Enterprise Architect

User Guide Series

Geospatial Models

Sparx Systems Enterprise Architect supports Geography Markup Language (GML) application schemas and ArcGIS geodatabase modeling to define geospatial data integrated with other data to form comprehensive views of information not possible in other tools.

Author: Sparx Systems

Date: 16/01/2019

Version: 1.0

CREATED WITH ENTERPRISE ARCHITECT

Table of Contents

Geospatial Modeling 3
Geodatabase Design for ArcGIS 4
ArcGIS UML Profile 5
ArcGIS Toolbox Pages 7
Connectivity Rule Examples 12
Topology Example 14
Relationship Rule Example 16
Setting ArcGIS Coordinate Systems 18
Applying ArcGIS Stereotypes to Abstract Classes 21
Export ArcGIS XML Workspace 24
Export Modular ArcGIS Schemas 26
Import ArcGIS XML Workspace 32
Validate an ArcGIS Workspace 34
MDG Technology For GML 35
UML Profile for GML 36
GML Toolbox Pages 38
Generate GML Application Schema 41

Geospatial Modeling

The popularity of the internet, the ubiquitous mobile phone and the prevalence of location-based services have resulted in almost every one interacting with location-based information in some form in their daily lives. It has also become critical for governments and organizations to embrace this type of information as part of strategic decision making. Geospatial information can be modeled in Enterprise Architect and also integrated with other data to form a single and comprehensive view of information not possible in other tools.

Enterprise Architect, through the use of MDG Technologies, supports the Geography Markup Language (GML) application schemas and the modeling of ArcGIS geodatabases. The information precursors to these models, such as community conceptual models, can also be modeled and traceability can be used to connect the models together.

Modeling Tools

Tool	Description
ArcGIS Profile	Enterprise Architect supports the design of geodatabases for the ArcGIS 10.0 suite of tools developed by Esri Inc.
Geography Markup Language	MDG Technology for GML is the implementation of the Open Geospatial Consortium's Geography Markup Language (GML) 3.3 for Enterprise Architect, which provides an XML grammar for geographical feature modeling capabilities within Enterprise Architect

Geodatabase Design for ArcGIS

The ArcGIS system, developed by Esri, supports the development and management of geodatabases. As for other databases, it is useful to model the design of a geodatabase using a standard notation such as UML. You can perform such modeling in Enterprise Architect, using the UML profile for ArcGIS, which is part of the built-in MDG Technology for ArcGIS. Once you have modeled an ArcGIS schema in Enterprise Architect, you can export the model to ArcGIS as an XML Workspace document. You can also visualize an existing ArcGIS geodatabase schema, by importing the ArcGIS XML Workspace document into Enterprise Architect.

Notice of Acknowledgement:

Support for modeling ArcGIS databases in Enterprise Architect was developed in collaboration with the Commonwealth Scientific and Industrial Research Organization (CSIRO), who defined mappings between UML 2 and ArcGIS concepts, and prototyped an automated import and export capability for ArcGIS geodatabase schemas represented in UML.

Notes

- The MDG Technology for ArcGIS is available in the Professional, Corporate, Unified and Ultimate editions of Enterprise Architect

ArcGIS UML Profile

The MDG Technology for ArcGIS is built in to the Enterprise Architect installer. A key component of the technology is the UML Profile for ArcGIS.

Access

Ribbon	Specialize > Technologies > ArcGIS
Context Menu	Right-click on Package | Specialize | ArcGIS

Features

Feature	Detail
Profile Support	The built-in MDG Technology for ArcGIS provides: • ArcGIS Toolbox pages that map ArcGIS concepts to appropriately stereotyped UML elements • A Model Pattern that helps you to start designing geodatabases quickly and to use the required Package structure in Enterprise Architect • Datatypes that are specific to the ArcGIS platform • Quick-linker capabilities that help you make valid connections between elements
ArcGIS Toolbox Pages	The ArcGIS Toolbox contains five core pages: • Domains - for coded value and range domains • Features and Tables - for custom feature types and tables • Network Features - for geometric network and topology Packages • Raster - for raster datasets • Workspace - for ArcGIS workspace and spatial reference information Two additional Toolboxes group the objects used specifically in creating Geometric Network and Topology diagrams.
Spatial References	Enterprise Architect helps you to model Spatial Reference information for your ArcGIS schema, including the selection of predefined coordinate systems and associated values.
Show/Hide System Attribute Fields	The ArcGIS elements provided through the Toolbox pages contain a number of system-assigned attributes that define the «AttributeIndex», «SpatialIndex» and «RequiredField» stereotypes. When you drag an element onto a diagram from the Toolbox, these attributes are not visible in the newly-created structure. If you want to show these system attributes, right-click on the element(s) and select the 'Specialize > Technologies > ArcGIS > Show or Hide ArcGIS System Fields' ribbon option. Similarly, if you have exposed the attributes and want to hide them, select the elements and select the menu option again.

	This option does not operate on attributes or stereotypes you have added to the selected elements, nor on elements that you have not selected. If you do not select any elements, the option is grayed out.

Notes

- The MDG Technology for ArcGIS is available in the Professional, Corporate, Unified and Ultimate editions of Enterprise Architect

ArcGIS Toolbox Pages

The ArcGIS Toolbox pages provide elements and connectors that you can use to model ArcGIS geodatabase concepts and relationships. The ArcGIS Toolbox consists of five Core pages:

- Domains - for coded value and range domains
- Features and Tables - for custom feature types and tables
- Network Features - to identify geometric network and topology Packages
- Raster - for raster datasets
- Workspace - for ArcGIS workspace and spatial reference information

Two additional Toolboxes group the objects used specifically in creating Geometric Network and Topology diagrams.

Access

On the Diagram Toolbox, click on to display the 'Find Toolbox Item' dialog and specify 'ArcGIS:

- Core'
- Geometric Network' or
- Topology'

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Alt+5

ArcGIS Toolbox Pages

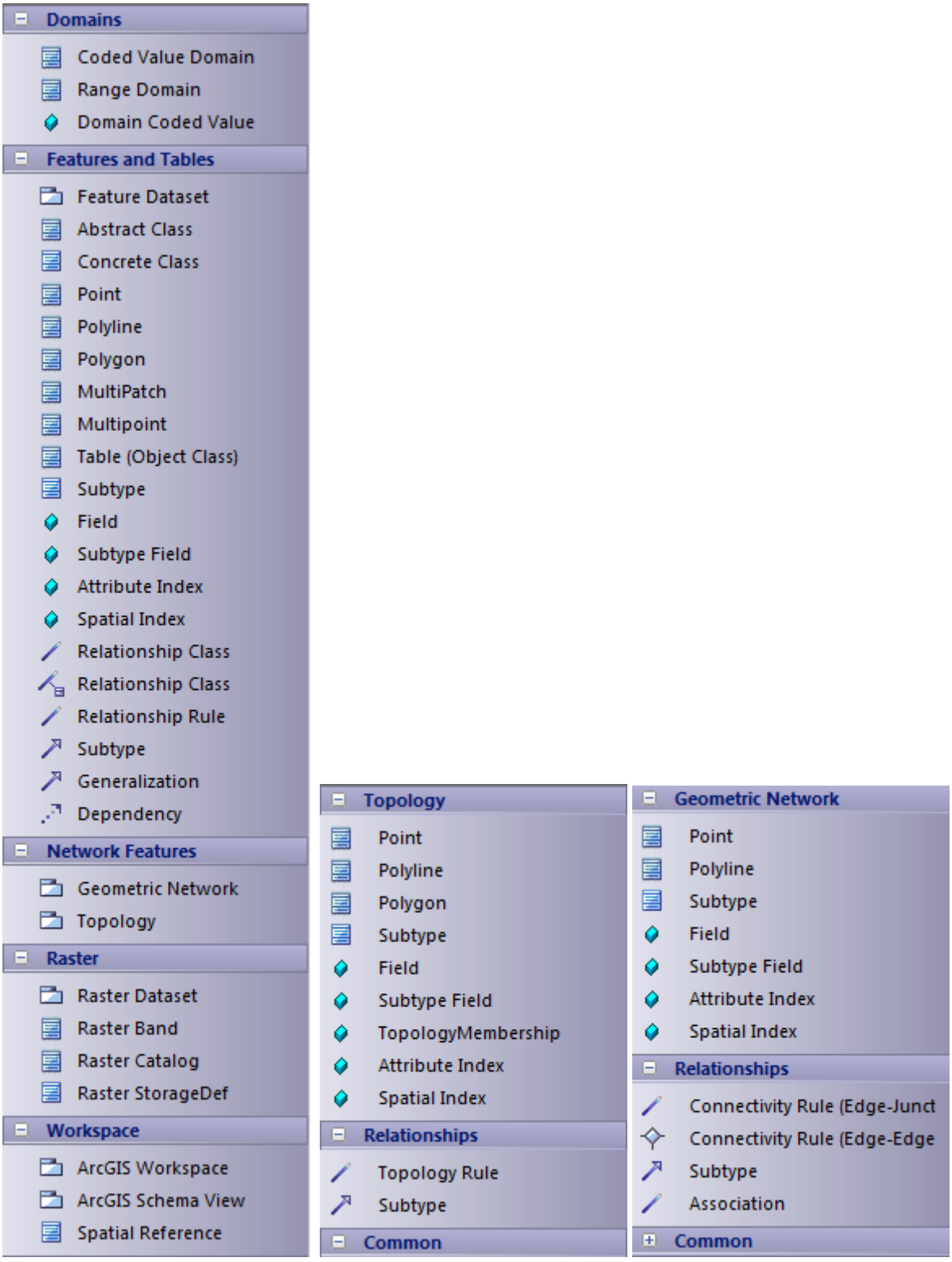


Diagram toolbox icons

Toolbox Icon	Description
Packages	

ArcGIS Workspace	The geodatabase workspace Package, which holds all the ArcGIS modeling elements. Export the contents of this Package to produce the Geodatabase XML Workspace Document, which can be imported to Esri ArcCatalog.
ArcGIS Schema View	A stereotyped Package that represents a subset of the geodatabase schema defined within the ArcGIS Workspace Package. ArcGIS Schema View Packages are useful if you need to create partial or modular schemas based on your complete geodatabase design. You can create any number of ArcGIS Schema View Packages under your ArcGIS Workspace Package. Add this element to a diagram under your workspace, then create a UML Dependency connector from it to each Package to include in the generated XML Workspace Document. For example, you could include only a subset of your Feature Datasets and Domains, by drawing UML Dependency connectors to the appropriate Packages. To export your ArcGIS Schema View for use with ArcCatalog, right-click on it and select the 'Specialize | ArcGIS | Export to ArcGIS Workspace XML' option. The system generates a Workspace XML document containing only the elements associated with the ArcGIS Schema View Package. See the *Export Modular ArcGIS Schemas* topic.
Feature Dataset	A stereotyped Package that holds or organizes Point, Polyline, Polygon or Multipatch elements with the same spatial reference, geometry type and attribute fields (that is, Feature Classes). The Feature Dataset is only created under the ArcGIS Workspace Package; it can not be created under another Feature Dataset Package. Feature Dataset Packages can contain other types of sub-Packages, however, which can be useful for organizing large Feature Datasets. When exported to an XML Workspace document, elements of any subPackages are included while the subPackages themselves are ignored, resulting in a 'flattened' model hierarchy. Although ArcGIS forbids Tables (ObjectClasses) being defined under Feature Datasets, Enterprise Architect lets you model Tables under Feature Datasets for convenience. On export, Tables are placed at the root level to create a valid schema.
Geometric Network	An extended UML Package that represents the logical relationships between features in a network system – implemented in ArcGIS as a geometric network.
Raster Dataset	A stereotyped Package that holds or organizes the raster data (as Raster Band elements).
Topology	An extended UML Package that represents the shared geometry of a set of Feature Classes from a Feature Dataset.
Elements (in alphabetical order)	
Abstract Class	A standard UML Abstract Class, representing a concept and set of fields, that can be shared by multiple Feature Classes. Feature Classes that connect to an Abstract Class via an Inheritance relationship gain all of its fields. Since the geodatabase does not directly support Abstract Classes, inherited fields are exported into the definition of each child Feature Class when generating a schema from the model.
Coded Value Domain	An extended UML Class, representing a set of valid values that might apply to any type of attribute.

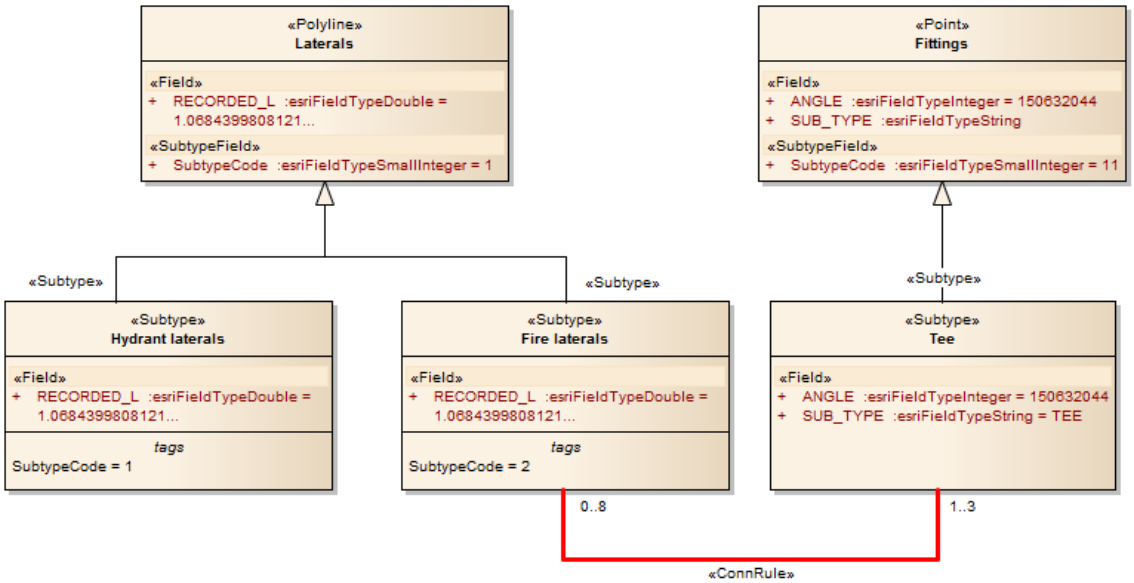
Concrete Class	A standard UML Class that can represent a Feature Class or a Table in ArcGIS, depending on the stereotype setting of its parent Class. If the element has no stereotyped parent Class, it is treated as an ArcGIS Table (Object Class) by default.
MultiPatch	An extended UML Class, representing the ArcGIS MultiPatch.
Multipoint	An extended UML Class, representing the ArcGIS Multipoint.
Point	An extended UML Class, representing the ArcGIS Point.
Polygon	An extended UML Class, representing the ArcGIS Polygon.
Polyline	An extended UML Class, representing the ArcGIS Polyline.
Range Domain	An extended UML Class, representing a valid range of numeric values that might apply to a numeric type of attribute.
Raster Band	An extended UML Class, representing one layer of a matrix of cell values. Every Raster Dataset contains one or more Raster Bands.
Raster Catalog	An extended UML Class, representing a collection of Raster Datasets in the geodatabase.
Raster StorageDef	An extended UML Class, representing the storage properties for a Raster value in the geodatabase; this information is required when a Raster Dataset Package element is created.
Spatial Reference	An extended UML Class that defines the spatial reference information of your schema, such as a coordinate system and XYTolerance. You can define one or more Spatial Reference elements, which you link to other ArcGIS elements via their Spatial Reference Tagged Value.
Subtype	An extended UML Class, holding a subset of the attributes of an element in the Feature Dataset.
Table (Object Class)	An extended UML Class, representing a collection of nonspatial data of the same type or Class.
Relationships (in alphabetical order)	
Association	A normal UML Association connector.
Connectivity Rule (Edge-Edge)	An extended UML N-ary Association that models the valid relationships between edge elements in a Geometric Network. For an example, see *Connectivity Rule Examples*.
Connectivity Rule (Edge-Junction)	An extended UML Association that models the valid relationships between edge and junction elements in a Geometric Network. For an example, see *Connectivity Rule Examples*.
Dependency	A normal UML Dependency connector.

Generalization	Indicates inheritance from the specific classifier to a general classifier.
Relationship Class	An extended UML Association, providing the relationship between: • Two elements in the Feature Dataset, or • An element in the Feature Dataset and an Object Class element
Relationship Class	An extended UML Association Class, providing the attributed relationship between: • Two elements in the Feature Dataset, or • An element in the Feature Dataset and an Object Class element
Relationship Rule	An extended UML Association that determines which subtypes can be related in the geodatabase.
Subtype	An extended UML Association, providing the relationship between a Feature Class element and a Subtype element.
Topology Rule	An extended UML Association that connects Feature Class and Subtype elements in the geodatabase.
Attributes (in alphabetical order)	
Attribute Index	An extended UML attribute that represents the ArcGIS Attribute Index.
Domain Coded Value	An extended UML attribute that specifies the value of an ArcGIS Coded Value Domain.
Field	An extended UML attribute that represents an ArcGIS field of the geodatabase, in a Table or Feature Class.
Spatial Index	An extended UML attribute that represents the ArcGIS Spatial Index.
Subtype Field	An extended UML attribute that represents the 'subtype' field of an ArcGIS Table or Feature Class.
TopologyMembership	An extended UML attribute that represents the accuracy ranks of a Feature Class.

Connectivity Rule Examples

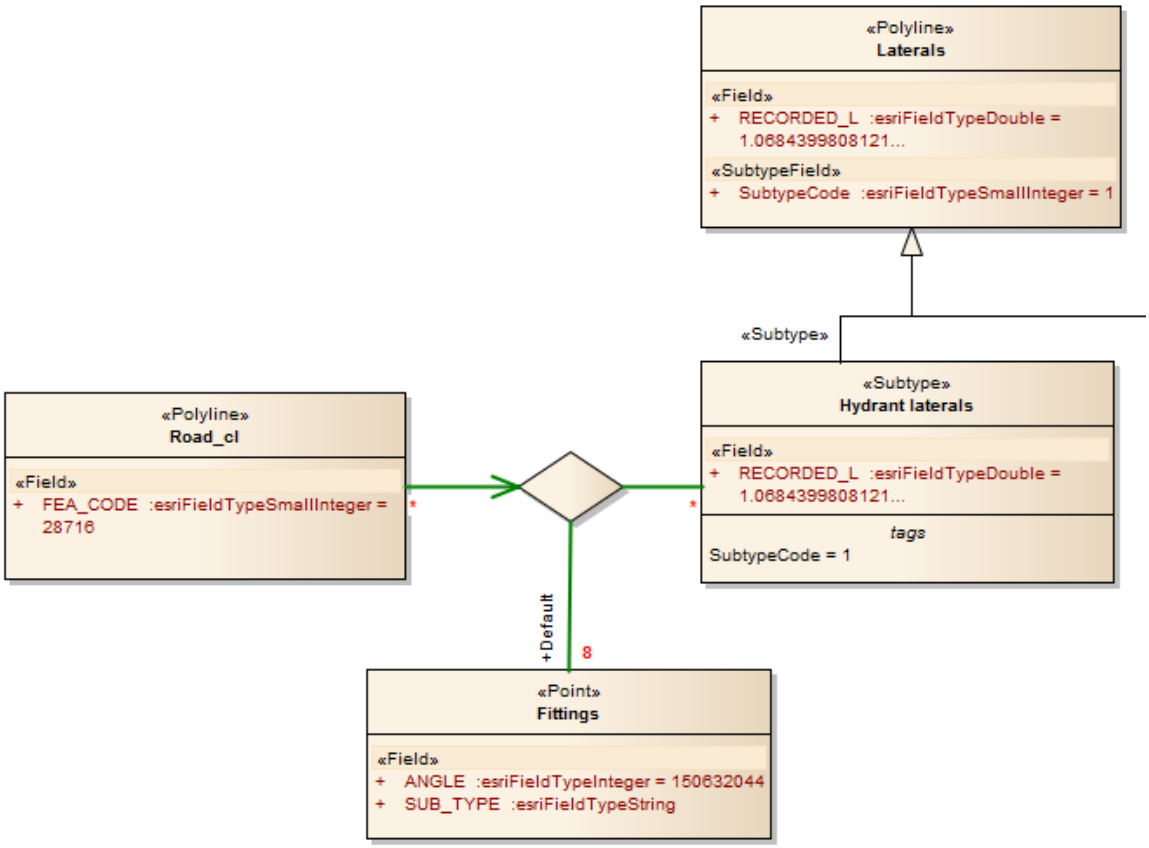
In an ArcGIS Geometric Network diagram, you can use one or other of the two Connectivity Rule relationships - Edge-Junction or Edge-Edge. These examples illustrate the use of each type.

Edge-Junction Connectivity Rule



- The Connectivity-Rule (Edge-Junction) connector is a UML binary Association connector
- The connection includes one edge element («Point», or «Subtype» with «Point» as parent) and one junction element («Polyline», or «Subtype» with «Polyline» as parent)
- Cardinality can be set from the source and target 'Multiplicity' fields on the connector 'Properties' dialog
- You can set the 'Source Role' or 'Target Role' fields to 'Default' on the connector 'Properties' dialog
- All the elements within this Edge-Junction rule must be held in the «GeometricNetwork» Package

Edge-Edge Connectivity Rule



- The Connectivity-Rule (Edge-Edge) connector is a UML N-ary Association connector
- The connection should include two edge elements («Polyline», or «Subtype» with «Polyline» as parent) and any number of junction elements («Point», or «Subtype» with «Point» as parent)
- It is recommended that you use a Direct Association connector, drawn from one of the edge elements to the N-ary element, to indicate the 'from' Class - in the diagram, Road_cl is the edge element that is set as the 'from' Class; for the rest of connection, you can use Association connectors to connect the edge or junction element and the N-ary element, drawn either from the edge or junction elements to the N-ary element, or from the N-ary element to the edge or junction elements
- Cardinality can be set from the source or target element 'Multiplicity' fields on the connector 'Properties' dialog; you only need to set the multiplicity of one end of the connector - if both ends are set, only the multiplicity of the target end is used
- You must mark one of the Junction-N-ary connections as Default, using the 'Source Role' or 'Target Role' field on the connector 'Properties' dialog
- All the elements within this Edge-Edge rule must be held in the «GeometricNetwork» Package

Topology Example

In geodatabases, topology defines the spatial relationship between geographic features; that is, how Point, Polyline, and Polygon features share coincident geometry. Topology is fundamental to data integrity in a GIS database. In the Enterprise Architect ArcGIS profile, you use a «Topology» Package to model data integrity among the Feature Classes.

Modeling topology in the Enterprise Architect ArcGIS model is simple:

1. Select a «FeatureDataset» Package in which to create topology relationships.
2. Open the diagram under the «FeatureDataset» Package.
3. From the Diagram Toolbox ArcGIS Network Features page, drag and drop a «Topology» Package icon onto the diagram; this creates a Package that will contain all the elements and relationships that are required for topology definition.

A Topology defined in Enterprise Architect has these characteristics:

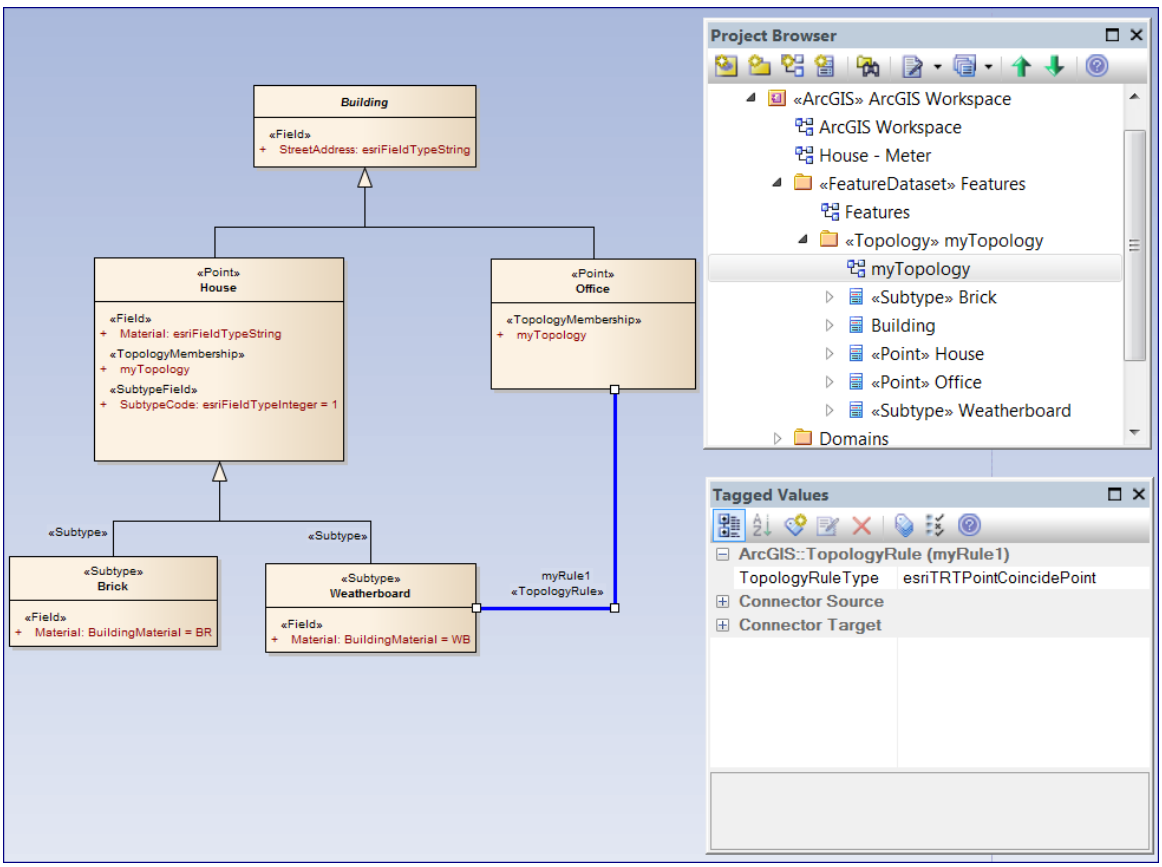
- The «Topology» Package cannot be created outside a « FeatureDataset» Package
- Within one «FeatureDataset» Package, multiple «Topology» Packages can be created
- A Feature Class (Point, Polyline or Polygon) can only participate in one «Topology» Package
- A Feature Class cannot participate in both a «Topology» Package and a «GeometricNetwork» Package

Elements of Topology

Element	Description
Name	You can define the name for the Topology as the «Topology» Package name.
List of Feature Classes	Either: • Create new Feature Classes from the Diagram Toolbox or • Drag existing Feature Classes from the Project Browser into the «Topology» Package
X,Y Cluster Tolerance and Z Cluster Tolerance	You define the cluster tolerance values using the ClusterTolerance and ZClusterTolerance Tagged Values of the «Topology» Package.
Accuracy ranks	Accuracy ranks are defined using the Tagged Values of the TopologyMembership attribute, which you can create using the 'TopologyMembership' icon on the 'Topology' page of the Diagram Toolbox. Add this stereotyped attribute to each Feature Class element and then set a value for each rank. • The name of the attribute should be the name of the «Topology» Package • You do not need to set the type of the attribute Each Feature Class only has one TopologyMembership attribute. If you do not add a TopologyMembership attribute to a Feature Class, the ArcGIS exporter will generate a set of default ranking values for you. The values for XYRank and ZRank are between 1 and 50.
Topology Rules	Topology Rules are represented by a UML Association connector that has the «TopologyRule» stereotype. You can create the connector using the 'Topology Rule' icon on the 'Topology' page of the Diagram Toolbox.

	Use this connector to link: • Two Feature Class («Point», «Polyline» or «Polygon») elements • Two «Subtype» elements • A Feature Class («Point», «Polyline» or «Polygon») element to a «Subtype» element • A Feature Class («Point», «Polyline» or «Polygon») itself, or • A «Subtype» element itself The TopologyRuleType tag is used to define the type of Topology Rule.

Example Topology Rule connection



Relationship Rule Example

In ArcGIS modeling, you can use relationship rules to refine the cardinality of a «RelationshipClass» connector between a source Feature Class or Table and a destination Feature Class or Table; a Relationship Class connector only defines the initial cardinality, such as one-to-many or many-to-many.

A relationship rule in Enterprise Architect is represented by a «RelationshipRule» connector, a stereotyped UML Association connector, which you can create using the Relationship Rule icon on the 'ArcGIS Core' page of the Diagram Toolbox. You set the cardinality from the source and target 'Multiplicity' fields on the connector 'Properties' dialog.

When creating a «RelationshipRule» connector between two objects, you must have:

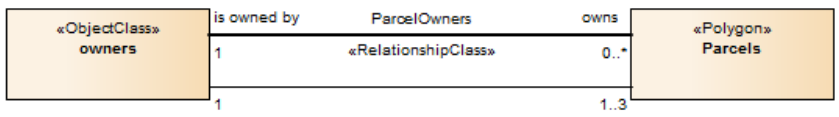
- An existing «RelationshipClass» connector between the two objects that you want to define the relationship rule for; if there is no connector, the «RelationshipRule» you create is ignored during ArcGIS schema generation
- A cardinality range at each end that is compatible with the cardinality of the parent «RelationshipClass»; for example, if you define a cardinality of 1-M in a «RelationshipClass» connector, the source end of the «RelationshipRule» connector must be 1, while you can set the target end of the «RelationshipRule» to a specific number such as 3 (see the example diagrams in this topic)

Relationship rules can also restrict the type of object in the source Feature Class or Table that can be related to a certain kind of object in the destination Feature Class or Table. For example, if the source Class has no subtype elements, the relationship rule applies to all features. If the source Class has subtype elements and the «RelationshipRule» is linked to one of the subtype elements, this means only the associated subtype element is related to the «RelationshipRule». The same restriction is also applied to the destination Feature Class or Table.

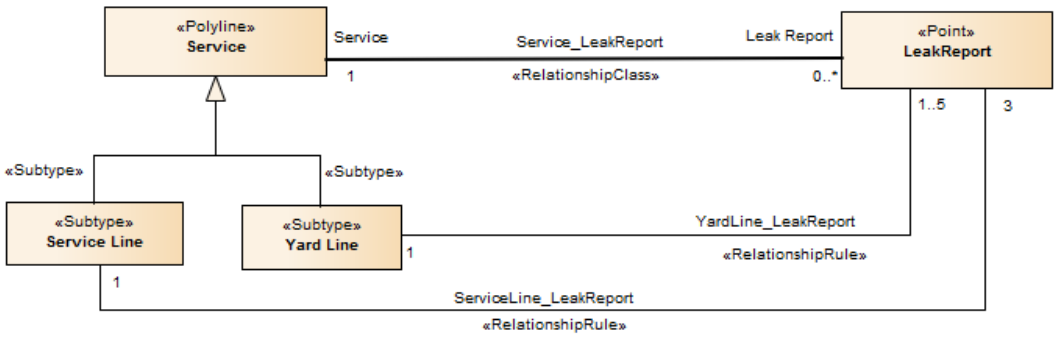
Examples

This diagram provides three examples of possible «RelationshipRule» connections in an ArcGIS model. A custom Line Thickness has been applied to highlight the Relationship Class connectors, and the «RelationshipRule» stereotype label has been hidden where appropriate:

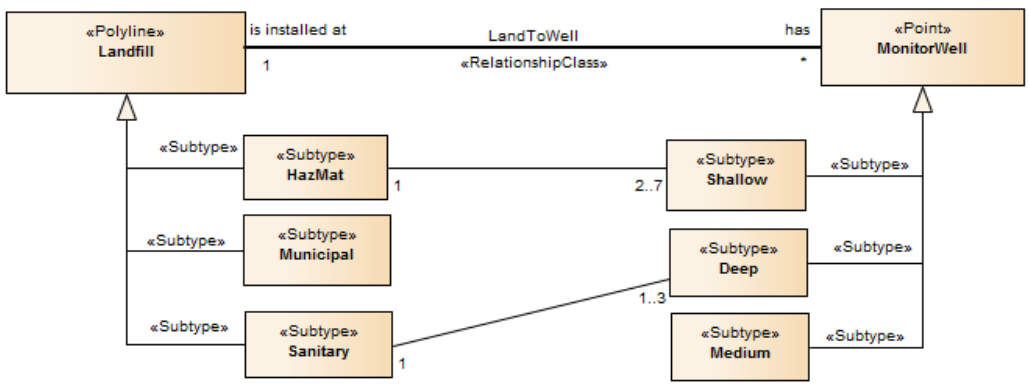
1) Relationship Rule Links feature class and object class/table.



2) Relationship Rule links feature class and subtype.



3) Relationship Rule links subtype and subtype.



Setting ArcGIS Coordinate Systems

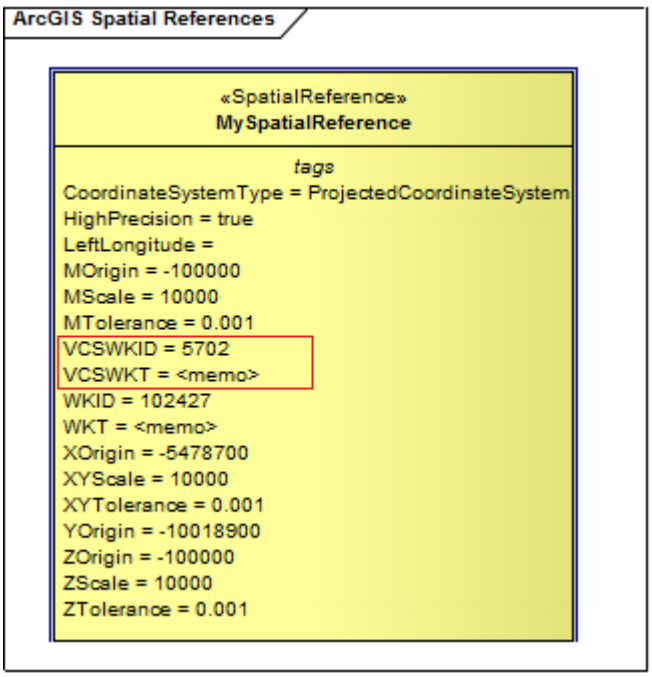
ArcGIS Feature Classes and Feature Datasets use spatial references, which consist of a coordinate system and associated values such as XY resolution and various tolerance values.

You can capture spatial reference properties using a Class stereotyped as «SpatialReference», which is available from the ArcGIS Toolbox pages. The ArcGIS model Pattern includes a Package named Spatial References, as a placeholder for creating such elements.

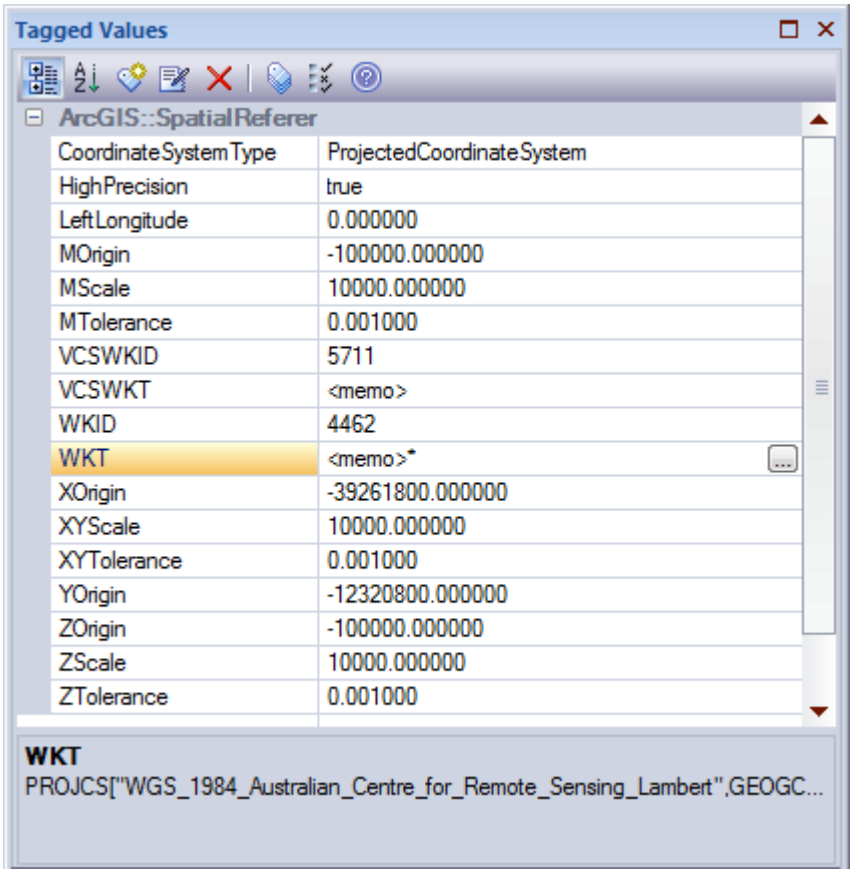
To help you model spatial reference properties, Enterprise Architect provides a dialog for selecting one of the predefined coordinate systems supported by ArcGIS. When you select a Geographic or Projected coordinate system, Enterprise Architect automatically inserts default values for the associated properties, such as Well Known Text (WKT), resolution, precision or tolerances. These values are held as Tagged Values on the «SpatialReference» element.

You can also add vertical coordinates to a selected Geographic or Projected coordinate system; the vertical coordinate is loaded to the VCSWKT and VCSWKID Tagged Values on the «SpatialReference» element.

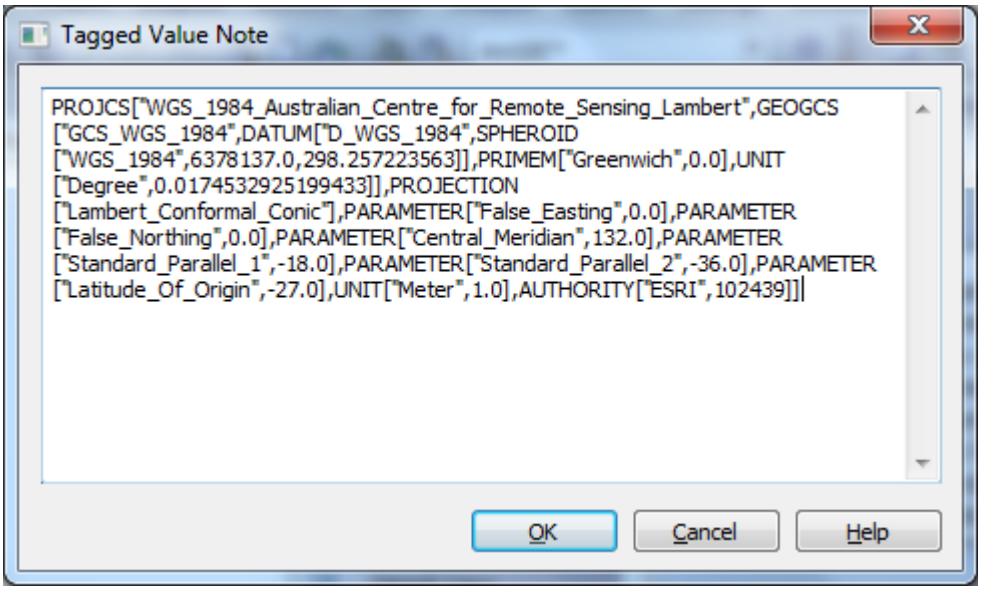
This is an example «SpatialReference» element:



Looking at the WKT Tagged Value in the Tagged Values window for this element, you can see that the 'WGS 1984 Australian Centre for Remote Sensing Lambert Projected Coordinate' system has been selected.



You can expand the information held in this Tagged Value by viewing its Tagged Value Note.



Define a Spatial Reference element

Step	Action
1	Open the diagram under the Spatial References Package of your ArcGIS model. (You can actually use any ArcGIS diagram in your model to define Spatial Reference elements; however,

	this diagram is a convenient placeholder created by Enterprise Architect's model Pattern for ArcGIS.)
2	Drag a Spatial Reference element from the 'Workspace' page of the ArcGIS Core Toolbox onto the diagram.
3	Right-click on the Spatial Reference element, and select the 'Specialize | ArcGIS | Set Coordinate System' menu option. The 'Set Coordinate System' dialog displays.
4	Expand the Geographic or Projected Coordinate Systems hierarchy as appropriate and click on the required coordinate system in the list.
5	If you want to also apply a vertical coordinate system, click on the button at the right of the 'Vertical Coordinate' field. The 'Set Vertical Coordinate System' dialog displays, containing a hierarchy that you again expand and from which you select a listed vertical coordinate system. Click on the OK button to return to the 'Set Coordinate System' dialog; the 'Vertical Coordinate' field now displays the system you selected.
6	Click on the OK button to close the dialog and return to the diagram. The Tagged Values for the Spatial Reference element are updated with the Coordinate System information you have selected.

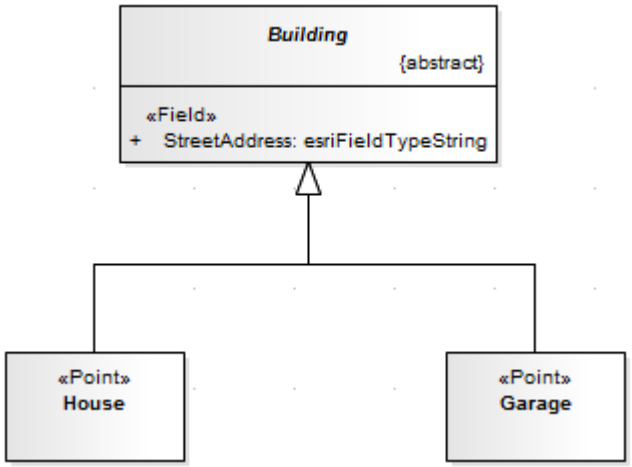
Notes

- You can refer to a «SpatialReference» Class from any other Feature Dataset or Feature Class in your model, using the SpatialReference Tagged Value; the «SpatialReference» Class thus provides a single point of control, should you need to change the Spatial Reference information later
- If a Feature Class element references a «SpatialReference» Class that contains a vertical coordinate, set the HasZ Tagged Value on that Feature Class element to true if you want this Feature Class element to store three-dimensional data
- If you do not refer to a «SpatialReference» Class from any Feature Dataset or Feature Class in your ArcGIS model, the system will generate an XML schema with the Unknown Spatial Reference type for these elements

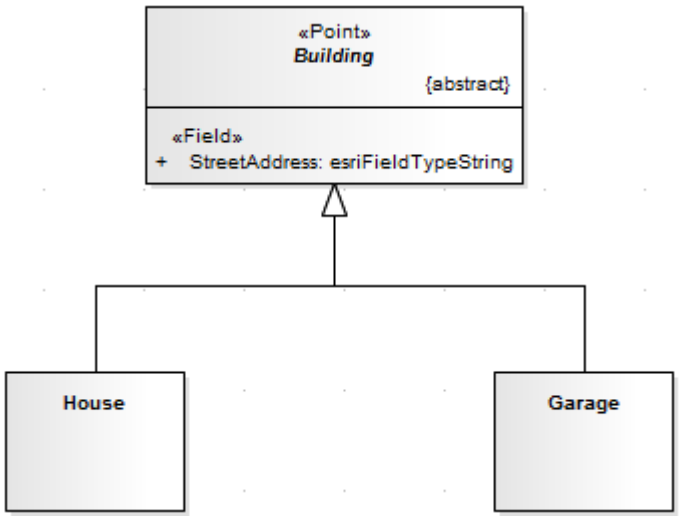
Applying ArcGIS Stereotypes to Abstract Classes

Using the Enterprise Architect UML Profile for ArcGIS, you can specify a geometry stereotype on the Feature Classes in your geodatabase schema. Geometry stereotypes include «Point», «Polyline», «Polygon» and «Multipoint», among others. The ArcGIS Toolbox provides convenient icons for each geometry so that you can drag and drop stereotyped Classes into your geodatabase design model that are immediately ready to export. These Classes are referred to as concrete Classes; since their UML property, IsAbstract, has a value of False, they will be implemented directly in the geodatabase schema.

However, sometimes it is useful to specify the geometry stereotype on an abstract UML Class so that multiple concrete Classes can inherit the geometry, as well as Tagged Values and any fields defined in the abstract Class. This example models houses and garages as Point Feature Classes. Both the 'House' and 'Garage' Classes inherit the 'StreetAddress' field from the abstract Class named 'Building'.



You can create an equivalent model by specifying the stereotype on the abstract Class and using unstereotyped concrete Classes for 'House' and 'Garage', as shown:



The advantages of stereotyping the abstract Class rather than each concrete Class (especially when you have many such derived Feature Classes) include:

- It is easier to change the geometry during design time; you make only one change to the abstract Class stereotype, which then automatically applies to each concrete Class
- It is quicker to create the model in the first place, because you have to edit only one set of Tagged Values associated with the stereotype; the concrete Class might not have to replicate (or override) any of the Tagged Values associated

with its inherited geometry stereotype

- For the same reason, the overall model is smaller and simpler

Create an abstract Class with geometry in the model

A Class is considered to be abstract when its UML property IsAbstract is set to True. When you create a Class using the 'Abstract Class' icon from the ArcGIS toolbox, the IsAbstract property is set to True automatically. You can also set the property manually for any Class, on the 'Details' tab of the Class 'Properties' dialog.

Step	Action
1	Open the relevant diagram in your model.
2	Select the 'Core' page of the ArcGIS Toolbox (click on to display the 'Find Toolbox Item' dialog and specify 'ArcGIS Core') and drag the 'Abstract Class' icon onto the diagram to create the element.
3	If the 'Properties' dialog does not automatically display, double-click on the Abstract Class element.
4	On the 'General' page of the 'Properties' dialog, click on the 'Stereotype' field button and, on the 'Stereotype for Class' dialog, set the 'Profile' to 'ArcGIS' and click on the checkbox against the required geometry stereotype.

Create a concrete Class that inherits an abstract Class's geometry stereotype

When you export your model as a geodatabase schema, the system applies the geometry stereotype from the abstract Class to any of its derived concrete Classes. Furthermore, the exporter will add any missing 'system level' fields. For example, a Class need not specify, nor inherit, a field named 'OBJECTID'. Similarly for the 'Shape', 'Shape_Length' and 'Shape_Area' fields. Although the exporter will use these fields if they are modeled somewhere in the inheritance hierarchy, it will automatically generate valid instances of them as required.

Step	Action
1	Open the diagram that contains the abstract Class.
2	Select the 'Core' page of the ArcGIS Toolbox (click on to display the 'Find Toolbox Item' dialog and specify 'ArcGIS Core') and drag the 'Concrete Class' icon onto the diagram to create the element.
3	Click on the Generalization icon in the Toolbox and then click and drag the cursor from the concrete Class to the abstract Class.
4	Save your diagram.

Notes

- Any concrete Class that does not have a stereotype, and does not inherit one, is exported as a Table (ObjectClass); its OBJECTID field is also inserted if it is not defined in the model
- Concrete Classes can only inherit geometry stereotypes or the «ObjectClass» stereotype from an abstract ancestor

Class; currently, Enterprise Architect does not support stereotype inheritance from other concrete Classes

- In addition to inheriting the stereotype, concrete Classes also inherit fields from ancestor abstract Classes
- You can inherit the stereotype from an abstract Class at any level in inheritance hierarchy; for example, the abstract Class that specifies the geometry can be the grandparent of the concrete Class, rather than the parent Class
- Multiple shapes for a single Feature Class are not supported by ArcGIS, nor by Enterprise Architect's ArcGIS profile; therefore, it would be a modeling error if a concrete Class inherited from more than one geometry-stereotyped abstract Class
- If you specify a given tag on a concrete Class that is already present in one of its parent abstract Classes, the concrete Class has precedence and its Tagged Value will be exported to the schema
- Enterprise Architect does not require you to show the Object and Feature Esri Classes on a diagram, nor even include them in your model, because the system implicitly applies their characteristics when you apply a geometry or ObjectClass stereotype to a Class
- It is not, however, an error to include the Object and Feature Esri Classes and model Generalization links to them, even though they are typically not marked as abstract

Export ArcGIS XML Workspace

When you have modeled your Geodatabase Workspace XML Document (containing the ArcGIS schema), you can export it to an external directory (using the Publish Model Package facility), from which you can then import it to the Esri ArcCatalog.

Access

Click on an ArcGIS stereotyped Package (your ArcGIS Workspace Package) in the Project Browser.

Ribbon	Specialize > Technologies > ArcGIS > Export to ArcGIS Workspace XML or Publish > Model Exchange > Publish As...
Context Menu	Right-click on Package | Specialize | ArcGIS | Export to ArcGIS Workspace XML
Keyboard Shortcuts	Ctrl+Alt+E : Publish

Export the Workspace

Option	Action
Root Package	Display the name of the selected ArcGIS Workspace Package.
Filename	Type in or browse for the file path into which the XML file is to be generated.
XML Type	Select 'ArcGIS' as the XML/XMI version to export the Package to.
Format XML Output	Format the output into readable XML (this takes a few more seconds at the end of the run).
Write Log File	Write a log of the export activity (recommended). The log file is saved to the directory into which the XML file is exported.
View XML	Click on this button to view the exported XML file.
Export	Click on this button to initiate the XML export.
Close	Click on this button to close this dialog.
Progress	Observe the progress of the XML export.

Notes

- The MDG Technology for ArcGIS is available in the Professional, Corporate, Unified and Ultimate editions of

Enterprise Architect

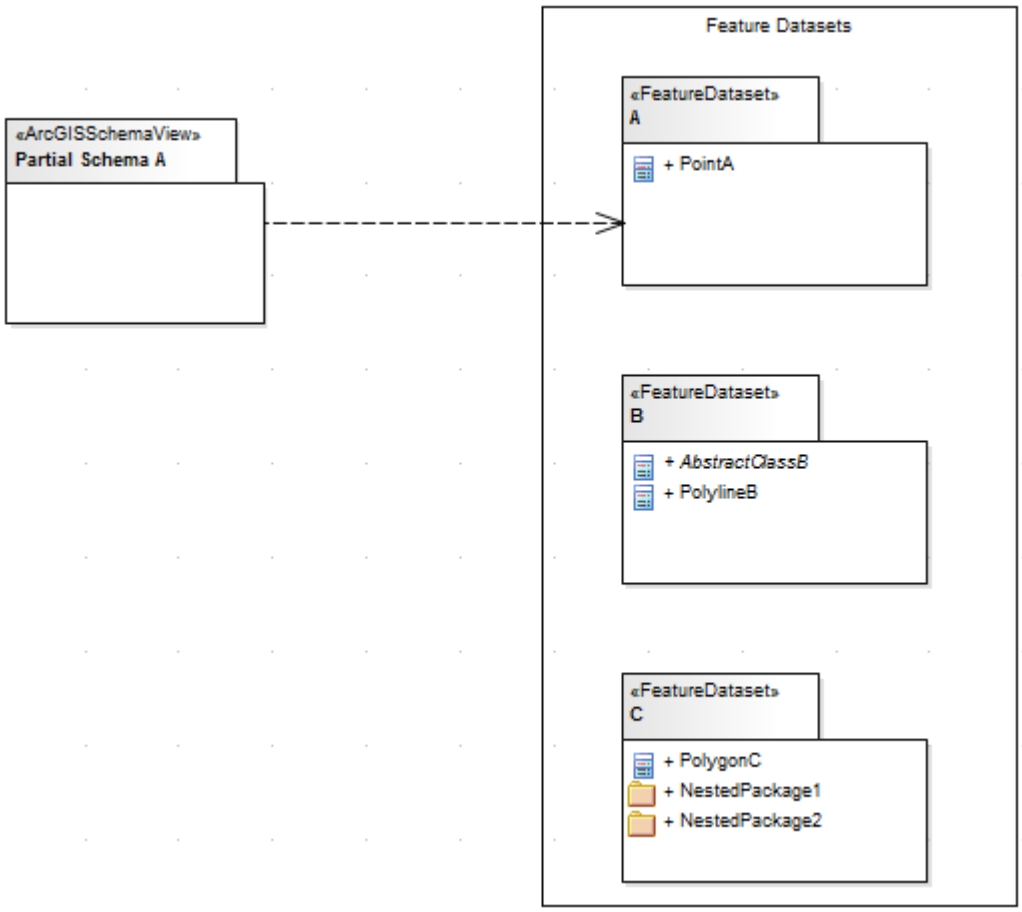
- In the Corporate, Unified and Ultimate editions of Enterprise Architect, if security is enabled you must have Export XMI permission to export to XML
- Before exporting your model to an ArcGIS schema, you must define at least one Spatial Reference element; Spatial Reference elements are referred to by other schema elements via a dynamically linked Tagged Value, named SpatialReference
- The DefaultSpatialReference tag on an ArcGIS Package is used to specify a Spatial Reference that can be applied to all Feature Datasets and Feature Classes in the workspace; therefore, you do not need to apply a Spatial Reference element to each Feature Dataset or Feature Class
- If you do not reference a Spatial Reference Class from any Feature Dataset or Feature Class in your ArcGIS model, Enterprise Architect by default will generate an XML schema with Unknown type of Spatial Reference for these elements

Export Modular ArcGIS Schemas

In Enterprise Architect, in addition to exporting your entire ArcGIS workspace, you can also export partial schemas. This is useful if you have a large geodatabase schema, as might be defined in an industry reference model. You might require the entire schema in some situations, but only require small parts of it for particular spatial applications, such as field data collection. In such a scenario, you would want to export a schema that contains only the Feature Classes, Tables and Domains that your field data application uses, without duplicating parts of your original schema model. You use the «ArcGISSchemaView» stereotyped Package for this purpose.

An «ArcGISSchemaView» Package is modeled as a subPackage of an ArcGIS Workspace Package. You can define any number of «ArcGISSchemaView» Packages - each representing a different subset of the geodatabase schema. You specify which parts of the schema are included by drawing a UML Dependency connector from the «ArcGISSchemaView» Package to each included Package. When you export the «ArcGISSchemaView» Package, the system includes any other Packages that your included Packages depend on (via Dependency connectors).

This figure shows a partial schema that includes only one of the three Feature Datasets from the complete schema.



Create an ArcGISSchema Package

Step	Action
1	Create or open an ArcGIS diagram within your ArcGIS Workspace.
2	Drag the ArcGIS Schema View icon from the Core Diagram Toolbox onto your diagram. A prompt displays to enter the Package name.

3	Type in a meaningful Package name and click on the OK button.
4	Drag onto the diagram any other Packages that you want to include in the exported schema. (You could achieve the same result using the child diagram of the «ArcGISSchema View» Package to draw the included Packages).
5	Draw a Dependency connector from the «ArcGISSchema View» Package to each of the other Packages.

Notes

- Defining the Dependency relationships on a diagram is convenient, but not necessary; as long as the dependencies are defined in the model – irrespective of whether they exist on a diagram – the ArcGIS schema exporter will use them
- You can arrange your dependency diagrams in whatever part of the ArcGIS Workspace seems appropriate – the diagrams can reside under the «ArcGISSchema View » Package itself or under any other element within the ArcGIS Workspace

Export an ArcGIS Schema View for use with ArcCatalog

Step	Action
1	Select the ArcGIS Schema View Package in a diagram or in the Project Browser.
2	Right-click and select the 'Specialize | ArcGIS | Export to ArcGIS Workspace XML' menu option.
3	Identify the target file and click on the Export button. The system generates a Workspace XML document containing only the elements associated with the ArcGIS Schema View Package.

Which related elements are included when you export an ArcGIS Schema View Package?

These rules apply when you export an ArcGIS Schema View Package:

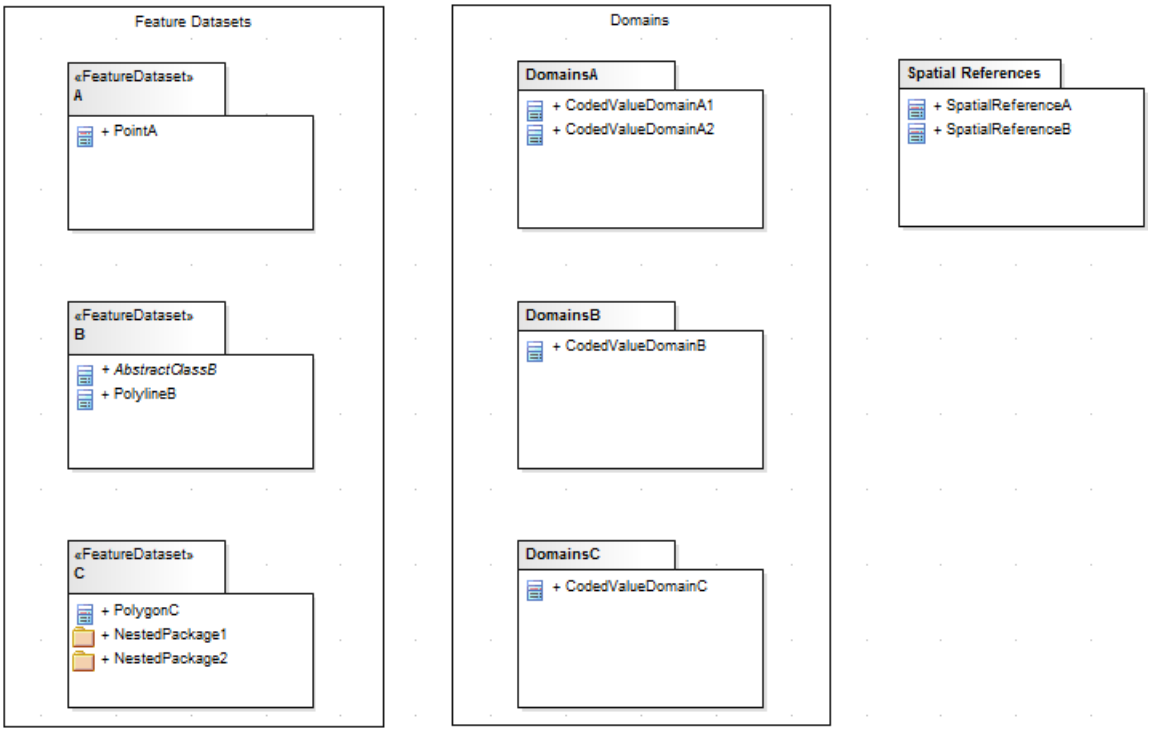
- Dependencies are modeled using the UML Dependency connector
- All elements of a Package that the ArcGIS Schema View depends on (directly or indirectly) are included in the generated schema
- All fields inherited from Abstract Classes by included elements are exported, regardless of the Package in which the Abstract Classes reside
- All Coded Value Domain elements to which included elements refer are exported, regardless of the Package in which the Coded Value Domain elements reside
- If an ArcGIS Schema View Package depends on one or more subPackages of a Feature Dataset Package, the Feature Dataset is exported with only those elements contained in the linked subPackages - no Feature Classes, Domains and Tables that are directly contained in the FeatureDataset Package are exported, because of the Dependency to one of its subPackages; therefore, if you want to export the entire Feature Dataset you must use a Dependency to the

Feature Dataset Package itself

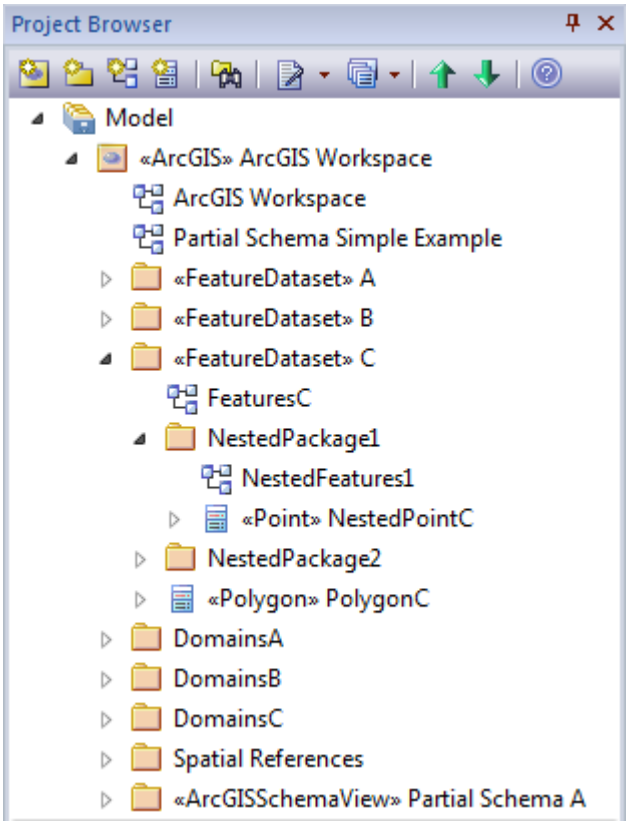
- If a field of an included element refers to a Coded Value Domain element, that Coded Value Domain element is exported, irrespective of whether the ArcGIS Schema View Package has an explicit dependency on the Coded Value Domain element's Package
- If an included element has a Relationship Class connector to another element X AND element X is not already included by the ArcGIS Schema View, neither element X nor the Relationship Class connector are exported; the log file will hold a list of names of the Relationship Class connectors that, for this reason, are not exported

Examples of modeling partial schemas

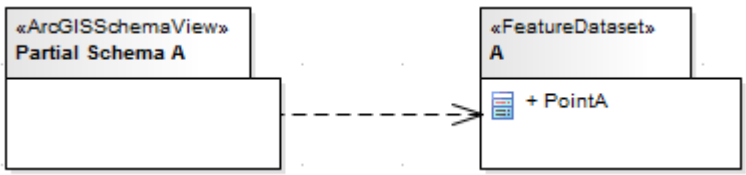
Consider this complete Workspace, which includes three Feature Datasets named A, B and C, and three Packages of Coded Value Domains named DomainsA, DomainsB and DomainsC:



The corresponding model hierarchy in the Project Browserresembles this:

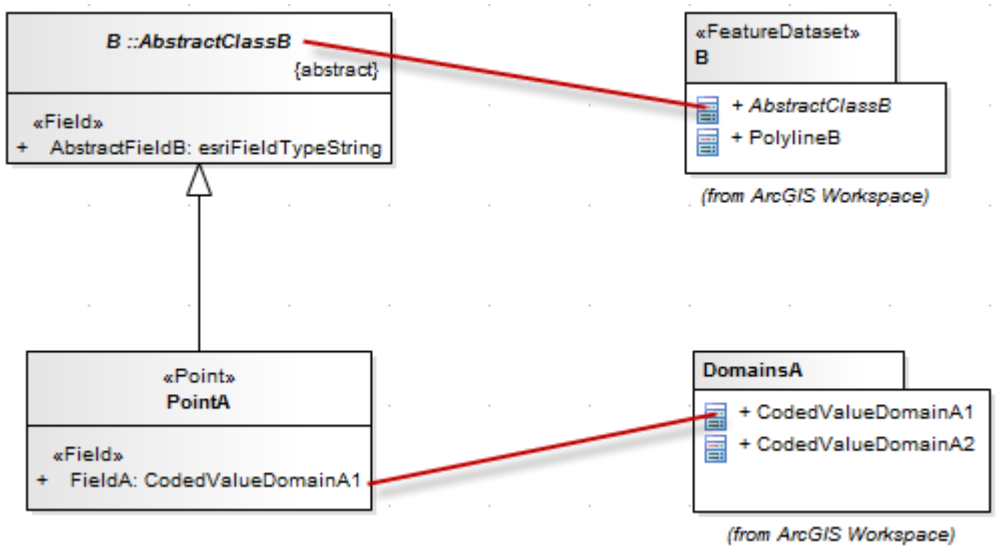


If you want to export only Feature Dataset A and its required elements, you can model the Schema as a partial schema that includes a single Feature Dataset, as shown:

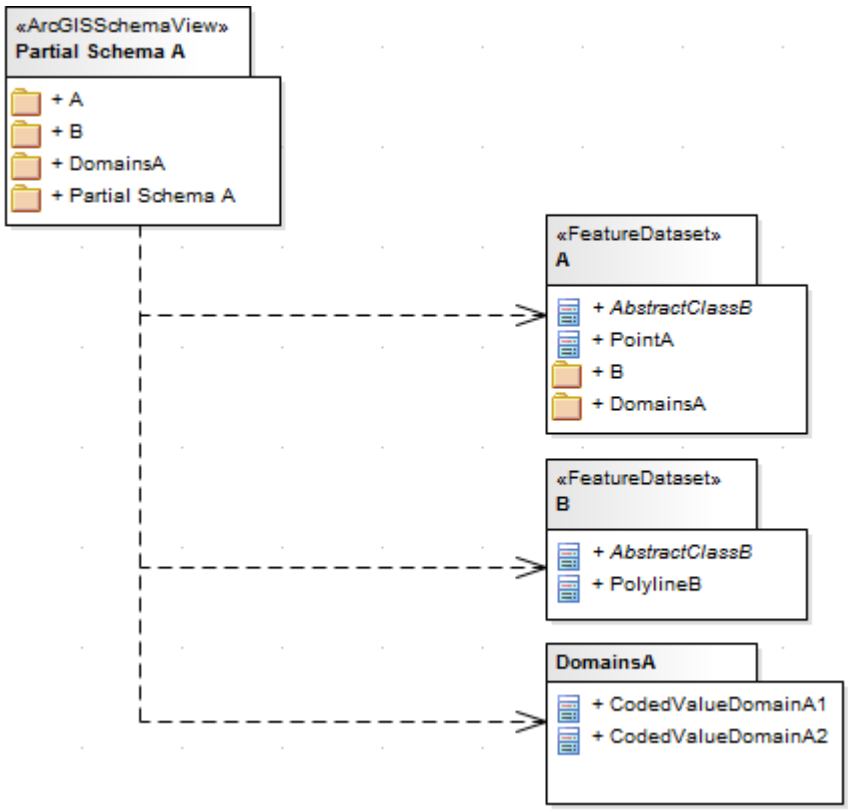


(This diagram is equivalent to the first diagram provided at the start of the topic.) Assuming that Point A depends on no other elements, the resulting schema would include only FeatureDataset A with its Feature Class, Point A.

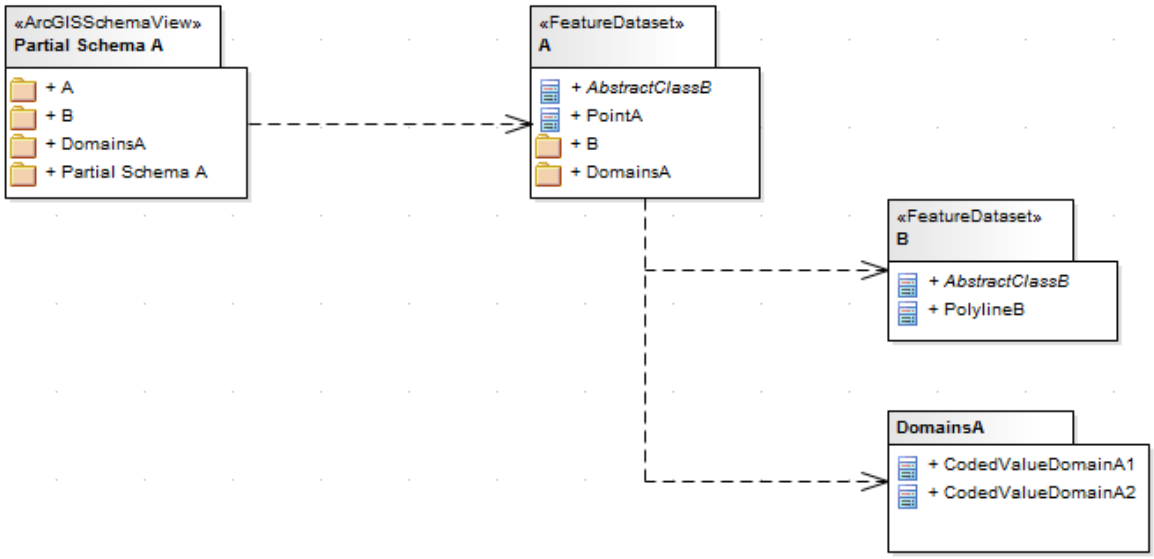
Now assume that Point A inherits from the Abstract Class AbstractClassB (defined in FeatureDataset Package B) and that one of A's fields has type CodedValueDomainA1 defined in the DomainsA Package (as in the next model diagram). Now, the same Partial Schema model would result in an exported schema that included the fields of AbstractClassB and CodedValueDomainA1, even though Partial Schema A does not explicitly depend on Package B or Package DomainsA, because partial schemas automatically include elements that are related by inheritance or are referred to by field types. The exporter thus helps you to generate valid ArcGIS schemas by including such required elements.



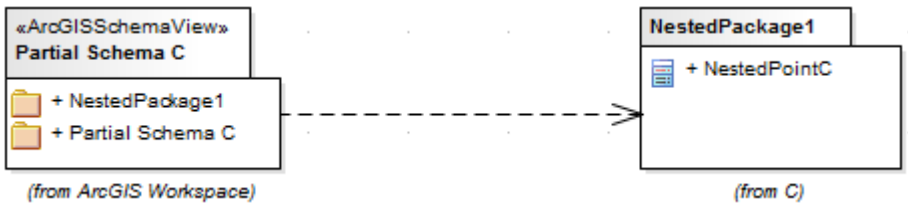
If you wanted to include all CodedValueDomains in DomainsA and all Feature Classes in FeatureDataset B (including any domains they depended on), you could model this situation to include entire Packages of elements in a partial schema via direct and indirect UML Dependency connectors, as shown.



You can also include Packages via indirect Dependency connectors. For example, you can achieve the same result as in the previous example by linking Packages to, say, Feature Dataset A instead of linking them directly to the ArcGIS Schema View.



Finally, if you want to create a partial schema that includes only the elements in, say, NestedPackage1, you can model the scenario as a partial schema Package that refers to nested Packages within a Feature Dataset.



The resulting schema would include a Feature Dataset named C that contained all elements within NestedPackage1. The elements in NestedPackage2 would be excluded, as would PolygonC (assuming no explicit relationships existed with the elements of NestedPackage1).

Import ArcGIS XML Workspace

If you have a Geodatabase Workspace XML Document (containing the ArcGIS schema) you can import it into your Enterprise Architect project as a UML model.

Before running the import, deselect the 'Sort Features Alphabetically' checkbox on the 'Objects' page of the 'Preferences' dialog (Start > View > Preferences). This ensures that the fields are imported and organized in Enterprise Architect in the same order as in the source.

Access

Click on the target Package in the Project Browser.

Ribbon	Publish > Technologies > Publish > ArcGIS > Import ArcGIS Workspace XML or Publish > Model Exchange > Import XMI > Import Model XMI : Other XML File Formats > ArcGIS
Context Menu	Right-click on Package | Specialize | ArcGIS | Import ArcGIS Workspace XML
Keyboard Shortcuts	Ctrl+Alt+I : Other XML Formats | ArcGIS

Import a Geodatabase Workspace XML document

Option	Action
Filename	Type in or browse for the name of the ArcGIS XML file to import.
Create Diagrams	Select the checkbox to create Class diagrams under the imported Packages.
Hide System-Level ArcGIS Fields on Diagrams	Select the checkbox to hide these stereotyped attributes: • RequiredField • AttributeIndex • SpatialIndex on these stereotyped Classes: • Point • Polyline • Polygon • MultiPatch The 'RequiredField' and 'AttributeIndex' attributes are also hidden for the Table (Object Class) Class. This option is enabled only when the 'Create Diagrams' checkbox is selected.
Strip GUIDs	The 'Strip GUIDs' feature is currently mandatory for ArcGIS imports, which means that elements are created 'as new' each time an ArcGIS schema is imported.
Write Log File	Select the checkbox to write a log of import activity (recommended).

	The log file is saved in the directory from which the file is being imported, with the same name as the imported file plus the suffix _import.log.
View XML	Click on this button to view the XML before import.
Import	Click on this button to import the ArcGIS XML file.
Close	Click on this button to close this dialog.
Help	Click on this button to display this Help page.
Import Progress	This field indicates the progress of the import.

Notes

- The MDG Technology for ArcGIS is available in the Professional, Corporate, Unified and Ultimate editions of Enterprise Architect

Validate an ArcGIS Workspace

When you have developed or imported an ArcGIS model, you can validate it against a set of rules in a system-provided ArcGIS validation table.

Access

Ribbon	Specialize > Technologies > ArcGIS > Validate ArcGIS Model
Context Menu	Project Browser | Right-click «ArcGIS» Workspace Package | Specialize | ArcGIS | Validate ArcGIS Model

Process

The option launches a validation script on the workspace. While running, the script logs information to the 'ArcGIS Model Validation' tab of the System Output window. Check the script output for errors and warnings.

There are two ways to investigate the errors reported by the model validation script:

- Expand the System Output window and review the errors and warnings directly; you can double-click on a warning or error line to highlight the element or attribute the message relates to, in the Project Browser
  or
- Copy all of the output to a text file and open the file using your preferred text editor; this is likely to provide cleaner formatting of the script's output

MDG Technology For GML

MDG Technology for GML is the implementation of the Open Geospatial Consortium's Geography Markup Language (GML) 3.3 for Enterprise Architect, which provides an XML grammar for geographical feature modeling capabilities within Enterprise Architect at or later than Release 10.

Through the MDG Technology for GML, you can:

- Apply a UML Profile for the Geography Markup Language (GML) 3.3
- Make use of customized diagram types and toolbox pages, for convenient access to elements and relationships to model geographical features effectively
- Generate GML Application Schema files

Notes

- The MDG Technology for GML is available in the Professional, Corporate, Unified and Ultimate editions of Enterprise Architect

UML Profile for GML

The MDG Technology for GML is built into the Enterprise Architect installer. A key component of the technology is the UML Profile for GML.

Access

Ribbon	Specialize > Technologies > GML
Context Menu	Right-click on Package | Specialize | GML

Features

Feature	Detail
Profile Support	You can develop GML constructs quickly and simply, through use of the built-in MDG Technology for GML facilities provided in the form of: • A GML diagram type, accessed through the 'New Diagram' dialog • GML pages in the Diagram Toolbox that map GML concepts to appropriately stereotyped UML elements • GML element and relationship entries in the 'Toolbox Shortcut Menu' and 'Quick Linker'
GML Toolbox Page	The GML Toolbox pages contain elements and connectors to model geographical features effectively.
UML Classes from ISO	(Optional) You can download the UML Classes implemented in ISO/TC 211 as an XMI file, then import the XMI file into Enterprise Architect as a Package containing diagrams and standard UML Classes, which you can reuse in your model. • Not all UML Classes implemented in ISO/TC 211 have a corresponding mapping in GML; the Classes that have a mapping (as specified in the GML 3.2.1 specification) are specified in the configurable file GMLClassMapping.xml in the 'Sparx Systems > EA > Config > GML' folder • The Namespace information for these Classes is specified in the configurable file GMLNamespaces.xml in the 'Sparx Systems > EA > Config > GML folder'
GML Application Schema Generation	Any model you create using the built-in MDG Technology for GML can be exported as a GML Application Schema. Using the configurable file GMLStereotypes.xml in the 'Sparx Systems > EA > Config > GML' folder, you can specify aliases for the standard GML stereotypes. The GML Application Schema Generator will also consider these aliases during Schema generation.

Notes

- The MDG Technology for GML is available in the Professional, Corporate, Unified and Ultimate editions of Enterprise Architect

GML Toolbox Pages

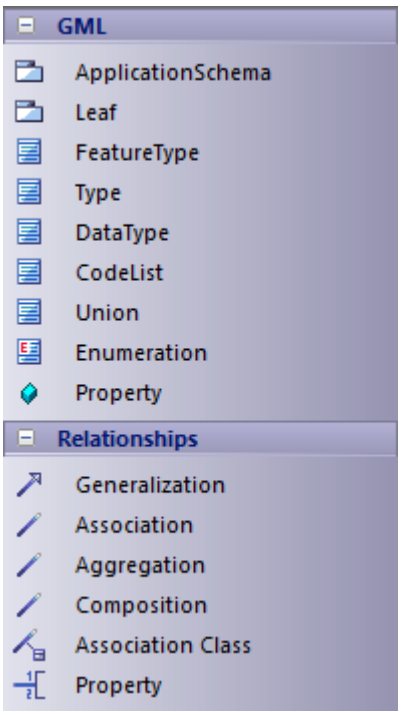
When you begin to model GML concepts and relationships, you can represent them on a GML diagram using the elements and connectors provided in the GML Toolbox pages.

Access

On the Diagram Toolbox, click on to display the 'Find Toolbox Item' dialog and specify 'GML'.

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Alt+5

GML Toolbox pages



GML Element Toolbox

Item	Description
FeatureType	An extended UML Class that maps to an XSD Global element, whose content model is a globally scoped XML Schema type derived by direct/indirect extension of gml:AbstractFeatureType.
Type	An extended UML Class that maps to an XSD element, whose content model is a

	globally scoped XML Schema type derived by direct/indirect extension of gml:AbstractGMLType.
DataType	An extended UML Class that maps to an XSD Global element, whose content model is a globally scoped XML Schema complexType.
CodeList	An extended UML Class, that maps to: • A union of an enumeration and a Pattern, or • an external Dictionary
Union	An extended UML Class that maps to an XSD Choice group, whose members are GML objects or features, or objects corresponding to DataTypes.
Enumeration	An extended UML Class that is a restriction of XSD string with enumeration values.

GML Package Toolbox

Item	Description
ApplicationSchema	The GML workspace Package, which holds all the GML modeling elements. You can export the contents of this Package to produce the GML Application Schema.
Leaf	A stereotyped Package that can be created under an Application Schema. Leaf is a convenient Package for grouping elements within an Application Schema. If the value of the tag xsdDocument on this Package is empty, the contents of this Package will be exported as a part of the parent Package whose tag xsdDocument is not empty.

GML Relationship Toolbox

Item	Description
Generalization	Identifies an element as a specialized descendant of another element, containing additional properties. This relationship is allowed between: • Feature Types • DataTypes and Unions • Types An element can have a maximum of one Generalization connector.
Association	Identifies the connection between two elements. Only navigable Association Ends will be mapped to GML Application Schema.

Aggregation	An extended UML Association that shows that an element is composed of other elements.
Composition	An extended UML Association that is a stronger form of Aggregation, indicating ownership of the whole over its parts.
Association Class	Identifies the attributed connection between two elements. Only navigable Association Ends can be mapped to GML Application Schema.
Attribute	
Property	An extended UML Attribute, whose type is either a property type (if the type is a complex type) or a simple type.
Association Role	
Property	An extended UML Association Role, whose type is always a property type.

Notes

- In GML, Association Class is specific to GML 3.3; when generating GML 3.2.1 Application Schema, the Association and Class of an Association Class are treated as separate entities
- The values for tags on the «Application Schema» and «Leaf» stereotyped Packages can be entered using either:
  - The Tagged Values window or
  - The Generate GML Application Schema dialog

Generate GML Application Schema

After you have created a GML model using the built-in MDG Technology for GML, you can generate GML Application Schema from the model Package.

Access

Ribbon	Specialize > Technologies > GML > Generate GML Application Schema
Context Menu	Right-click on the Package | Specialize | GML | Generate GML Application Schema

Generate GML Application Schema

Option	Action
Source Package	Displays the name of the selected «Application Schema» stereotyped Package.
Directory	Type in or browse for the directory into which the application schema file is to be generated.
GML Version	Click on the drop-down arrow and select the appropriate GML version for the GML Application Schema. Currently versions 3.3 and 3.2.1 are supported.
XML Encoding	Click on the drop-down arrow and select the XML encoding scheme to be used.
Ignore Element and Property Visibility	Click on this option to include all Classes, Attributes and Association End roles in the GML Application Schema, irrespective of their visibility (or Scope). Otherwise, only those with Public visibility will be considered for generation.
Stylesheet	(Optional) Select an XSL stylesheet to post-process the Schema before saving it to file. Import the required stylesheet into the project through the Resources window before selecting it in this field.
Application Schema(s)	Lists all the Application Schema Packages under the selected source Package. Click on a Package to display its details, as well as any of its child «Leaf» stereotyped Package details, in the Package Details panel. GML Application Schema will not be generated for any Package in the list that is unchecked.
Package Details	Against each of the child «Application Schema» and (optionally) «Leaf» stereotyped Packages (if any), enter a filename. To do this, double-click on the Package name in the list and, on the 'Package Details' dialog, review or complete these fields: • 'Stereotype' - The stereotype of the selected Package

	• 'Filename' - (Optional for a «Leaf» stereotyped Package) Click on the ... button and type the name of the .xsd file into which the GML Application Schema is to be generated; the filename must be in the same file path as selected in the 'Directory' field on the 'Generate GML Application Schema' dialog • 'Namespace' - (Not required for a «Leaf» stereotyped Package) Type in the namespace for the selected Package • 'Namespace Prefix' - (Not required for a «Leaf» stereotyped Package) Type the abbreviated value that represents the Namespace Note that: • The value of the 'Filename' field will be saved as the value of the tag xsdDocument on the Package • The value of the 'Namespace' field will be saved as the value of the tag targetNamespace on the Package • The value of the 'Namespace Prefix' field will be saved as the value of the tag xmlns on the Package
Generate	Click on this button to generate the GML Application Schema for each of the checked Packages in the 'Package Details' panel.
View Schema	Click on this button to display the generated GML Application Schema for the selected Package in the 'Package Details' panel.
Close	Click on this button to close the 'Generate GML Application Schema' dialog.
Help	Click on this button to display this Help topic.

Notes

- The MDG Technology for GML is available in the Professional, Corporate, Unified and Ultimate editions of Enterprise Architect
- A «CodeList» stereotyped Class with the tag 'asDictionary' set to True will be exported as a separate XML file to the directory specified in the 'Directory' field, with the filename CodeList_<classname>.xml (where <classname> is the name of the «CodeList» stereotyped Class)
- Using the configurable file GMLStereotypes.xml in the Sparx Systems > EA > Config > GML folder, you can specify aliases for the standard GML stereotypes; the GML Application Schema Generator will also consider these aliases during Schema generation
- Association Class is specific to GML 3.3; when generating GML 3.2.1 Application Schema, the Association and Class of an Association Class are treated as separate entities