UML-Editor Referenzhandbuch
the Art of modelling...

Peter Hirzel

V1.0.4 (30th December 2003)

Contents

List of Figures

Preface

The wish to enhance application of the *model-based method* motivated the creation of the UML-Editor .

In this sense the editor is unique, since it permits the modeling of UML/INTERLIS, i.e. the synthesis of two standards within the scope of data-modeling.

INTERLIS is a specific form of UMLwhich permits the automatic derivation of different formats (amongst others the XML-Schema).

This tool is an attempt to facilitate the application and thus the mastery of the very complex matter of UML and INTERLIS thanks to a intuitive device and hence to render it accessible to a greater number of users.

Thus we express due thanks to KOGIS because without their support this UML/INTERLIS-editor could never have been achieved.

Chapter 1

Regarding This Document

- Chapter 2 gives general information concerning the UML-editor.
- Chapter 3 describes function and interface of the UML-editor.
- Chapter 4 describes specific model elements (objects) of the UML-editor.

Chapter 2

General Remarks

This reference manual has been conceived as an accompanying document for the UML-editor. Thus it offers assistance in the use of functions and explain procedures of the tool. At the same time it displays the technical possibilities of the editor in the modeling of UML and INTERLIS.

Modeling with the UML-editor requires a certain knowledge in UML (siehe Kap. B) [1] and INTERLIS (siehe Kap. C.1). Therefore we do not enter into these topics, because they would be beyond the scope of this reference manual. You will find further informationen concerning UML in the Internet and for INTERLIS there is an *INTERLIS Reference Manual*[6] which offers ample explication concerning the use of INTERLIS by means of practical examples.

In addition to this manual there is also an *Introduction to the UML/INTERLIS-Editor* [7] with a step-by-step explanation of modeling with the UML-editor.

2.1 Installation

For further instructions concerning the installation of the UML-editor see [7].

2.2 Licence Terms

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

2.3 Mandator

KOGIS c/o Bundesamt für Landestopographie
Seftigenstrasse 264, Postfach, CH-3084 Wabern
Telephon: +41 31 963 21 11 – Fax: +41 31 963 23 25
http://www.kogis.ch – E-mail: kogis@swisstopo.ch

2.4 Developer

Eisenhut Informatik AG
Claude Eisenhut, Dipl. Informatik-Ingenieur HTL
Rosenweg 14, CH-3303 Jegenstorf
Telephon: +41 31 762 06 62 – Fax: +41 31 762 06 64
http://www.eisenhutinformatik.ch – E-Mail: info@eisenhutinformatik.ch

Assistants:

*soft*Environment
Peter Hirzel, Dipl. Informatik-Ingenieur HTL, NDS Umwelt
Rüttiweg 7, CH-3047 Bremgarten
Telephon: +41 79 746 67 40
http://www.softenvironment.ch – E-Mail: info@softenvironment.ch

2.5 Software applied

- Java SDK (cf. http://java.sun.com)
- JHotDraw (Graphic) (cf. http://sourceforge.net/projects/jhotdraw)
- Apache Software Foundation (XML) (cf. http://www.apache.org)

Chapter 3

User Interface

On principle the UML-Editor siehe Abb. 3.1 represents the most important model elements (siehe Kap. 4) hierarchically in the Navigationsbereich as a tree-structure (siehe Kap. 3.3). Some few elements (e.g. generalization, dependencies, syntax, etc.) do not really make sense in the Navigationsbereich and hence will be suppressed by the UML-Editor . However these model elements can still be maintained via the specification dialogs (siehe Kap. 4.2) .

A possible element in the Navigationsbereich is the Klassendiagramm (siehe Kap. 4.1), which permits *graphic modeling*. Thus it is possible to represent model elements from the Navigationsbereich in a class diagram. The same model element may appear in several diagrams in order to further visualize different aspects of the same data model. However graphic representation is not possible for all model elements. Class diagrams can be generated in the Navigationsbereich and then opened in the Modellierbereich (siehe Kap. 3.4) .

3.1 Menüleiste

The menu list contains the names of the menus. By clicking a menu name a list of commands will appear, which in turn controle a series of functions of the UML-editor.

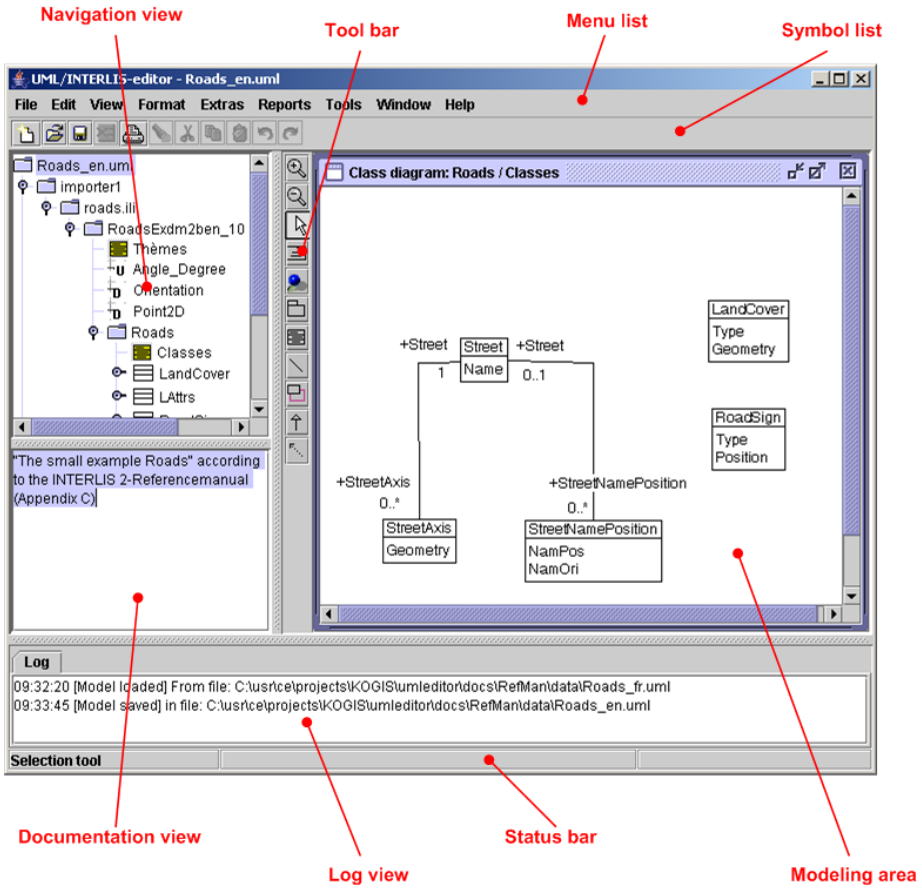

Figure 3.1: UML-Editor with example-model ROADS

3.1.1 File

Funktion	Beschreibung
New	Generates a new Modell.
Open...	Opens a file Dialog for the selection of a model file.
Save	Saves the present Modell with the file name indicated.
Save as...	Opens a file Dialog for the saving of a file name with a different name.
Print...	Opens a print Dialog.
Close	Closes the program.

3.1.2 Editing

Funktion	Beschreibung
Undo	(Diese Funktion ist z.Z. nicht implementiert.)
Restore	(Diese Funktion ist z.Z. nicht implementiert.)
Cut	(Diese Funktion ist z.Z. nicht implementiert.)
Copy	(Diese Funktion ist z.Z. nicht implementiert.)
Insert	(Diese Funktion ist z.Z. nicht implementiert.)
Select all	(Diese Funktion ist z.Z. nicht implementiert.)
Search/Replace...	Opens a search-Dialogsiehe Abb. 3.2, to find elements that correspond to the search criteria. By selecting an element in the *search result list* the relevant element is automatically selected in the Navigationsbereich .

3.1.3 View

Funktion	Beschreibung
Look & Feel	Various representation managers can be selected (independent of platform).
Symbol lists	The standard tool bar (siehe Kap. 3.2) can be activated/ deactivated.
Status bar	The Statusleiste can be activated/ deactivated.

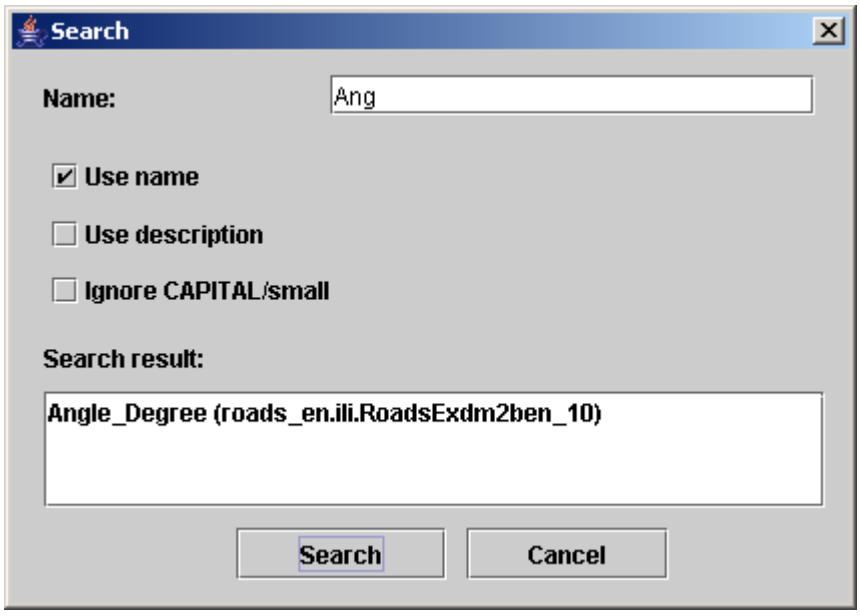

Figure 3.2: Dialog – Search/Replace

3.1.4 Formatting

Funktion	Beschreibung
Diagramm anordnen	The contents of the present diagram are automatically adjusted. The function attempts to distribute the model elements as evenly as possible in the diagram and at the same time to avoid as much as possible crossings of lines. In general when using this function it is necessary to manually improve this arrangement.

3.1.5 Extras

Funktion	Beschreibung
Options...	An options dialog siehe Abb. 3.3, siehe Abb. 3.4 is opened. The necessary configurations are stored in the file *.umleditor* in your personal directory (${user.home}) .

Feld	Beschreibung
Work directory	Standard directory for the saving and opening of files.
Import directory	Standard directory for the import of data.

Feld	Beschreibung
Width (Standard)	Standard width for new class diagrams.
Height (Standard)	Standard height for new class diagrams.

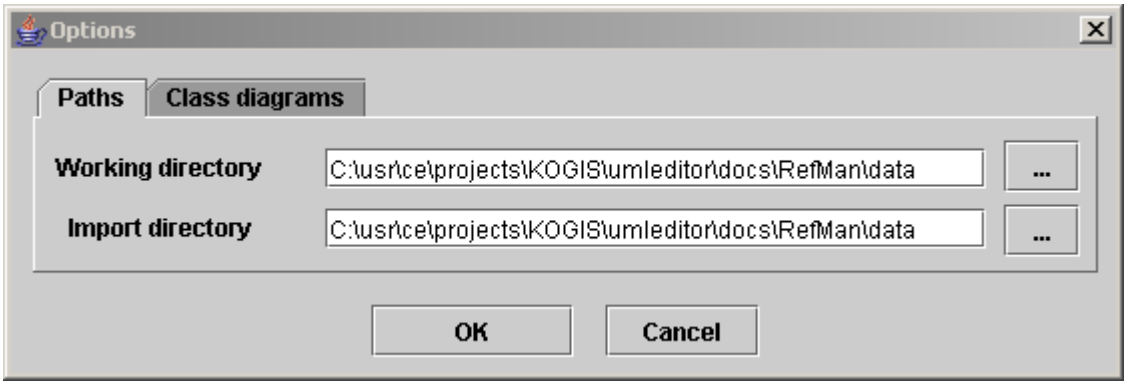

Figure 3.3: Dialog – Options (Reiter *paths*)

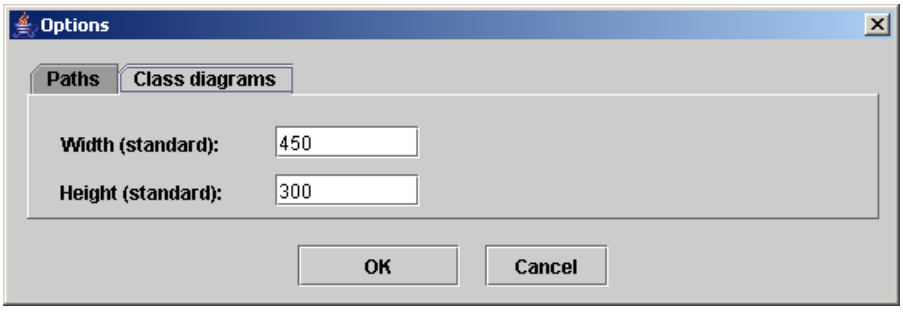

Figure 3.4: Dialog – Options (Reiter *Class diagrams*)

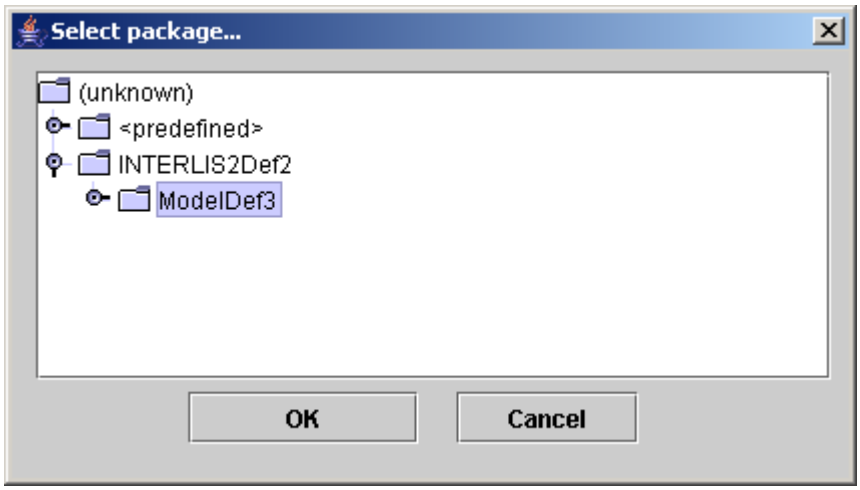

Figure 3.5: Dialog – Package selectionDialog

These values define the minimal size of a diagram and thus also the size of a new empty diagram. Depending on the size of your screen it may be sensible to alter these values. If some model elements are situated at the lower or right edge of a diagram, this diagram is automatically enlarged even without previously altering the corresponding values.

3.1.6 Reports

Funktion	Beschreibung
Object catalog...	Opens a package selection dialog siehe Abb. 3.5 for the selection of a package. For the package thus selected, model objects are catologued in an HTML-report siehe Abb. 3.6.
Structure...	Opens a package selection dialog siehe Abb. 3.5 for the selection of a package. For the package thus selected the corresponding package structure is generated in an HTML-report siehe Abb. 3.7.

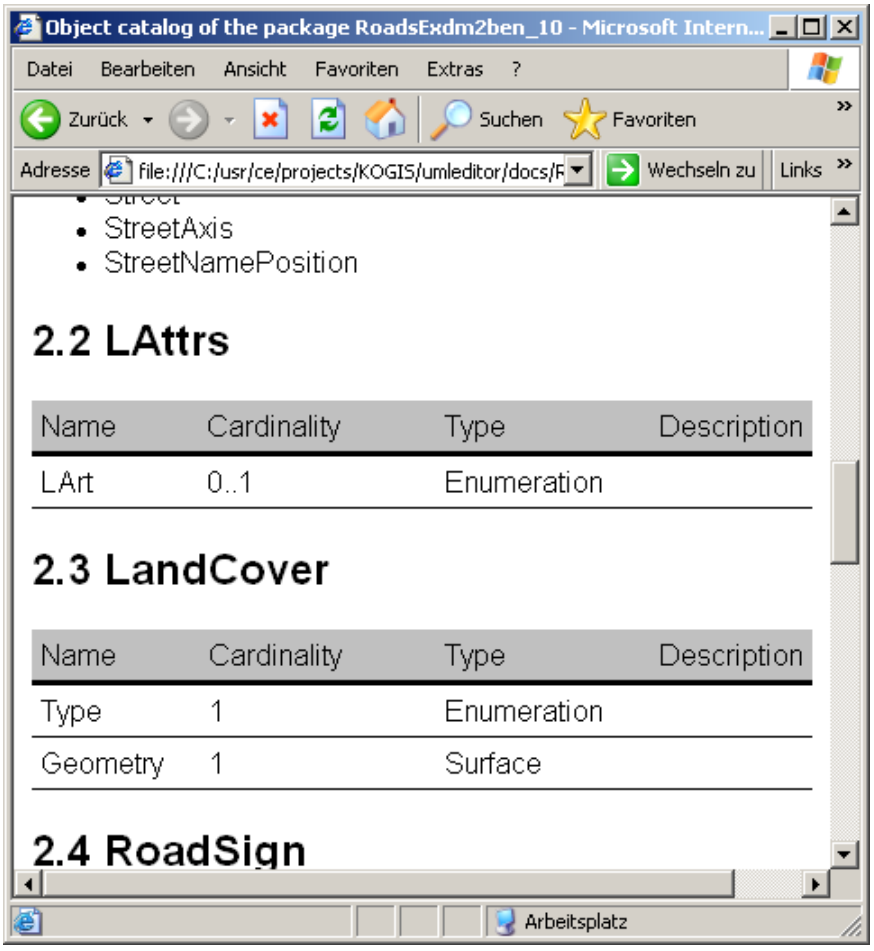

Figure 3.6: Dialog – Report *Object catalog*

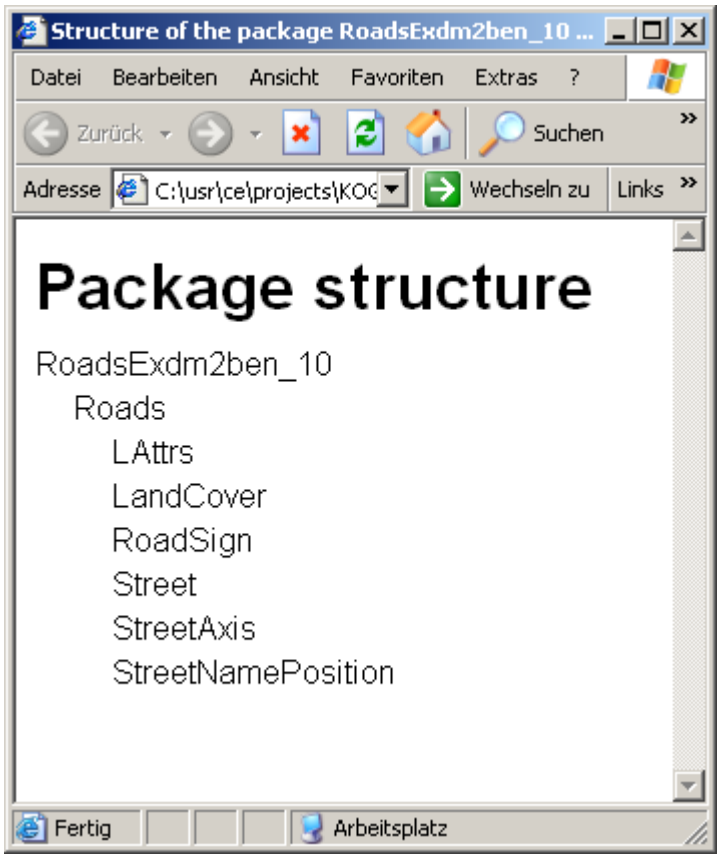

Figure 3.7: Dialog – Report *Structure*

3.1.7 Tools

INTERLIS

Funktion	Beschreibung
Import...	Opens a fileDialogin order to import INTERLIS-model files into the Modell (siehe Kap. D.4).
Import groups...	(Diese Funktion ist z.Z. nicht implementiert.)
Export...	Generates the INTERLIS-model files (siehe Kap. D.4) according to the present Modell in the work directory (siehe Kap. 3.1.5).
Export XML-Schema ...	Opens a file-Dialogfor the export of the XML-Schema (XSD) (siehe Kap. D.2). The XML-Schema thus generated describes the transfer format.
Model check	The model is examined by means of the INTERLIScompiler (siehe Kap. C.2). Possible errors appear in the Logbereich with a Identifikationsnummer (siehe Kap. 3.6).

XMI/ROSE

Funktion	Beschreibung
Import...	Opens a file Dialogto import a model file exported with Rational-Rose via XMI .

3.1.8 Window

Funktion	Beschreibung
Cascade	Several windows in the Modellierbereich will be super-positioned one behind the other.
Distribute	Several windows in the Modellierbereich will super-positioned one below the other.

3.1.9 Help

Funktion	Beschreibung
Help...	Opens a help-line in a browser.
Info...	Opens an info Dialog with information concerning the program.

3.2 Symbolleiste

Depending on the context, individual functions are activated or deactivated (by activating the symbols in the symbol list). By positioning the mouse cursor on a symbol, the corresponding description of the function appears in text form.

 *New* (siehe Kap. 3.1.1).
 *Open...* (siehe Kap. 3.1.1).
 *Save* (siehe Kap. 3.1.1).
 *Print* (siehe Kap. 3.1.1).

3.3 Navigationsbereich

The Navigationsbereich represents the entire Modell as a tree structure . Thus the user may see how the model is organized and the view can be adjusted to the current needs of the user by opening/closing sub-trees A context-sensitive menu will show the possible functions for each selected Modell-Element in the tree:

Funktion	Beschreibung
New	Permits the inserting of a new model element. Depending on the selection various elements are available (siehe Kap. 4).
Modify...	Opens the specification dialog for the corresponding element Modell-Element (siehe Kap. 4.2).
Activate diagram	Opens or displays the selected Diagramm in the foreground within the modeling range.
Sort	Arranges the tree structure according to the possible selection, criterias being *Name* or *Type/Name*.
Print...	Opens a print dialog.
Delete (in the model)	Deletes the selected Modell-Element from the Modell.
Rename	Permits direct renaming of the selected element.
Insert in diagram	Inserts the selected Modell-Element into the momentarily active Diagramm .

3.4 Modellierbereich

In the Modellierbereich diagrams (siehe Kap. 4.1) that typically display a prominent detail from the Modell can be represented in their own (internal) windows.

3.4.1 Werkzeugleiste

The Werkzeugleiste is dynamically adjusted to the currently selected diagram (siehe Kap. 4.1), i.e. the tools that are admissible for the corresponding diagram type are automatically displayed and activated.

The following tools are supported by the UML-Editor :

 *Enlarges* the current Diagramm.

 *Reduces* the current Diagramm.

 Permits the *selection* of a model element.

 Permits inserting of a *note*.

 Permits *connecting a note* with another (knot) Modell-Element.

 Permits inserting of a *package*.

 Permits inserting of a *Klasse* .

 Permits *connecting* two classes by means of a relationship.

 Permits the creation of a *reflexive relationship* of a Klasse.

 Permits the *inheritance (generalization)* between two model elements.

 Permits the creation of a *dependency* between two model elements.

3.5 Dokumentationsbereich

The documentation view displays descriptions concerning an Modell-Element , e.g. by means of selection:

- in the Navigationsbereich(siehe Kap. 3.3)
- in the current Diagramm

A popup menu permits all common editing operations (siehe Kap. 3.1.2).

3.6 Logbereich

The Logbereich siehe Abb. 3.8 features run-time messages (e.g. when saving the model or during model check (siehe Kap. 3.1.7)). Depending on the selection the following functions of the popup menu are possible:

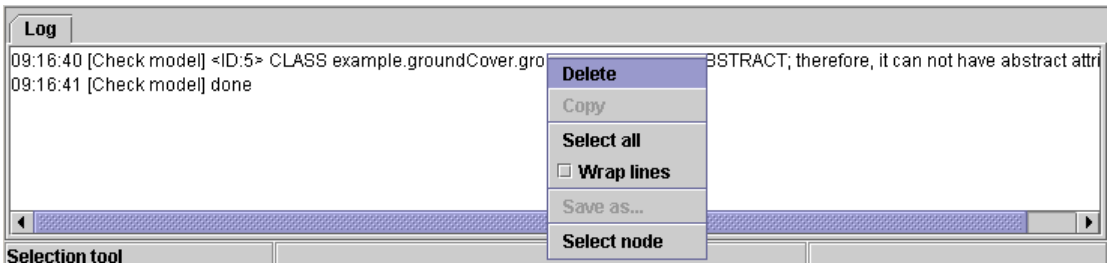

Figure 3.8: Logbereich Select knots (according to Identifikationsnummer)

Funktion	Beschreibung
Delete	Deletes the entire display in the Logbereich.
Copy	Inserts the selected messages in the copy buffer of the system.
Select all	Selects all messages.
Make up lines	Messages that are longer than the current window width are made up into two lines by the UML-Editor .
Save as...	Permits the saving of all messages in the Logbereich of a file.
Select knots	Applies only to special messages with Identifikationsnummer. This function selects the Modell-Element concerned by this message in the Navigationsbereich.

3.7 status bar

The Statusleiste is set up in three sections:

- *Left box* indicates which tool from the Werkzeugleiste is currently activated.
- *Middle box* (Diese Funktion ist z.Z. nicht implementiert.)
- *Right box* (Diese Funktion ist z.Z. nicht implementiert.)

Chapter 4

Modeling Elements

Hereafter we describe model elements that can be modeled by means of the UML-Editor . Based upon the language definition and resulting rules of INTERLIS there is a valid selection of sub-elements for each Modell-Element , These will be automatically supported by the UML-Editor .
This is the particular strong point of the UML-editor, a user need not worry whether he is developping a valid model or not. Whatever is rendered possible by the editor, is also valid within the scope of UML and INTERLIS . Special cases can always be verifies by means of the INTERLIS-Compiler (siehe Kap. 3.1.7).

4.1 Klassendiagramm

The class diagrams strictly respect the rules of UML (siehe Kap. B). All possible tools are automatically activated when opening a class diagram in the *Werkzeugleiste* (siehe Kap. 3.4.1).

A typical use of class diagrams is the representation of packages siehe Abb. 4.1 or classes siehe Abb. 4.2:

4.1.1 Popup-Menu for the Diagramm

The following functions, affecting the entire Diagramm , are at your disposal:

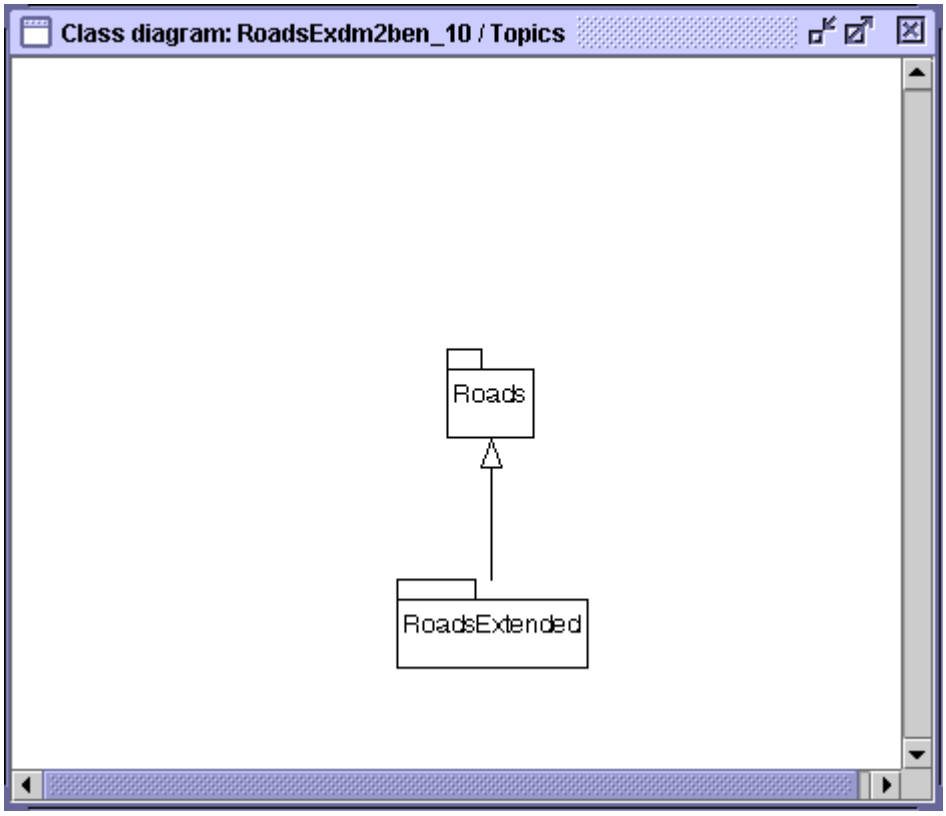

Figure 4.1: Klassendiagramm – Representation of packages

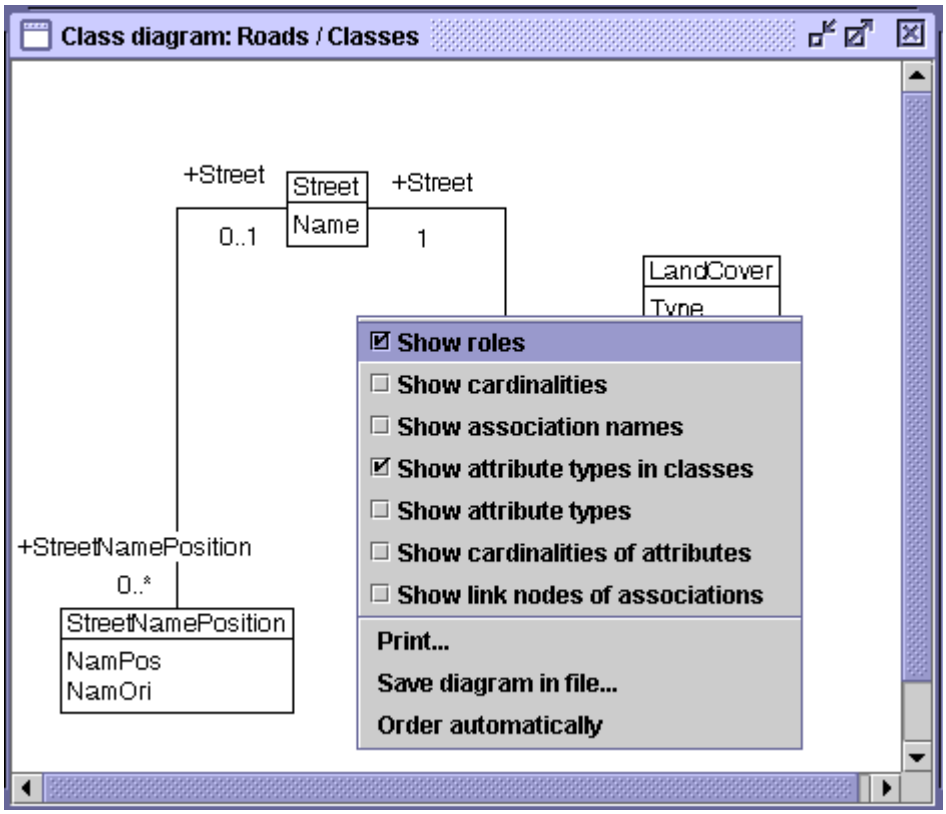

Figure 4.2: Klassendiagramm – Representation of classes with Popup-Menu zum Diagramm

FUNKTION	BESCHREIBUNG
Display role	Represents the roles (siehe Kap. 4.2.11) of a relationship or not. Rollen werden mit einem fuehrenden Pluszeichen (z.B. *+Street* siehe Abb. 3.1) dargestellt.
Display cardinality	Represents the cardinalities of relationships or not (e.g. ***0..**** siehe Abb. 3.1).
Display association names	Represents the names of all relationships in the Diagramm dar or not.
Display attributes in classes	Represents the attributes of all classes in the diagram or not.
Display type of attribute	Represents the data type for all attributes displayed in all classes in the diagram or not.
Display cardinality of attributes	Represents the cardinality of all attributes displayed in all classes in the diagram or not .
Display link knot of associations	The link knot is an "artificial device", that enables graphic links between particular cases of relationships (siehe Kap. 4.2.9) in the Diagramm .
Print...	Opens a print dialog for the printing of the diagram.
Save...in diagram file	Opens a file-Dialog for the saving of the diagram.
Automatic reorganization	Automatic reorganization of the contents of a diagram.

4.1.2 Popup-Menu of Modell-Element

General Functions

For each Modell-Element the following *general functions* are available siehe Abb. 4.3:

FUNKTION	BESCHREIBUNG
Modify...	Opens the specification Dialog (siehe Kap. 4.2) for an Modell-Element.
Selecting in the navigation pane	Indicates the selected graphic object in the Navigationsbereich.

Sub-menu *Formatting*:

FUNKTION	BESCHREIBUNG
Font type...	(Diese Funktion ist z.Z. nicht implementiert.).
Line color...	Opens a color dialog for the selection of a line color (e.g. for the coloring of associations).
Fill-in color...	Opens a color dialog for the selection of a fill-in color (e.g. for the coloring of class symbols).

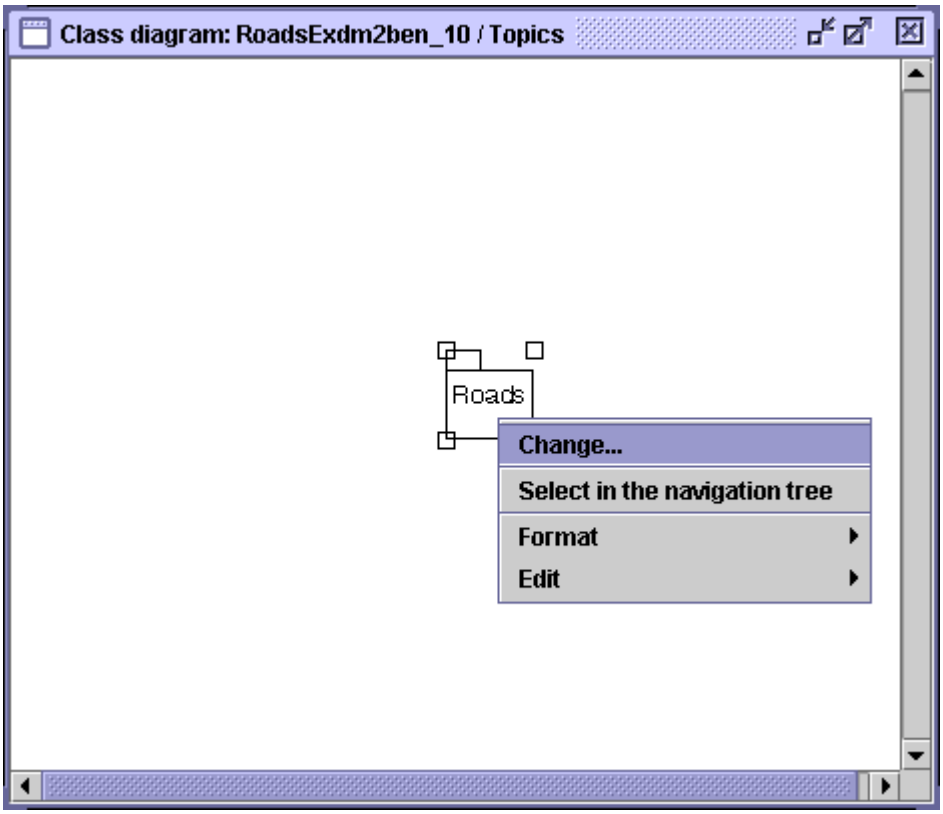

Figure 4.3: Klassendiagramm – General remarks Popup-Menu about Modell-Element

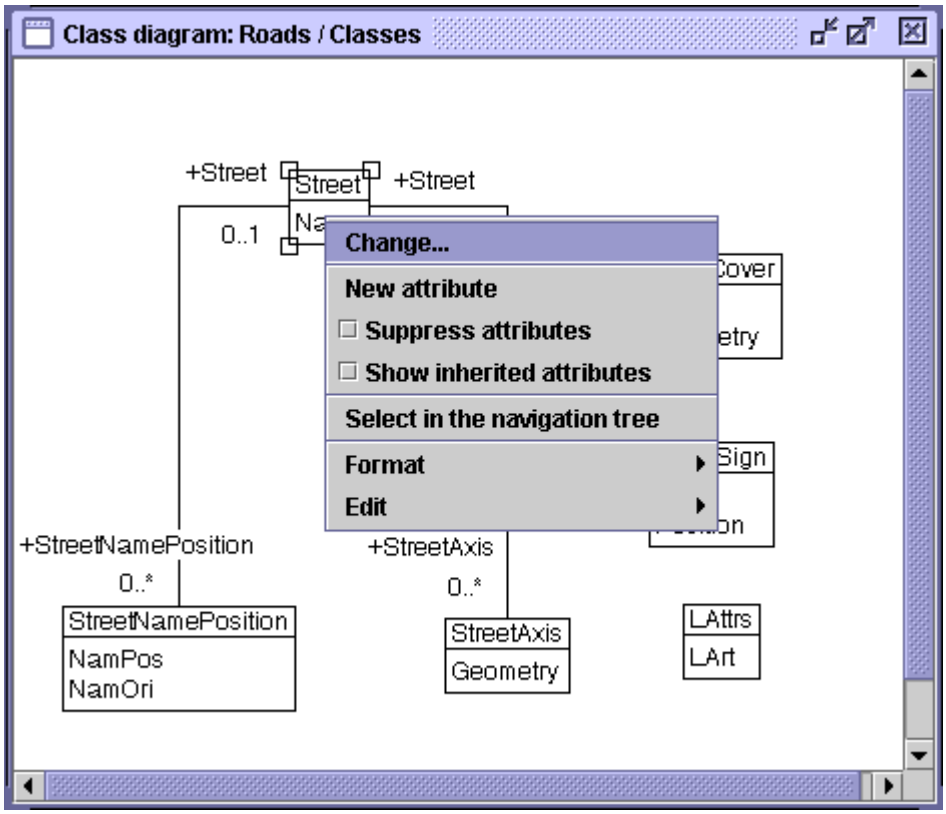

Figure 4.4: Klassendiagramm – Special functions of a Klasse

Sub-menu *Editing*:

Funktion	Beschreibung
Cut	(Diese Funktion ist z.Z. nicht implementiert.).
Copy	(Diese Funktion ist z.Z. nicht implementiert.).
Paste	(Diese Funktion ist z.Z. nicht implementiert.).
Delete	The selected Modell-Element is only graphically deleted in the current diagram, i.e. it remains in the Navigationsbereich (and thus in the model).
Delete in the model	The selected Modell-Element is deleted graphically in the current diagram as well as definitively in the model. In addition all model dependencies are deleted (e.g. Roles in classes with connected associations).

Class-specific functions

siehe Abb. 4.4

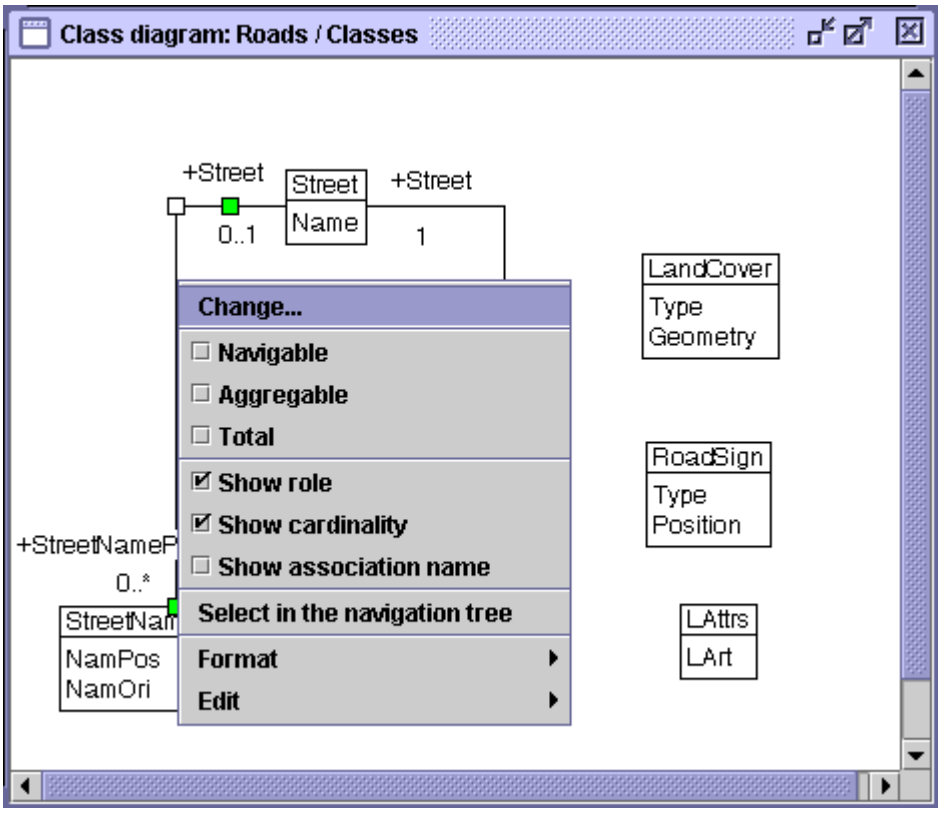

Figure 4.5: Klassendiagramm – Special functions of an Beziehung

Funktion	Beschreibung
New attribute	Adds a new attribute (siehe Kap. 4.2.10) to the class.
Hide attributes	Activates/ Deactivates representation of attributes for this class.
Display inherited attributes	In addition to the attributes defined in this class, the attributes of the basic class(es) are also displayed.

Relation-specific functions

siehe Abb. 4.5

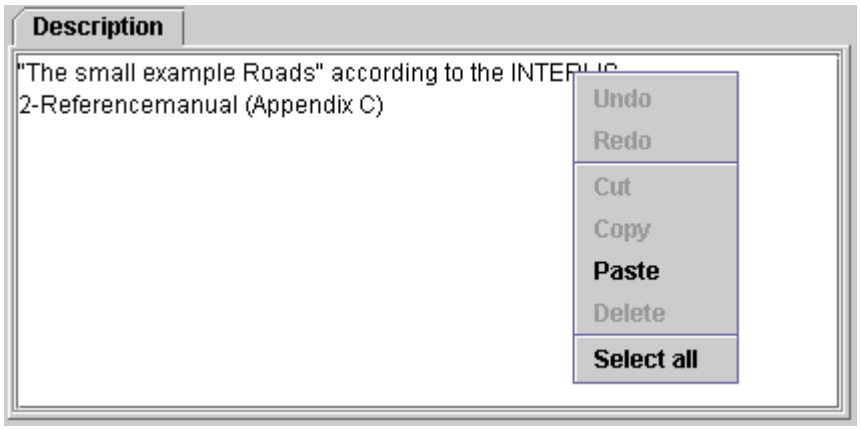

Figure 4.6: Dialog (Reiter *description*)

FUNKTION	BESCHREIBUNG
Association	Defines the *direction of navigation* towards a role (siehe Kap. 4.2.11) on the corresponding Beziehung. In accordance with UML this is represented optically by an open arrow in the Klassendiagramm .
Aggregation	Characterizes the role (siehe Kap. 4.2.11) of an Beziehung as *aggregation*. In accordance with UML this is represented optically with a void rhombus in the Klassendiagramm .
Composition	Characterizes the role (siehe Kap. 4.2.11) of an Beziehung as *composition*. In accordance with UML this is represented optically with a filled in rhombus in the Klassendiagramm .
Display role	Activates / Deactivates the role name (siehe Kap. 4.2.11) of a relationship (siehe Kap. 4.1.1).
Display cardinality	Represents the *cardinality* of the roles (siehe Kap. 4.2.11) of an Beziehung or not .
Display name of association	Represents the name of an Beziehung or not.

4.2 Dialogs

On principle there is a specific dialog for each Modell-Element to view or modify the properties of this element. Each Modell-Element-Dialog features a minimum of the following tabs siehe Abb. 4.6:

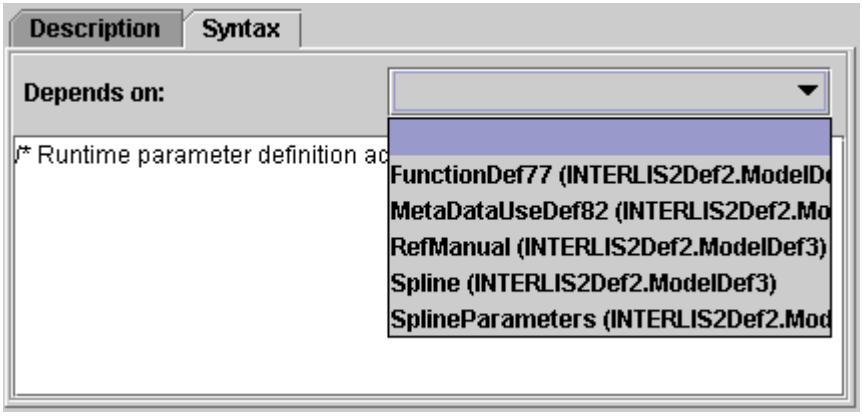

Figure 4.7: Dialog (Reiter *Syntax*)

Feld	Beschreibung
Description	Text entry with Popup-Menu featuring common text functions (siehe Kap. 3.1.2). This text is taken into consideration when generating an object catalog

siehe Abb. 4.7

Feld	Beschreibung
Depends of	Possible dependencies of other *Model elements* can be selected from a list.
INTERLIS Syntax	Text entry with Popup-Menu featuring common text functions (siehe Kap. 3.1.2). Permits entry of syntax code (in accordance with INTERLISrules). This code can be verified by means of the function *Model check* (siehe Kap. 3.1.7).

Each Dialog features the following functions:

Funktion	Beschreibung
OK	Saves all modifications and closes dialog.
Cancel	Rejects all modifications and closes dialog.
Accept	Saves all modifications without closing dialog.

4.2.1 Baskets of reference systems/signatures

siehe Abb. 4.8

Feld	Beschreibung
File name	Assigns an XML-file .
Beschreibung	(siehe Abb. 4.6).

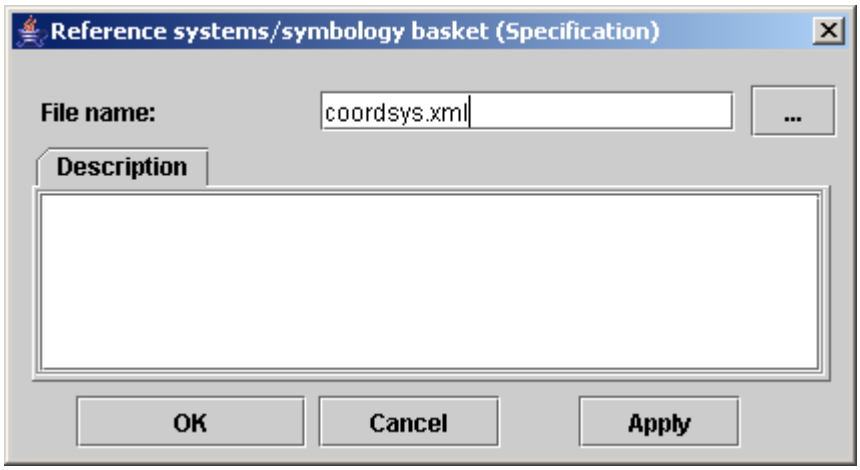

Figure 4.8: Dialog – Containers of reference systems/signatures

4.2.2 UML-Package

A UML-package siehe Abb. 4.9 corresponds to a folder, in which other elements can be stored. A UML-package is a model element which does not exist in INTERLIS and hence does not appear in the exported INTERLIS-model.

Feld	Beschreibung
Name	Name of the *UML-package*
Beschreibung	(siehe Abb. 4.6).

4.2.3 INTERLIS 2-file

A *INTERLIS 2-file* siehe Abb. 4.10 corresponds to a package in accordance with UML.

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.3.

Feld	Beschreibung
Table	Lists language and corresponding file names. A model element *INTERLIS 2-file* represents an INTERLIS-model file (siehe Kap. D.4). If the field file name is a relative file path, then it refers to a directory, in which the UML-editor file is stored. By means of the Popup-Menu-function *Modify...* a table entry can be modified.
Linguistic version	We refer to the version of the INTERLIS-specification applied. This version of the editor supports version 2.2.
Beschreibung	(siehe Abb. 4.6).

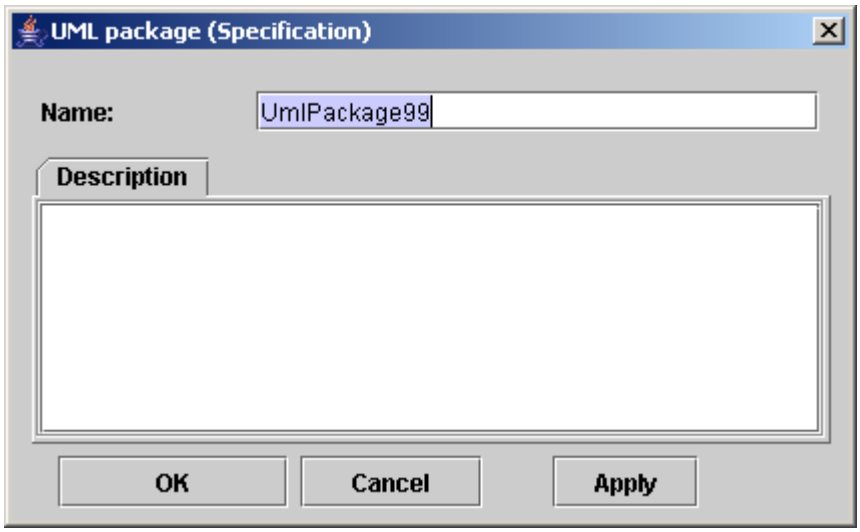

Figure 4.9: Dialog – UML-Paket

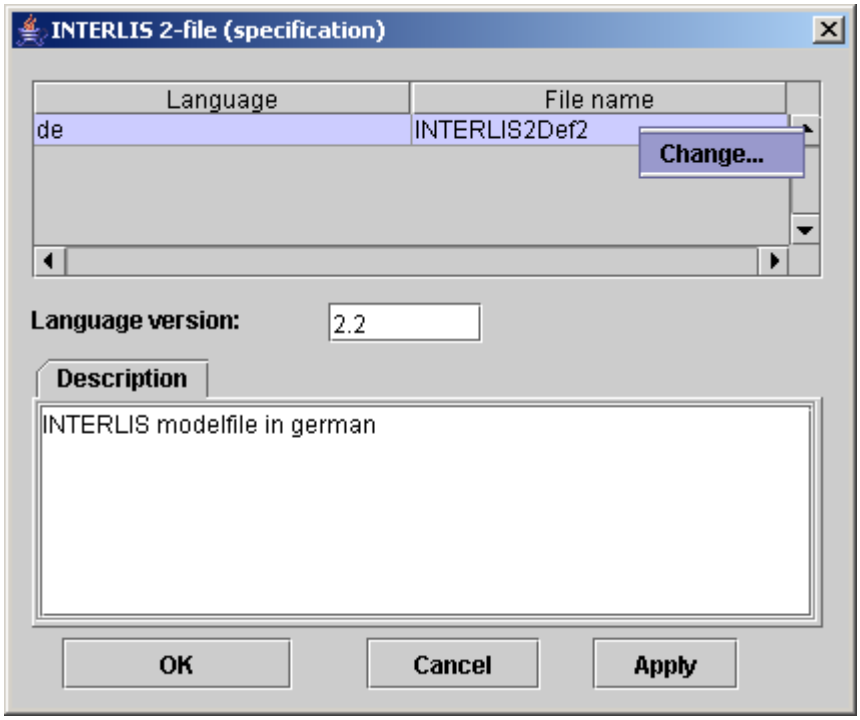

Figure 4.10: Dialog – *INTERLIS 2-file*

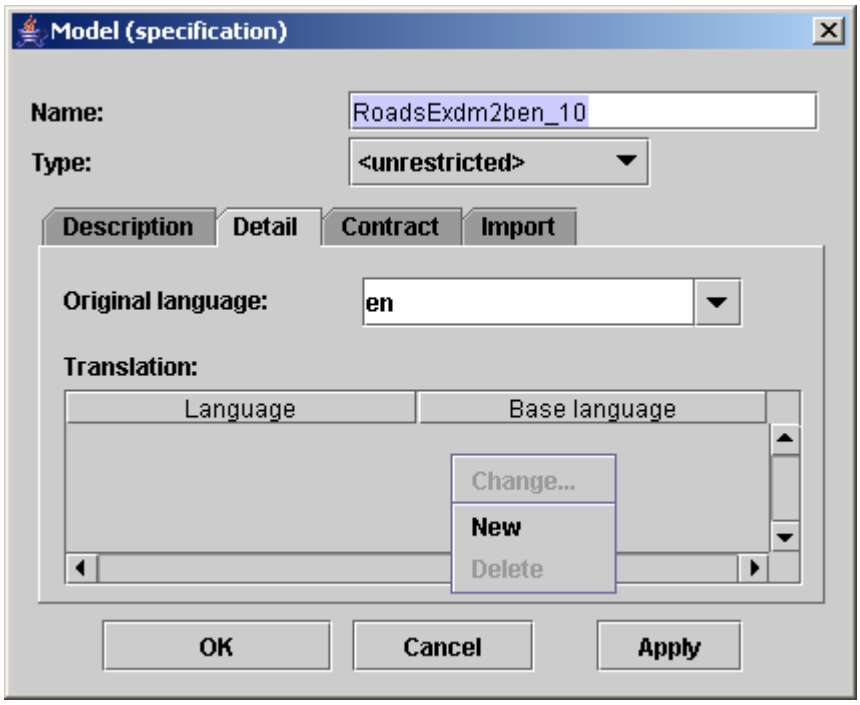

Figure 4.11: Dialog – *Model* (Reiter *Detail*)

4.2.4 Model

A *model* siehe Abb. 4.11 corresponds to a package) in accordance with UML.

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.5.1.

Feld	Beschreibung
Name	Name of the *model*
Type	Characterization (possible options being *types*, *reference systems*, *symbologies* or *unlimited*).
Beschreibung	(siehe Abb. 4.6).
Original language	Initial language of the *Model* .
Translation (table)	A translation Dialog can be opened via a Popup-Menu (siehe Kap. 4.2.5). All translations concerning the *Model* are listed in the table.

siehe Abb. 4.12

Feld	Beschreibung
Contract (Table)	Via Popup-Menu you can open a contract Dialog (siehe Kap. 4.2.6). In this table all *authors* of contracts are listed.

Figure 4.12: Dialog – *Modell* (Reiter *Vertrag*)

Figure 4.13: Dialog – *Modell* (Reiter *Import*)

siehe Abb. 4.13

Feld	Beschreibung
Import (Tabelle)	Via Popup-Menu and by means of a assignation Dialog other *models* can be assigned and administered. In the table all imported *models* are listed. Import relationships can be represented in a package diagram.

4.2.5 Translation of the model

At present multilingual models are not supported by the UML-Editor .

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.5.1. siehe Abb. 4.14

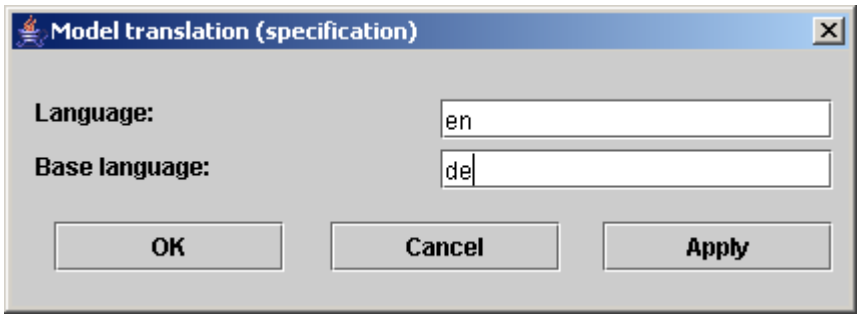

Figure 4.14: Dialog – Translation of model

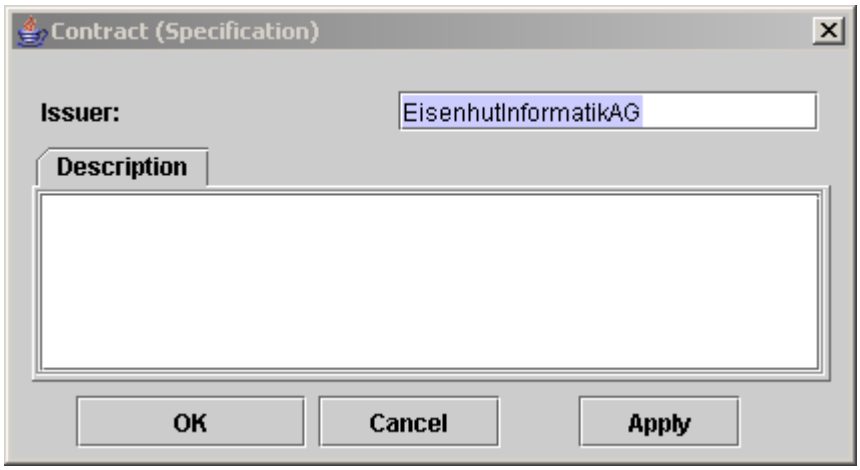

Figure 4.15: Dialog – Contract

Feld	Beschreibung
Language	Target language of the translation.
Basic language	Initial language before the translation.

4.2.6 Vertrag

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 1.7. siehe Abb. 4.15

Feld	Beschreibung
Editor	Author of a contract.
Beschreibung	(siehe Abb. 4.6).

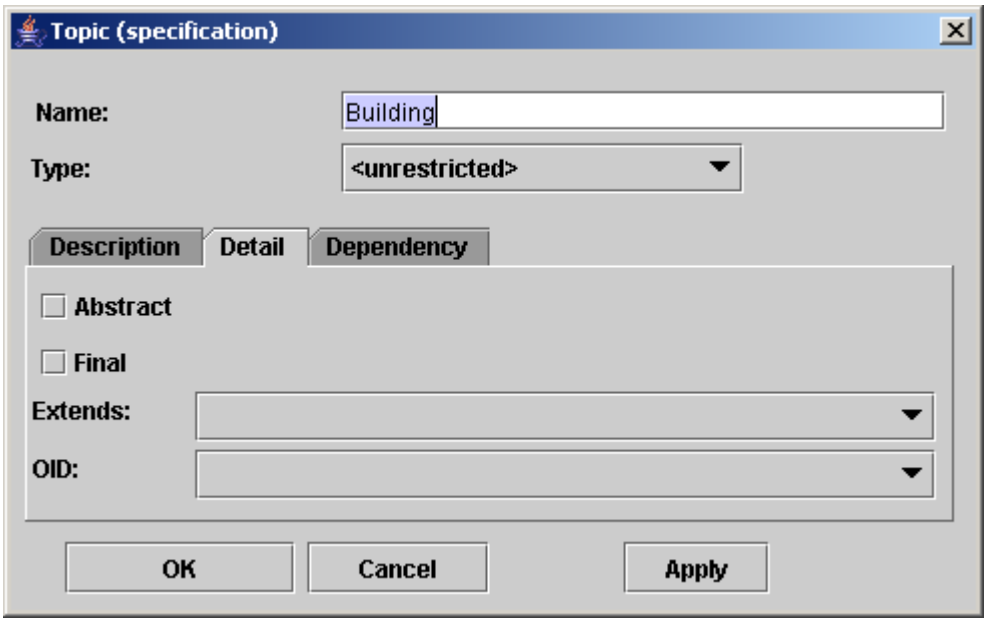

Figure 4.16: Dialog – *Thema* (Reiter *Detail*)

4.2.7 Topic

A *topic* siehe Abb. 4.16 corresponds to a package in accordance with UML.

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.5.2.

Feld	Beschreibung
Name	Name of the *topic*
Type	Characterization (by *views* or *unlimited*).
Beschreibung	(siehe Abb. 4.6).
Abstrakt	Setzt Modell-Element als *Abstrakt* oder nicht.
Abschliessend	Setzt Modell-Element als *Abschliessend* oder nicht.
Erweitert	Erlaubt die Selektion eines allenfalls zu erweiternden *Modell-Elementes* aus einer Liste von Modell-Elementen, die für eine *Spezialisierung* in Frage kommen (Liste wird automatisch vom UML-Editor aufbereitet).

siehe Abb. 4.17

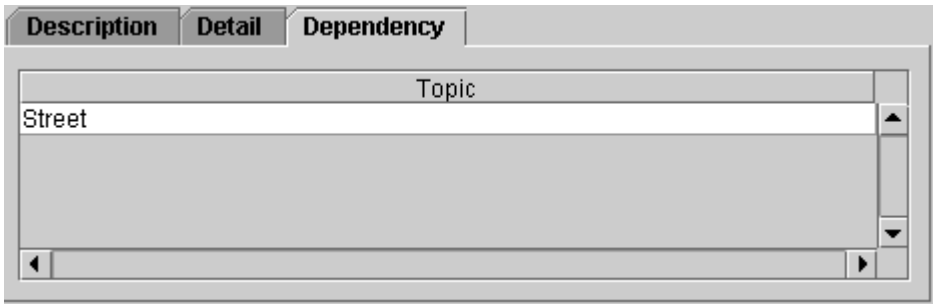

Figure 4.17: Dialog – *Thema* (Reiter *Dependency*)

Feld	Beschreibung
Dependency (Table)	Via Popup-Menu and by means of an assignation Dialog other *topics* can be assigned and processed. In the table all dependent *topics* are listed. The dependency relationships can be represented in a package diagram.

4.2.8 Klasse

A Klasse siehe Abb. 4.18 describes similar objects by means of attributes .

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.5.3.

Feld	Beschreibung
Name	Name of the *class*
Beschreibung	(siehe Abb. 4.6).
Abstrakt	Setzt Modell-Element als *Abstrakt* oder nicht.
Abschliessend	Setzt Modell-Element als *Abschliessend* oder nicht.
Erweitert	Erlaubt die Selektion eines allenfalls zu erweiternden *Modell-Elementes* aus einer Liste von Modell-Elementen, die für eine *Spezialisierung* in Frage kommen (Liste wird automatisch vom UML-Editor aufbereitet).
Typ	A Klasse can either be characterized as a *class* (Default) or as a *structure*.

siehe Abb. 4.19

Feld	Beschreibung
Attribute (Tabelle)	Via Popup-Menu a list of *attributes* (siehe Kap. 4.2.10) can be processed. All defined *attributes* are listed in the table.

siehe Abb. 4.20

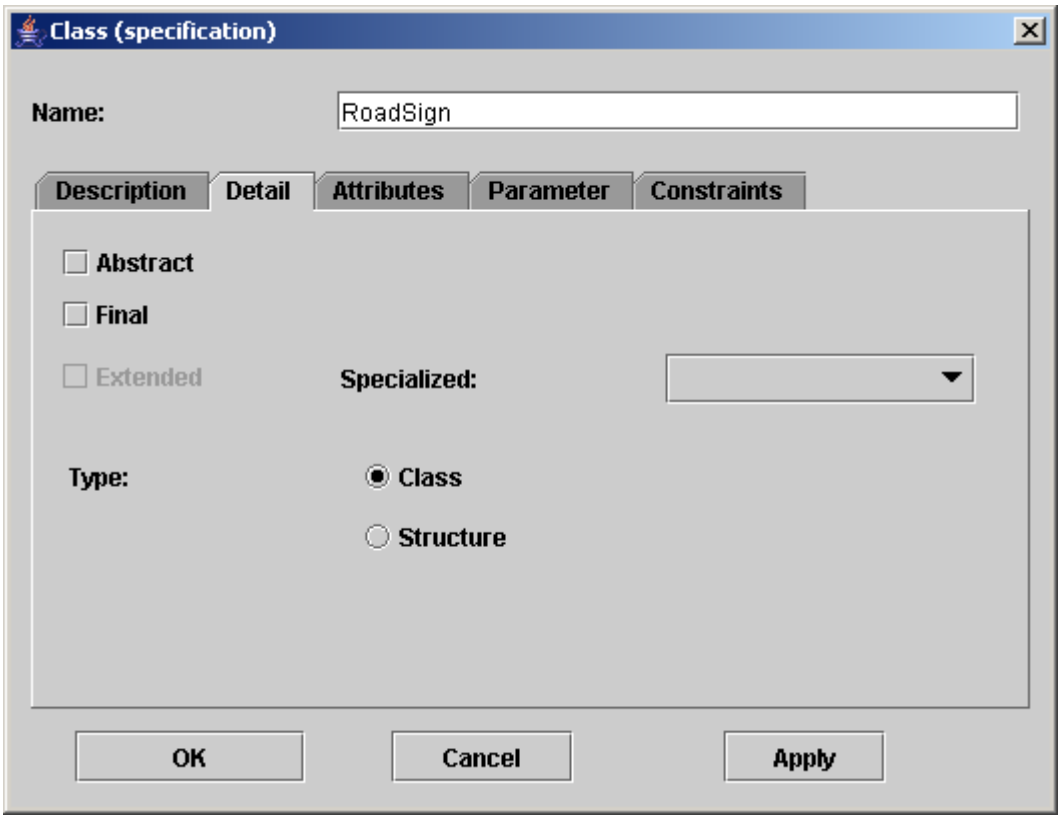

Figure 4.18: Dialog – *Class* (Reiter *Detail*)

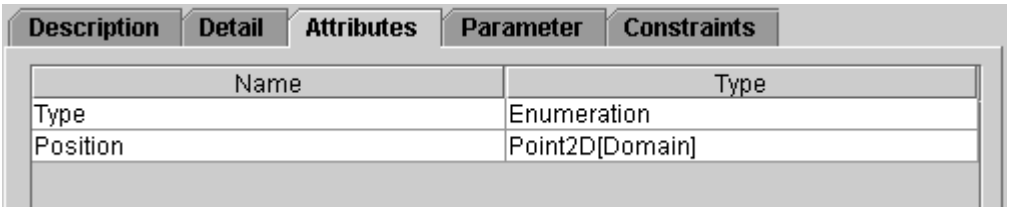

Figure 4.19: Dialog – *class* (Reiter *Attribute*)

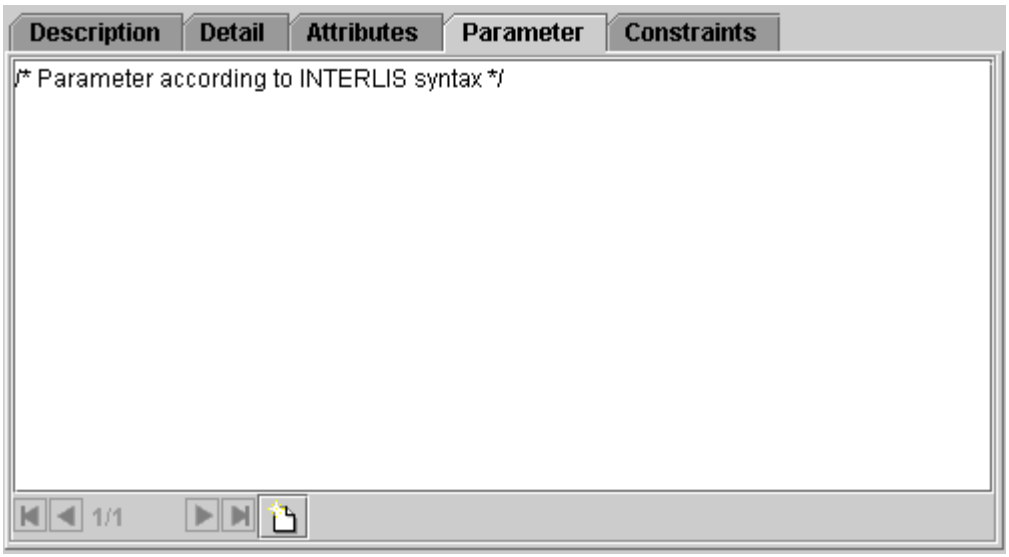

Figure 4.20: Dialog – Klasse (Reiter *Parameter*)

Feld	Beschreibung
Parameters	Es können mehrere *Parameters* in Form von Syntax-Code (siehe Abb. 4.7) angegeben werden. Dabei hilft eine Symbolleiste unterhalb des Syntax-Feldes beim Browsen, Erfassen und Löschen der Parameters.

siehe Abb. 4.21

Feld	Beschreibung
Constraints	Es können mehrere *Constraints* in Form von Syntax-Code (siehe Abb. 4.7) angegeben werden. Dabei hilft eine Symbolleiste unterhalb des Syntax-Feldes beim Browsen, Erfassen und Löschen der Constraints.

4.2.9 Beziehung

An Beziehung siehe Abb. 4.22 describes similar connections between individual objects.

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.7.

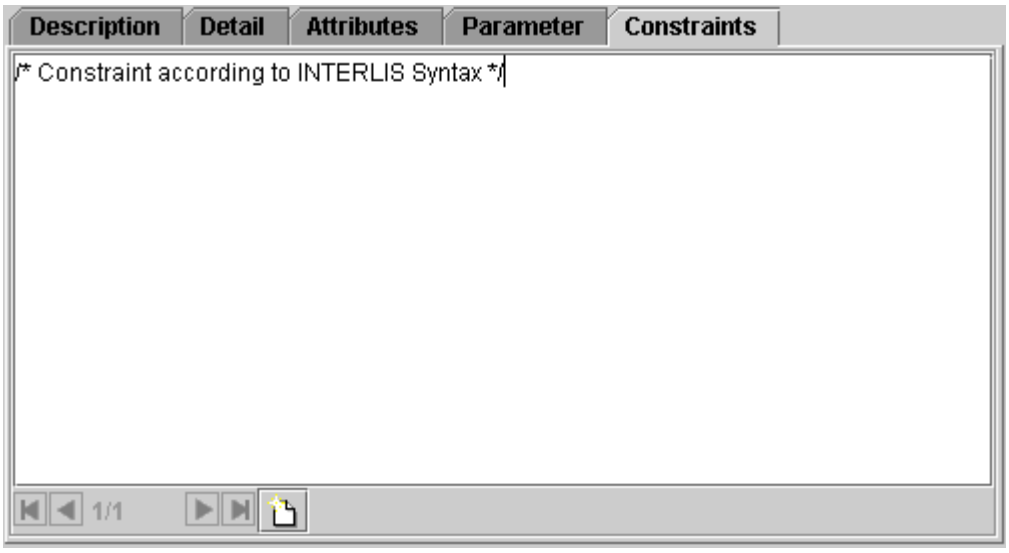

Figure 4.21: Dialog – Klasse (Reiter *Contraints*)

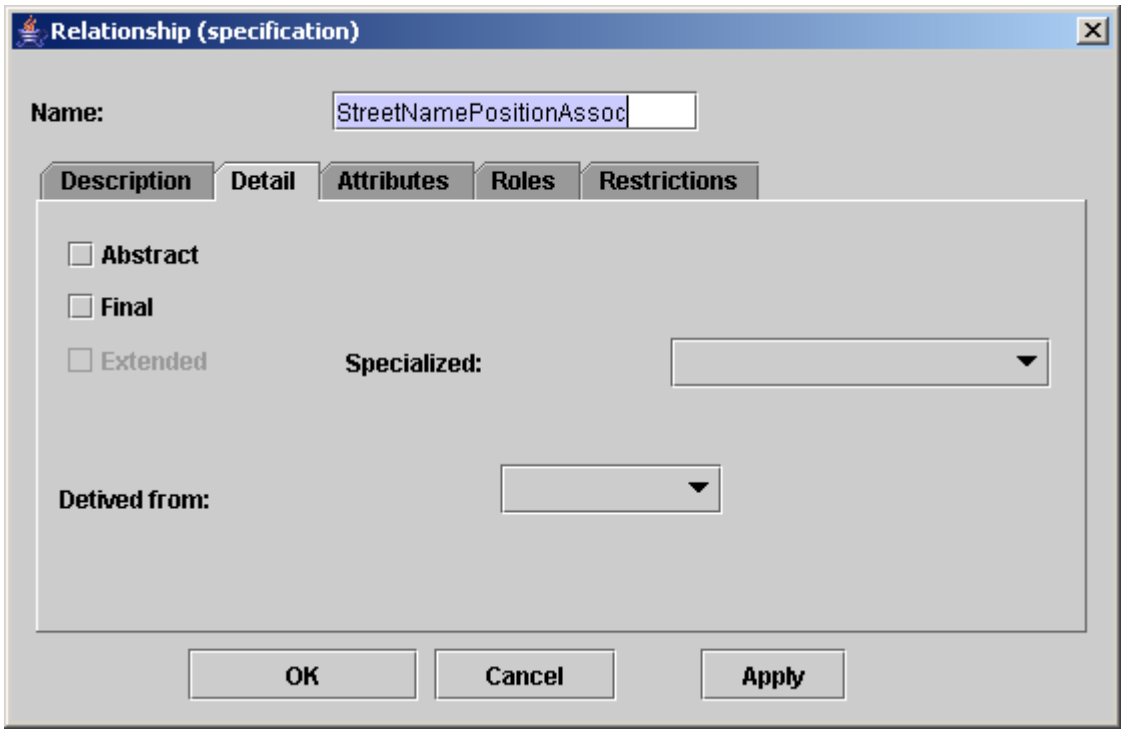

Figure 4.22: Dialog – Beziehung (Reiter *Detail*)

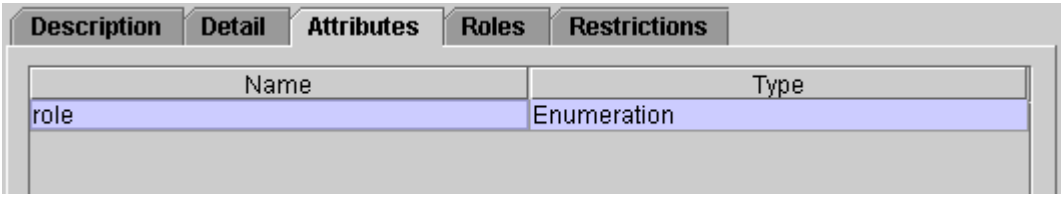

Figure 4.23: Dialog – Beziehung (Reiter *Attribute*)

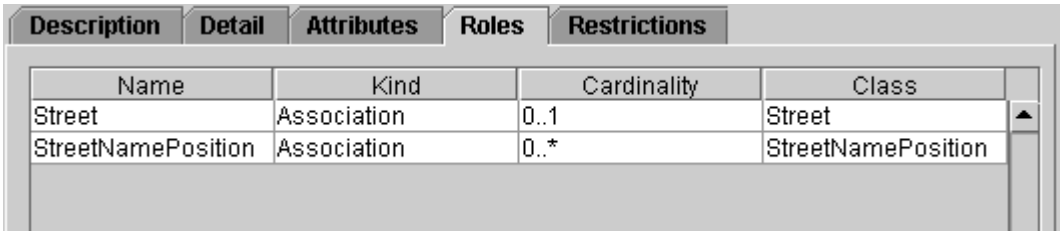

Figure 4.24: Dialog – Beziehung (Reiter *Rollen*)

Feld	Beschreibung
Name	Name of the *association*
Beschreibung	(siehe Abb. 4.6).
Abstrakt	Setzt Modell-Element als *Abstrakt* oder nicht.
Abschliessend	Setzt Modell-Element als *Abschliessend* oder nicht.
Erweitert	Erlaubt die Selektion eines allenfalls zu erweiternden *Modell-Elementes* aus einer Liste von Modell-Elementen, die für eine *Spezialisierung* in Frage kommen (Liste wird automatisch vom UML-Editor aufbereitet).
Derived from	From a list an Modell-Element (in general a view) can be selected, from which the Beziehung is to be derived.

siehe Abb. 4.23

Feld	Beschreibung
Attributes (Table)	Via Popup-Menu a list of *attributes* (siehe Kap. 4.2.10) can be processed. In the table all defined *attributes* can be listed.

siehe Abb. 4.24

Feld	Beschreibung
Roles (Table)	Via the table all defined *rolles* are listed.

siehe Abb. 4.25

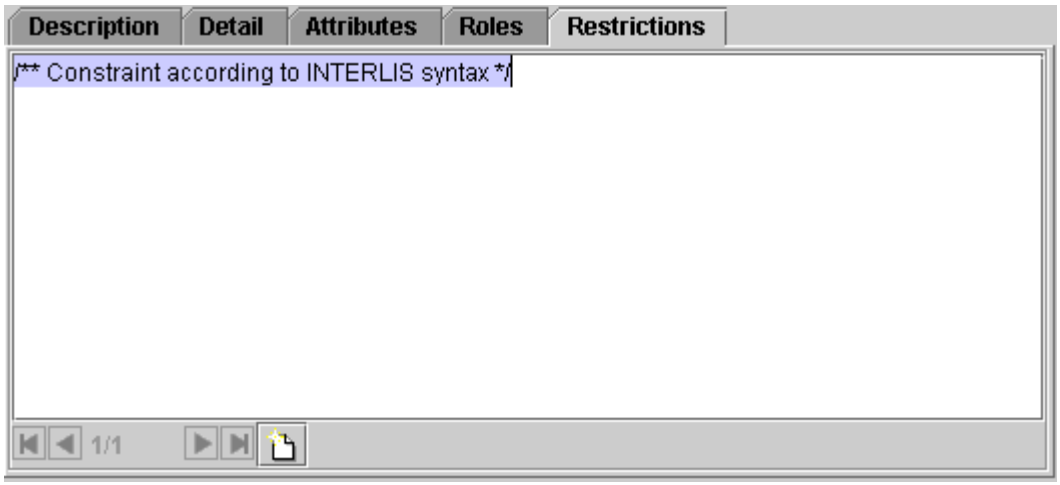

Figure 4.25: Dialog – Beziehung (Reiter *Constraints*)

Feld	Beschreibung
Constraints	Es können mehrere *Constraints* in Form von Syntax-Code (siehe Abb. 4.7) angegeben werden. Dabei hilft eine Symbolleiste unterhalb des Syntax-Feldes beim Browsen, Erfassen und Löschen der Constraints.

There are a few special cases of associations (besides the common binary association), which the UML-Editor supports graphically and thus also in the model.

Reflexive relations

siehe Abb. 4.26

Inherited associations

siehe Abb. 4.27

Hinweis:

- For graphic modelling it is necessary to visualize first the *link knots* in the Klassendiagramm (void rhombus in the middle of the association) (siehe Kap. 4.1).

Multiple relations

siehe Abb. 4.28

Hinweis:

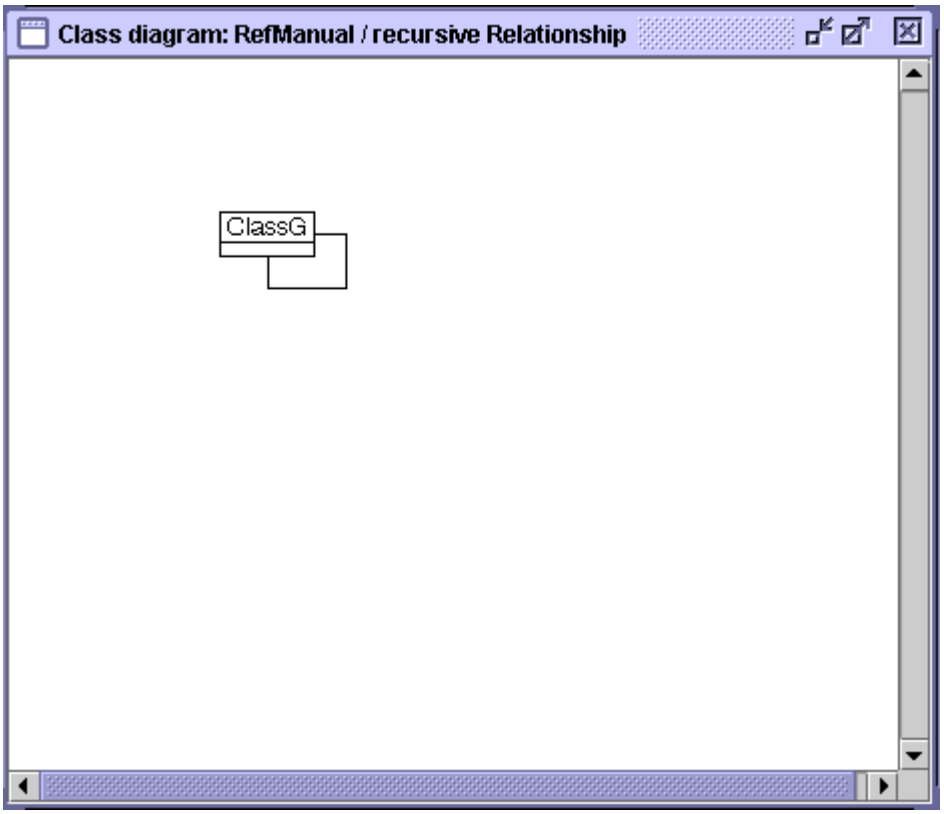

Figure 4.26: Reflexive relations

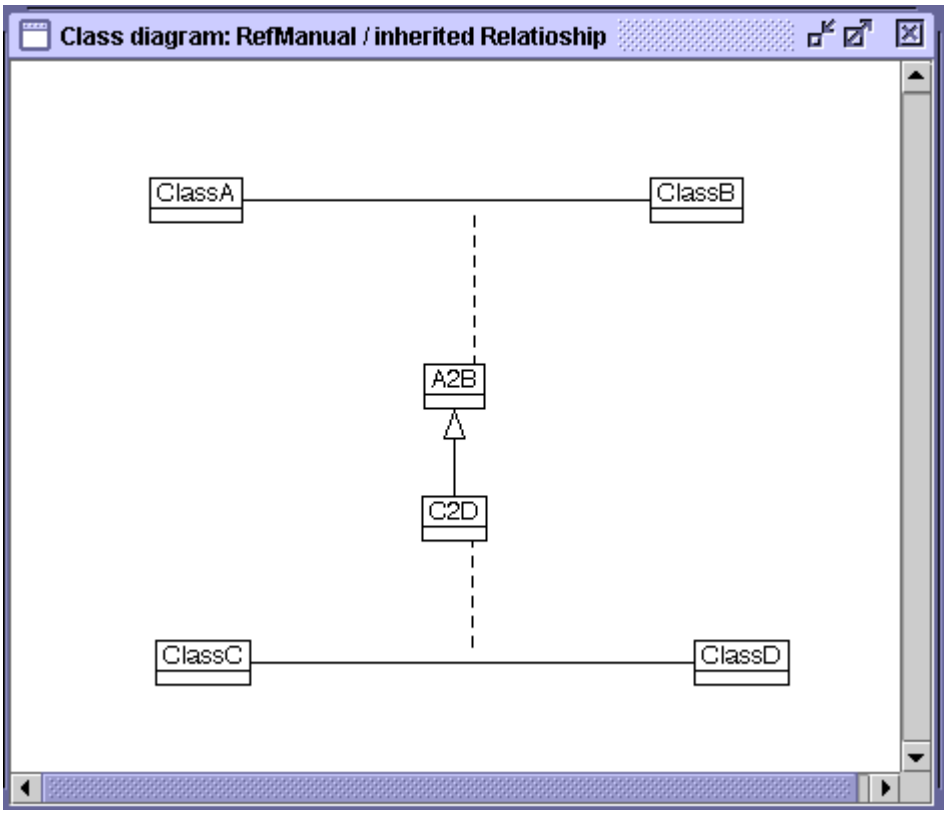

Figure 4.27: Inherited relations

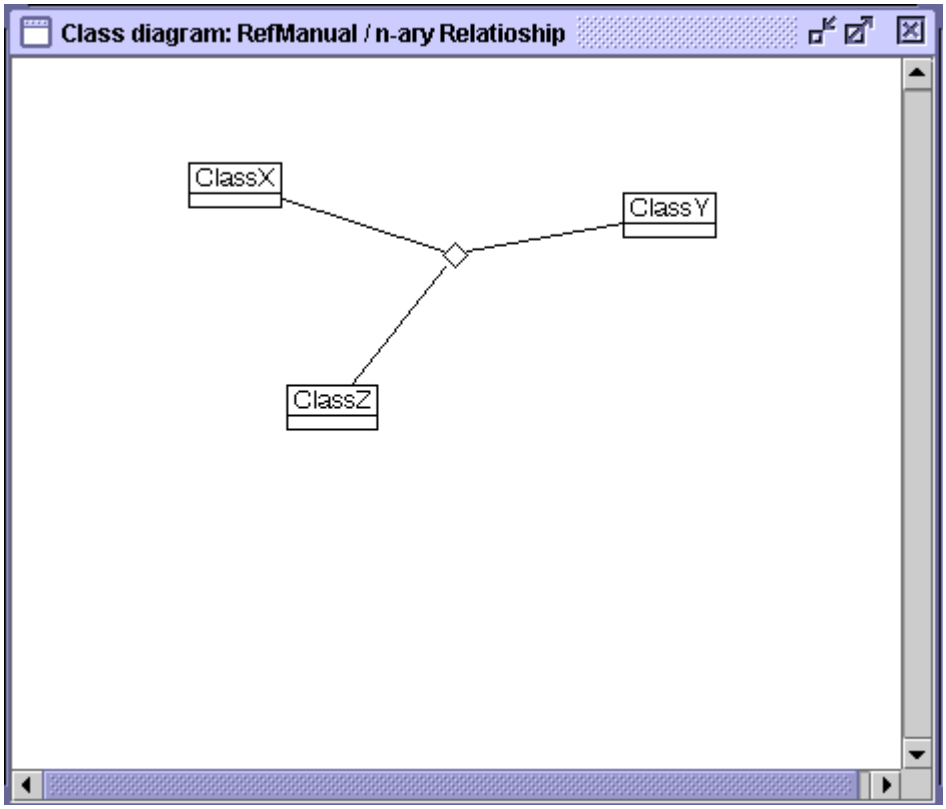

Figure 4.28: Multiple relations

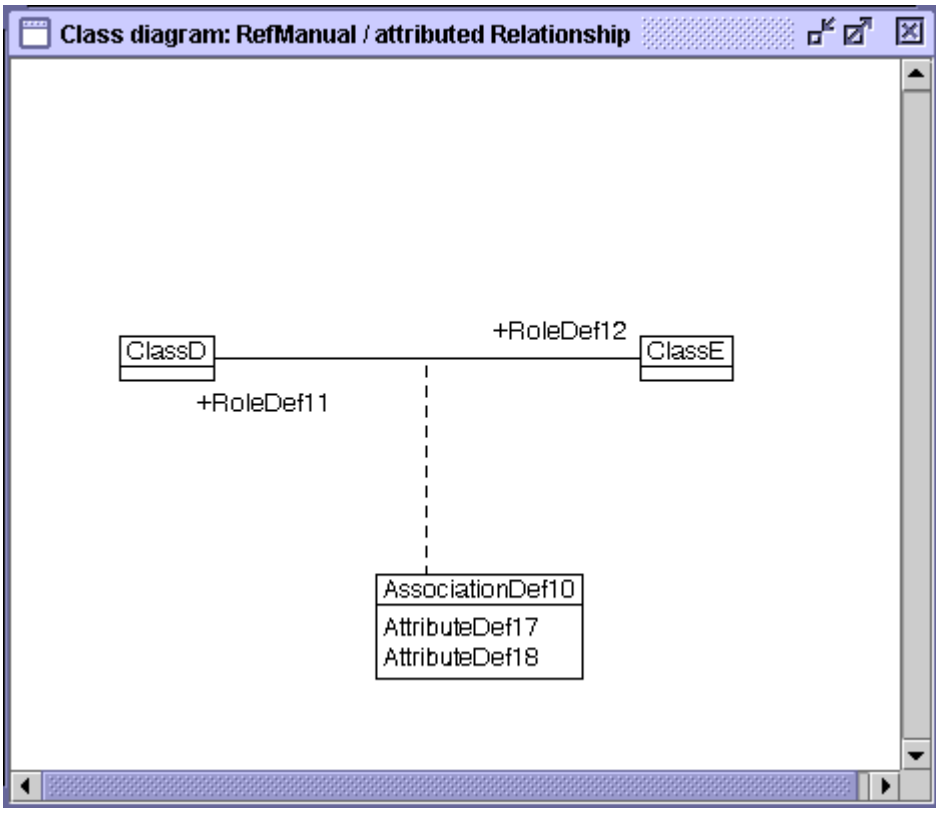

Figure 4.29: Association with attributes

- For graphic modelling it is necessary to visualize first the *link knots* in the Klassendiagramm (void rhombus in the middle of the association) (siehe Kap. 4.1).

Association with attributes

siehe Abb. 4.29

Hinweis:

- Via the specification Dialog *Attributes* can be processed in the tab *Attribute* . The representation in the Klassendiagramm ensues automatically.

4.2.10 Attribute

An attribut (engl. feature or also property) siehe Abb. 4.30 is a data element of a class. An attribute possesses both name and data type. As data types all types that have been defined by INTERLIS are at your disposal.

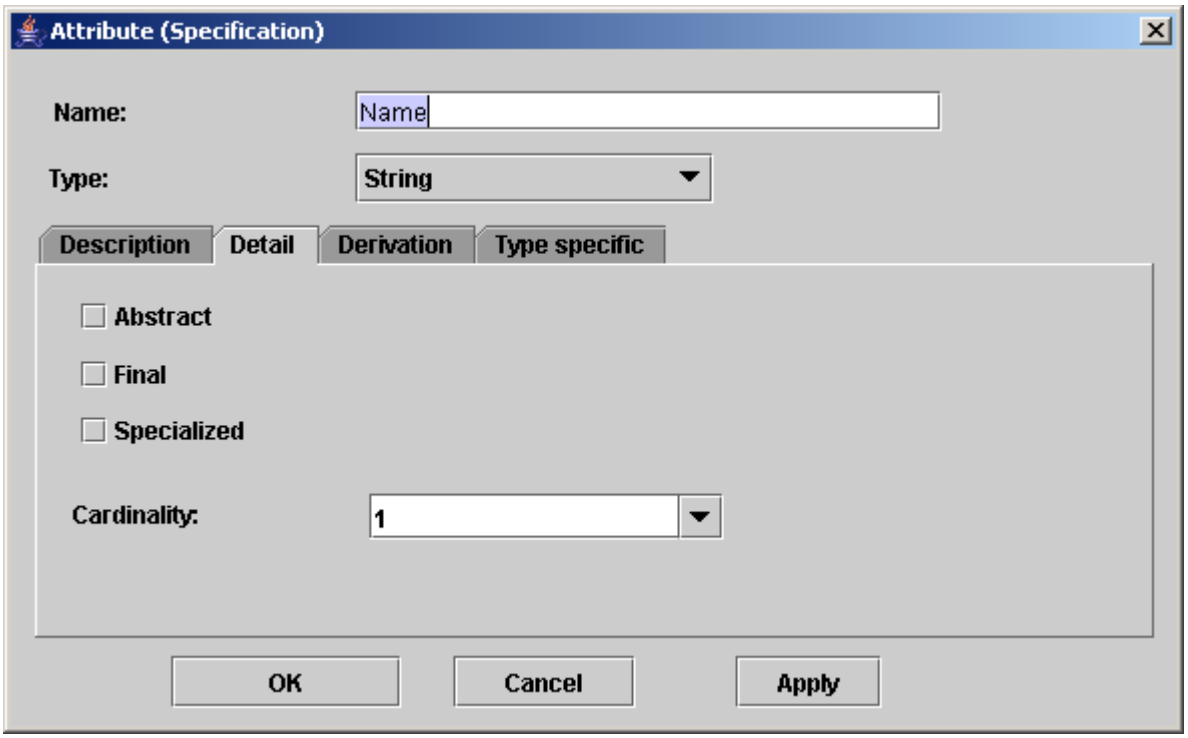

Figure 4.30: Dialog – Attribut (Reiter *Detail*)

Attributes can be processed in three different ways, these being Navigationsbereich (siehe Kap. 3.3), Klassendiagramm siehe Kap. 4.1.2 or this Dialog.

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.6.

Feld	Beschreibung
Name	Name of the *Attributes*
Typ	Depending on the *type* selected particular *Typ spezifisch* information can be indicated in the tab (see below).
Beschreibung	(siehe Abb. 4.6).
Abstrakt	Setzt Modell-Element als *Abstrakt* oder nicht.
Abschliessend	Setzt Modell-Element als *Abschliessend* oder nicht.
specialized	Setzt Modell-Element als *specialized* oder nicht.
Cardinality	Selection of the cardinality is possible in a list. Attributes with an anonymous domain can only possess cardinality 0..1 or 1. Attributes which refer to a domain definition, can have maximum cardinality greater than 1.

siehe Abb. 4.31

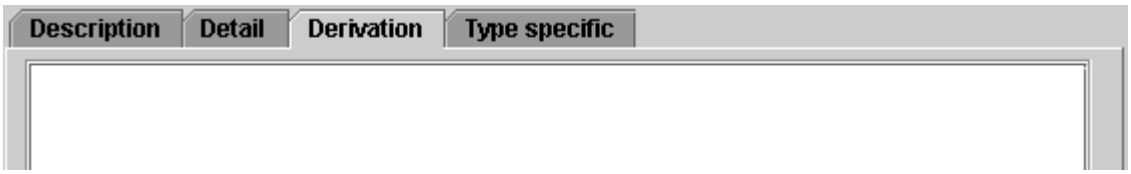

Figure 4.31: Dialog – Attribut (Reiter *Ableitung*)

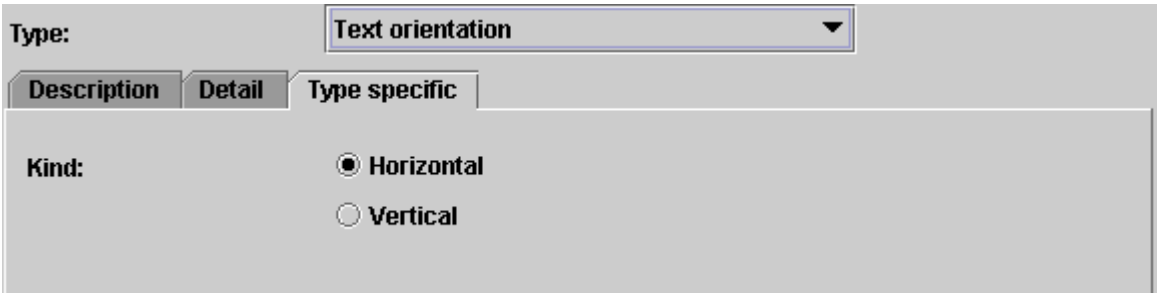

Figure 4.32: INTERLIS-Basistyp – Textasurichtung

Feld	Beschreibung
Derivation	Permits the indication of a syntax code (siehe Abb. 4.7) , e.g. a function call or a constant.

Depending on the *type* the representation of information in the tab *Typ spezifisch* varies:

Boolean

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.8.4.

For the type *Boolean* there is no special view.

Text Orientation

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.8.3. siehe Abb. 4.32

Feld	Beschreibung
Type	Characterizes the text orientation *Horizontal* or *Vertical*.

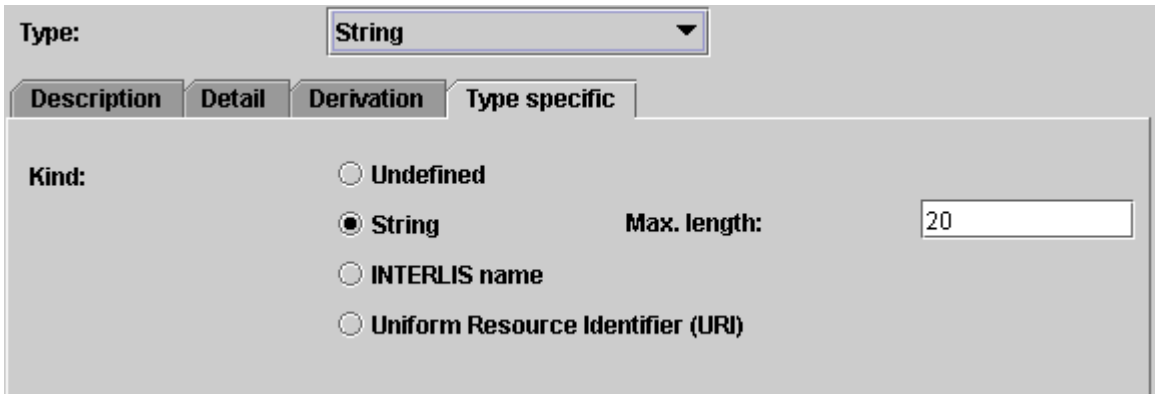

Figure 4.33: INTERLIS-Basistyp – Zeichenkette

String

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.8.1. siehe Abb. 4.33

Feld	Beschreibung
Type	Characterizes the string as *Undefined*, *String* (Default), *INTERLIS-name* or *Uniform Resource Identifier (URI).*
Max. Length (only for the selection of "string")	entry of a whole number for limiting the maximum string length.

Enumeration

With the type *enumeration* siehe Abb. 4.34 it is possible to define enumerations or sub-enumerations as a tree-structure in the field *elements*. Furthermore each enumeration element can be commented upon in the domain *element description*.

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.8.2.

Feld	Beschreibung
Type	Charakterizes the string as *Undefined* (Default), *Ordered*, oder *Circular*.
Elements	A Popup-Menu permits the processing of enumerations in a tree structure (by means of sub-enumeration).
Element description	for each enumeration elementa corresponding comment can be formulated (siehe Abb. 4.6).

Figure 4.34: INTERLIS-basic type – enumeration

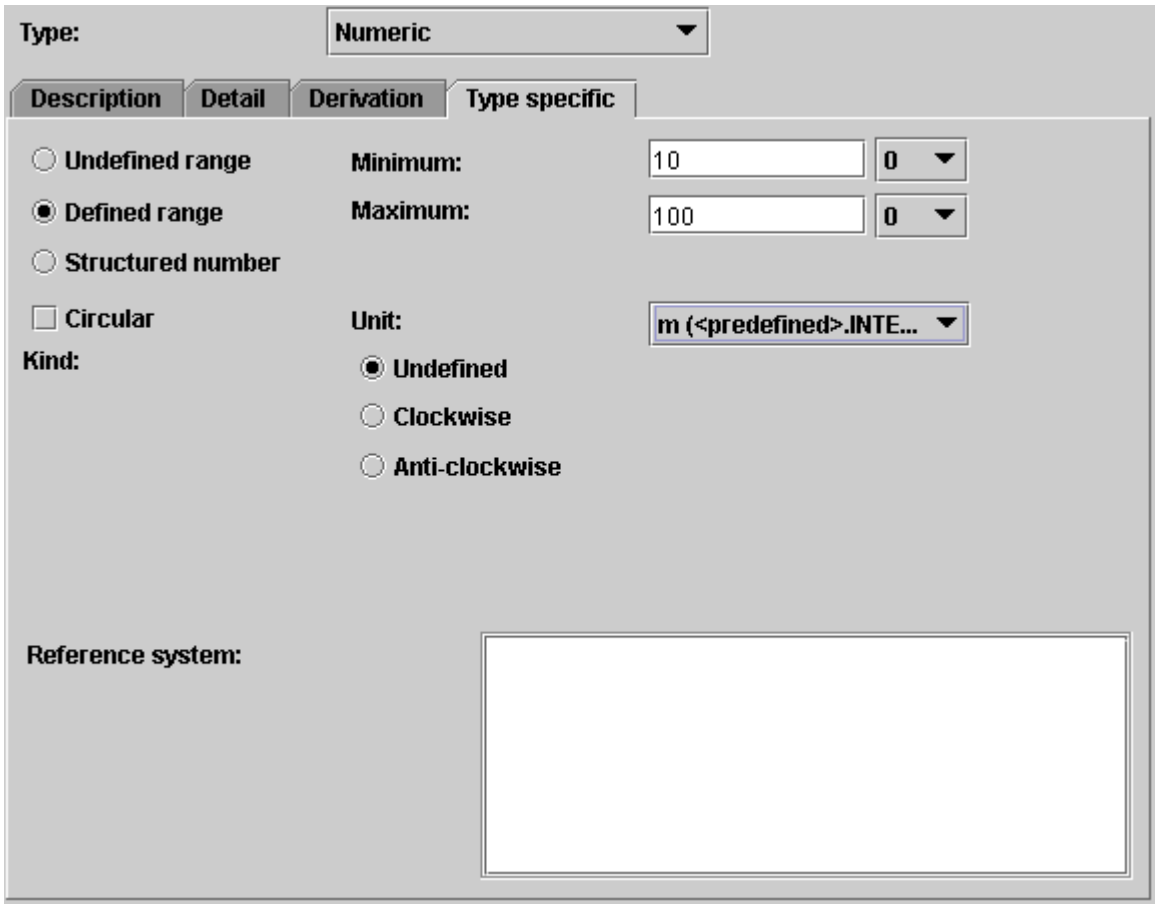

Figure 4.35: INTERLIS-basic type – Numeric

Numeric

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.8.5. siehe Abb. 4.35

Feld	Beschreibung
Domain	Characterizes the numeric domain as an *Undefined domain* (Default), as a *Defined domain* (permits the entry of a minimal or maximal domain as a value with flowing comma, whereby accuracy can be defined via the selection list), or as a *structured number*.
Cirkular	Setzt Modell-Element als *Cirkular* oder nicht.
Unit	A list permits the selection of a *unit* existing in the model (siehe Kap. 4.2.14). Diese Liste wird automatisch vom UML-Editor aufbereitet.
Type	Characterizes the type as *Undefined* (Default), as *clockwise* or as *anti-clockwise*.
Reference system	Permits the indication of syntax code (siehe Abb. 4.7) .

Coordinate

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.8.7. siehe Abb. 4.36

Feld	Beschreibung
Dimensions	Defines the number of dimensions of the *coordinate*.
Circular	Setzt Modell-Element als *Circular* oder nicht.
1/2/3D (Numeric)	For each dimension it is possible to indicate a *numeric value* (siehe Kap. 4.2.10) moeglich.
Rotation	Setzt Modell-Element als *Rotation* oder nicht.
Main axis	Provided the *rotation* has been determined, the *main axis* can be defined.
PI-main axis	Provided the *rotation* has been determined, the *PI-main axis* can be defined.

Basket

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.8.9. siehe Abb. 4.37

Feld	Beschreibung
Type	Characterizes the type as *Undefined* (Default), *Data basket*, *Sicht-Behaelter*, *Basis-Behaelter fuer Grafik* oder als *Grafik-Behaelter*.
According to	Permits the indication of a topic (siehe Kap. 4.2.7). Diese Liste wird automatisch vom UML-Editor aufbereitet.

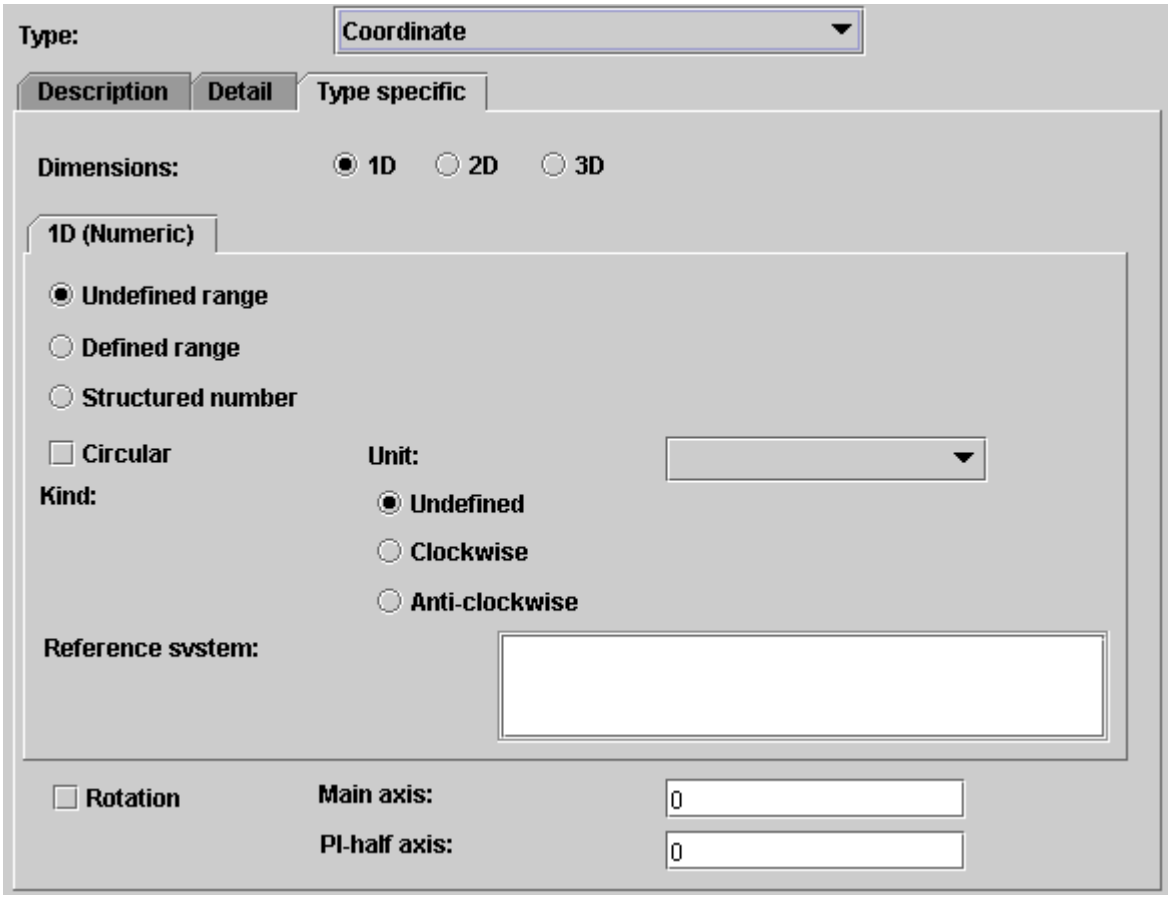

Figure 4.36: INTERLIS-basic type – coordinate

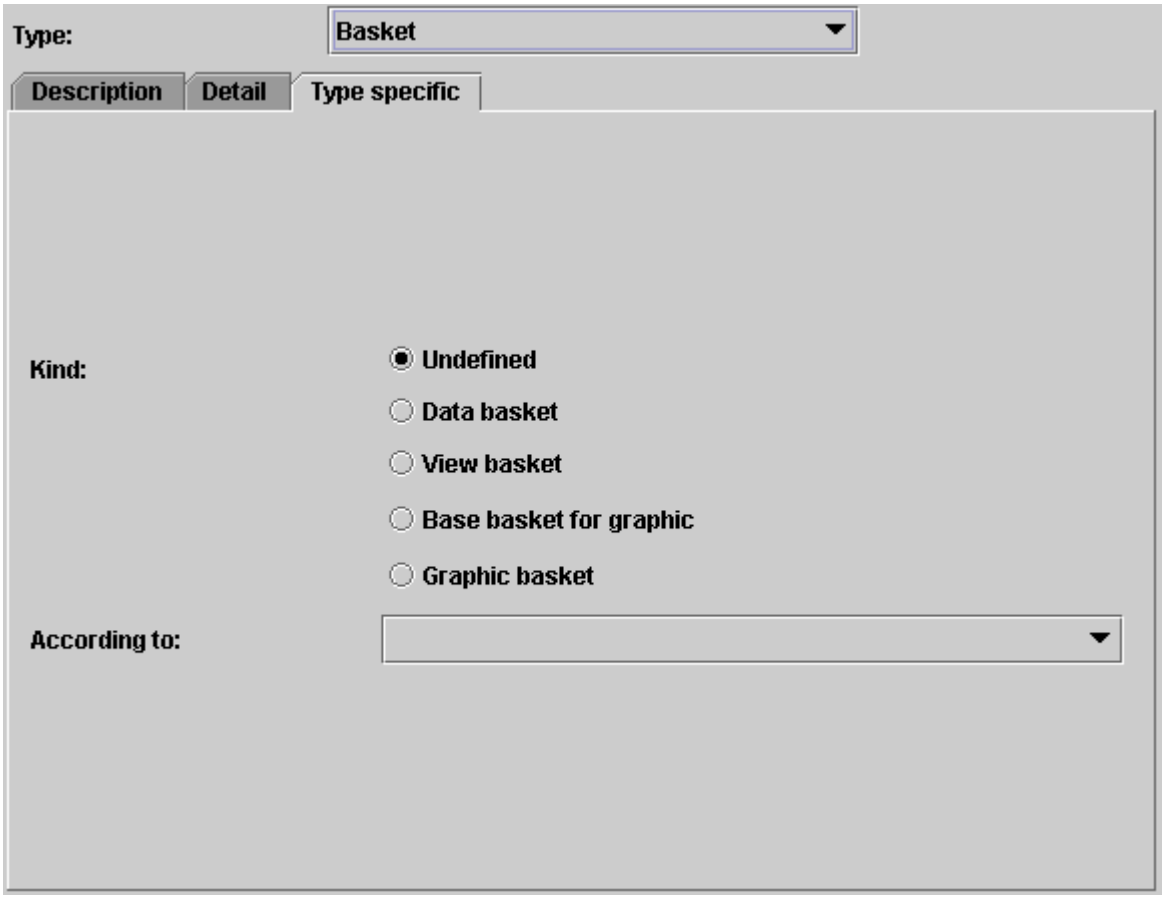

Figure 4.37: INTERLIS-basic type – Basket

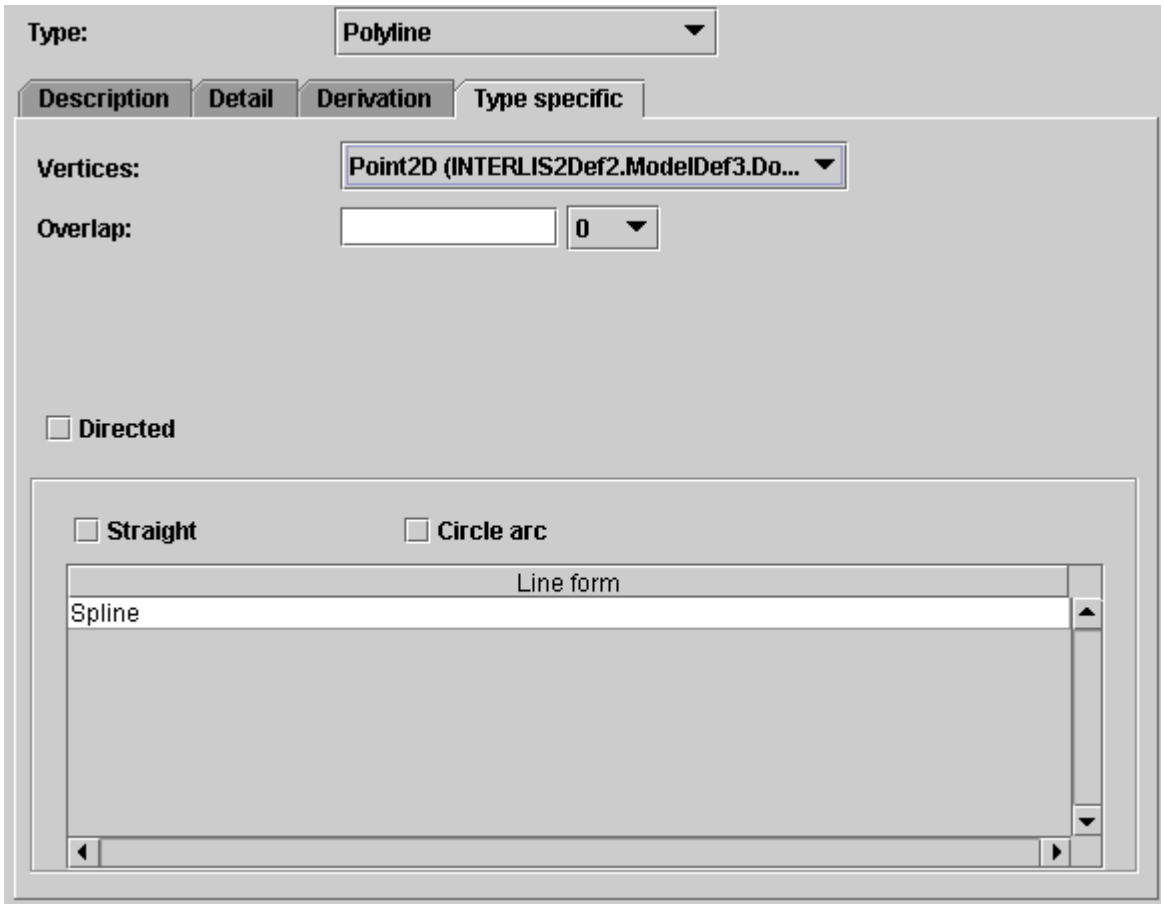

Figure 4.38: INTERLIS-basic type – polyline

Polyline

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.8.11.2. siehe Abb. 4.38

Feld	Beschreibung
Vertices	Permits the assignation of a domain (siehe Kap. 4.2.12) for the vertices of a polyline. Diese Liste wird automatisch vom UML-Editor aufbereitet.
Overlap	Permits the entry of a decimal value (definition of accuracy by means of selection list).
Directed	Defines the polyline as *directed* or not.
Straight	Defines *straights* as admissible line types.
Circle arc	Defines *circle arcs* as admissible line types.
Linienform (Tabelle)	Ein Popup-Menu der Tabelle erlaubt die Zuordnung von *Linienformen* (siehe Kap. 4.2.15).

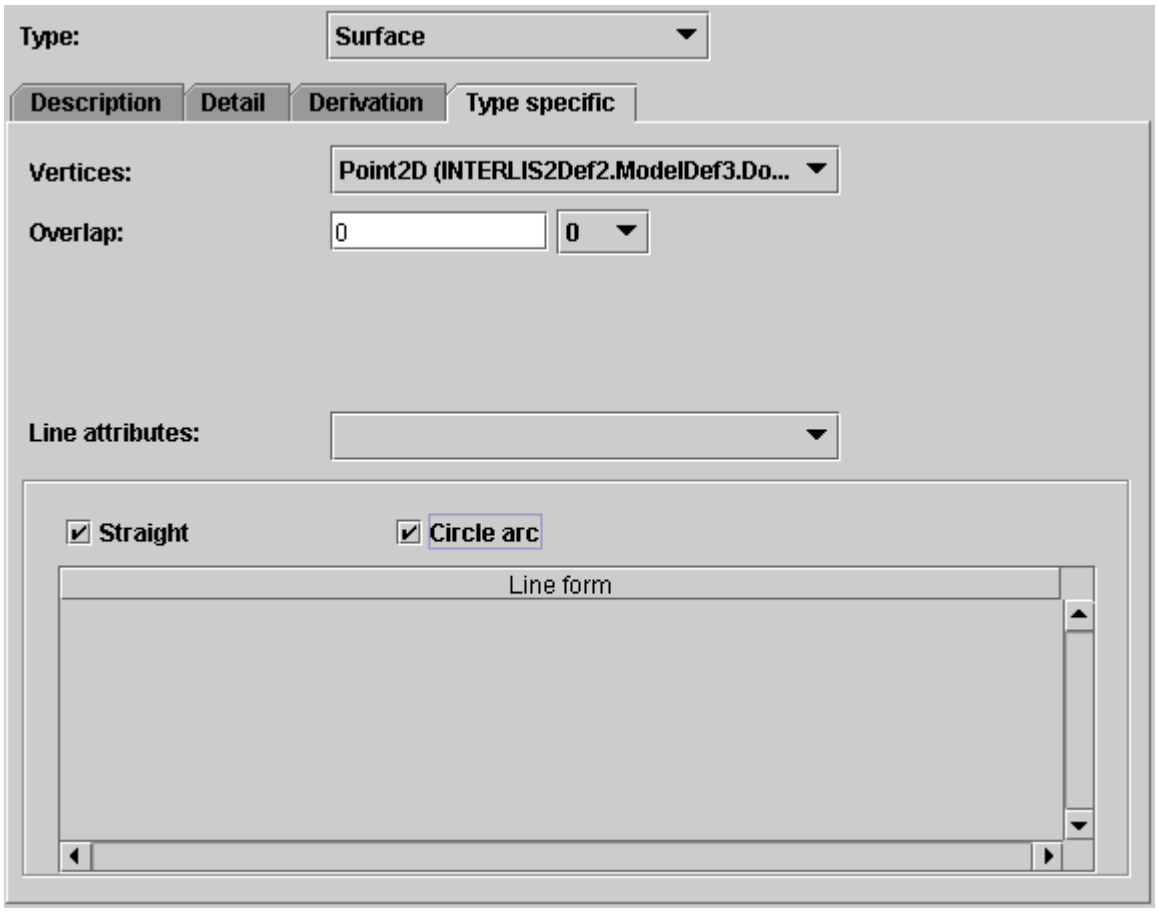

Figure 4.39: INTERLIS-basic type – Surface

Surface

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.8.12.2. siehe Abb. 4.39

Feld	Beschreibung
Vertices	Permits the assignation of a domain (siehe Kap. 4.2.12) for the vertices of the surfaces. Diese Liste wird automatisch vom UML-Editor aufbereitet.
Overlap	Permits the enty of a decimal value (definition of accuracy by means of a selection list).
Line attributes	Permits the selection of a structure (siehe Kap. 4.2.8) for the line attributes. Diese Liste wird automatisch vom UML-Editor aufbereitet.
Straight	Defines *straights* as admissible line types.
Circle arc	Defines *circle arcs* as admissible line types.
Linienform (Tabelle)	Ein Popup-Menu der Tabelle erlaubt die Zuordnung von *Linienformen* (siehe Kap. 4.2.15).

Area Tessellation

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.8.12.3. siehe Abb. 4.40

Feld	Beschreibung
Vertices	Permits the assignation of a domain (siehe Kap. 4.2.12) for the vertices of the area tessellation. Diese Liste wird automatisch vom UML-Editor aufbereitet.
Overlap	Permits the entry of a decimal value (Definition of accuracy by means of a selection list).
Line attributes	Permits the selection of a structure (siehe Kap. 4.2.8) for the line attributes. Diese Liste wird automatisch vom UML-Editor aufbereitet.
Straights	Defines *straights* as admissible line types.
Circle arcs	Defines *circle arcs* as admissible line types.
Linienform (Tabelle)	Ein Popup-Menu der Tabelle erlaubt die Zuordnung von *Linienformen* (siehe Kap. 4.2.15).

Domain definition

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.6.2. siehe Abb. 4.41

Feld	Beschreibung
Name	Permits the assignation of the *domain* (siehe Kap. 4.2.12). Diese Liste wird automatisch vom UML-Editor aufbereitet.

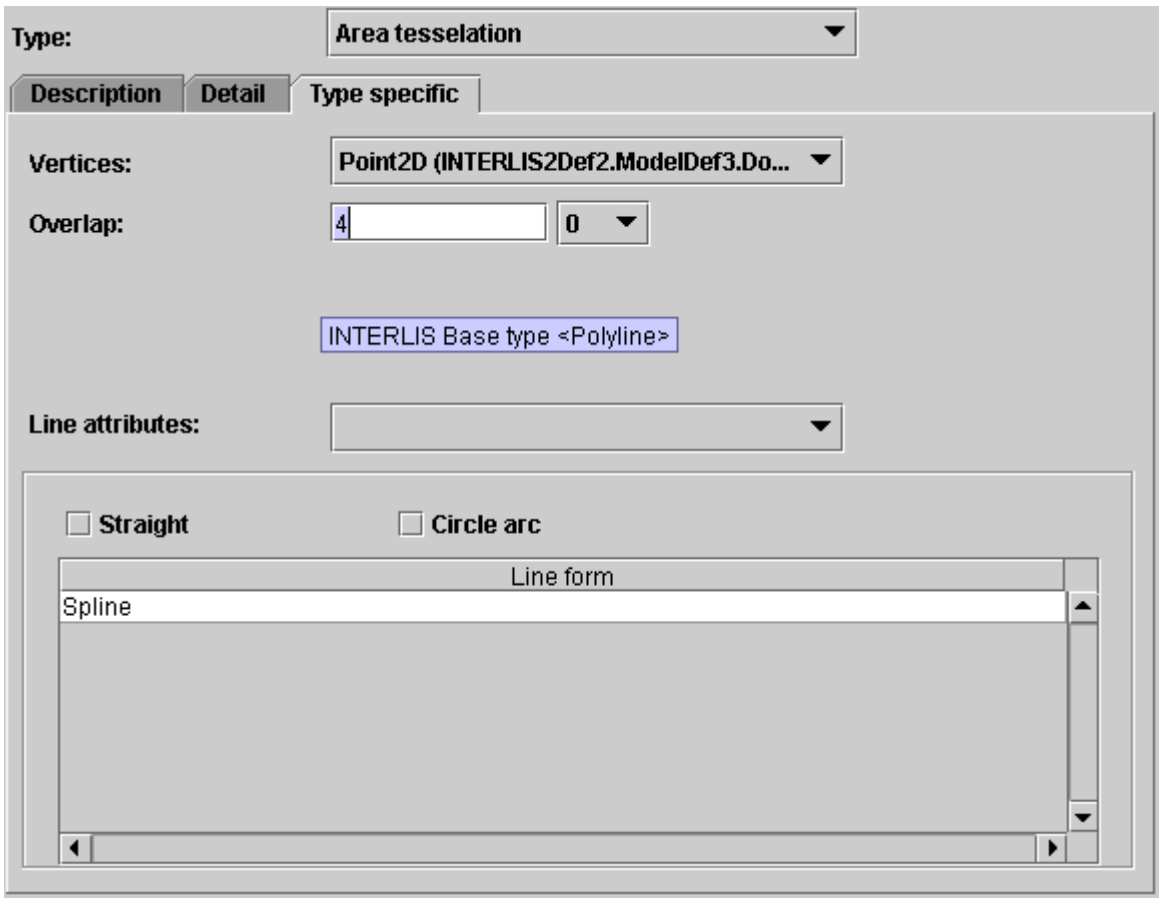

Figure 4.40: INTERLISbasic type – Area tessellation

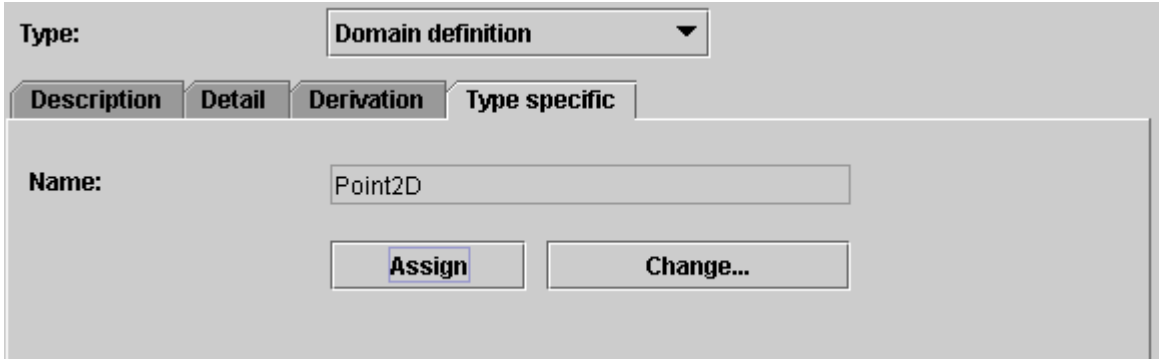

Figure 4.41: INTERLIS-basic type – Domain definition

4.2.11 Role

A role siehe Abb. 4.42 is on end of a relationship. When dealing with a relationship between the class *school* and the class *person*, *teacher* would be a possible role for a *Person*.

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.7.

Feld	Beschreibung
Name	Name of the *role*
Type	Permits the indication if *Association* (Default), *Aggregation* or *Composition* (Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.7.2.). Depending on the selection the specific representation (rhomboid) ensues in the Klassendiagramm in accordance with UML.
Beschreibung	(siehe Abb. 4.6).
Abstrakt	Setzt Modell-Element als *Abstrakt* oder nicht.
Abschliessend	Setzt Modell-Element als *Abschliessend* oder nicht.
Specialised	Setzt Modell-Element als *Specialised* oder nicht.
Ordered	Setzt Modell-Element als *Ordered* oder nicht.
Navigierbar	Setzt Modell-Element als *Navigierbar* oder nicht.
Cardinality	(Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.7.3.)
Classes concerned	Permits the selection of the *Klasse* concerned with this *Role* (siehe Kap. 4.2.8). Diese Liste wird automatisch vom UML-Editor aufbereitet.
Constraints (Table)	Permits the assignation of corresponding *classes* (siehe Kap. 4.2.8 and Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.7.5.).
Type of reference	Permits the selection of the *Role* as *Association* (Default), *Structure* or *Reference*.

4.2.12 Domain

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.8. siehe Abb. 4.43

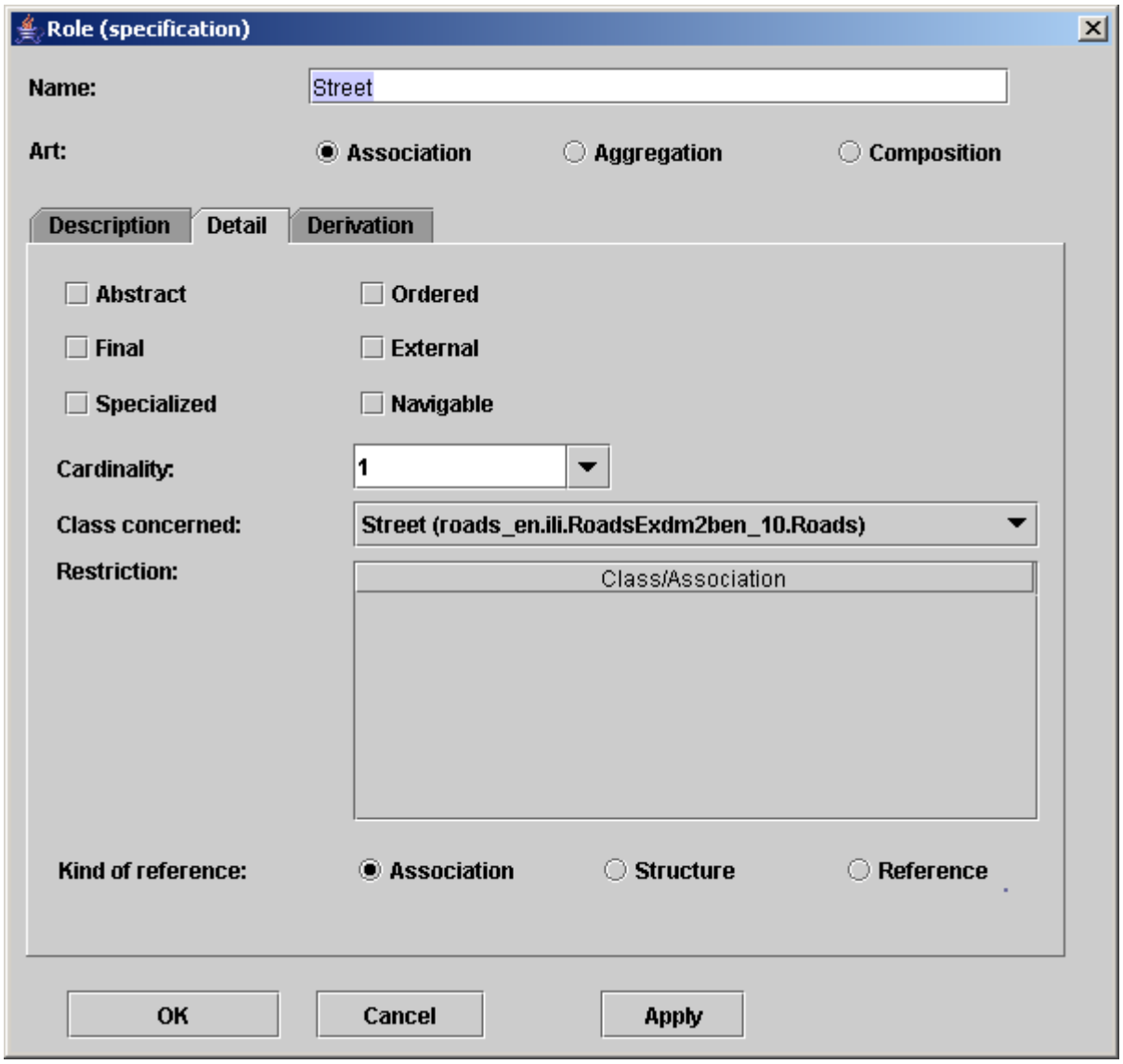

Figure 4.42: Dialog – Role (Reiter *Detail*)

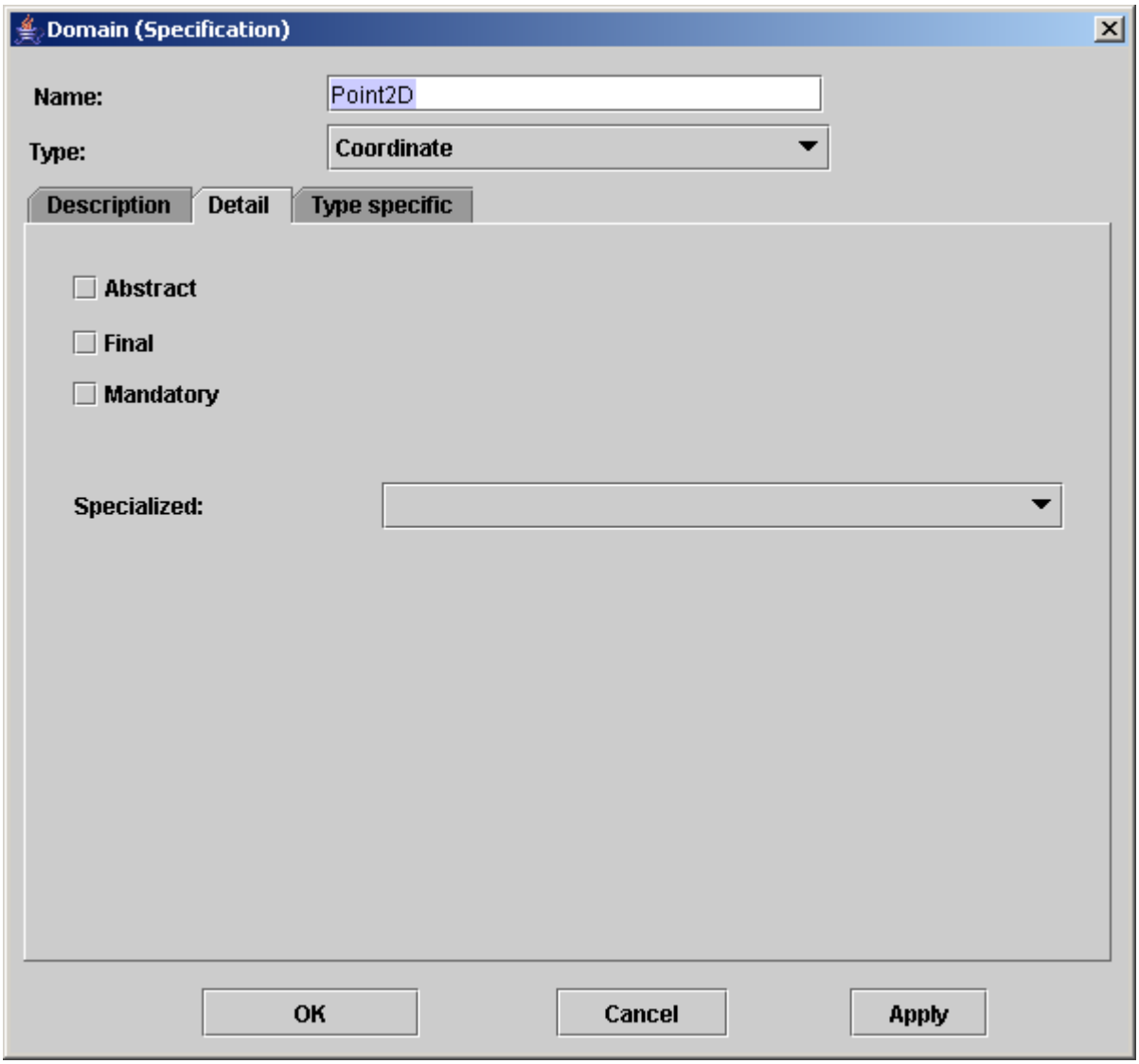

Figure 4.43: Dialog – Domain

Feld	Beschreibung
Name	Name of the *domain*
Typ	Depending on the *type* selected it is possible to indicate special information in the tab *Typ spezifisch* (analogous siehe Kap. 4.2.10).
Beschreibung	(siehe Abb. 4.6).
Abstrakt	Setzt Modell-Element als *Abstrakt* oder nicht.
Abschliessend	Setzt Modell-Element als *Abschliessend* oder nicht.
Mandatory	Setzt Modell-Element als *Mandatory* oder nicht.
Specialised	Permits the selection of a basic -*domain* .

4.2.13 Reference systems/Symbology baskets - Agreement

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.10. siehe Abb. 4.44

Feld	Beschreibung
Name	Name of the *reference system/symbology basket - Agreement*
Basket identification (BID)	Indication of the BID.
Type	Selection as *symbology basket* (Default) or *reference system basket*.
Beschreibung	(siehe Abb. 4.6).
Definition	(siehe Abb. 4.7)

4.2.14 Unit

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.9. siehe Abb. 4.45

Feld	Beschreibung
Name (Abbreviation	Abbreviation of the *Unit*.
Description	Name written in full of the *Unit*.
Beschreibung	(siehe Abb. 4.6).
Depends on	Selection of dependency on another Modell-Element. Diese Liste wird automatisch vom UML-Editor aufbereitet..
Syntax	(siehe Abb. 4.7).

4.2.15 Line form type

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.8.11.3. siehe Abb. 4.46

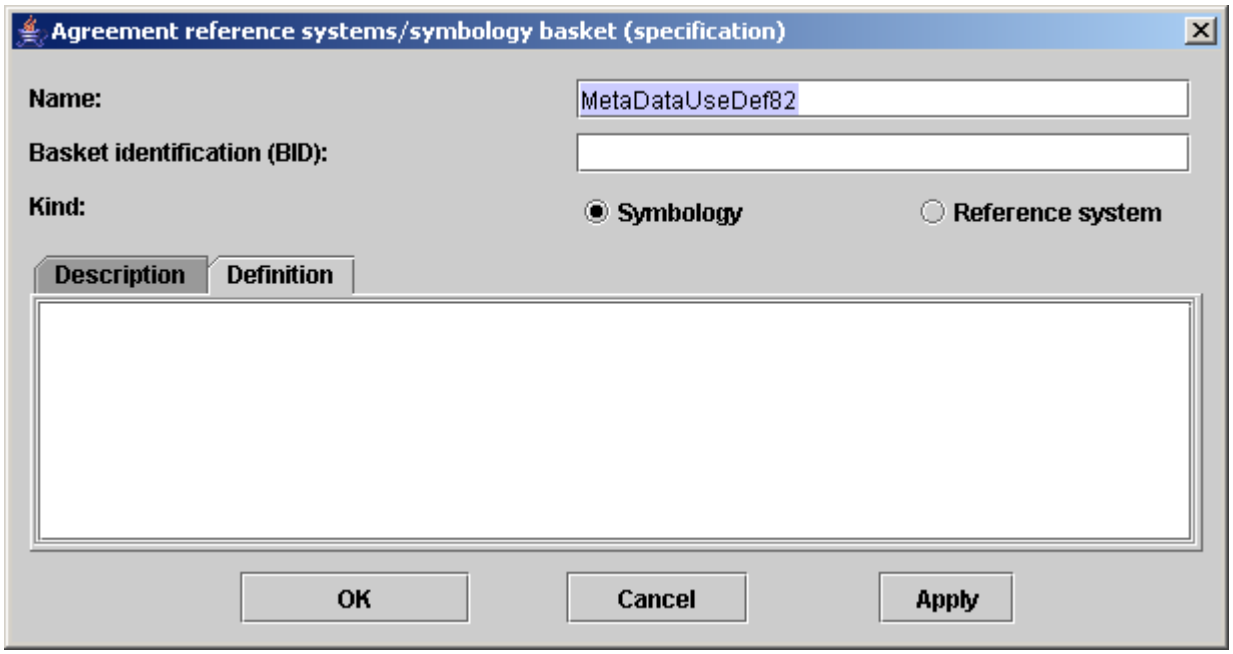

Figure 4.44: Dialog – Reference systems/Symbology baskets

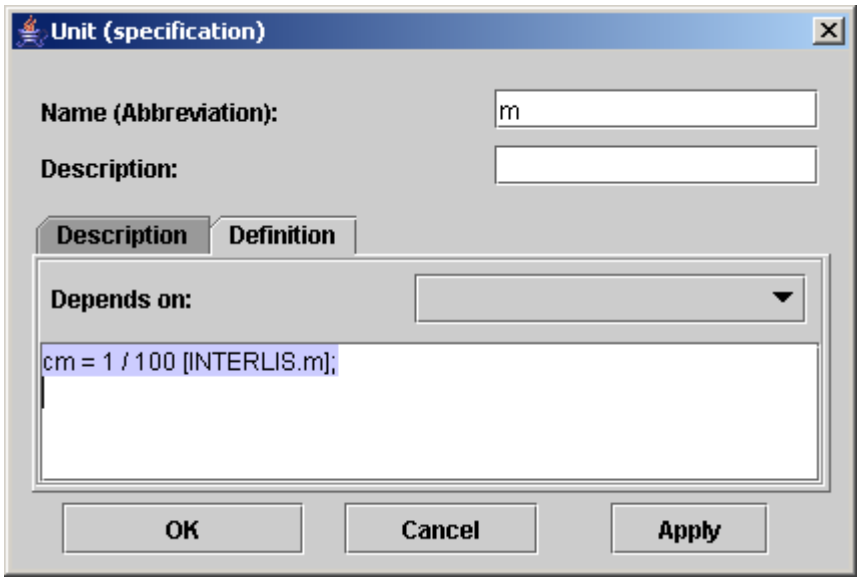

Figure 4.45: Dialog – Unit

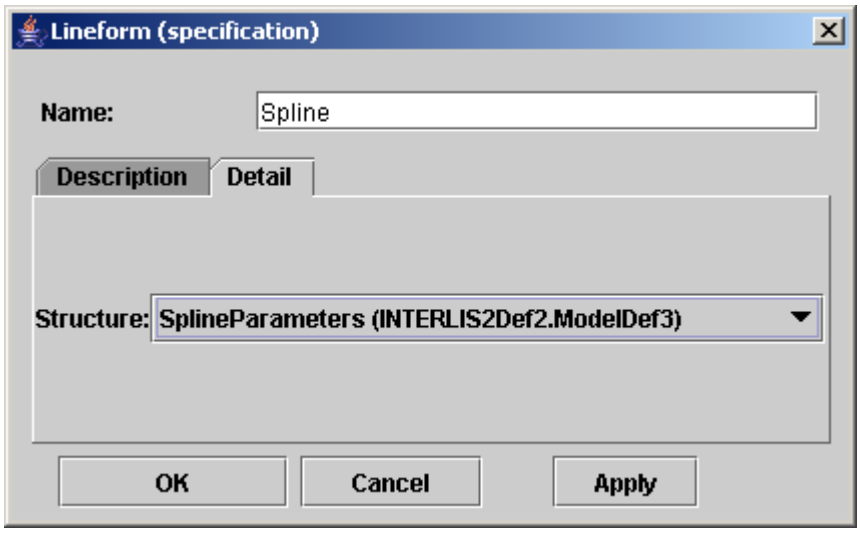

Figure 4.46: Dialog – Line form

Feld	Beschreibung
Name	Name of the *Line form*
Beschreibung	(siehe Abb. 4.6).
Syntax	(siehe Abb. 4.7).

4.2.16 Run time parameter

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.11. siehe Abb. 4.47

Feld	Beschreibung
Name	Name of *Run time parameter*
Beschreibung	(siehe Abb. 4.6).
Syntax	(siehe Abb. 4.7).

4.2.17 Function

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.14. siehe Abb. 4.48

Feld	Beschreibung
Name	Name of the *function*
Beschreibung	(siehe Abb. 4.6).
Syntax	(siehe Abb. 4.7).

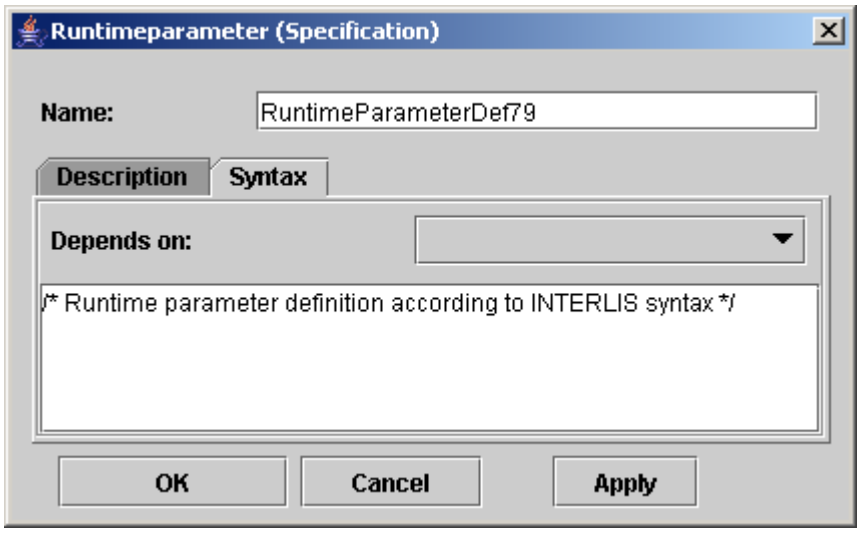

Figure 4.47: Dialog – Run time parameter

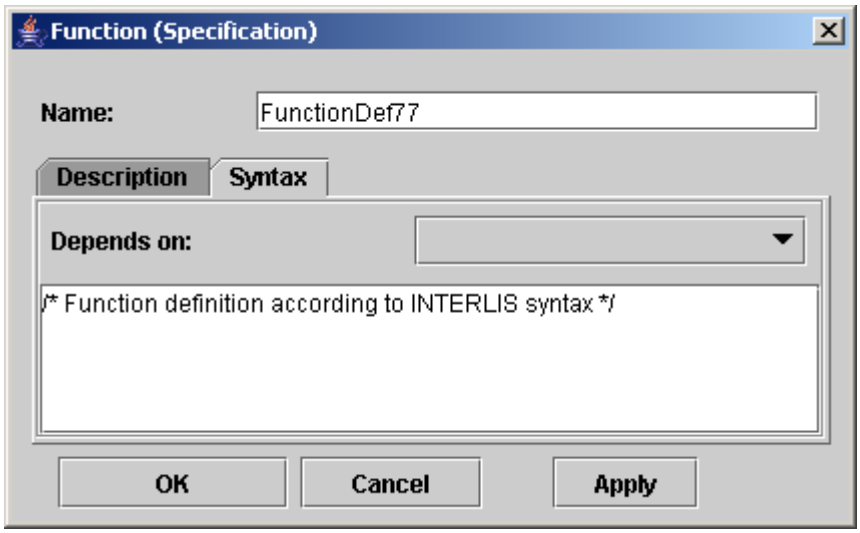

Figure 4.48: Dialog – Function

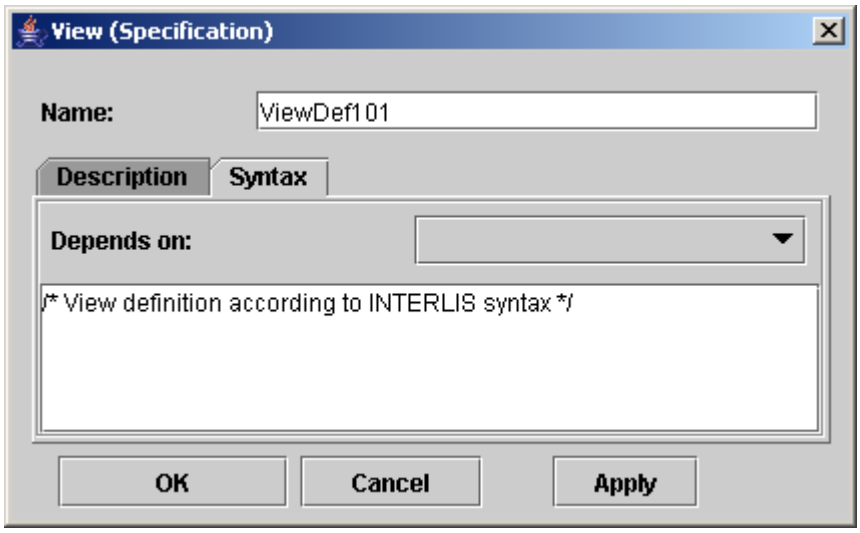

Figure 4.49: Dialog – View

4.2.18 View

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.15. siehe Abb. 4.49

Feld	Beschreibung
Name	Name of the *view*
Beschreibung	(siehe Abb. 4.6).
Syntax	(siehe Abb. 4.7).

4.2.19 Graphic

Für weitere Angaben zu Regeln und Eigenschaften siehe INTERLIS 2 – Referenzhandbuch 2.16. siehe Abb. 4.50

Feld	Beschreibung
Name	Name of the *graphic*
Beschreibung	(siehe Abb. 4.6).
Syntax	(siehe Abb. 4.7).

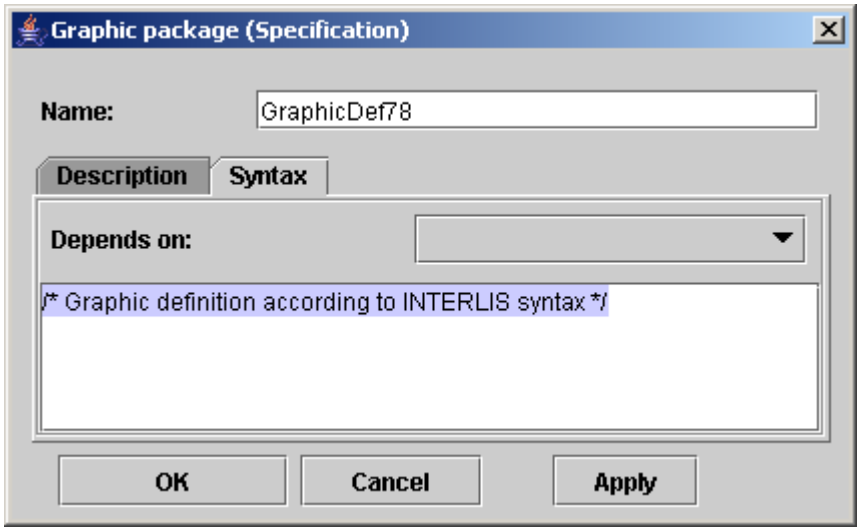

Figure 4.50: Dialog – Graphic

Appendix A

Technical Background

The UML-Editor is based entirely on the UML meta model of the Object Managment Group (OMG). Thus it is guaranteed that all models created with this UML-Editor will be compatible in all respects of the UML-specification (siehe Kap. B.1). Any upgrade of the UML-meta model design on the part of OMG should be possible with very little effort.

An INTERLIS-Plugin extends and concretizes the UML-meta model by adding INTERLISelements. Thus the UML-Editor becomes the ideal tool for users that intend to model INTERLIS (siehe Kap. C.1) (i.e. geomatic engineers).

The UML-Editor has been entirely conceived in Java (s. http://java.sun.com). Java was considered the ideal device for this project because besides other advantages it offers the possibility to implement the object-oriented design of UML and INTERLIS . Furthermore Java is deemed platform-independent. The present release is based upon JRE 1.3, but on principle it can also be applied in improved versions of the Virtual Machine (VM).

For its greater part the model has been designed with Rational Rose (a commercial UML-editor) and then generated with a specially extended Java-Code-Generator. Thus the design will always be up-to-date with the source code.

This document was drafted in LaTeX.

Appendix B

UML

OMG's Unified Modeling Language™(UML) is of use when specifying, visualizing and documenting models in connection with software systems (including their structure and design).

UML can be used in business-modeling as well as in non-software-systems. In the present case UML is especially adapted to the demands of modeling of INTERLIS-data models.

B.1 Specification

The present version of the UML-editor is determined by the *UML Specification Version 1.4*.

It is possible to look into detailed information concerning the contents of the UML specification under http://www.omg.org/uml .

B.1.1 UML Meta Model

OMG makes the UML meta modell available in the form of a download .

The UML-specification in accordance with OMG is very extensive. In the present version of the UML-Editor only those possibilities have been implemented with priority which are relevant within the scope of modeling with INTERLIS.

Appendix C

INTERLIS

C.1 Specification

The present version of the UML-Editor is in accordance with the *INTERLIS specification version 2*.

For more detailed information concerning this reference manual see http://www.interlis.ch

C.2 INTERLIS Compiler

The INTERLIScompiler developped by *Eisenhut Informatik AG* (siehe Kap. 2.4) on behalf of KOGIS (cf. http://www.interlis.ch) is an integrated component of the UML-Editor and permits e.g. the examination of INTERLIS language definitions within an INTERLIS model (siehe Kap. 3.1.7).

Appendix D

Formate

D.1 UML-Editor-Format

The UML-Editor uses its own format for saving (siehe Kap. 3.1.1) models with the ending *.uml*.

D.2 XML-Schema

ASCII-Code nach *Export of an XML-Schema* (siehe Kap. 3.1.7) into a file with suffix *.XSD*.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.interlis.ch/INTERLIS2.2"
targetNamespace="http://www.interlis.ch/INTERLIS2.2"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xsd:element name="TRANSFER" type="Transfer"/>
    <xsd:complexType name="Transfer">
      <xsd:sequence>
        <xsd:element name="HEADERSECTION" type="HeaderSection"/>
        <xsd:element name="DATASECTION" type="DataSection"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="HeaderSection">
      <xsd:sequence>
        <xsd:element name="ALIAS" type="Alias"/>
        <xsd:element name="COMMENT" type="xsd:anyType" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="VERSION" type="xsd:decimal" use="required" fixed="2.2"/>
      <xsd:attribute name="SENDER" type="xsd:string" use="required"/>
```

```
</xsd:complexType>
<xsd:complexType name="Alias">
  <xsd:sequence>
    <xsd:element name="ENTRIES" type="Entries" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Entries">
  <xsd:sequence>
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="TAGENTRY" type="Tagentry"/>
      <xsd:element name="VALENTRY" type="Valentry"/>
      <xsd:element name="DELENTRY" type="Delentry"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="FOR" type="xsd:string" use="required"/>
</xsd:complexType>
<xsd:complexType name="Tagentry">
  <xsd:attribute name="FROM" type="xsd:string" use="required"/>
  <xsd:attribute name="TO" type="xsd:string" use="required"/>
</xsd:complexType>
<xsd:complexType name="Valentry">
  <xsd:attribute name="ATTR" type="xsd:string" use="required"/>
  <xsd:attribute name="FROM" type="xsd:string" use="required"/>
  <xsd:attribute name="TO" type="xsd:string" use="required"/>
</xsd:complexType>
<xsd:complexType name="Delentry">
  <xsd:attribute name="TAG" type="xsd:string" use="required"/>
</xsd:complexType>
<xsd:complexType name="BasketValue">
  <xsd:attribute name="TOPIC" type="xsd:string" use="required"/>
  <xsd:attribute name="KIND" type="xsd:string" use="required"/>
  <xsd:attribute name="BID" type="xsd:string" use="required"/>
</xsd:complexType>
<xsd:complexType name="CoordValue">
  <xsd:sequence>
    <xsd:element name="C1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:decimal"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="C2" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:decimal"/>
      </xsd:simpleType>
    </xsd:element>
```

```
    <xsd:element name="C3" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:decimal"/>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArcPoint">
  <xsd:sequence>
    <xsd:element name="C1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:decimal"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="C2">
      <xsd:simpleType>
        <xsd:restriction base="xsd:decimal"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="C3" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:decimal"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="A1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:decimal"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="A2">
      <xsd:simpleType>
        <xsd:restriction base="xsd:decimal"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="R">
      <xsd:simpleType>
        <xsd:restriction base="xsd:decimal"/>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RoleType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="REF" type="xsd:string"/>
```

```
          <xsd:attribute name="EXTREF" type="xsd:string"/>
          <xsd:attribute name="BID" type="xsd:string"/>
          <xsd:attribute name="NEXT_TID" type="xsd:string"/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>

    <xsd:complexType name="DataSection">
      <xsd:sequence>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
        </xsd:choice>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:schema>
```

D.3 INTERLIS-Compiler Configuration

Compiler configuration files use the format **.ilc* (siehe Kap. 3.1.7).

D.4 INTERLIS Model File

In general an INTERLIS-model file ends in *.ili* and contains ASCII-signs. When importing/exporting (siehe Kap. 3.1.7) from an INTERLIS Modell INTERLIScode in an ASCII-file is expected as follows.

An example according to the Roads Model in siehe Abb. 3.1):

```
\textbf{INTERLIS} 2.2;

\textbf{MODEL} ModelDef3 (de) =

  \textbf{DOMAIN}

    Point2D= \textbf{COORD NUMERIC CIRCULAR, NUMERIC};

  \textbf{TOPIC TOPIC} Roads =

    \textbf{CLASS} StreetAxis =
      Geometry : \textbf{TEXT*}20;
    \textbf{END} StreetAxis;
```

```
  \textbf{CLASS} LandCover =
    Type : \textbf{TEXT*}20;
    Geometry : \textbf{TEXT*}20;
  \textbf{END} LandCover;

  \textbf{CLASS} PointObject =
    Type : \textbf{TEXT*}20;
    Position : \textbf{TEXT*}20;
  \textbf{END} PointObject;

  /** Position of a StreetName
   */
  \textbf{CLASS} StreetNamePosition =
    NamPos : \textbf{TEXT*}20;
    NamOri : \textbf{TEXT*}20;
  \textbf{END} StreetNamePosition;

  \textbf{UNIT}

    PI / 180 [rad]

  \textbf{CLASS} Street =
    Name : \textbf{TEXT*}20;
  \textbf{END} Street;

  \textbf{ASSOCIATION} StreetNamePositionAssoc =
    StreetNamePosition -- {0..*} StreetNamePosition;
    Street -- {1} Street;
  \textbf{END} StreetNamePositionAssoc;

  \textbf{ASSOCIATION} StreetAxisAssoc =
    Street -- {1} Street;
    StreetAxis -- {9223372036854775807..*} StreetAxis;
  \textbf{END} StreetAxisAssoc;

\textbf{END TOPIC} Roads;

\textbf{TOPIC TOPIC} RoadsExtended =

  \textbf{CLASS} PointObjectExtended
  \textbf{EXTENDS} ModelDef3.\textbf{TOPIC} Roads.PointObject =
  \textbf{END} PointObjectExtended;

  \textbf{CLASS} StreetAxisExtended
  \textbf{EXTENDS} ModelDef3.\textbf{TOPIC} Roads.StreetAxis =
```

```
      Precision : \textbf{TEXT*}20;
    \textbf{END} StreetAxisExtended;

  \textbf{END TOPIC} RoadsExtended;

\textbf{END} ModelDef3.
```

Appendix E

Country-specific Differences

Depending on the country or region there are different names for the same technical term. All texts (strings) concerning the UML-Editor are stored in so-called *ressource-files (with the ending .property)* . This bears the advantage, that the terms used in the UML-Editor can be altered without having to recompile the entire program. This is why it may be possible that e.g. field names differ from the ones printed in this manual.

Bibliography

[1] OMG, *Object Management Group*, http://www.omg.org, The Object Management Group (OMG) is an open membership, non-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications. Our membership includes virtually every large company in the computer industry, and hundreds of smaller ones. Most of the companies that shape enterprise and Internet computing today are represented on our Board of Directors.

[2] UML, *Cetus-Links*, http://www.cetus-links.org, Various links to *Objects & Components*

[3] Martin Fowler & Kendall Scott, *UML konzentriert*, Addison-Wesley

[4] James Martin & James Odell, *Object-oriented methods a foundation*, Prentice-Hall

[5] KOGIS, *INTERLIS-Reference Manual*, http://www.interlis.ch

[6] KOGIS, *INTERLIS-User Manual*, http://www.interlis.ch

[7] KOGIS, *Introduction to the UML-Editor*, http://www.eisenhutinformatik.ch/umleditor

[8] KOGIS, *UML-Editor*, http://www.eisenhutinformatik.ch/umleditor