ELF WP2 – Modelling guidelines

Author: Morten Borrebæk

Date: 20.11.2014

Version: 1.15

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

Change Summary

Version	Date	Author/Editor	Change Summary
1.0	09/07/2013	Morten Borrebæk	First version based upon ESDIN project
1.1	11/07/2013	Anja Hopfstock	Revised version based on telecon AHO and MBO
1.2	18/09/2013	Anja Hopfstock	Revised due to implementation of central SVN repository
1.3	01.10.2013	Morten Borrebæk	Revised due to ELF glossary and UML model structure, with input from Anja and Kent.
1.4	21.11.2013	Heidi Vanparys/Anja Hopfstock	Comments and remarks
1.5	27.11.2013	Morten Borrebæk / Kent Jonsrud	Revised on UML profile for ELF
1.6	27.11.2013		No change
1.7	27.11.2013		No change
1.8	19.12.2013	Morten Borrebæk	Update according to discussion at telecon (Anja, Kent, Marcel and Morten), and feedback from Heidi. The 'tagged values' always void and never void is moved to feature types in the table in clause2.
1.9	20.12.2013	Morten Borrebæk	Update on tagged vales from Marcel, and guidelines for documentation
1.10	12.06.2014	Anja Hopfstock	Revised version based on telecon with Clemens Portele, Heidi Vanparys, Helen Eriksson, Marcel Heimann (26.05.2014)
1.11			As 1.10?
1.12	14.06.2014	Morten Borrebæk	Revised version based upon comments in 1.10
1.13	02.09.2014	Morten Borrebæk	Figures updated after updated modelling approach. Some change in text due to the UML figures.
1.14	15.09.2014	Morten Borrebæk	Final revision by Anja, Heidi and Morten. Revised document structure.
1.15	20.11.2014	Anja Hopfstock	Editorial changes

References

Ref.	Title/Version/Publication Date/Author
[1]	ESDIN_WP7_ExM_modelling_guidelines_V1.2

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

Contents

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

LIST OF FIGURES

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

Scope

This document gives an introduction to modelling of application schemas for the ELF specifications. The application schemas shall build upon the INSPIRE specifications and should therefore adhere to the requirements and recommendations in the INSPIRE Generic Conceptual Model (GCM).

However, during the project period it was stated that the previous current modelling guidelines did not conform to the rules for extensions specified by INSPIRE, nor following standard practise, and lead to an extra overhead for service implementers.

The revised modelling guidelines in this document are formed in such a way that an existing INSPIRE implementation by default is conformant to an ELF specifications for the themes that are in the remit of the ELF data specifications. The impact of such a precondition is that all ELF additions have to be optional (not even voidable), and that there should be no constraints on the INSPIRE that affects the INSPIRE GML application schema.

1 Definition and abbreviations

1.1 Definitions

ELF data specification	specification according to EN-ISO 19131 Geographic information -- Data product specifications for **ELF data** according to the INSPIRE themes that the NMCA's support Note: The specifications provide a detailed description of spatial datasets provided by NMCAs as harmonized geospatial reference data for Europe at different levels of detail together with additional information.
	Note: In the introduction of ISO 19131 Data product specification it is stated that A data product specification may be created and used on different occasions, by different parties and for different reasons. It may, for example, be used for the original process of collecting data as well as for products derived from already existing data. It may be created by producers to specify their product or by users to state their requirements. The ELF project is not a project to capture new data, so our main focus is to specify products derived from existing data.
ELF product specification	specification according to ISO 19131 Geographic information -- Data product specifications for data combined of one or more ELF/INSPIRE themes. Note: The specifications provide a detailed description of a dataset or dataset series together with additional information that will enable it to be created, supplied to and used by another party. These specifications are based on the ELF data specifications.
Classifiers	generalization of a class that includes other class like elements, such as data types, actors and components. A UML class has a name, a set of

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

attributes, a set of operations and constraints. A class may participate in associations.

1.2 Abbreviations

UML	Unified modeling language
ELF	The European Location Framework
GCM	INSPIRE Generic Conceptual Model
INSPIRE	Infrastructure for Spatial Information in the European Community
NMCA	National Mapping and Cadastre Autority
GCM	INSPIRE Generic Conceptual model
OCL	Object Constraint Language
DIS	Draft International Standard

2 ELF modelling approach

2.1 Model structure

2.1.1 Introduction

The ELF model consists of a hierarchical structure of packages and application schemas. The <<applicationSchema>><theme> will show all imported feature types, data types, codelists and enumerations of interest for ELF. This would define the “NMCA profile for INSPIRE”.

The ELF models includes the INSPIRE consolidated UML model, with its foundation schemas (ISO/TC 211-, EarthResourceML- , GeoSciML- and OGC models), The INSPIRE Generic Conceptual model and the INSPIRE theme models).

To be able to use and extend the INSPIRE schemas the relevant INSPIRE themes are imported to the ELF model by using package control in Enterprise Architect. See ELF_UML_repositoryInstruction at Projectplace (https://service.projectplace.com/pp/pp.cgi/r936673615).

This is done by creating a new diagram <theme> containing the packages under discussion (drag them in) and draw a UML dependency from the ELF package to the relevant INSPIRE application schema.

2.1.2 ELF data specification packages

The ELF data specification package shall contain all ELF application schemas.

The <<applicationSchema>><theme> will show all imported feature types, data types, codelists, enumerations and associations of interest for ELF. This would define the “NMCA profile for INSPIRE”.

All packages containing application schemas shall have the <<applicationSchema>> stereotype from the ELF UML profile.

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

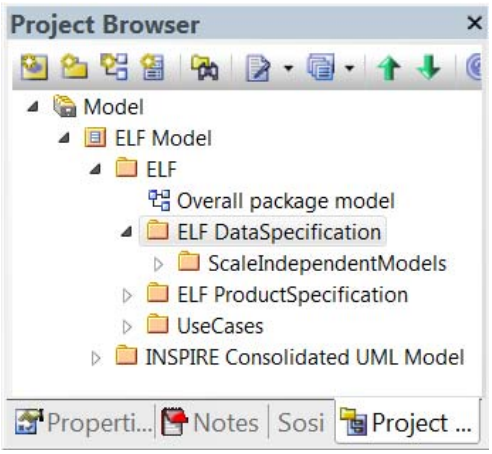
2.1.3 ELF Product Specification package

The ELF data product package will contain the application schemas for the envisaged ELF products and services, e.g. ELF Basemap, ELF International Boundaries, etc. There might be additional products and services as we go along in the project.

With this approach each ELF application schema specifies its feature types and related types in detail and can be used as a source for validating data.

2.1.4 ELF Use cases

The ELF use cases packages describes some use cases that verifies the additional ELF concepts.



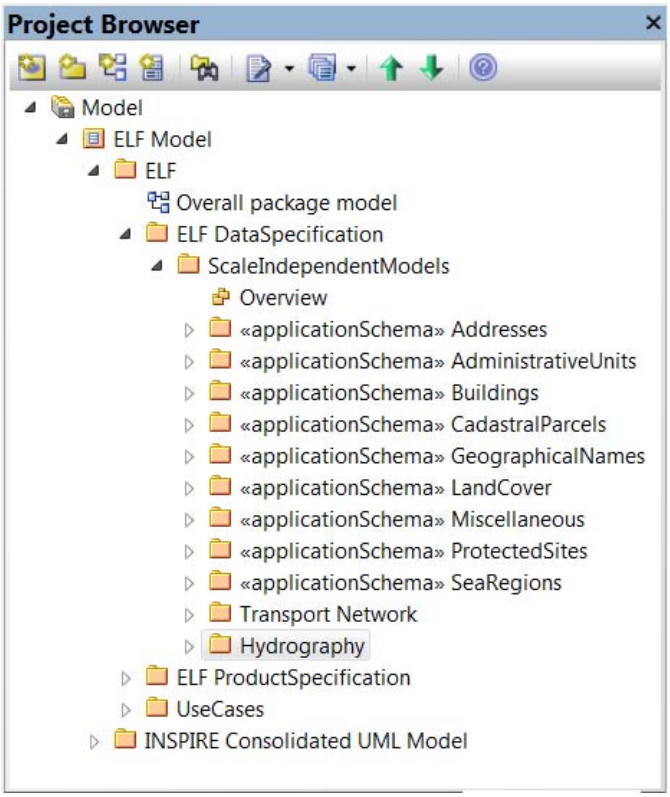


Figure 1 ELF UML structure (high level) **Figure 2 ELF UML structure with some application schema**

Figure 1 shows the high level model structure while Figure 2 goes down to the level of application schema.

2.2 UML modelling principles

2.2.1 Introduction

The ELF data specifications are developed as a set of profiles of the INSPIRE data specifications for different levels of detail comprising the parts of INSPIRE data specifications that are in the remit of the NMCAs and for which a pan-European and cross-border coverage is of interest.

The analysis of the ELF matching tables revealed the matching concepts between INSPIRE and NMCA data holdings. This results in the common part of ELF/INSPIRE.

Requirements for the ELF specifications may go beyond those for the INSPIRE schemas and it is expected that additional properties are required for ELF feature types. New feature types that are not subtypes of INSPIRE feature types are also allowed. Using the inheritance pattern it is easy to

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

extend the INSPIRE models by adding properties to the subtypes defined for the ELF application schema.

Additional content in ELF with respect to INSPIRE is coming from user requirements survey and/or from specifications of existing data with respect to data definitions and classification of features, data quality requirements, multilingual aspects, and data representation.

To achieve INSPIRE compliancy in data modelling the following principles should be observed

- Extensions shall not
  - Change the specification but normatively reference it with all its requirements
  - Set any additional requirements that break any requirement of the INSPIRE data specification
  - Add concepts that overlap with existing INSPIRE concepts.
  - Make a pure INSPIRE implementation non-conformant to the ELF specifications.
- Extensions may
  - Add new application schemas importing INSPIRE or other schemas as needed
  - Add new types and constraints in the new application schemas
  - extend INSPIRE code lists if not centrally managed

For each matching concept of INSPIRE and ELF identified from the analysis of the matching tables, a corresponding concept is created in ELF. Where possible, these concepts should be sub-classes of existing INSPIRE concepts (feature or data type, code lists etc.) by:

- Define additional optional attributes that are present in existing data or required by users but missing in INSPIRE
- Add constraints to ensure that ELF meets the user requirements where applicable
- Define new ELF feature types for concepts that are present in existing data or required by users but missing in INSPIRE
- For code lists in INSPIRE, identify matching codes and define additional codes where missing – reuse as many values from INSPIRE as possible and define new values only if no existing value can be matched. Describe any additional constraints (e.g. sometimes a code list value cannot be mapped or a code list value depends based on the value of another property).
- Add optional associations where required
- Avoid the stereotype <voidable> for new attributes and associations, to ensure that a 'pure' INSPIRE implementation conforms to ELF.

Further:

- Non-voidable properties from INSPIRE cannot be made voidable in the ELF specifications.
- See also GCM UML modelling conventions in INSPIRE, clause 9.4.

Each ELF application schema contains feature types, data types and enumerated types defined for the relevant theme / specification. Feature types for the ELF specifications shall be based on the feature types defined in INSPIRE (if possible). This is done by creating the ELF feature types as

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

subtypes of the corresponding INSPIRE feature types via a generalization relationship in the application schemas. The ELF feature types will then inherit the properties (attributes, associations and constraints) from the INSPIRE feature types. These properties from INSPIRE can be restricted for the ELF feature types by adding OCL constraints. The ELF feature types can extend the INSPIRE feature types by adding optional properties.

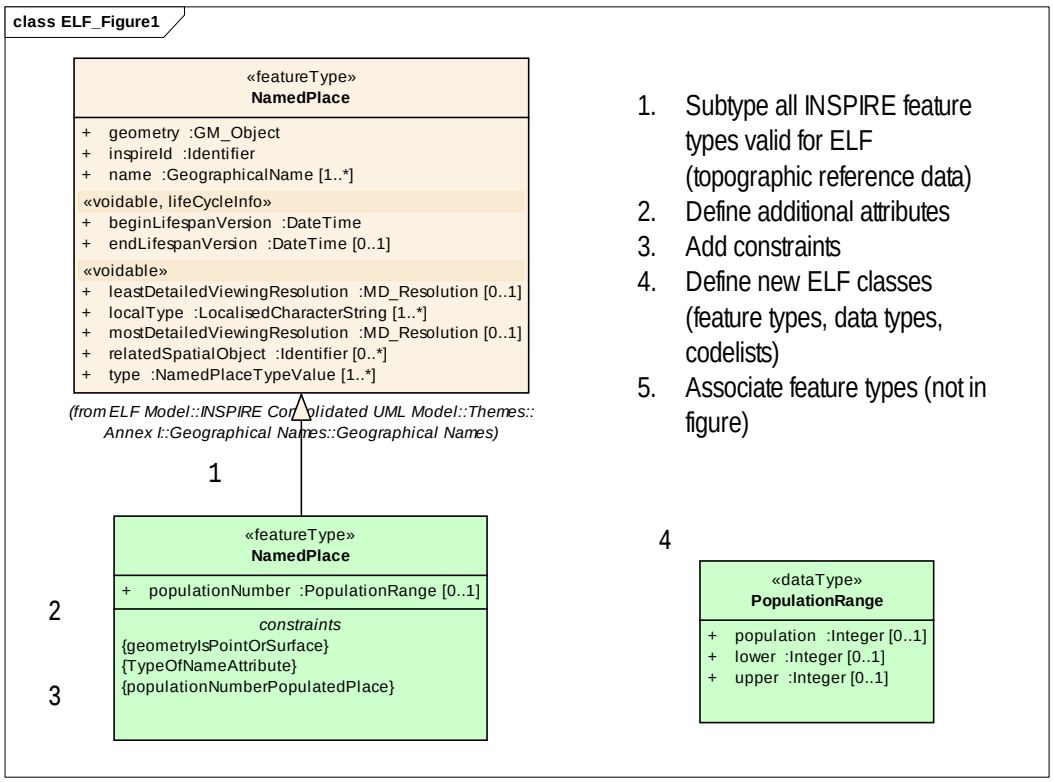


Figure 3 ELF UML modelling principles

Figure 3 ELF UML modeling principles explains the methods for extending the INSPIRE models

2.2.2 Subtyping of UML classifiers in INSPIRE (feature types, data types and codelists)

The common part of ELF/INSPIRE (1) is realised by sub-typing, a simple inheritance/ specialisation of INSPIRE feature types.

Create a class diagram for the ELF application schema. The relevant classes from INSPIRE schemas can be dragged into this class diagram for the ELF models. Note that dragging types into a class diagram does not add it to the application schema, only to a view of the model. Create the new ELF feature type in the ELF package and make it a subtype of the corresponding INSPIRE feature type.

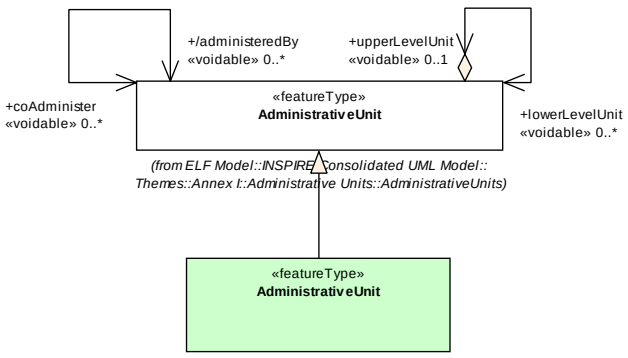


Figure 4 – Subtyping INSPIRE feature types (overview)

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

Figure 4 shows an example where a new feature type AdminsitrativeUnit has been added to the ELF application schema and made a subtype of AdministrativeUnit from INSPIRE. A model can be looked at from different views and this view describes the relationship between the ELF feature type and the INSPIRE feature type. Adding another class diagram we can focus on the specific ELF feature types themselves by adding only the ELF feature types and showing all defined properties, inherited properties is shown in the INSPIRE class.

2.2.3 Define additional attributes

Additional attributes are added to the ELF feature types where appropriate.

Names of UML elements should combine multiple words as needed to form precise and understandable names without using any intervening characters (such as "_", "-"or space).

Capitalize only the first letter of each word after the first word that is combined in a name. Capitalize the first letter of the first word for each name of a class, package, type specification and association names.

Figure

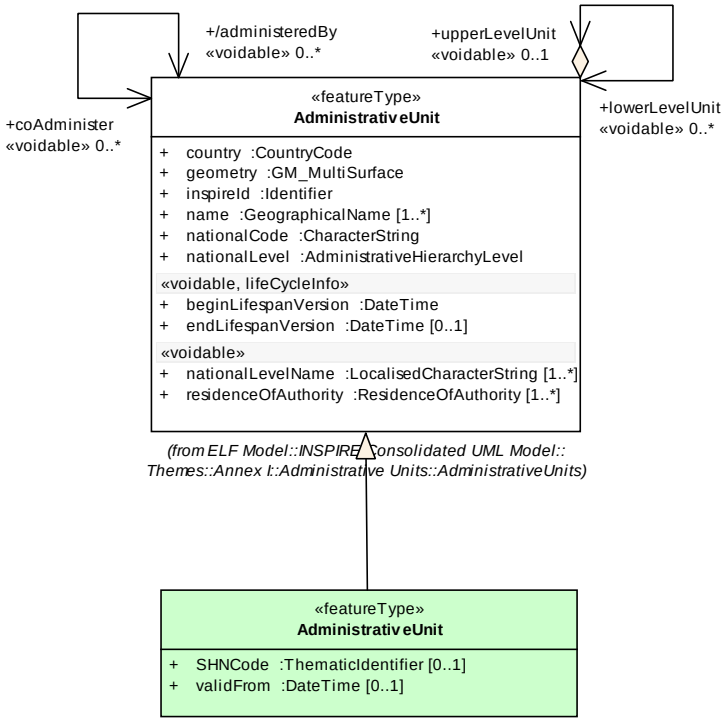


Figure 5 – Adding new attribute to ELF feature type AdministrativeUnit

Figure 5 shows two additional attributes (SHNCode og validFrom) for the ELF feature type AdministrativeUnit.

2.2.4 Add constraints

Restrictions of INSPIRE are expressed through constraints in the UML data model.

Constraint can be added to all ELF classifiers(feature types, data types, codelists, enumeration and association roles (association ends) thar are not inherited from INSPIRE.

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

This is achieved by adding OCL constraints to the feature types. For instance, for NamedPlace the inherited Geometry is GM_Object. This is constrained to point or surface by the following OCL expression (See also Figure 6):

inv: self.geometry.oclIstypeOff(GM_Point) or self.geometry.oclIsTypeOf(GM_Surface).

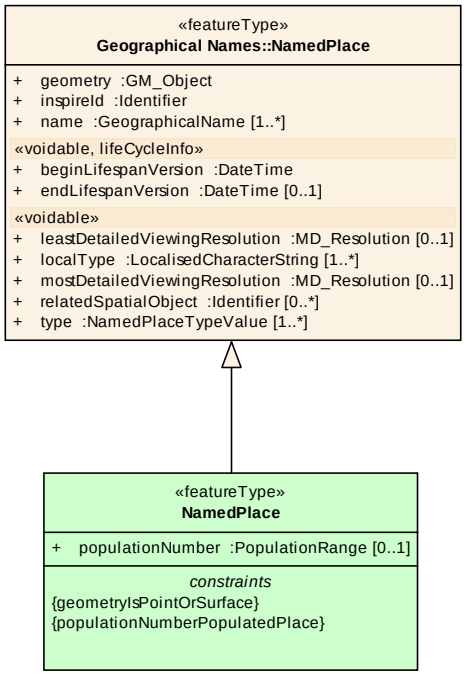


Figure 6 – Adding constraints to feature types

Figure 6 describes two new constraints, the first on geometry, the second related to the new attribute on the ELF feature type.

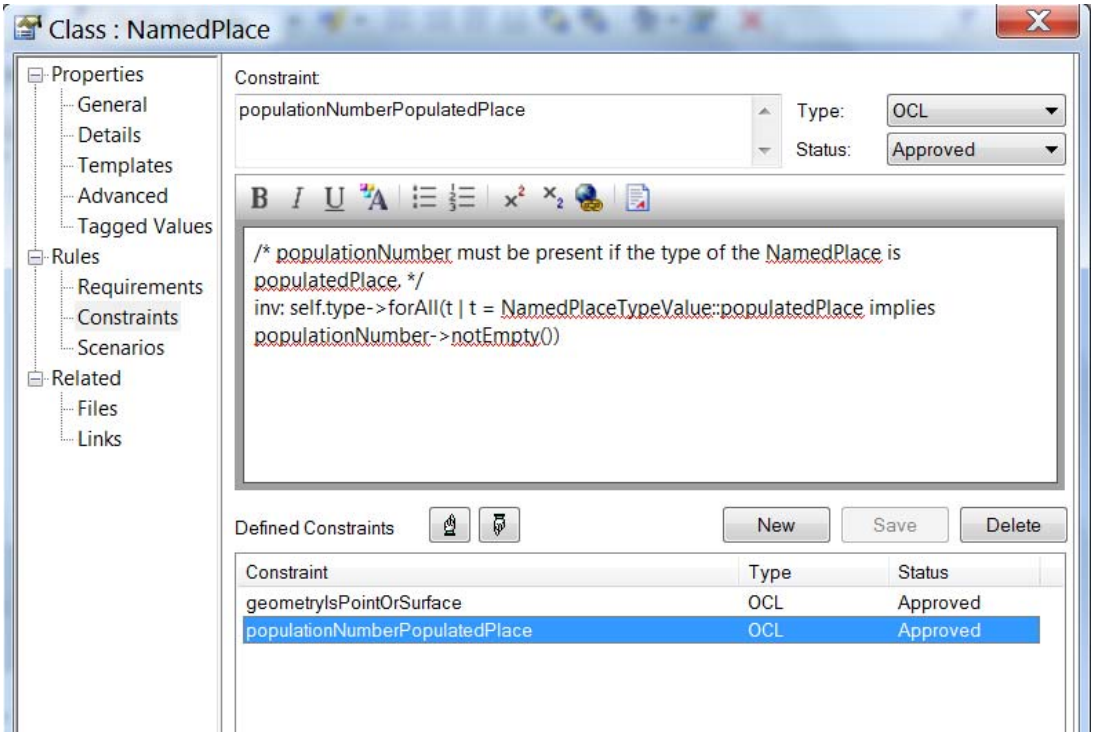


Figure 7 – Adding constraints to feature types

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

Figure 7 shows two new ELF-constraints added. The constraint name should be self explaining. The second constraint is shown with its addition descrioption and OCL statement.

Like the name of feature types and attributes, the name of the contraint should be self explanatory.

One of the rules mentioned earlier is that it is not legal to constraint the INSPIRE classes. However, there are some exception from this rule. For example, where INSPIRE models geometry with GM_Object, we can constrain this to a geometry element that is more suitable for a certain scale. However, these kind of constraints do not have any impact on the GML applications schema, but will be documented in the feature catalogue.

It is also possible to specify constraints on a profile level. For this, we could include the profile name in the constraint name, e.g. using [profile-name ":"] constraint-name. If the profile-name is missing, the constraint applies to all profiles applicable to the feature type. No name may include a colon. Some hypothetical examples:

class NamedPlace
MasterLoD1:geometryIsPoint
MasterLoD2:geometryIsPoint
RegionalGlobal:geometryIsPointOrSurface

2.2.5 Additional tagged values for ELF

ELF applies two predefined 'tagged values' in the ELF applications schema, shown as 'tags' in the UML models.

suppress — Subtypes that are only created in the ELF application schemas to include an INSPIRE feature type in an ELF data specification and/or to attach a constraint will be suppressed in the GML application schemas as these modelling constructs are not relevant for the XML encoding). This tagged values can be assigned to feature types.

To signal this behaviour to the process deriving the GML application schemas for ELF, a tagged value "suppress" with the value "true" is set for these classes. In the ShapeChange configuration for derivation of the XML Schemas, the encoding rules needs to include the conversion rule "rule-xsd-cls-suppress". See http://shapechange.net/targets/xsd/extensions/#rule-xsd-cls-suppress.

profiles — Associates a model element specified in an ELF application schema package to an ELF specification. This tagged values can be assigned to classes, attributes (including enumerated values) and association roles.

For example:

class AdministrativeBoundary: profiles=MasterLoD1,MasterLoD2
class AdministrativeUnit: profiles=MasterLoD1,MasterLoD2,Regional,Global
attribute SHNcode: profiles=MasterLoD1,MasterLoD2
role adminUnitArea: profiles=MasterLoD1,MasterLoD2,Regional,Global

It is also possible to support different versions of the profiles.

Example: class NamedPlace: profiles=MasterLoD1[2.0-],MasterLoD2[2.0-],RegionalGlobal[3.0-]

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

See http://shapechange.net/transformations/profiler/ for details. . http://shapechange.net/transformations/ has general details about the transformation concept in ShapeChange.

2.2.6 Define new ELF classes

Extensions to INSPIRE are realised by adding specific features types, attributes, codelists and/or associations. An example is shown in **Fehler! Verweisquelle konnte nicht gefunden werden.**.

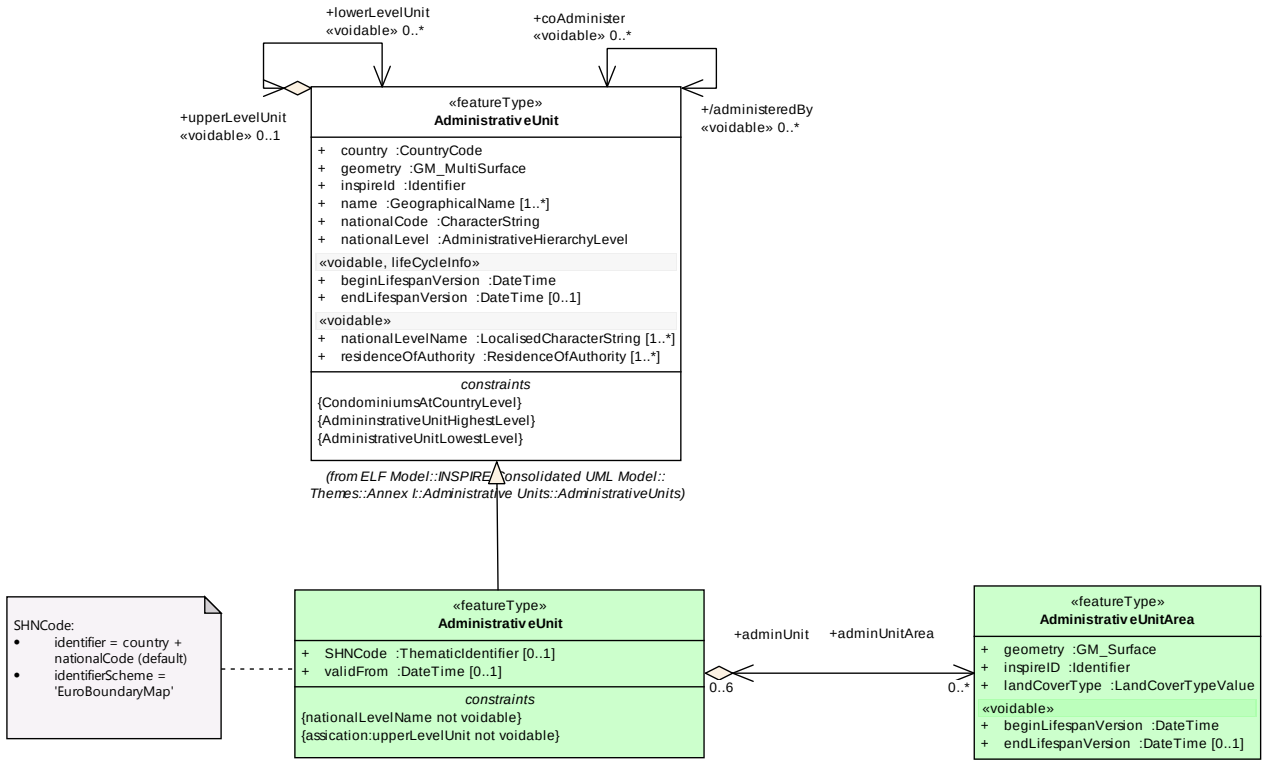


Figure 8 – Adding a new feature type associated to 'AdminsitrativeUnit.

Figure 8 shows an example of adding a new feature type (AdministrativeUnitArea) associated to the ELF feature type AdministrativeUnit. The association to the new feature type is also optional, and shall not be stereotyped <voidable>. See the general rules explained in clause 2.2.1 Introduction to the modelling principles.

However, the stereotype <voidable> can still be used on attributes on new feature types, as long as the association to them is optional.

2.3 Naming conventions

The usual naming convention for ELF feature type is to use the name of the INSPIRE feature type. Similar naming convention is also applied for other UML classes, like datatypes, codelists and enumerations.However, this implies that the definition of the ELF feature type must not redefine the definition of the INSPIRE feature type. If this is the case, it is not really the same feature type, and a new feature type should be added to the ELF specification, without subtyping any INSPIRE feature type.

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

3 Documentation

3.1 Definitions

All additional ELF classifiers shall contain definitions sufficient for understanding of all classes, attributes, associations, operations and appropriate data type definitions.

UML elements should use documentation fields (in Enterprise Architect) to further explain the meanings (semantics) of named elements.

Definitions from INSPIRE feature types and properties cannot be changed, but if additional information is required or appropriate it may be added to the existing definition (e.g. as a note or additional description for a profile) in the ELF application schema.

Redefinitions of properties should be avoided. Use OCL constraints instead to achieve the same result. If used, redefinitions cannot break compliance with the INSPIRE model.

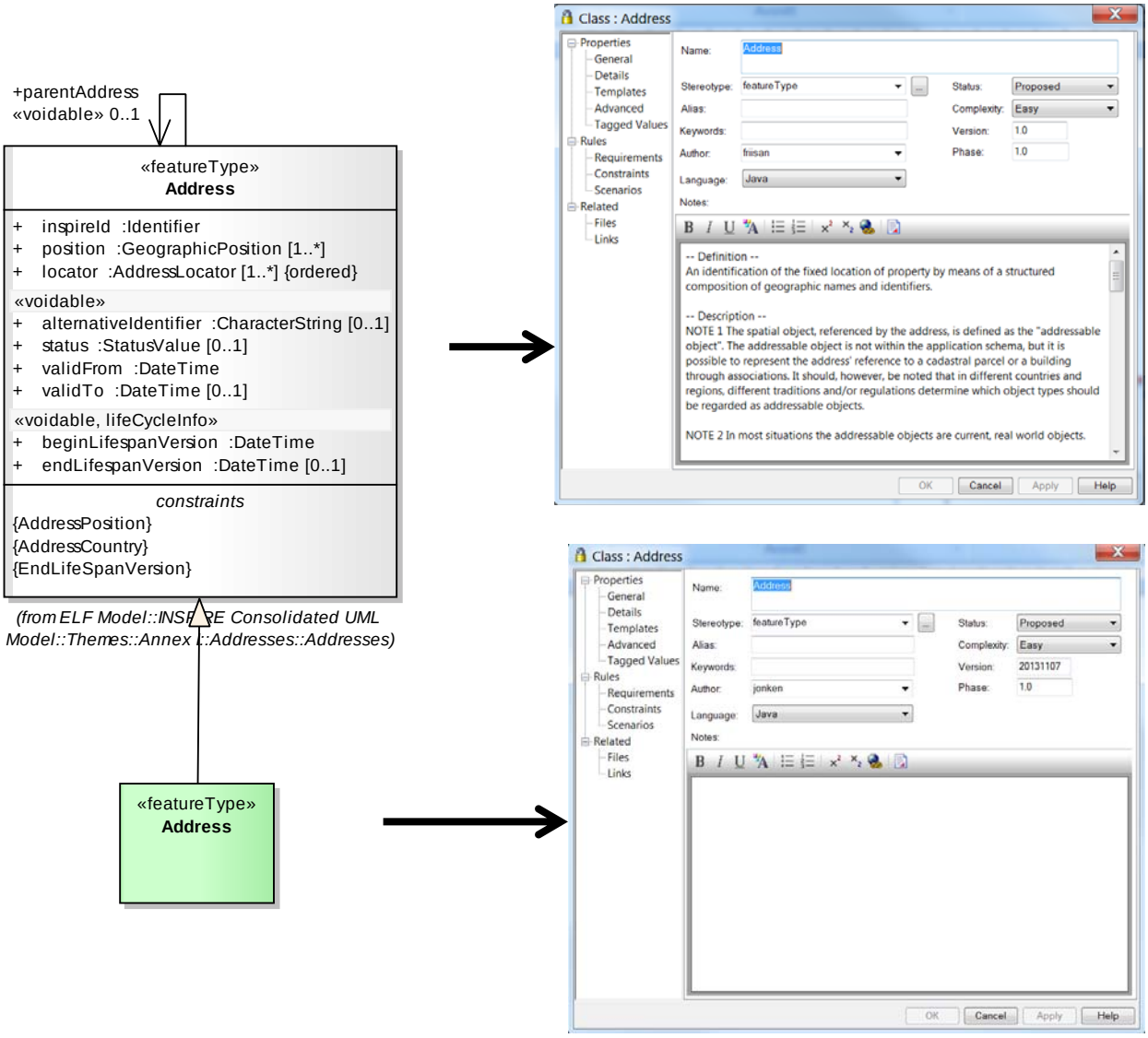


Figure 9 - Subtyping INSPIRE feature types

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

Figure 9 - Subtyping INSPIRE feature types shows the definition of the feature type Address from INSPIRE and the missing definition in the ELF address, meaning that the definition from INSPIRE will be applied for ELF.

3.2 ELF Diagrams

3.2.1 Application schemas diagram

Diagram that shows all application schemes that are part of ELF. Use'boundary' to assign these to themes.

3.2.2 Package depencency diagram

All package dependencies shall be shown in one of more package diagrams for each application schema.

Fully qualified namespaces should be stated in the figures.

Naming convention - Package dependencies <Name of applicaton schema>

Figure 10 shows an example of a package dependency diagram for AdministrativeUnitsFigure 10 – Package dependency for AdministrativeUnits

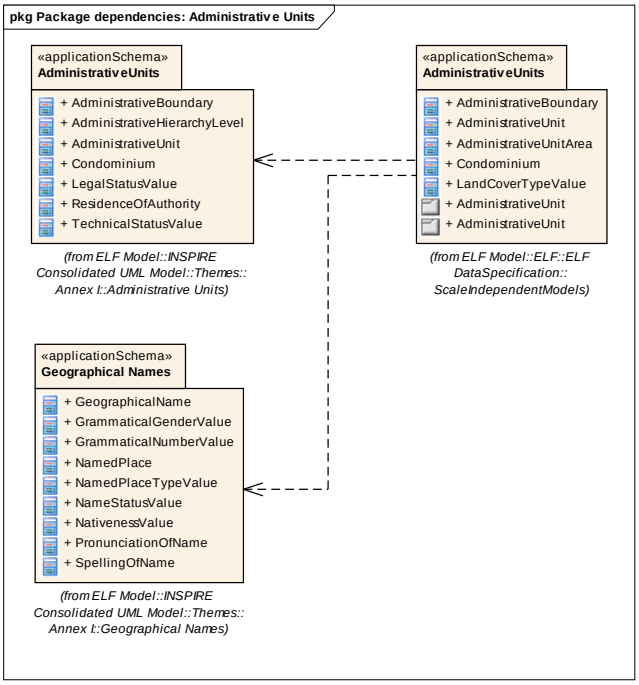


Figure 10 – Package dependency for AdministrativeUnits

3.2.3 Overview diagram

For each application schema there should be a class diagram for the entire package, including all feature types (and where they are inherited from) and main associations without role names (operations and attributes shall not be shown). The definition of main associations is the responsibility of the domain expert.

Fully qualified namespaces should be stated in the figures.

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

Naming convention - Overview <Name of applicaton schema>

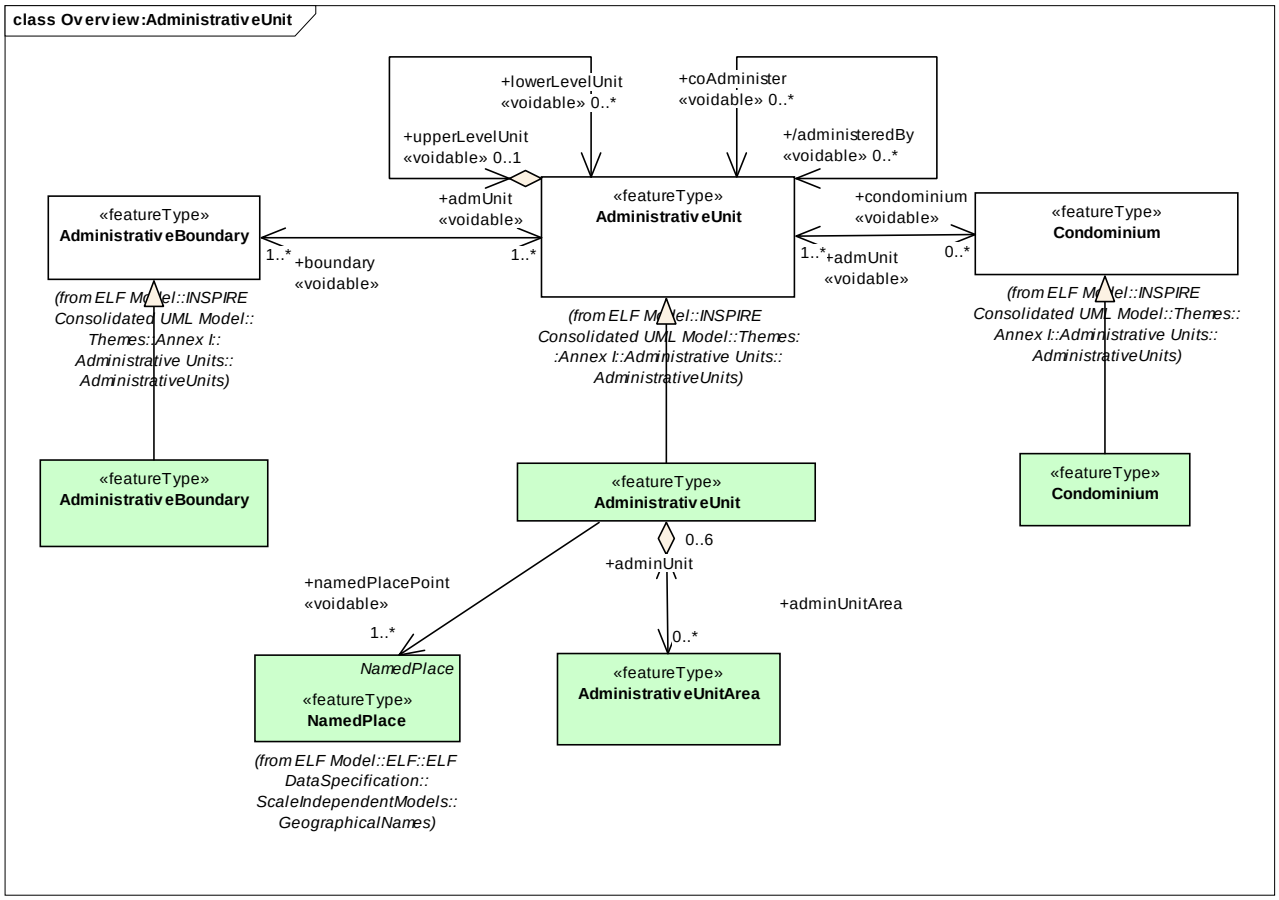


Figure 11 – ELF overview diagram for AdministrativeUnits

Figure 11 is an example of an overview diagram, showing the feature types and the associations between feature types for a specific theme here AdministrativeUnit in ELF.

3.2.4 Context diagram

All classifiers (feature types, data types, codelists, enumerations, associations) shall be documented in a "context diagram" where all attributes, operations and all relationships that are navigable from the central classifier are displayed ISO 19103 (DIS) . Supertypes of the ELF feature types shall also be part of the diagram(feature types, data types, codelists). Associated feature types and datatypes do not need to show the Supertype.

Note that closely related classifiers can share a context diagram as long as the combined diagram is neither overly cluttered nor difficult to read.

Fully qualified namespaces should be stated in the figures, but from a pragmatic point of view they could be omitted in some figures (not to clutter the figures).

Naming convention - Context diagram <name>

Note: Qualifiers defined in foundation schemas of the ELF models are not shown.

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

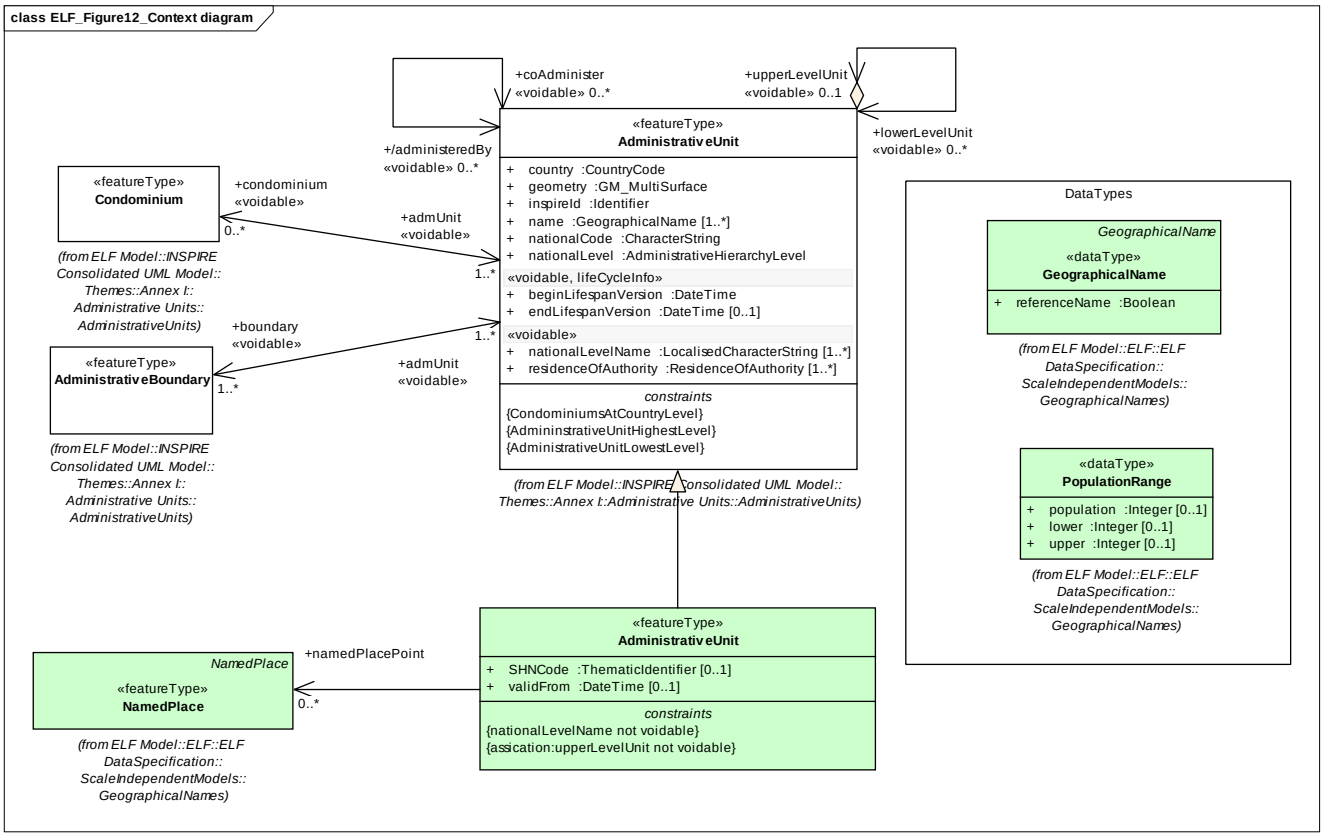


Figure 12 – ELF context diagram for AdministrativeUnits

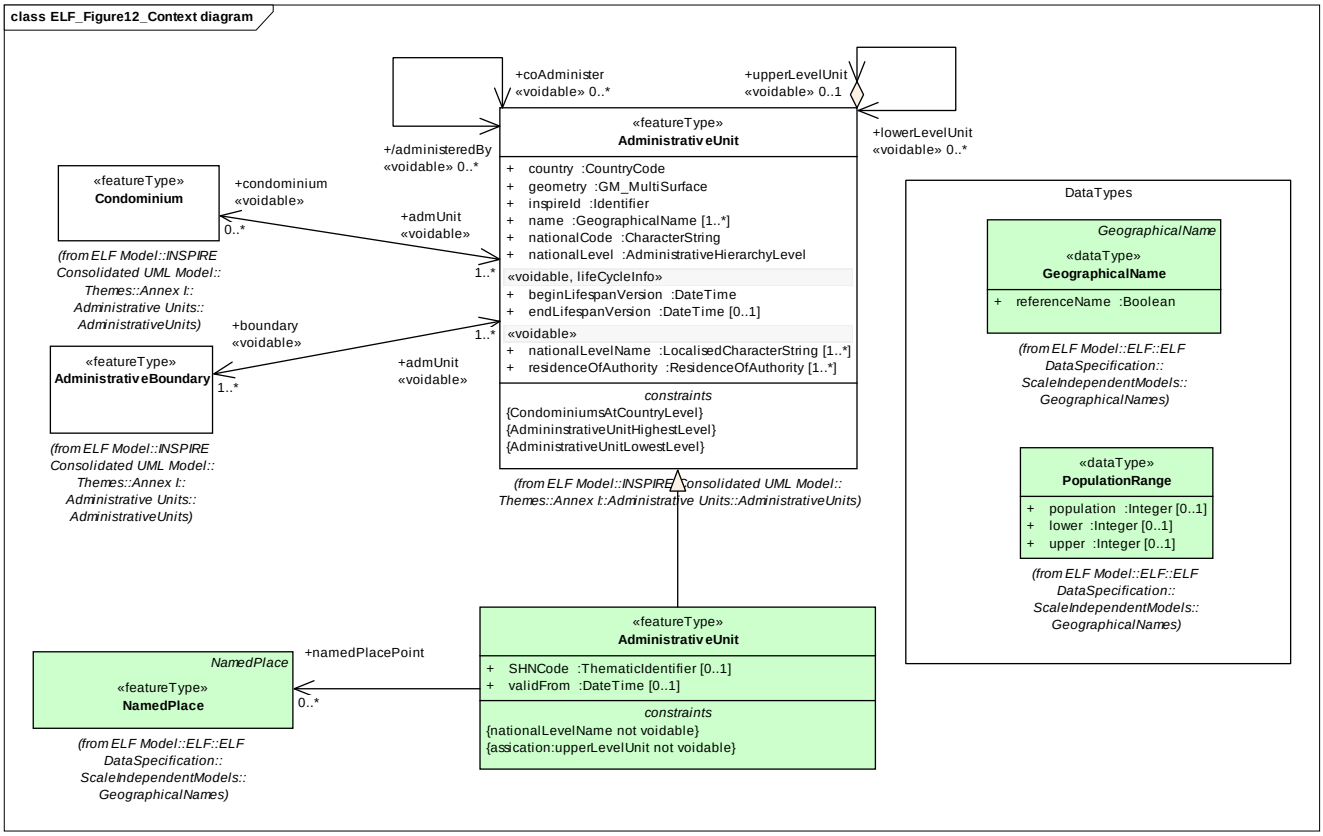


Figure 12 shows a context diagram for the classifier AdministrativeUnit. The diagram contains all attributes, associated classes, datatypes, codelists and unspecified qualifiers, both in ELF and INSPIRE.

In addition, all constraints are shown.

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

Note: Data types, codelists and enumerated types defined in the INSPIRE Consolidated UML models (including the INSPIRE generic conceptual model and the foundation schemas are not shown in these context diagrams, but will be documented in the ELF feature catalogue.)

3.3 Feature catalogue

The description of the ELF data specifications will be documented in a feature catalogue, conformant to EN-ISO 19110 Methodology for feature cataloguing.

The documentation tool in ShapeChange will be applied for this documentation.

Be aware that definitions / descriptions for classes are not inherited by subtyping INSPIRE. Up to now, this has to be done mauually by copying the text from INSPIRE, potentially narrowing it if this is required to ELF.

The new approach is that documentation on subtyped feature types may be left empty, if it is unchanged. In this case, ShapeChange will derive the documentation from the supertype. (Note: So far, ShapeChange does not support this, it is only supported that documentation may be "imported" from a class with a dependency relationship, but we would add support for supertypes with a new conversion rule "rule-all-all-documentation-from-supertype").

If the text needs to be narrowed, this must still be done manually.

Due to automatic generation of a feature catalogue / feature concept dictionary, the property of the application schemas should have sufficient textual description.

4 ELF UML Profile

4.1 Description

The INSPIRE GCM specifies a profile of UML to be used in INSPIRE context. For example, a stereotype <<applicationSchema>> is specified in the profile as a stereotype applicable to packages denoting that it is an application schema, similar for <<featureType>>. Such a profile can be added to EA giving direct access to these stereotypes when modelling. The ELF UML profile adds some additional tagged values to the INSPIRE UML profile.

The full ELF UML profile consists of the following elements:

Stereotype	Tagged Value	Description	Remark	Value and examples
applicationSchema	targetNamespace	Target XML namespace of the application schema [ISO 19136]	Mandatory	http://www.locationframework.eu/schemas/<theme>/<version> Example: http://www.locationframework.eu/schemas/LandCover/ 1.0

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

Stereotype	Tagged Value	Description	Remark	Value and examples
				This structure allows one application schema for each theme. Also, all schemas have their own absolute and nonrecurring path which excludes redundancy.
	xmlns	Namespace prefix to be used as short form of the target namespace [ISO 19136]	Mandatory	Add "elf-" to the INSPIRE namespace. Example on theme TN Roads: elf-tn-ro
	version	Current version of the application schema [ISO 19136] It is the same number which is used in the targetNamespace.	Mandatory, but not critical	<version number> Example: 0.1 Within the same version, there may be separate "builds". This is visible in the model and inside the schema, as x.y.build)
	gmlProfileSchema	URL of the schema location of a GML profile (where applicable) [ISO 19136]	Optional	can be left empty
	xsdDocument	Name of an XML Schema document to create representing the content of this package [ISO 19136]	Mandatory	<theme>.xsd (Version will be visible in the namespace inside the schema document.)
	xsdEncodingRule	XML Schema encoding rule to apply	Mandatory	iso19136_2007_ELF_Extensions
featureType				
	xsdEncodingRule	XML Schema encoding rule to apply (iso19136_2007, iso19136_2007_INSPIRE_Extensions or iso19139_2007) [D2.7]	Mandatory	iso19136_2007_ELF_Extensions
	noPropertyType	Suppress creation of a standard property type that supports inline or by-reference encoding (applies to ISO 19136:2007 encoding rule). Always set to false in INSPIRE. [ISO 19136]	Mandatory	false
	byValuePropertyType	Create a property type that requires that the instance is encoded inline (applies to ISO 19136:2007 encoding rule). Always set to false in INSPIRE. [ISO 19136]	Mandatory	false

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

Stereotype	Tagged Value	Description	Remark	Value and examples
	isCollection	Identifies the feature type as a feature collection. [ISO 19136]	Optional	if set should be “false”
	gmlMixin			false
	suppress	When true, the feature type is suppressed in the ELF GML application schemas. This is used on the feature types that (1) are included in the ELF data specification, (2) inherit from an INSPIRE feature type and (3) do not define properties in addition to the ones already defined in their INSPIRE superclass.		values="true | false" default="false"
	profiles	Comma-separated list of profile indicators which associates this model element to one or more levels of detail.		MasterLoD0 MasterLoD1 MasterLoD2 Regional Global If empty, it applies to all LoD's [It is also possible to add version, see class 2.2.5]
dataType	xsdEncodingRule	XML Schema encoding rule to apply	Mandatory	iso19136_2007_ELF_Extensions
	noPropertyType	Suppress creation of a standard property type that supports inline or by-reference encoding (applies to ISO 19136:2007 encoding rule). Always set to false in INSPIRE. [ISO 19136]	Mandatory	false
	isCollection	Identifies the feature type as a feature collection. [ISO 19136]	Optional	if set should be “false”
	profiles	Comma-separated list of profile indicators which associates this model element to one or more levels of detail.		MasterLoD0 MasterLoD1 MasterLoD2 Regional Global
codeList	xsdEncodingRule	XML Schema encoding rule to apply	Mandatory	iso19136_2007_ELF_Extensions
	asDictionary	GML dictionary	Mandatory	Default: true
	codeList	Reference to external codelist.	Optional	If set should be the URI of the external code list dictionary]. May in addition use GML-specific TaggedValue defaultCodeSpace

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

Stereotype	Tagged Value	Description	Remark	Value and examples
	defaultCodeSpace		Optional	Alternative to codeList
	profiles	Comma-separated list of profile indicators which associates this model element to one or more levels of detail.		MasterLoD0 MasterLoD1 MasterLoD2 Regional Global
association.end	profiles	Comma-separated list of profile indicators which associates this model element to one or more levels of detail.		MasterLoD0 MasterLoD1 MasterLoD2 Regional Global

4.2 Implementation

To add the profile to your EA file go to menu Project ->Resources. In the Resource window go down to 'UML profiles' and right click. Choose 'Import profile' and point to your local version, available for download from:

https://service.projectplace.com/pp/pp.cgi/r1025631019

All stereotypes required for ELF shall now be available when modelling.

4.2.1 Import UML profile

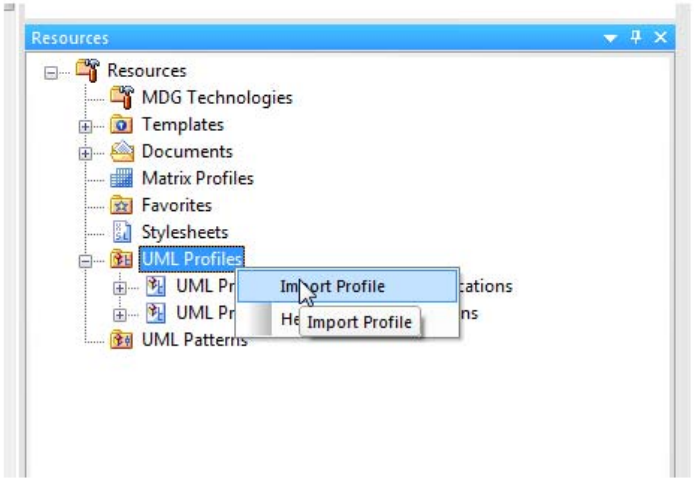


Figure 13 –Import UML profile

Figure 13 shows the Engterprise Architect function for importing profiles

NB. If the Resources toolbar is not available under the project menu, go to 'Tools' and choose "Customize".

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

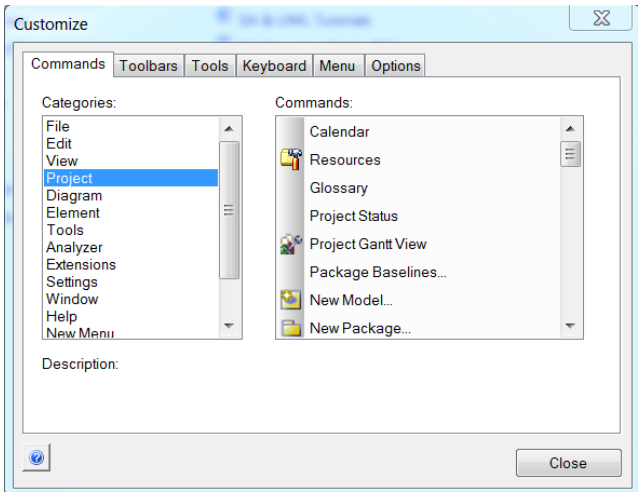


Figure 14 –Resource toolbar

Figure 14 shows how to get access to the resource toolbar

From her you drag the function to the toolbar.

4.2.2 Open UML Profile

This may look slightly different in your EA depending on the configuration. Click on “More Tools…” in the Toolbox pane and select ‘UML Profile for ELF data specifications’.

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

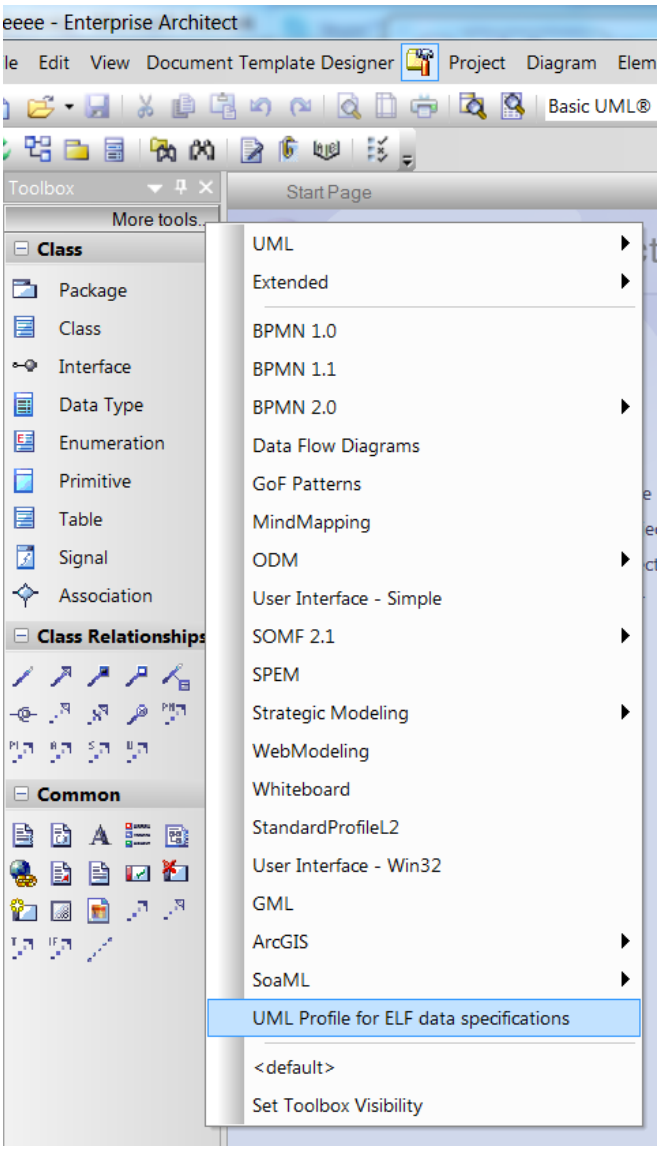


Figure 15 – Open UML profile

Figure 15 shows how to get access to the resource toolbar

If you want the Toolbox to be always visible, then go to “Set to Toolbox Visibility (see also figure above) and tick ‘UML Profile for ELF data specifications’.

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

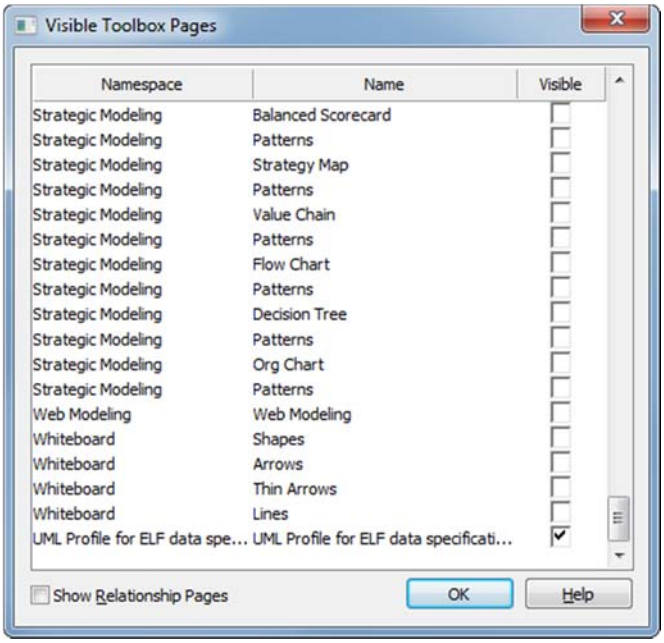


Figure 16 –Setting the ELF toolbox to be always visible

Figure 16 shows how to set the ELF toolbox always visible

4.2.3 UML Profile Toolbox

The toolbox is now available for use.

To create a new applicationSchema, create a UML package diagram if not yet present and drag-and-drop the applicationSchema icon from the toolbox to the diagram. This will create a new package with stereotype <<applicationSchema>> and the tagged values defined in the UML profile.

To create a new featureType, dataType, etc., create a UML class diagram if not yet present and drag-and-drop the appropriate icon from the toolbox to the diagram.

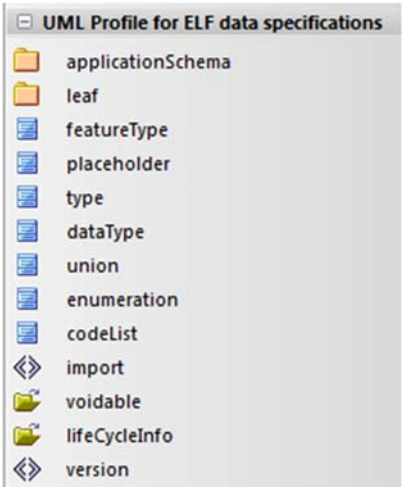


Figure 17 –ELF UML profile for ELF data specifications

Figure 17 shows the ELF UML profile for ELF data specifications

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

If you created e.g. a new featureType by first creating a regular UML class and then giving it stereotype <<featureType>> via the Properties dialog of the class, it will not yet have the tagged values of the UML Profile yet.

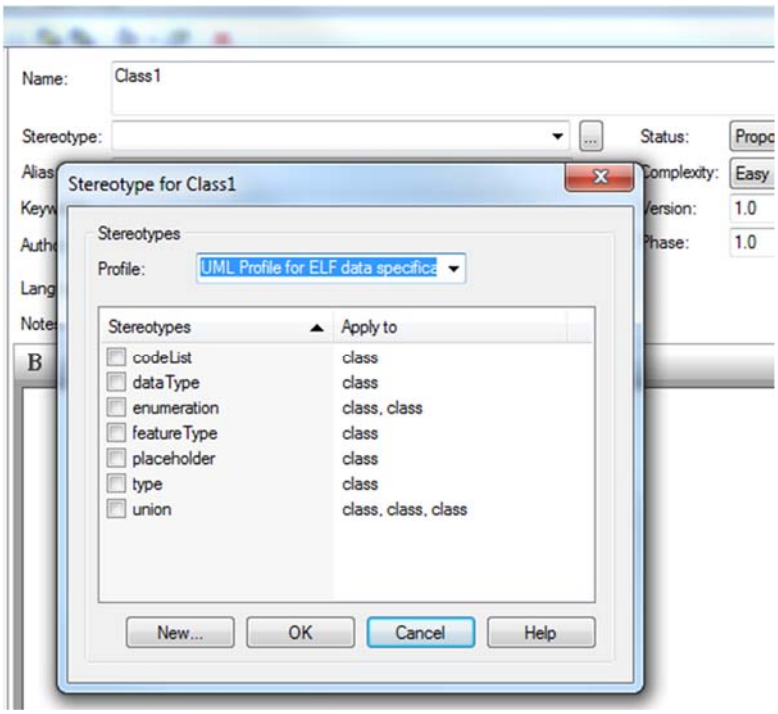


Figure 18 –Adding tagged values

Figure 18 shows how to add tagged values

To give this class the correct tagged values, right-click on the stereotype in the toolbox and choose 'Synchronize stereotype'.

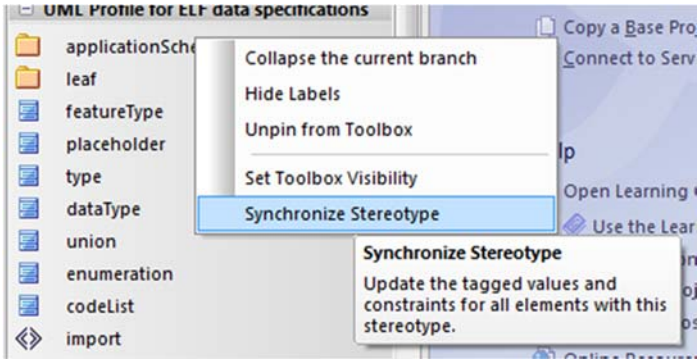


Figure 19 –synchronizing stereotypes

Figure 19 shows synchronization of stereotypes

NOTE: Enterprise Architect gives the impression that the tagged values on all classes in the whole model that are of the stereotype you changed are updated, but it is important to remember that you can only make changes in packages that you have checked out. So in reality, only the tagged values on the model elements you have checked out are changed. You can verify this by checking in your changes, and then issuing the 'Get All Latest command' using the 'Always import' option (beware that this process can take a long time).

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

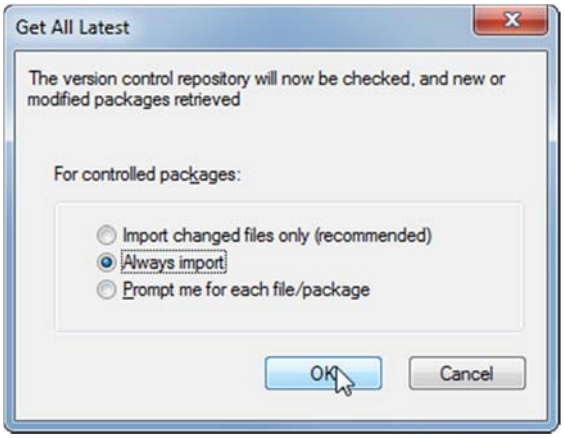


Figure 20 –Get all latest

Figure 20 shows how to update the local model from the repository using the function get all latest.

4.3 Colour scheme

Part of the the ELF model is also a predefined colour scheme, applying different colours for easier human interpretation. The colours applied have no impact on the models as such, the GML application schema will not be affected by colours. The colours are only applied for human beings to easier understand what is already described in the model / data specification.

Overview/Feature Types & data types	Apply color coding system for Feature Types and data types
	- INSPIRE white
	- ELF light green
codelists	Apply color coding system for codelists
	- INSPIRE codelists light yellow
	- ELF additional codelists yellow

5 Step by step approach

From a practical point of view, the modelling and the documentation is handled in parallel.

This clause describes a step by step approach to define an ELF feature type based on an INSPIRE feature type:

- Start with base model or model including INSPIRE model
- Create separate package for ELF models (if missing)
- Add subpackages for ELF (if missing)
- Import relevant INSPIRE theme by adding dependency from ELF package to INSPIRE package (package dependency diagram):
    - Add new diagram
    - Drag packages into diagram
    - Create dependency
- Drag INSPIRE feature types from INSPIRE application schemas into new diagram

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

- Create new feature types required for ELF specification
- Make the new feature types subtypes of the corresponding INSPIRE feature types.
- Use OCL constraints to constrain properties for the ELF feature types that are inherited from the INSPIRE feature types
- Add additional attributes required for ELF specification to the ELF feature types
- "Model" the ELF application schema based on requirements.
    - ELF additional attributes are optional and have a multiplicity of 0..1 or 0..*
- For all subtypes, attributes and association roles fill the tagged values as given in claus 2.2.5 above
- Create overview and context diagram for ELF application schemas (if missing)
- Check and amend the documentation (definition, description, etc.) for additional ELF model elements

6 Bibliography

D2.5 GCM, INSPIRE Generic Conceptual Model version 3.4 april 2014

D2.7 Encoding

User guide for Enterprise Architect (part of Enterprise Architect)

ISO 19103 Conceptual Schema Language (draft)

ISO 19110 Feature Catalogue Methodology.

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)

Annex A – ELF configuration file

ELF Configuration file for generating GML schema and feature catalogue from ShapeChange is provided upon request (accessible in the ELF SVN repository).

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)