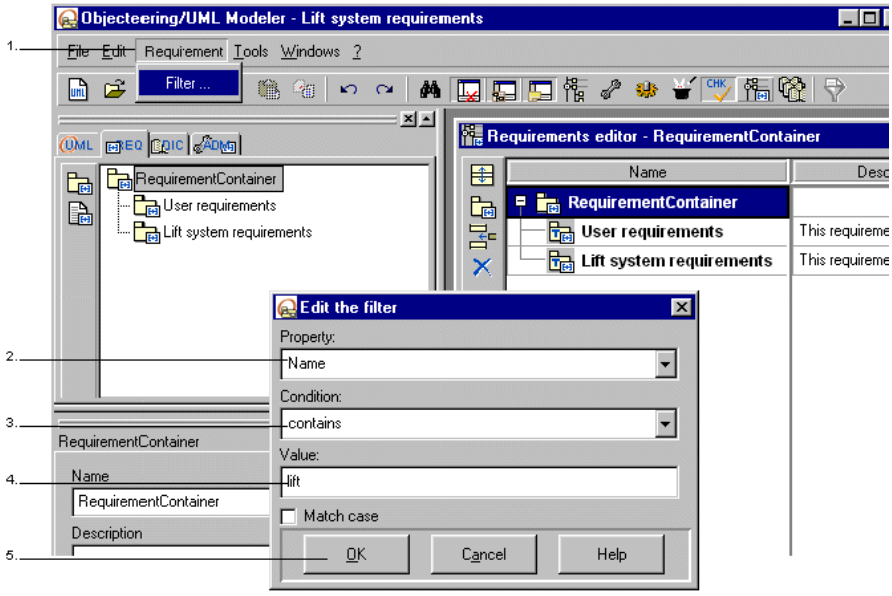


Figure 4-20. Running the "*Requirement/Filter...*" command

Steps:

1 - In the Objecteering/UML menu bar, select "*Requirement/Filter...*". The "*Edit the filter*" window then appears.

2 - In the "*Property*" field, select the property which you want to use as the basis for your new filter.

3 - In the "*Condition*" field, select the expression which fits what you want the filter to do (in our example, "*contains*").

4 - In the "*Value*" field, enter the value to be used with the condition in the filter.

5 - Click on "*OK*" to confirm and apply the new filter. Once the filter has been applied, the  button appears pressed down.

Note: If you check the "*Match case*" tickbox, then upper and lower case will be respected by the filter.

The result of this operation is shown in Figure 4-21 below.

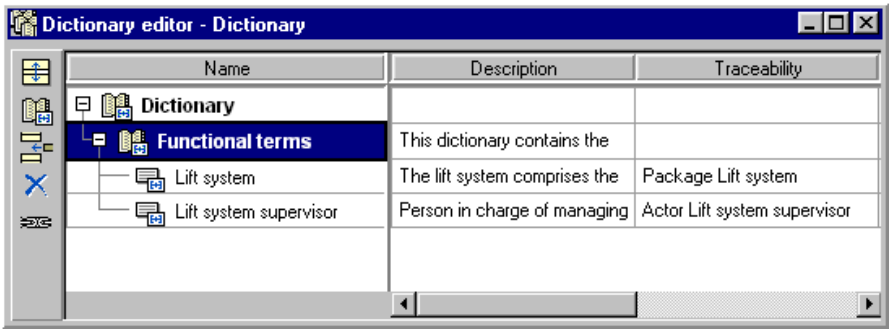


Figure 4-21. The same dictionary editor after filtering

As can be seen in Figure 4-21, the dictionary editor now only shows dictionaries and terms whose names contain the character string "*lift*".

Possible filter conditions

The conditions which can be used when creating or editing filters are as follows:

- contains
- is equal to
- is different from
- is greater than
- is greater than or equal to
- is less than
- is less than or equal to
- is empty
- is not empty

Note: Elements are filtered according to a specified property. In the "*Property*" field of the "*Edit the filter*" window, you can choose between the various properties provided by the property set implemented.

Deactivating a filter

To deactivate a filter, simply click on the  icon in the Objecteering/UML toolbar (which appears pressed down, to indicate that a filter has been applied and is active). The  icon, which appears non-pressed down, then appears in the toolbar, to indicate that no filter is currently active.

Alternatively, deactivate the filter through the "*Requirements/Activate/deactivate filter*" command in the context menu available in the dictionary and requirements spreadsheet editor.

Reactivating a filter

Similarly, to reactivate the filter, click on the  icon. The "*Requirements/Activate/deactivate the filter*" command can also be used.

The "*Edit the filter*" window which then appears displays the parameters of the previous filter to be applied. These can either be reused or modified.

Note: The same operation can be carried out using the "*Edit the filter*" command in the context menu available from the requirements spreadsheet editor and the dictionary spreadsheet editor.

Generating documentation

The "Generate documentation" command

The "*Requirements/Generate documentation*" command is used to generate documentation on requirements or dictionaries.

Documentation can be generated in HTML or RTF format.

As with standard documentation generation, a document work product is created on the element, before the generation process itself is launched.

Note: Before generating requirements or dictionary documentation, you must first set certain module parameters, such as the documentation generation root directory and the editors you are going to use. For further information, please see chapter 9, "*Parameterizing Objecteering/UML Requirements*", of this user guide.

Generating requirements documentation

To generate requirements documentation, carry out the steps shown in Figure 4-22.

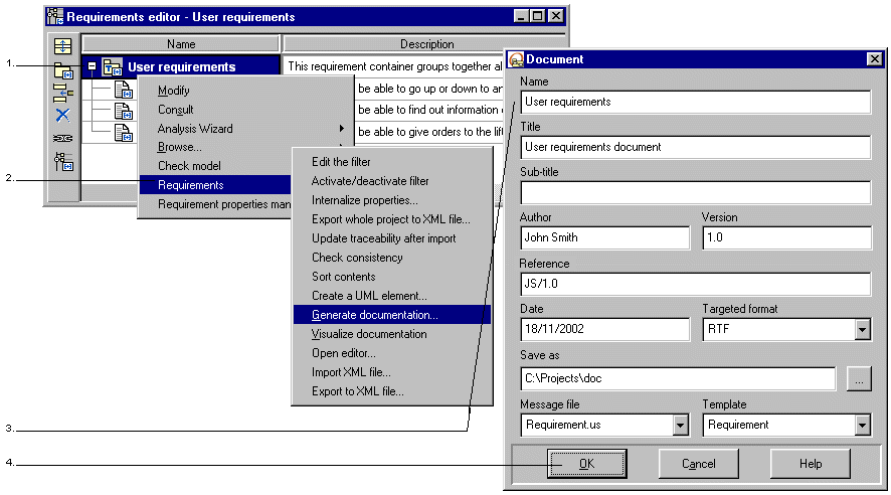


Figure 4-22. Generating documentation for requirements

Steps:

1 - Right-click on the requirement container for which you wish to generate documentation, in order to display the context menu.

2 - Run the "*Requirements/Generate documentation*" command. The "*Document*" window then appears.

3 - In the "*Document*" window, enter the necessary information (the name, title and subtitle of your document, the name of the author, the version number, the date, the desired format, the message file and the document template to be used). Remember to choose the "*Requirements*" document template if you are generating documentation for requirements!

4 - Click on "*OK*" to confirm and to launch generation.

The result of this operation is shown in Figure 4-23.

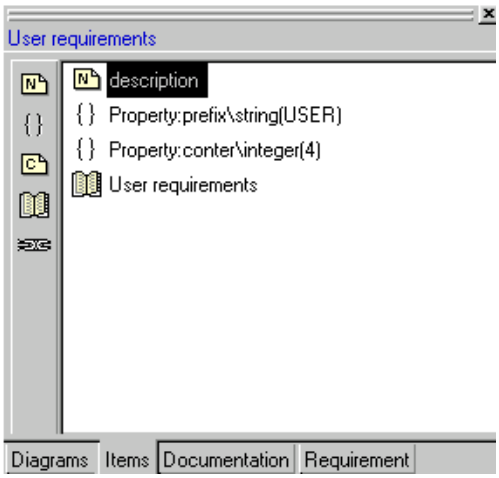


Figure 4-23. The newly created document in the "*Items*" tab of the properties editor

Generating dictionary documentation

The procedure for generating dictionary is very similar to that used to generate requirements documentation (see Figure 4-24).

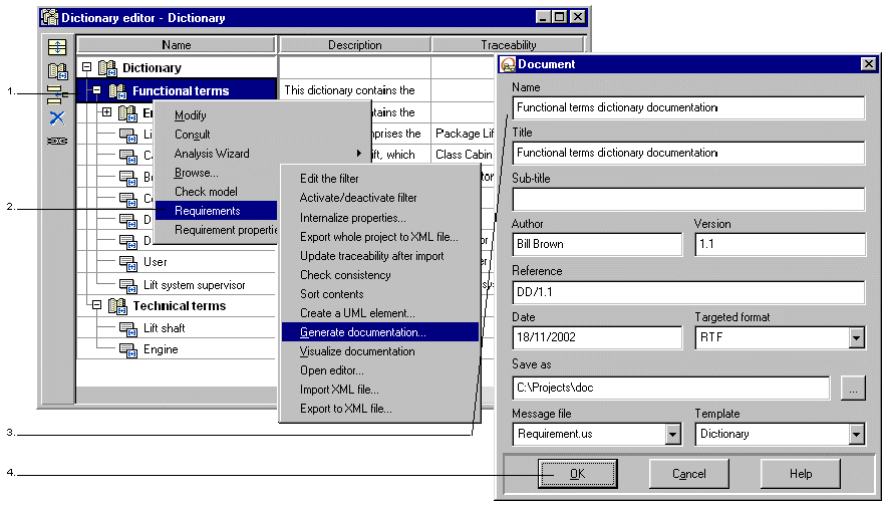


Figure 4-24. Generating documentation for a dictionary

Steps:

1 - Right-click on the dictionary for which you wish to generate documentation, in order to display the context menu.

2 - Run the "*Requirements/Generate documentation*" command. The "*Document*" window then appears.

3 - In the "*Document*" window, enter the necessary information (the name, title and subtitle of your document, the name of the author, the version number, the date, the desired format, the message file and the document template to be used). Remember to choose the "*Dictionary*" document template if you are generating documentation for a dictionary!

4 - Click on "*OK*" to confirm and to launch generation.

Chapter 5: The user interface

Overview of the Objecteering/UML Requirements user interface

Introduction

When the *Objecteering/UML Requirements* tool is selected, the dictionary spreadsheet editor and the requirements spreadsheet editor appear, in addition to the classic Objecteering/UML windows (as shown in Figure 5-1).

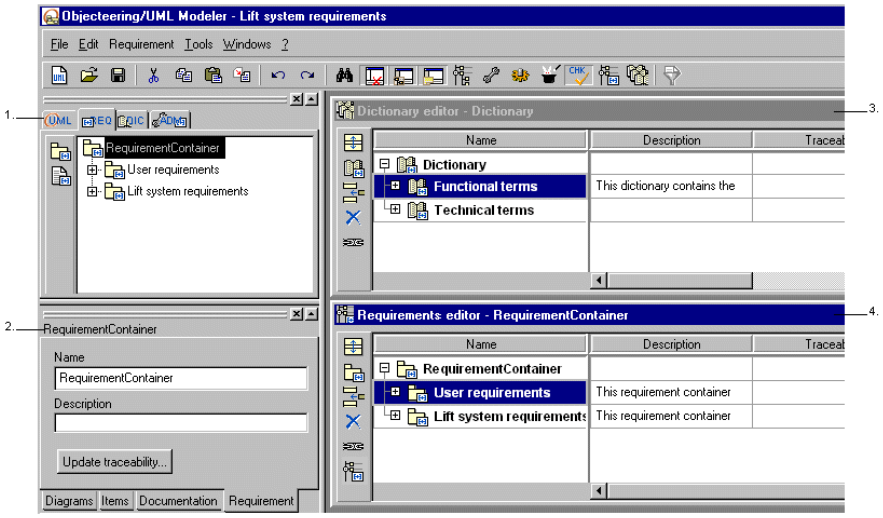


Figure 5-1. The Objecteering/UML user interface for working with requirements

Key:

1 - The explorer. When the *Objecteering/UML Requirements* tool is selected, a set of tabs used to browse between the different explorer views becomes available to the user. The  tab displays the classic UML model element explorer, the  tab displays the Requirements explorer, the  tab displays the Dictionary explorer and the  tab displays the Requirement Properties explorer. For further general information on the explorer, please refer to the "*The explorer*" section in chapter 3 of the *Objecteering/UML Modeler* user guide. For further information on the Requirement properties explorer, please refer to chapter 6 of this user guide.

2 - The properties editor. Figure 4-1 shows the properties editor with the "*Requirements*" tab selected. For further information on the properties editor for the *Objecteering/UML Requirements* tool, please refer to the "*The Objecteering/UML properties editor*" section in the current chapter of this user guide.

3 - The requirements spreadsheet editor. For further information on the requirements spreadsheet editor, please refer to the "*The requirements spreadsheet editor*" section in the current chapter of this user guide.

4 - The dictionary spreadsheet editor. For further information on the dictionary spreadsheet editor, please refer to the "*The dictionary spreadsheet editor*" section in the current chapter of this user guide.

The requirements spreadsheet editor

Introduction

The requirements spreadsheet editor is used to manage requirements and traceability. Requirements can be created, modified or deleted, and traceability links can be added, modified or destroyed.

By default, the requirements spreadsheet editor contains three columns, "*Name*", "*Description*" and "*Traceability*".

The requirements spreadsheet editor is launched by clicking on the  "*Open the requirements editor*" icon in the Objecteering/UML toolbar.

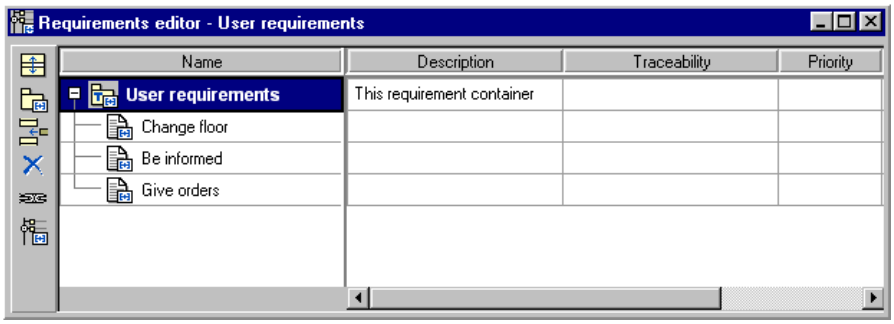


Figure 5-2. The requirements spreadsheet editor

Note: Requirement containers appear in **bold** in the requirements spreadsheet editor, for greater visibility.

Icons in the requirements spreadsheet editor

The ... icon	is used to ...
	change the spreadsheet editor mode to expand. In expand mode, the full contents of fields is visible.
	change the spreadsheet editor mode to compact. In compact mode, only the first line of each field's contents is visible.
	add a new requirement container to the selected requirement container.
	insert a new requirement into the requirements spreadsheet editor. Requirements can be created under the root or inside requirement containers. Note: requirement containers cannot be created using the "*Insert*" icon.
	delete a requirement or a requirement container (potentially containing one or several requirements) from the requirements spreadsheet editor.
	create a traceability link for a requirement or requirement container. When this icon is activated, the "*Managing traceability*" window appears, in which you can create new traceability links or modify existing ones.
	open an editor on the selected requirement container.

The dictionary spreadsheet editor

Introduction

The dictionary spreadsheet editor is used to manage dictionaries, terms and traceability. Terms can be created, modified or deleted, and traceability links can be added, modified or destroyed.

By default, the dictionary spreadsheet editor contains a number of columns which present the properties of the dictionaries in question.

The dictionary spreadsheet editor is launched by clicking on the  "*Open the dictionary editor*" icon in the Objecteering/UML toolbar.

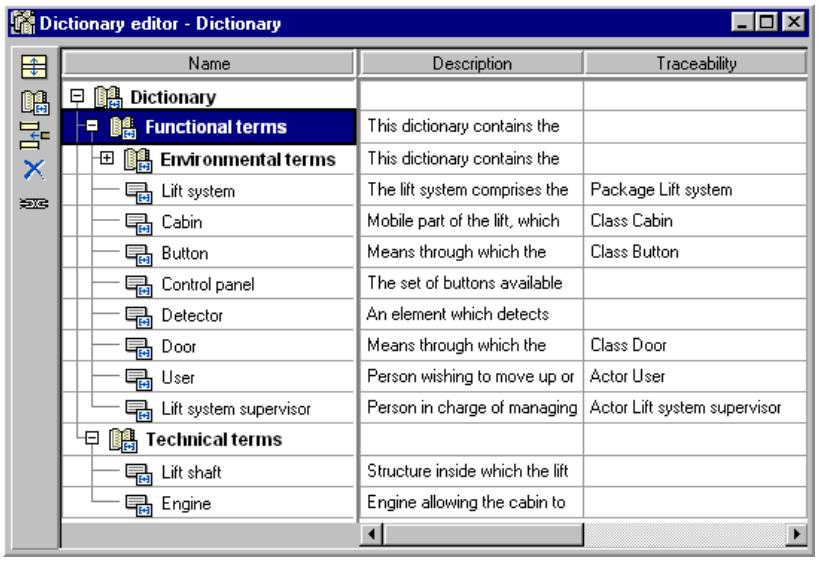


Figure 5-3. The dictionary spreadsheet editor

Note: Dictionaries appear in **bold** in the dictionary spreadsheet editor, for improved visibility.

Icons in the dictionary spreadsheet editor

The ... icon	is used to ...
	change the spreadsheet editor mode to expand. In expand mode, the full contents of fields is visible.
	change the spreadsheet editor mode to compact. In compact mode, only the first line of each field's contents is visible.
	add a new dictionary to the selected dictionary.
	insert a new term into the dictionary spreadsheet editor. Terms can be created under the root or inside specific dictionaries. Note: dictionaries themselves cannot be created using the "*Insert*" icon.
	delete a term or a dictionary (potentially containing one or several terms) from the dictionary spreadsheet editor.
	create a traceability link for a term or dictionary. When this icon is activated, the "*Managing traceability*" window appears, in which you can create new traceability links or modify existing ones.

The Objecteering/UML properties editor

The "Requirements" tab of the properties editor on a requirement model

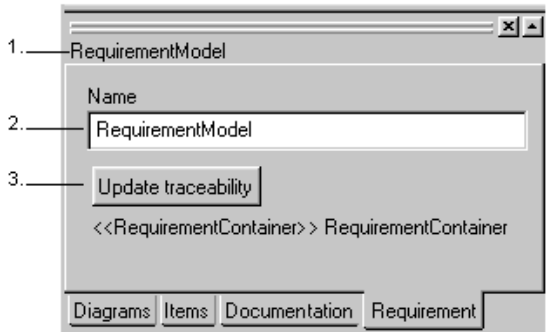


Figure 5-4. The "*Requirements*" tab of the properties editor on a requirement model

Key:

1 - This indicates the name of the requirement model selected in the explorer or in a graphic editor.

2 - This field is used to modify the name of the requirement model.

3 - This button is used to update traceability links. When this button is pressed, the "*Create a traceability link*" window opens, through which you can create new traceability links or modify existing ones.

Note 1: The properties editor displays these same fields for most other model elements, including packages, classes, actors and attributes.

Note 2: A requirement model is a package stereotyped <<RequirementModel>>. For further information on this stereotype, please see the "*Stereotypes*" section in chapter 7 of this user guide.

The "Requirements" tab of the properties editor on a use case

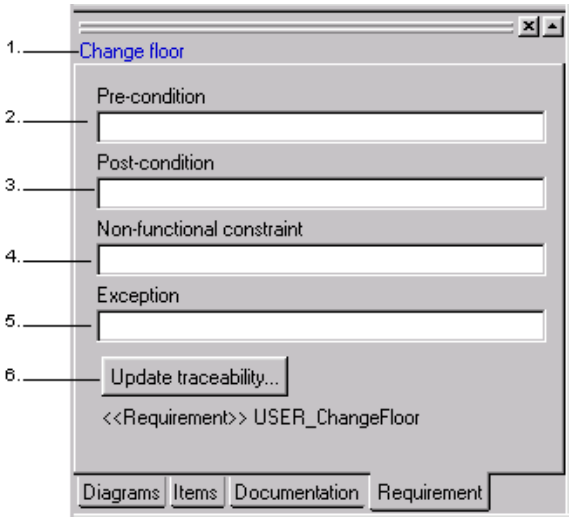


Figure 5-5. The "*Requirements*" tab of the properties editor on a use case

Key:

1 - This indicates the name of the use case selected in the explorer or in a graphic editor.

2 - This field is used to indicate a pre-condition for the use case.

3 - This field is used to indicate a post-condition for the use case.

4 - This field is used to specify a non-functional constraint for the use case.

5 - This field is used to indicate an exception for the use case.

6 - This button is used to update traceability links. When this button is pressed, the "*Create a traceability link*" window opens, through which you can create new traceability links or modify existing ones.

The "Requirements" tab of the properties editor on a requirement scenario

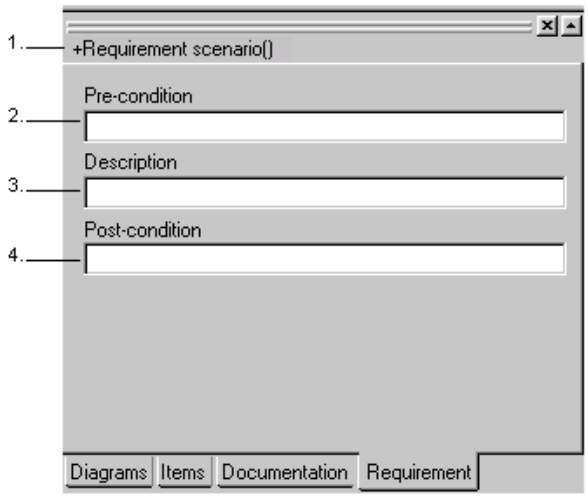


Figure 5-6. The "*Requirements*" tab of the properties editor on a requirement scenario

Key:

1 - This indicates the name of the requirement scenario selected in the explorer or in a graphic editor.

2 - This field is used to enter a pre-condition for the requirement scenario.

3 - This field is used to provide a description of the requirement scenario.

4 - This field is used to specify a post-condition for the requirement scenario.

Note: A requirement scenario is an operation stereotyped <<RequirementScenario>>. For further information on this stereotype, please see the "*Stereotypes*" section in chapter 7 of this user guide.

The "Requirements" tab of the properties editor on a requirement container

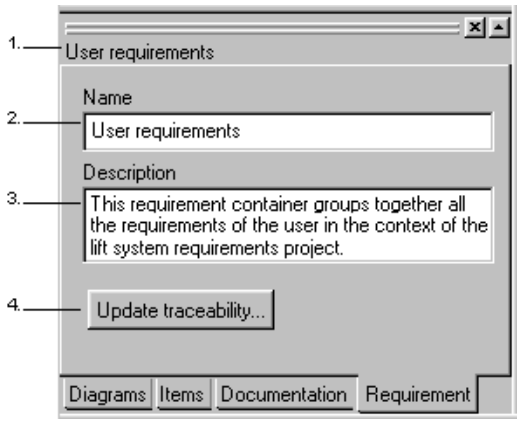


Figure 5-7. The "*Requirements*" tab of the properties editor on a requirement container

Key:

1 - This indicates the name of the requirement container selected in the explorer or in the requirements spreadsheet editor.

2 - This field is used to modify the name of the requirement container.

3 - This field is used to enter a description of the requirement container.

4 - This button is used to update traceability links. When this button is pressed, the "*Create a traceability link*" window opens, through which you can create new traceability links or modify existing ones.

The "Requirements" tab of the properties editor on a requirement

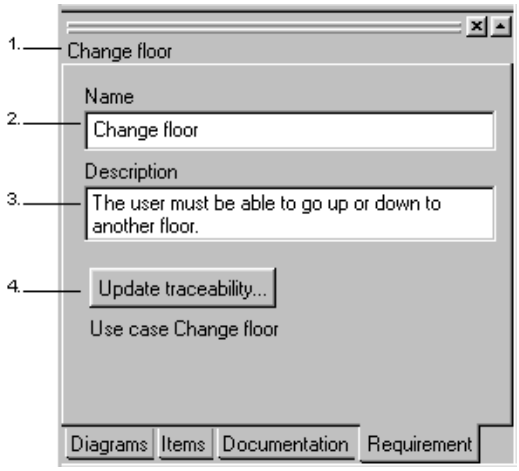


Figure 5-8. The "*Requirements*" tab of the properties editor on a requirement

Key:

1 - This indicates the name of the requirement selected in the explorer or in the requirements spreadsheet editor.

2 - This field corresponds to the "*Name*" column in the requirements spreadsheet editor, and is used to modify the name of the requirement.

3 - This field corresponds to the "*Description*" column in the requirements spreadsheet editor, and is used to enter a description of the requirement.

4 - This button is used to update traceability links. When this button is pressed, the "*Create a traceability link*" window opens, through which you can create new traceability links or modify existing ones.

The "Requirements" tab of the properties editor on a dictionary or a term

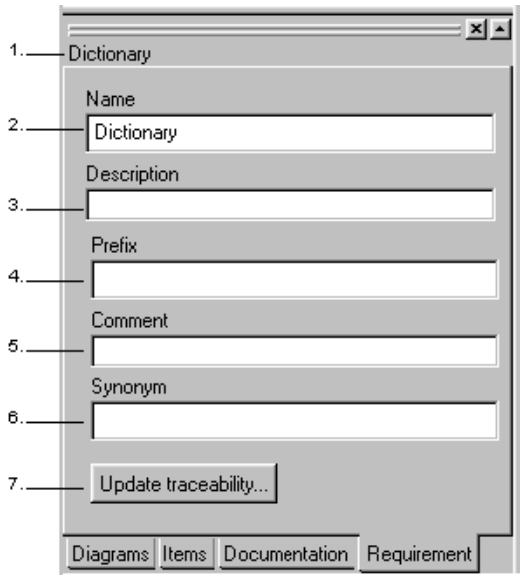


Figure 5-9. The "*Requirements*" tab of the properties editor on a dictionary

Key:

1 - This indicates the name of the dictionary or term selected in the explorer or in the dictionary spreadsheet editor.

2 - This field corresponds to the "*Name*" column in the dictionary spreadsheet editor, and is used to modify the name of the dictionary or term.

3 - This field corresponds to the "*Description*" column in the dictionary spreadsheet editor, and is used to enter a description for the dictionary or term.

4 - This field corresponds to the "*Prefix*" column in the dictionary spreadsheet editor, and is used to enter a prefix for the dictionary or term.

5 - This field corresponds to the "*Comment*" column in the dictionary spreadsheet editor, and is used to enter a comment for the dictionary or term.

6 - This field corresponds to the "*Synonym*" column in the dictionary spreadsheet editor, and is used to enter a synonym for the dictionary or term.

7 - This button is used to update traceability links. When this button is pressed, the "*Create a traceability link*" window opens, through which you can create new traceability links or modify existing ones.

Configuring your interface ergonomically

Introduction

Objecteering/UML Requirements features dockable windows, in other words, windows which you can position and dock as you wish within the tool's workspace or even outside it.

It can be practical for the user to re-arrange these dockable windows in different positions and configurations, according to the development situation. In this section, we are going to illustrate two possible configurations, specific to the development of a requirements project.

Defining several requirements

In the case of the definition of several requirements one after the other, we recommend that you arrange your windows as shown in Figure 5-10 below.

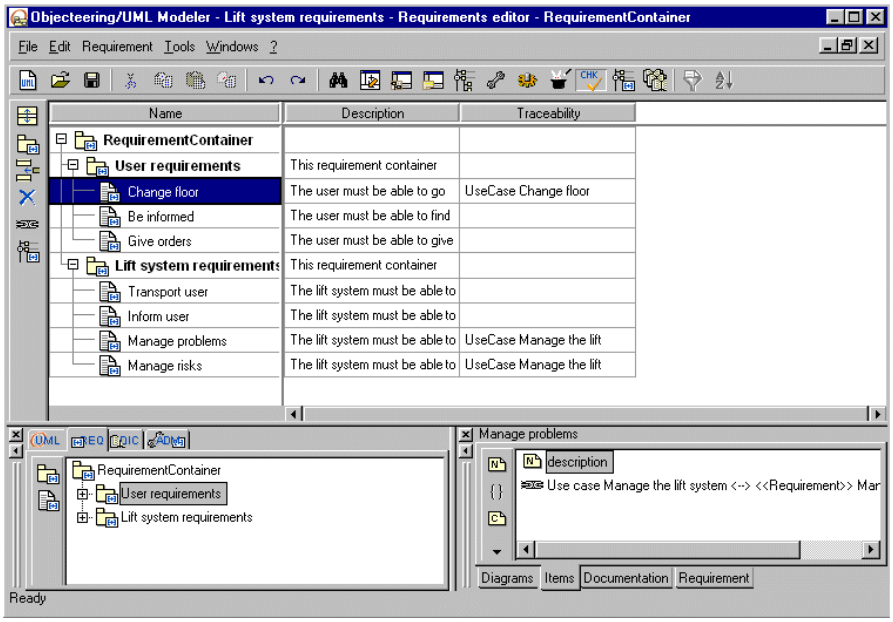


Figure 5-10. Ideal interface configuration when entering several requirements

To organize your interface as shown above, carry out the following steps:

1 - First, close the console by clicking on the  "*Hide the console*" icon in the tool bar.

2 - Next, open the requirements spreadsheet editor (if it is not already open), by clicking on the  "*Open the requirements editor*" icon. Continue by maximizing this window using the  "*Maximize*" button in the top right-hand corner.

3 - Finally, click on the double parallel lines at the top of the explorer window, and drag it to the bottom of your workspace, until you reach the desired position. Once you are in position, simply drop the window into place. Repeat with the properties editor, if you think you will need it (if not, close it by clicking on the  "*Hide the properties editor*" icon in the tool bar.

The advantages of this interface configuration in the context of the definition of several requirements are:

- the requirements spreadsheet editor has been expanded to fill the entire screen, thereby making it easier for the user to enter and define new requirements.
- the explorer is still visible, in case the user needs to create new requirement containers, which, as structuring units, can only be created in an explorer and not the requirements spreadsheet editor.
- visibility is increased.

Defining several terms

We recommend the same interface configuration when defining a number of terms as the one just described for defining several requirements, with the obvious difference that the requirements spreadsheet editor should be closed, and the dictionary spreadsheet editor open.

Similarly, the explorer should show the dictionary, so as to allow the creation of dictionary structuring units. To change the focus of the main explorer, or open a new explorer, click on the  "*Browse the model*" icon in the tool bar.

Creating UML elements from spreadsheet editors

Where are you going to be using the "*Requirements/Create a UML element*" command to create new model elements from a spreadsheet editor, the interface configuration shown in Figure 5-11 is ideal.

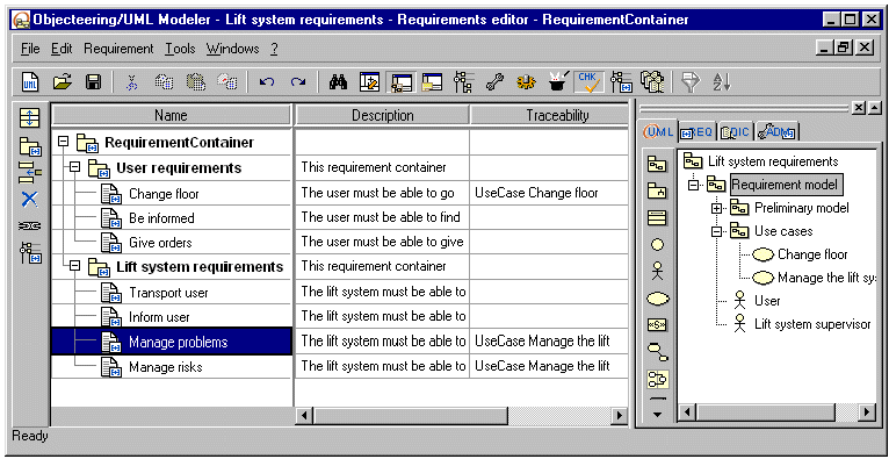


Figure 5-11. Ideal interface configuration when creating UML elements from a spreadsheet editor

To organize your interface as shown above, carry out the following steps:

1 - First, close the console by clicking on the "*Hide the console*" icon in the tool bar. Continue by closing the properties editor using the "*Hide the properties editor*" icon in the tool bar.

2 - Next, open the requirements spreadsheet editor (if it is not already open), by clicking on the "*Open the requirements editor*" icon. Continue by maximizing this window using the "*Maximize*" button in the top right-hand corner.

3 - Finally, click on the double parallel lines at the top of the explorer window, and drag it to the extreme right-hand side of your workspace, until you reach the desired position. Once you are in position, simply drop the window into place.

The advantages of this interface configuration in the context of the creation of UML elements from a spreadsheet editor are:

- the requirements spreadsheet editor has been expanded to take up most of the workspace, thereby making it easy for the user to access the "*Requirements/Create a UML element*" command.
- the position of the model explorer on the right-hand side of the screen means that it is easy to drag and drop elements from there into the "*Parent*" zone of the "*Create a UML element*" window.
- visibility is increased.

Note: We recommend the same interface configuration when using the "*Create a UML element*" command from the dictionary spreadsheet editor.

Creating traceability links

When creating traceability links through the  "*Traceability*" icon in the spreadsheet editors, we recommend the same interface layout as shown in Figure 5-11, as this provides easy access to both the spreadsheet editor and the explorer in question.

Chapter 6: Managing requirement properties

Working with the requirement properties explorer

Using the requirements properties explorer

In order to be able to use the Requirement properties explorer, the *Objecteering/ReqPropertiesAdmin* module must have been selected. For further information on selecting modules, please refer to the "*Working with the Objecteering/UML Requirements module*" section in chapter 2 of this user guide.

The requirement properties explorer

The Requirement properties explorer is used to create and manage properties and property sets.

The Requirement properties explorer can be accessed by clicking on the  tab of the explorer (see Figure 6-1 below).

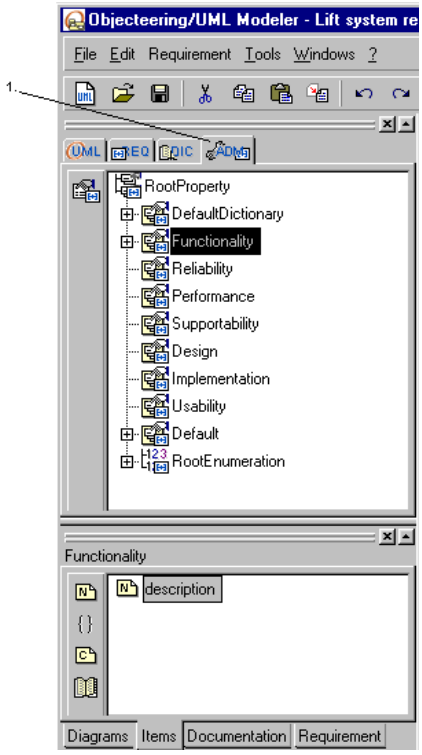


Figure 6-1. The Requirements properties explorer

Steps:

1 - Click on the  tab in the explorer.

Creating property sets and properties

Creating a new property set

To create a new property set, follow the steps shown in Figure 6-2 below.

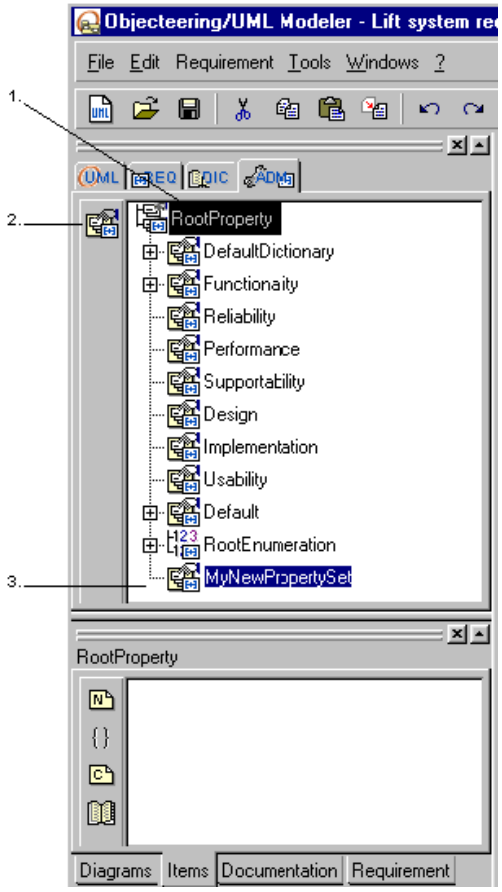


Figure 6-2. Creating a new property set

Steps:

1 - In the  tab of the explorer, select the property set inside which you want to create your new property set.

2 - Click on the  "*Create a property set*" icon.

3 - Enter a name for your new property set.

Modifying a property set

To modify a property set, either double-click on it or right-click and run the "*Modify*" command. The window shown in Figure 6-3 then appears.

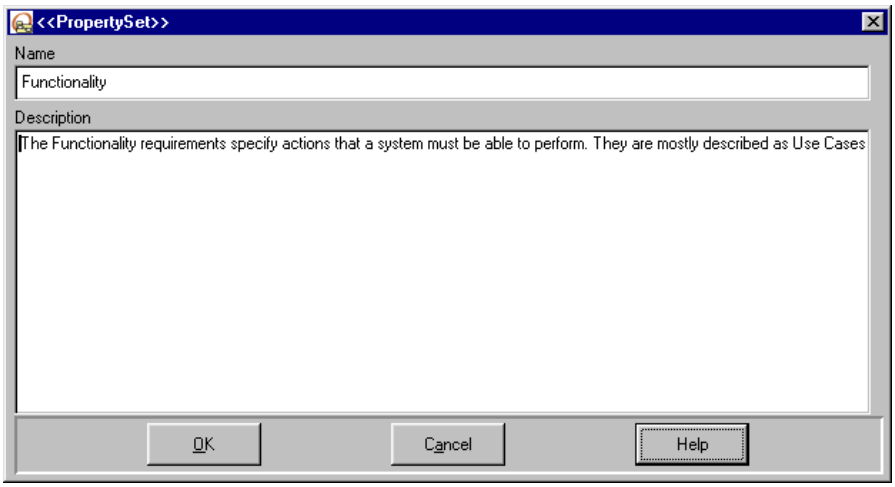


Figure 6-3. The window used to modify property sets

Creating a new property

To create a new property, follow the steps shown in Figure 6-4 below.

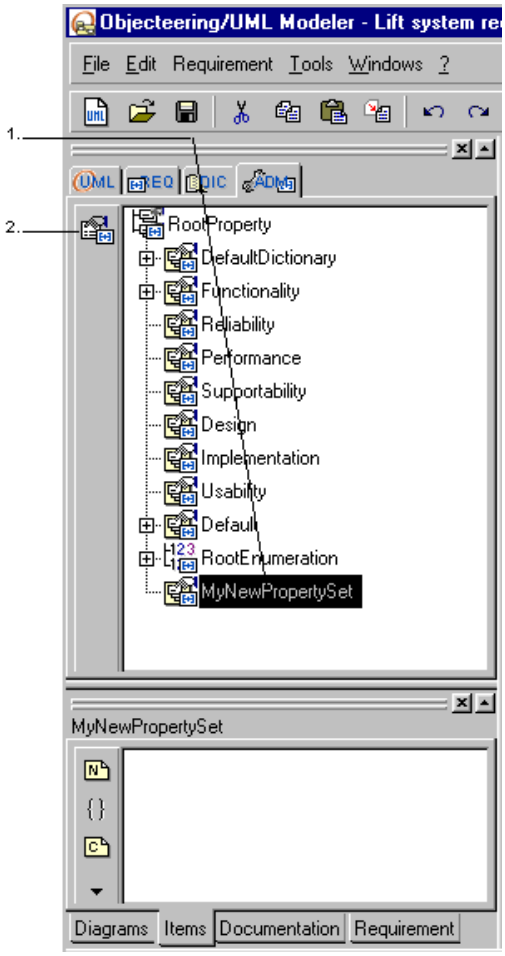


Figure 6-4. Creating a new property - stage 1

Steps:

1 - Click on the property set inside which you want to create a new property.

2 - Click on the  "*Create a property*" icon. The window shown in Figure 6-5 then appears.

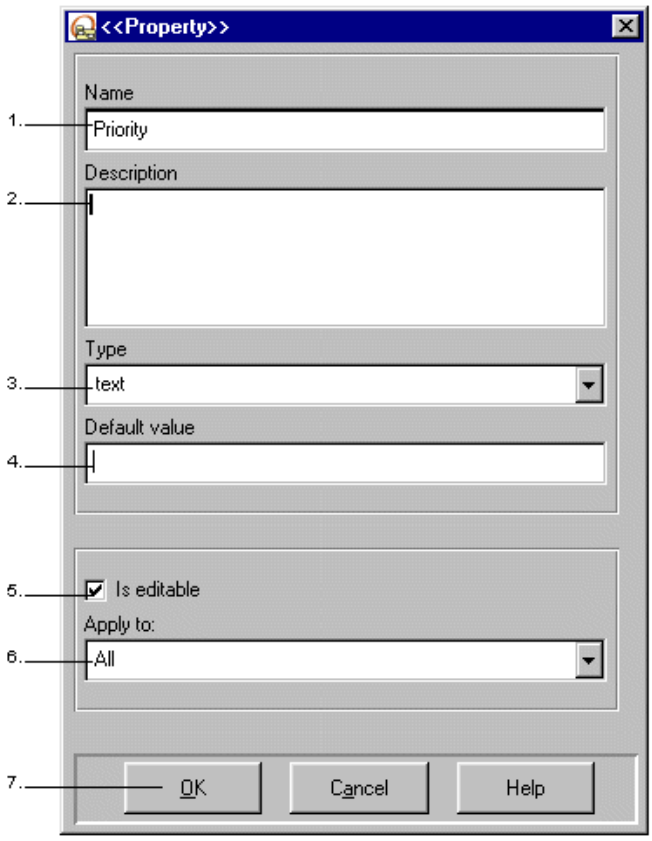


Figure 6-5. Creating a new property - stage 2

Steps:

1 - Enter a name for the new property.

2 - Enter a description of the new property.

3 - Select a type from the scrolling list.

4 - Define a default value.

5 - Indicate whether or not you want this new property to be editable.

6 - Choose whether you want this new property to be applicable to requirement containers and dictionaries, requirements and terms or everything.

7 - Click on "*OK*" to confirm.

Implementing property sets

Introduction

After creating properties sets containing various properties in the tab of the explorer, you can now implement them on your requirement containers, dictionaries, requirements and terms.

To implement a new property set, follow the steps shown in Figure 6-6 below.

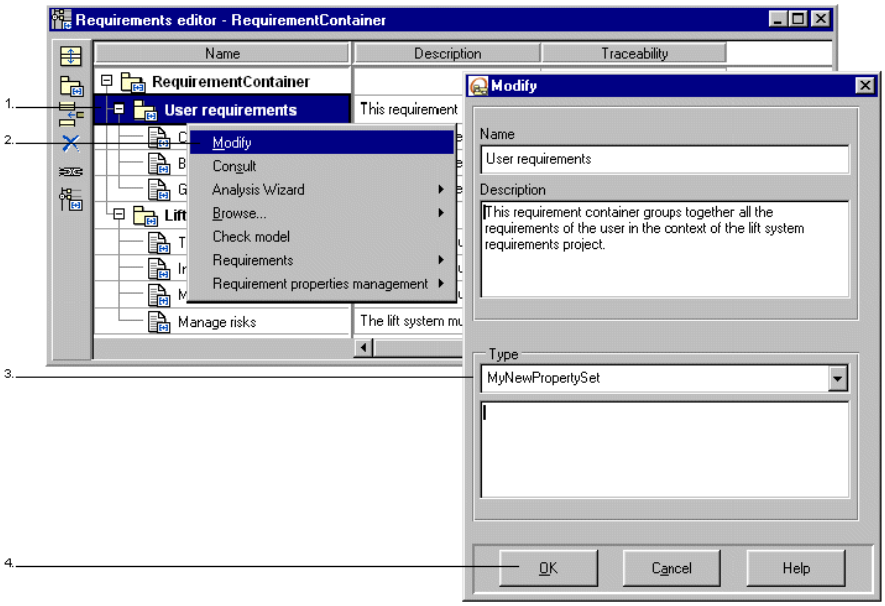


Figure 6-6. Implementing a new property set

Steps:

1 - In either the dictionary spreadsheet editor or the requirements spreadsheet editor, right-click on the dictionary or requirement container on which you want to apply the new property set.

2 - From the context menu which then appears, select "*Modify*".

3 - In the "*Type*" field of the "*Modify*" window, select the new property set from the dropdown list.

4 - Click on "*OK*" to confirm.

The result of this operation can be seen in Figure 6-7 below.

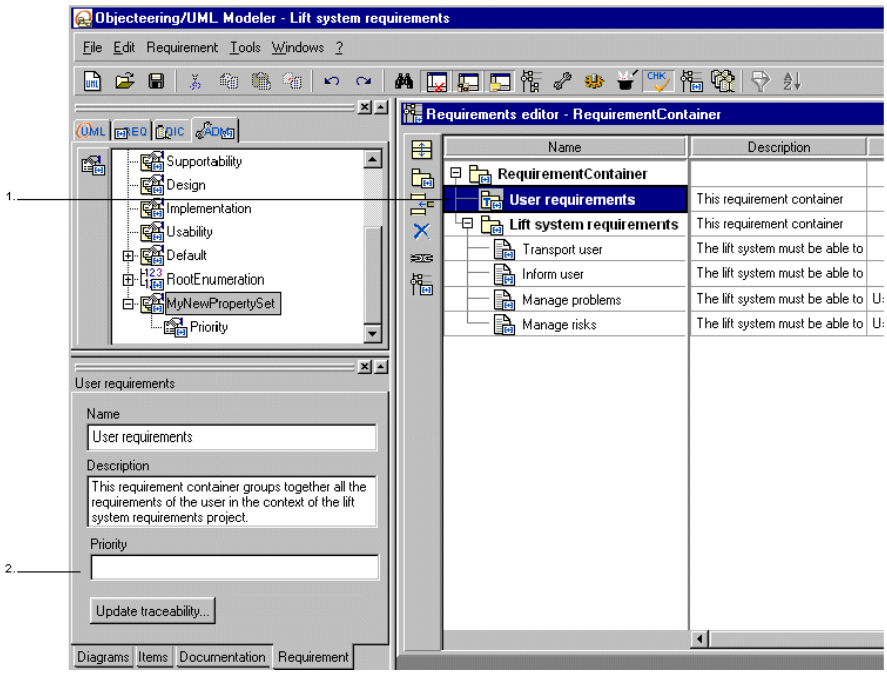


Figure 6-7. The result of the implementation of a property set

Key:

1 - In the requirements spreadsheet editor, the requirement container which has been typed with a property set is identified by the  icon.

2 - In the "*Requirement*" tab of the properties editor on the typed requirement contained, an additional property field appears.

Note 1: If other requirement containers or dictionaries are created under a typed requirement container or dictionary, they will have the same type as their parent.

Note 2: If a parent requirement container or dictionary is not typed, its children can be of different types.

Externalizing requirement properties

The "*Requirement properties management/Externalize properties*" command allows you to externalize properties you have developed, in order to make them available to other developers.

To externalize your properties, simply carry out the steps shown in Figure 6-8 below.

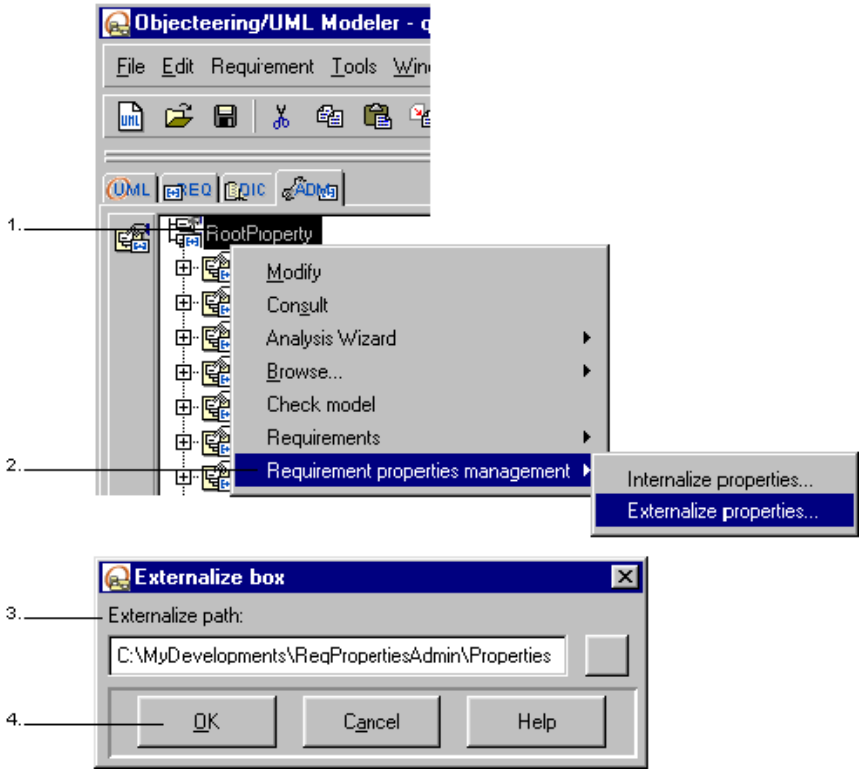


Figure 6-8. Externalizing properties

Steps:

1 - Select an element in the explorer by right-clicking to open the context menu.

2 - Run the "*Requirement properties management/Externalize properties*" command.

3 - The "*Externalizing properties*" window then opens. In the "*Externalization directory*" field, enter the path to the properties you wish to externalize.

4 - Click on "*OK*" to confirm.

The properties externalization operation is then run. You can follow its progress in the console.

Internalizing requirement properties

The "*Requirement properties management/Internalize properties*" command allows you to internalize properties developed by a third party, in order to make them available in your project.

This command can be used irrespective of whether or not you have already developed requirements in your project.

To internalize properties and be able to use them in the context of your project, simply carry out the steps shown in Figure 6-9 below.

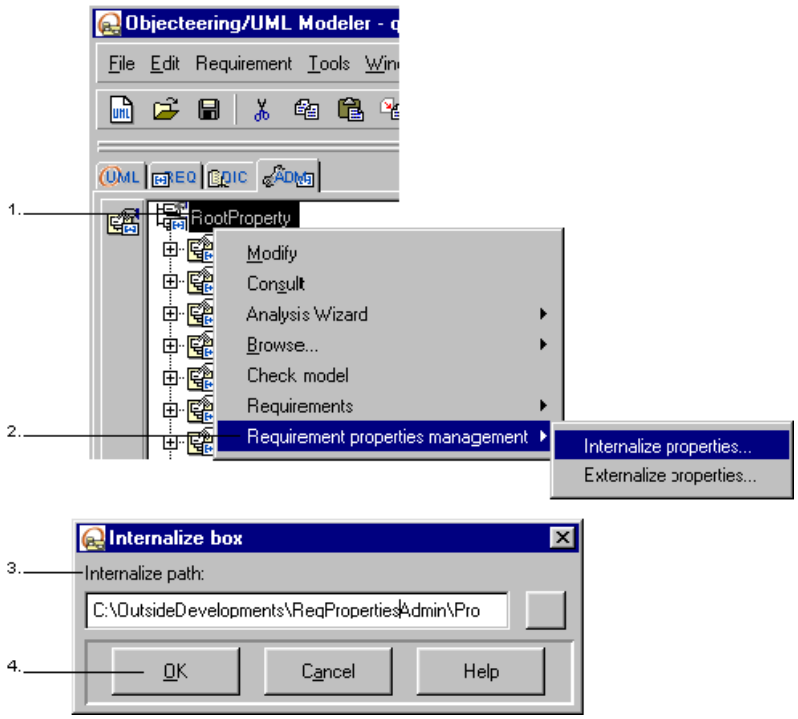


Figure 6-9. Internalizing properties

Steps:

1 - Select an element in the explorer by right-clicking to open the context menu.

2 - Run the "*Requirement properties management/Internalize properties*" command.

3 - The "*Internalizing properties*" window then opens. In the "*Internalization directory*" field, enter the path to the properties you wish to internalize.

4 - Click on "*OK*" to confirm.

The properties internalization operation is then run. You can follow its progress in the console.

Note: The "*Show internalized files*" tickbox is used to display in the console those files which have been internalized.

Chapter 7: Word and Objecteering/UML Requirements

Introduction to using Word with Objecteering/UML Requirements

Overview

The main purpose of coupling Word to *Objecteering/UML Requirements* is to provide users with a means of marking elements in an existing Word document as being *Objecteering/UML Requirements* project elements. The Word document can then be generated as an XML file, which can subsequently be imported into *Objecteering/UML Requirements*, so as to automatically build the corresponding *Objecteering/UML Requirements* project.

Using the "*Requirements For Objecteering Model.dot*" model, it is possible to mark elements in the Word document as being:

- requirement containers
- requirements
- dictionaries
- terms

Getting started

To be able to prepare your existing Word document for export into *Objecteering/UML Requirements*, the first step is to add the "*Requirements for Objecteering Model.dot*" model to the document, as shown in Figure 7-1 below.

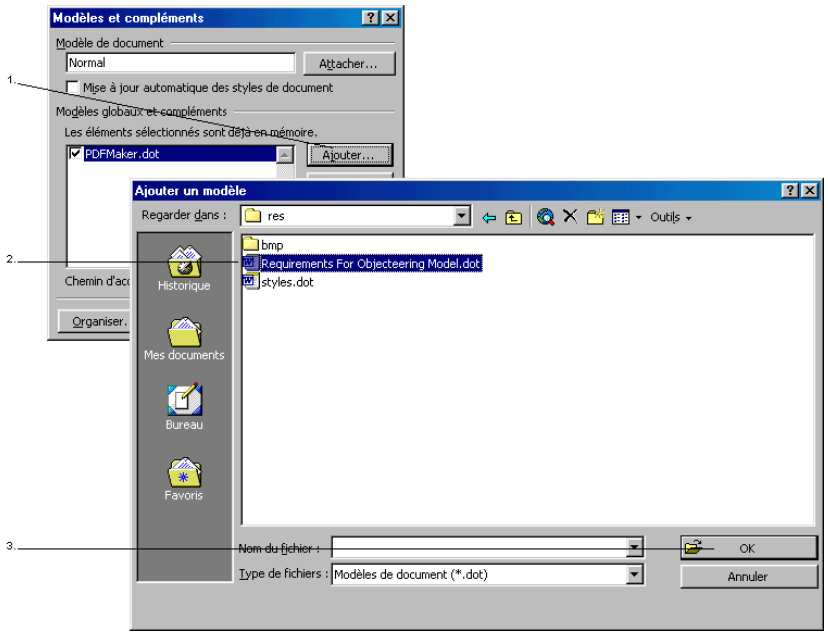


Figure 7-1. Adding the "*Requirements for Objecteering Model.dot*" file to your existing Word document

Steps:

1 - After clicking on the "*Tools/Models and complements*" menu in Word, the "*Models and complements*" window appears. To add the required .dot file, click on the "*Add*" button. The "*Add a model*" window then appears.

2 - Browse the explorer to locate the "*Requirements for Objecteering Model.dot*" file, which is found in the "*res*" directory of the "*Requirements*" module, in your Objecteering/UML installation.

The result of this operation is shown in Figure 7-2.

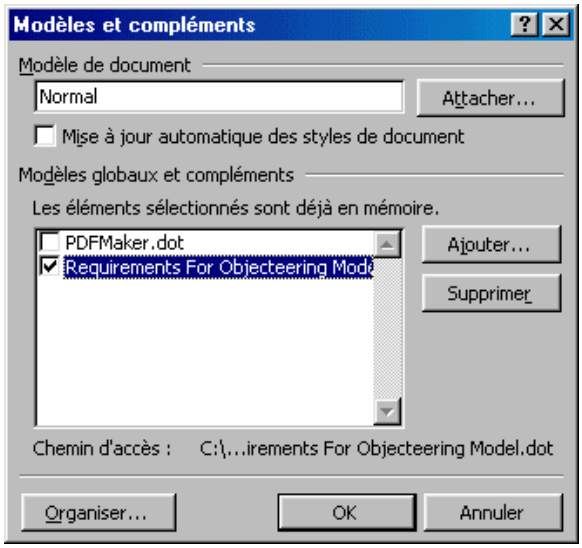


Figure 7-2. The "*Requirements For Objecteering Model.dot*" model can now be selected.

After making sure that the associated tickbox is checked, simply click on "*OK*" to apply this model to your existing Word document.

The result of this operation in Word is shown in Figure 7-3.

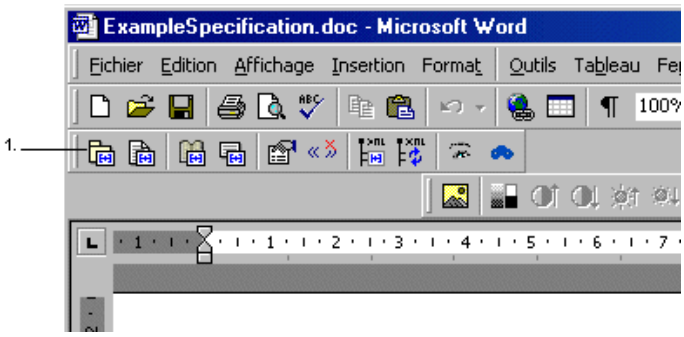


Figure 7-3. The Requirements icons available in your Word document

Key:

1 - The "*Requirements*" tool bar appears in your Word document.

Tagging elements in a Word document

Marking an element as being a Requirements element

Elements in a Word document can be marked as being the following:

- requirement containers
- requirements
- dictionaries
- terms

To mark an element in a Word document as being a requirement container, requirement, dictionary or term, the user highlights in the Word document the description of the element to be created and then clicks on the corresponding marker icon, before defining a name for the element. By default, the first word in the description is suggested as the element name.

Figure 7-4 below shows an example of a requirement container being marked.

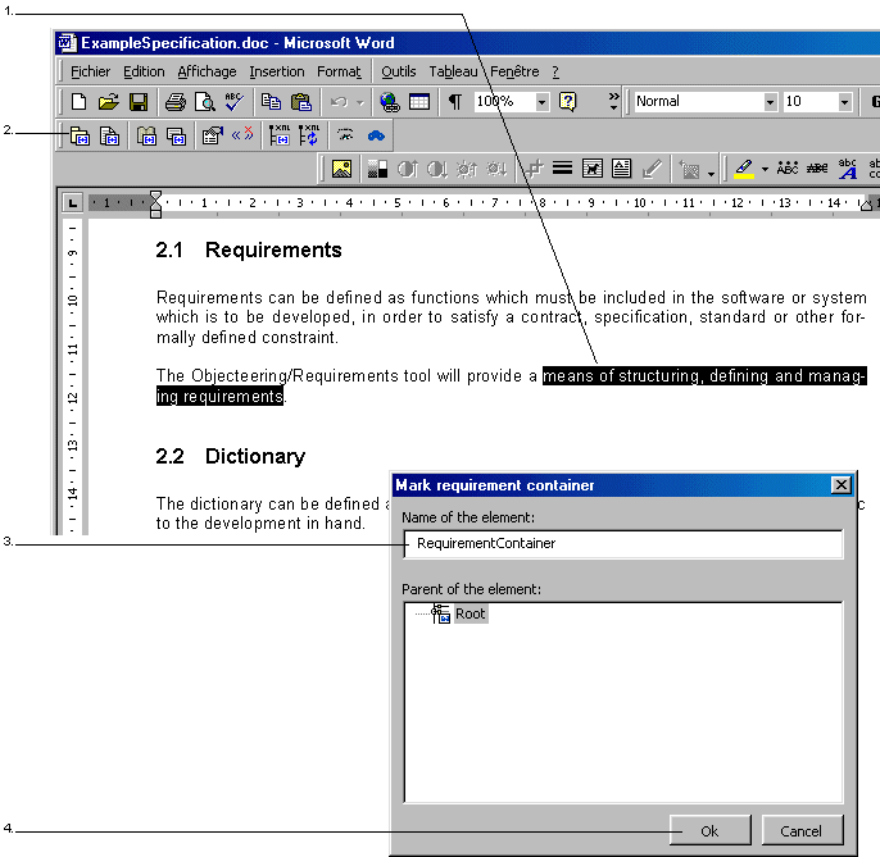


Figure 7-4. Marking an element as being a requirement container

Steps:

1 - Select in Word the phrase that you wish to use as the definition of the requirement container you are about to create.

2 - In the tool bar, click on the  "*Requirement container*" icon. The "*Mark requirement container*" window then appears.

3 - In the "*Name of the element*" field, a default name is proposed. If you wish to change this default name, enter a new name.

4 - Click on "*OK*" to confirm.

The result of this operation is shown in Figure 7-5.

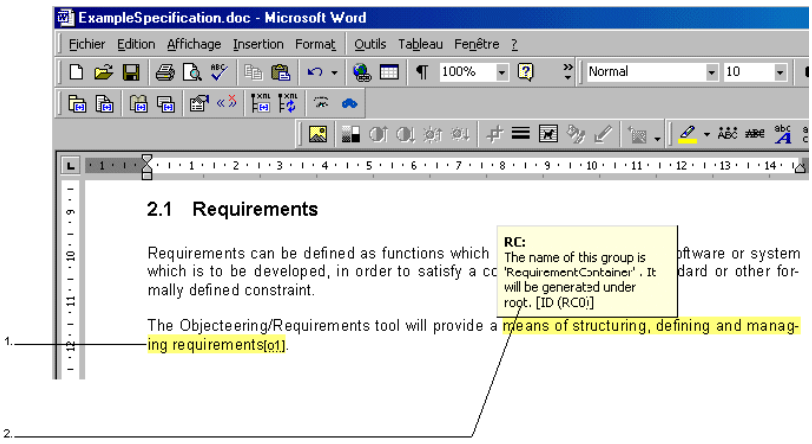


Figure 7-5. The newly marked requirement container

Steps:

1 - The selected element is then displayed over a yellow background and with a reference in brackets after it.

2 - If you position the cursor over the highlighted element, an information panel appears, informing you of the name of the element and its position in the hierarchy.

Exactly the same procedure as shown in Figure 7-4 is used to mark elements as being:

- requirements, using the  "*Requirement*" icon in the toolbar
- dictionaries, using the  "*Dictionary*" icon in the toolbar
- terms, using the  "*Term*" icon in the toolbar

---

Managing requirement properties

Basic requirements hierarchy

Before marking any elements in your Word document, a basic requirements hierarchy already exists, as shown in Figure 7-6 below.

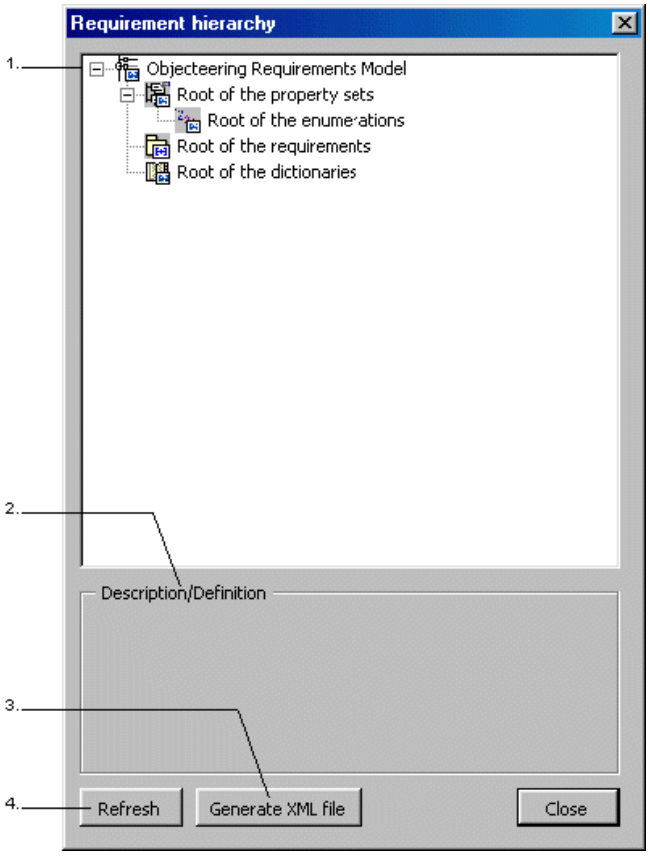


Figure 7-6. The basic "*Hierarchy*" window

Key:

1 - The basic hierarchy of elements, consisting of a root for property sets, a root for enumerations, a root for requirements and a root for dictionaries.

2 - A field which shows the description of an element.

3 - A "*Generate XML file*" button, used to launch generation of the XML file which can then be imported into Objecteering/UML.

4 - A "*Refresh*" button, used to refresh the view.

Defining the properties of a marked element

When marking elements in a Word document, you can select where you wish the element to be located.

- Requirement containers can be located either under the root requirement container or under a requirement container already marked in the document.
- Dictionaries can be located either under the root dictionary or under a dictionary already marked in the document.
- Requirements can be located either under the root requirement container or under a requirement container already marked in the document.
- Terms can be located either under the root dictionary or under a dictionary already marked in the document.

For example, Figure 7-7 illustrates the marking of a requirement and the selection of its location.

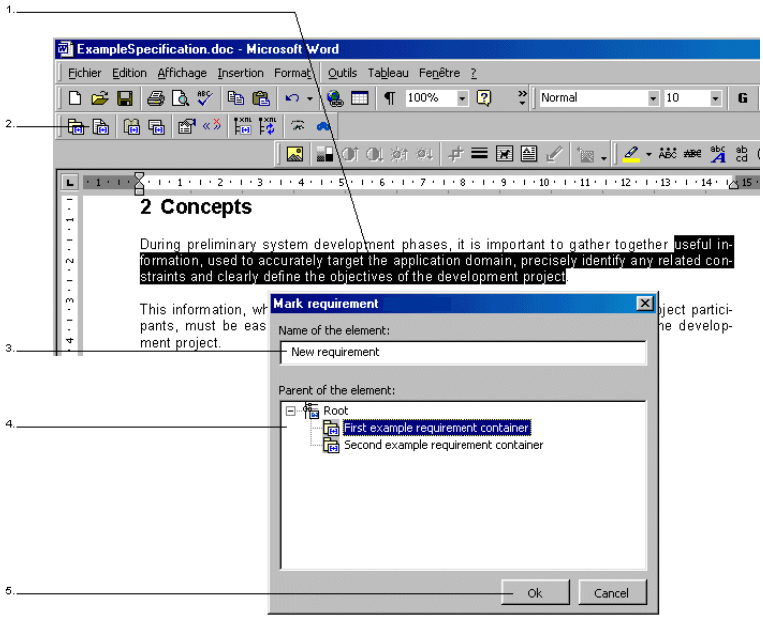


Figure 7-7. Marking a requirement in the "*First example requirement container*" requirement container

Steps:

1 - In the Word document, select the text you wish to use as the definition of the new requirement.

2 - In the toolbar, click on the  "*Requirement*" icon. The "*Mark requirement*" window appears.

3 - Enter a name for the new requirement.

4 - Select the requirement container which will contain the new requirement.

5 - Click on "*OK*" to confirm.

The new requirement is then located inside the selected requirement container in the "*Requirement hierarchy*" window, as shown in Figure 7-8 below.

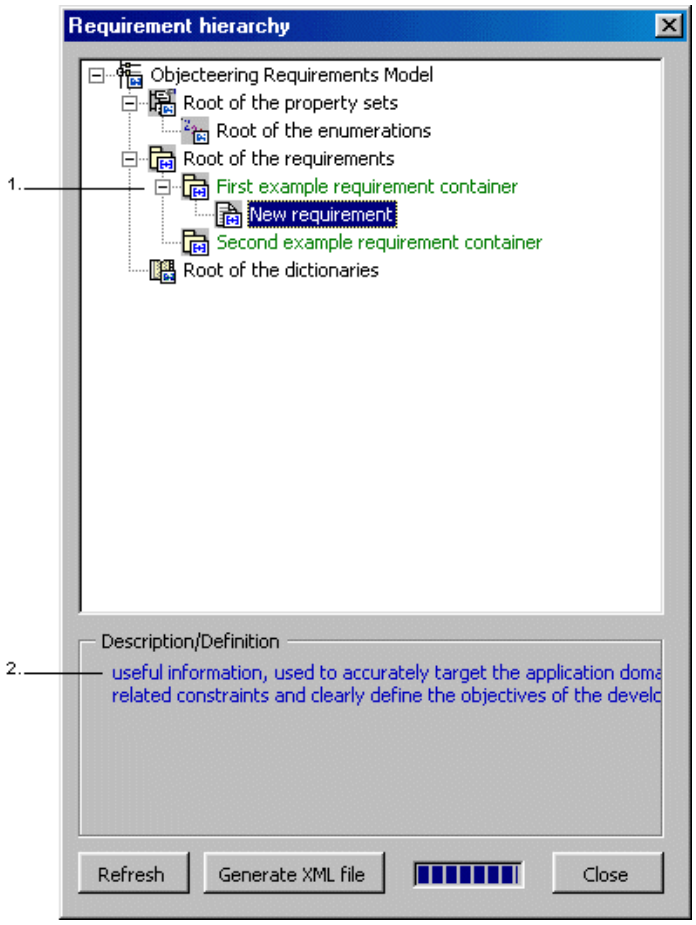


Figure 7-8. The "*Requirement hierarchy*" window now shows the new requirement

Key:

1 - Everything marked by the user (in other words, everything which is not the basic requirements hierarchy) appears in green.

2 - The description of the marked element appears in the "*Description/Definition*" field.

Note: If you click on an element in the "*Requirement hierarchy*" window, the element is automatically selected in the Word document. This can be particularly useful if you are working with large documents.

Modifying the properties of a marked element

If you want to change the properties (name and location) of a marked element, the "*Requirement properties*" window is used. This window is opened by clicking on a marked element and then clicking on the  "*Display properties*" icon in the toolbar.

For example, let's imagine we want to change the name and location of the requirement marked in the previous example. Simply carry out the steps shown in Figure 7-9.

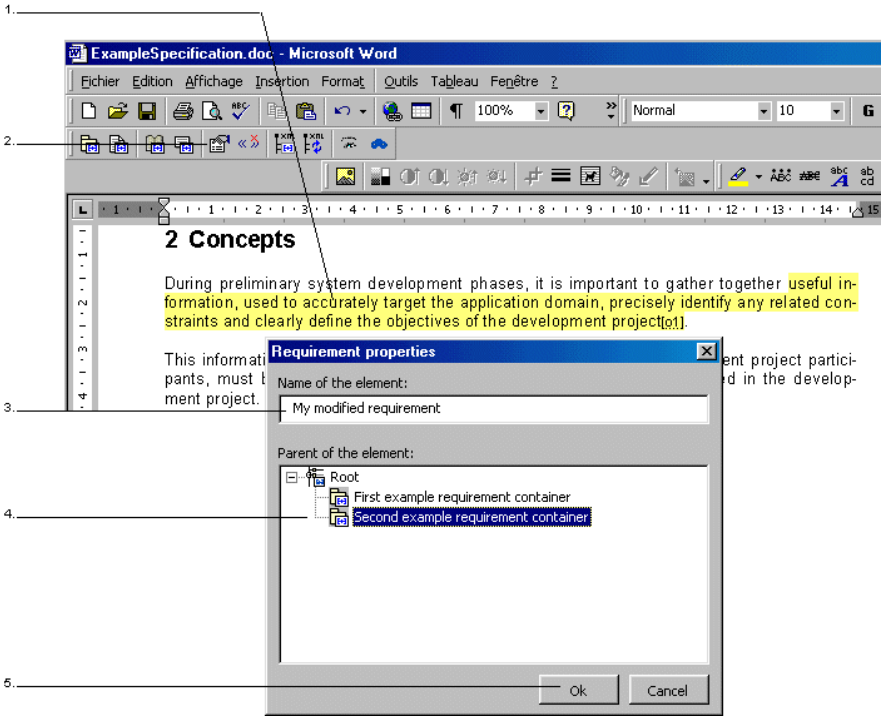


Figure 7-9. Modifying the properties of a requirement

Steps:

1 - Click on the highlighted text in the Word document.

2 - Click on the  "*Display properties*" icon in the toolbar. The "*Requirement properties*" window then appears.

3 - Modify the name of the requirement in the "*Name of the element*" field.

4 - Modify the location of the requirement from "*First example requirement container*" to "*Second example requirement container*".

5 - Click on "*OK*" to confirm.

These changes can be seen in the "*Requirement hierarchy*" window.

Note: If the "*Requirement hierarchy*" window is already open when modifications are made, it must be closed and then re-opened before changes become visible.

Generating the XML file

Generating the XML file

Once you have completed the marking of Word elements as being *Objecteering/UML Requirements* elements, the next step is to generate the corresponding XML file which will be imported into Objecteering/UML.

To generate the XML file, simply carry out the steps shown in Figure 7-10 below.

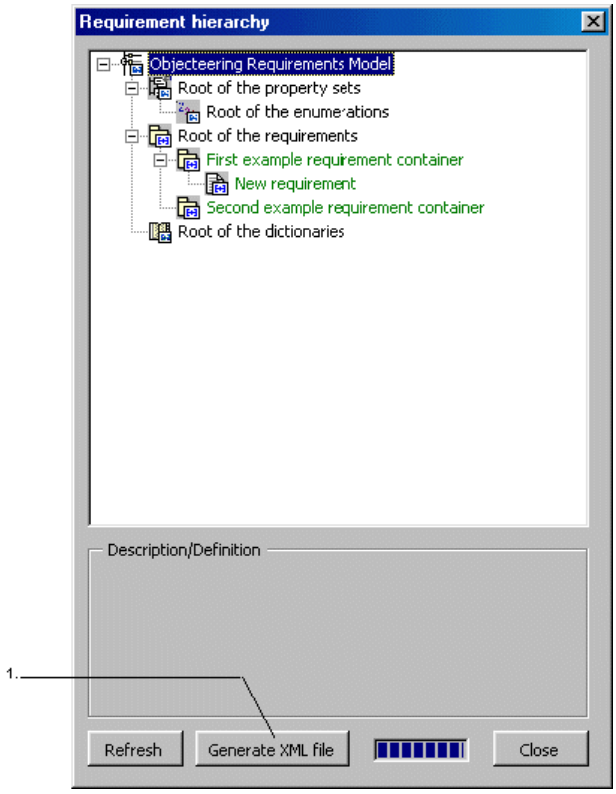


Figure 7-10. Generating the XML file

Steps:

1 - Click on the "*Generate XML file*" button. Generation of the XML file corresponding to the marked Word document is then launched.

Note: The XML file is generated where the original Word document is located.

Importing the generated XML file into Objecteering/UML Requirements

Once you have generated the XML file, the next step is to import it into *Objecteering/UML Requirements*. For this, the "*Requirements/Import XML file*" command is used, as shown in Figure 7-11 below.

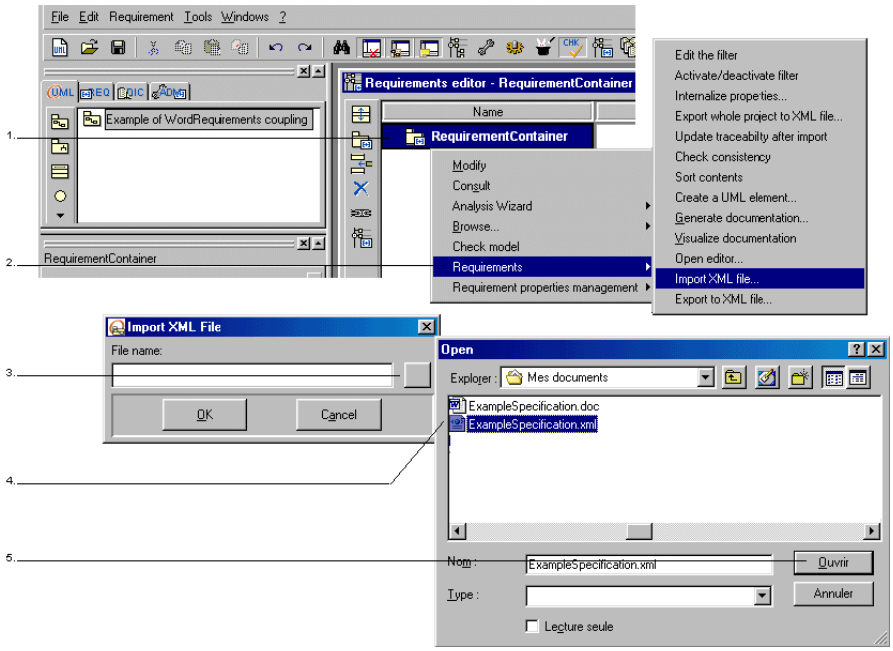


Figure 7-11. Importing the generated XML file into Objecteering/UML Requirements

Steps:

1 - Open your *Objecteering/UML Requirements* project and right-click over the element into which you wish to import the XML file.

2 - Run the "*Requirements/Import XML file*" command. The "*Import XML file*" dialog box then appears.

3 - In the "*Import XML file*" dialog box, click on the  icon to open a file browser.

4 - Using the file browser, select the XML file you want to import.

5 - Click on "*Open*". The file browser is then closed. Click on "*OK*" in the "*Import XML file*" dialog box to launch the import operation.

Note: The DTD allowing the import and export of requirements is described in the "*DTD used in the import/export of requirements*" section of chapter 10 of this user guide.

Other Requirements functions available in Word

Deleting markers

To delete markers added to elements in a Word document, the  "*Delete marker*" icon is used. Simply position the cursor over a marked element or highlight a marked element, and then click on this icon. The selected marked element is deleted.

Selecting elements in the requirements hierarchy

To select elements from the Word document in the "*Requirements hierarchy*" window, the  "*Select in requirement hierarchy*" icon is used. Simply click on a marked element, and then click on this icon. The focus automatically jumps to the selected element in the requirement hierarchy.

Showing markers

If markers are not visible in a Word document, the  "*Show markers*" icon can be used to make them appear. Simply click on this icon, and all marked elements in the document will be highlighted.

Masking markers

If markers are visible in a Word document, the  "*Mask markers*" icon can be used to no longer display them. Simply click on this icon, and marked elements in the document will no longer be highlighted.

Requirements icons in Word

The following table describes the function of each of the icons which appear when the "*Requirements For Objecteering Model.dot*" model is attached to a Word document.

The ... icon	is used to ...
	mark an element in a Word document as being a requirement container.
	mark an element in a Word document as being a requirement.
	mark an element in a Word document as being a dictionary.
	mark an element in a Word document as being a term.
	manage the hierarchy of your marked elements.
	delete a marker indicating that an element in a Word document is an Objecteering/UML Requirements element.
	view and manage the hierarchy of your marked elements.
	jump from the element selected in the Word document to the same element in the "*Requirement hierarchy*" window.
	show elements marked as being Objecteering/UML Requirements elements.
	mask elements marked as being Objecteering/UML Requirements elements.

Chapter 8: Objecteering/UML Requirements annotations

Stereotypes

Introduction

Stereotypes are specific model extensions used to adapt development to a particular context, in this case, the context of a requirements project.

Note: The *Objecteering/UML Profile Builder* tool can be used to develop your own stereotypes, thereby allowing you to significantly customize your development environment. Please see the related user guide for further details.

Stereotypes on an item

The ... stereotype	is used to ...
<<Requirement>>	define a requirement.
<<RequirementContainer>>	define a requirement container.
<<Dictionary>>	define a dictionary.
<<Term>>	define a term.
<<Property>>	define a property.
<<PropertyContainer>>	define a property container.

For further information on items, please see the "*Item class*" section in chapter 4 of the *Objecteering/Metamodel* user guide.

Stereotypes on a package

The ... stereotype	is used to ...
<<RequirementModel>>	define a package as being a requirement model, used in the context of a requirements project.

For further information on packages, please see the "*Package class*" section in chapter 5 of the *Objecteering/Metamodel* user guide.

Stereotypes on a use case

The ... stereotype	is used to ...
<<RequirementUseCase>>	define a use case as being a requirement use case, used in the context of a requirements project.

For further information on use cases, please see the "*Use case class*" section in chapter 6 of the *Objecteering/Metamodel* user guide.

Stereotypes on an operation

The ... stereotype	is used to ...
<<RequirementScenario>>	define an operation as being a requirement scenario, used in the context of a requirements project.

For further information on operations, please see the "*Operation class*" section in chapter 5 of the *Objecteering/Metamodel* user guide.

Stereotypes on a constraint

The ... stereotype	is used to ...
<<RequirementPreCondition>>	designate a constraint as being a pre-condition expressed in the context of a requirements project.
<<RequirementPostCondition>>	designate a constraint as being a post-condition expressed in the context of a requirements project.
<<RequirementException>>	designate a constraint as being an exception expressed in the context of a requirements project.
<<RequirementConstraint>>	designate a constraint as being a constraint expressed in the context of a requirements project.

For further information on constraints, please see the "*Constraint class*" section in chapter 4 of the *Objecteering/Metamodel* user guide.

Stereotypes on a dependency

The ... stereotype	is used to ...
<<Traceability>>	define the dependency as being a traceability link, used in the context of a requirements project.

For further information on dependencies, please see the "*Dependency class*" section in chapter 4 of the *Objecteering/Metamodel* user guide.

Tagged values

Tagged value on a document

The ... tagged value	is used to ...
{Requirement}	define that a document is a requirement document, used in the context of a requirements project.

Notes

The ... note	on an ...	is used to ...
description	element	provide a description of the element in question.

Chapter 9: Parameterizing Objecteering/UML Requirements

Overview of module parameterization

Introduction

The *Objecteering/UML Requirements* module provides the user with the possibility of customizing the following parameters:

- general documentation generation parameters
- the editors used to visualize generated documentation
- RTF generation parameters

The "Modifying configuration" window

The window through which *Objecteering/UML Requirements* can be configured is opened either by clicking on the  "*Modify module parameter configuration*" (Figure 9-1) icon or by clicking on the "*Tools/Modify configuration...*" menu in the Objecteering/UML menu bar.

Figure 9-1. Launching the "*Modifying configuration*" window

Parameter sets

The "General" parameter set

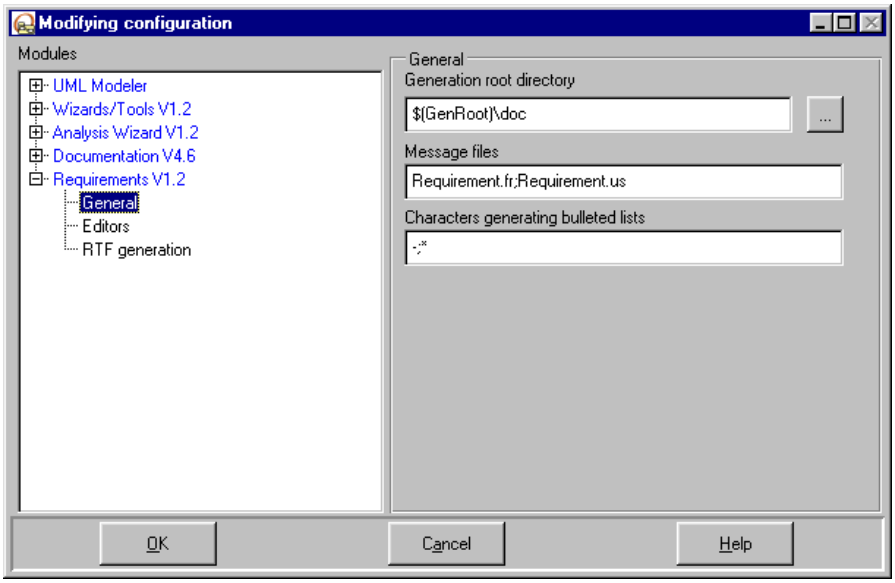


Figure 9-2. The "*General*" parameter set

The ... field	indicates ...
Generation root directory	the directory in which all the documentation files will be stored by default.
Message files	the files that contain the headers of the titles and messages of the documentation generation. This list will be proposed when the product is created. Each file must be separated by the ";" character.

The "Editors" parameter set

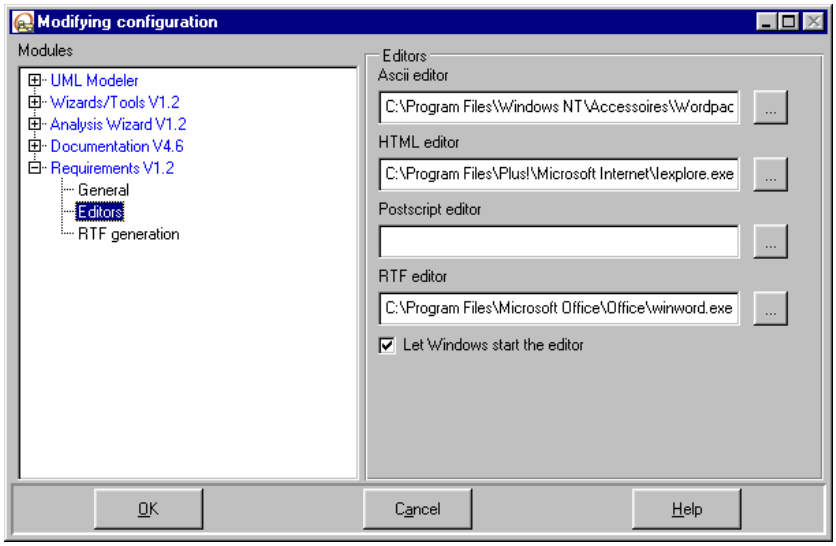


Figure 9-3. The "*Editors*" parameter set

The ... field	indicates ...
Ascii editor	the ASCII edition tool.
HTML editor	the tool which is used to visualize HTML documents (Netscape, Mosaïc, Internet Explorer)
Postscript editor	the tool which is used to visualize a postscript document (ghostview).
RTF editor	the rtf edition tool.
Let Windows start the editor	whether Windows should determine which editor to run. If this tickbox is checked, the parameters which define editors are not used and Windows determines which editor is to be run, according to the type of file and on what has been defined in the registry.

The "RTF generation" parameter set

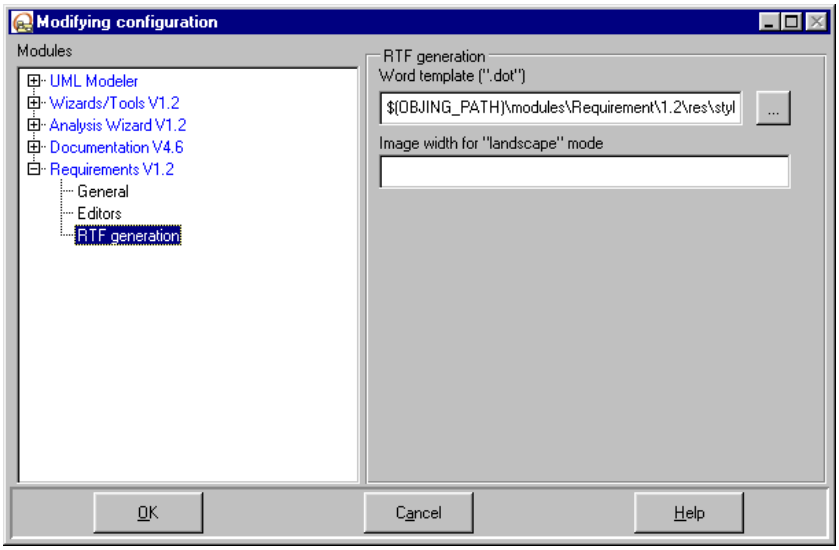


Figure 9-4. The "*RTF generation*" parameter set

The ... field	indicates ...
Word Template (".dot")	the Word model associated with the generated document.
Image width for "landscape" mode	the minimum width expressed in pixels, from which the page containing the diagram is generated in "landscape" mode. This parameter is only used if the width is greater than the height of the diagram.

Chapter 10: The Objecteering/UML Requirements DTD

DTD used in the import/export of requirements

Introduction

This section of the *Objecteering/UML Requirements* user guide provides details of the DTD used in the import and export of requirements to and from outside sources. This DTD is notably used when generating an XML file after marking a Word document (for further details on using Word with *Objecteering/UML Requirements*, please refer to chapter 7 of this user guide).

DTD structure

The following is the DTD used in the import/export of requirements.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  ******************************
  ** XML Requirement exchange DTD      **
  ** Copyright Softeam 2002            **
  ******************************
  -->
<!ELEMENT ObjecteeringRequirementsModel (Info*,
RootPropertySet, RequirementContainer?, Dictionary?, Info*)>
<!ATTLIST ObjecteeringRequirementsModel
      Origin CDATA #REQUIRED
>
<!ELEMENT RootPropertySet (PropertySet*, RootEnumeration)>
<!ELEMENT PropertySet (Info*, Description?, Property*)>
<!ATTLIST PropertySet
      ID ID #REQUIRED
      Name CDATA #REQUIRED
>
<!ELEMENT Property (Info*, Description?)>
<!ATTLIST Property
      ID ID #REQUIRED
      Name CDATA #REQUIRED
      Type (integer | text | multiText | date | enumeration)
            #REQUIRED
      DefaultValue CDATA #IMPLIED
      EnumerationTypeID IDREF #IMPLIED
      isEditable (true | false) #IMPLIED
      applicableFor (Containers | Requirements |
                     ContainersAndRequirements) #IMPLIED
>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Info (#PCDATA)>
<!ATTLIST Info
      Vendor CDATA #REQUIRED
      Key CDATA #REQUIRED
      Value CDATA #IMPLIED
>
<!ELEMENT RootEnumeration (Enumeration*)>
<!ELEMENT Enumeration (Description?, (Info |
EnumerationLitteral)*)>
<!ATTLIST Enumeration
      ID ID #REQUIRED
      Name CDATA #REQUIRED
>
```

```
<!ELEMENT EnumerationLitteral (Info*, Description?)>
<!ATTLIST EnumerationLitteral
      ID ID #REQUIRED
      Name CDATA #REQUIRED
>
<!ELEMENT RequirementContainer (Info*, Description?,
 (PropertyValue | Info | RequirementContainer |
  Requirement)*, Traceability*)>
<!ATTLIST RequirementContainer
      ID ID #REQUIRED
      Name CDATA #REQUIRED
PropertySetID IDREF #IMPLIED
>
<!ELEMENT Dictionary (Info*, Description?, (Info |
Dictionary
 | Term |PropertyValue)*, Traceability*)>
<!ATTLIST Dictionary
      ID ID #REQUIRED
      Name CDATA #REQUIRED
>
<!ELEMENT Requirement (Info*, Description?, (Info |
 PropertyValue | Traceability)*)>
<!ATTLIST Requirement
      ID ID #REQUIRED
      Name CDATA #REQUIRED
>
<!ELEMENT Term (Info*, Description?, (Info | PropertyValue |
 Traceability)*)>
<!ATTLIST Term
      ID ID #REQUIRED
      Name CDATA #REQUIRED
>
<!ELEMENT PropertyValue (Info*)>
<!ATTLIST PropertyValue
      PropertyID IDREF #REQUIRED
      Value CDATA #REQUIRED
>
<!ELEMENT Traceability (Info*)>
<!ATTLIST Traceability
      DestinationID IDREF #REQUIRED
>
```

Index