TSMAD27-4.3.1E

S-100 – Part 5

Feature Catalogue

Contents

5-1 Scope 1.1
5-2 Conformance 2.1
5-3 Normative References 3.1
5-4 Principal Requirements 4.2
5-4.1 Feature Catalogue 4.1.2
5-4.2 Information Elements 4.2.2
5-4.2.1 Introduction 4.2.1.2
5-4.2.2 Named Types 4.2.2.2
5-4.2.3 Properties 4.2.3.3
5-4.2.4 Feature Associations 4.2.4.4
5-4.2.5 Bindings 4.2.5.4
5-4.2.6 Definitions and source references 4.2.6.4
5-4.2.7 Completeness 4.2.7.4
A-1 Feature Catalogue Model 4.2.7.6
~~5-1 Scope ..... 1~~
~~5-2 Conformance ..... 1~~
~~5-3 Normative References ..... 1~~
~~5-4 Principal Requirements ..... 2~~
~~5-4.1 Feature Catalogue ..... 2~~
~~5-4.2 Information Elements ..... 2~~
~~5-4.2.1 Introduction ..... 2~~
~~5-4.2.2 Named Types ..... 2~~
~~5-4.2.3 Properties ..... 3~~
~~5-4.2.4 Feature Associations ..... 4~~
~~5-4.2.5 Bindings ..... 4~~
~~5-4.2.6 Definitions and source references ..... 4~~
~~5-4.2.7 Completeness ..... 4~~
~~A-1 Feature Catalogue Model ..... 6~~

5-1 Scope

This Part provides a standard framework for organizing and reporting the classification of real world phenomena in a set of geographic data. It defines the methodology for classification of the feature types and specifies how they are organized in a feature catalogue and presented to the users of a set of geographic data. This methodology is applicable to creating catalogues of feature types in previously uncatalogued domains and to revising existing feature catalogues to comply with standard practice. It applies to the cataloguing of feature types that are represented in digital form. Its principles can be extended to the cataloguing of other forms of geographic data.

A feature catalogue shall be defined for each product specification.

This Part is applicable to the definition of geographic features at the type level but not applicable to the representation of individual instances of each type.

5-2 Conformance

This profile conforms to conformance class 2 of ISO 19106. The following is a brief description of the specializations and generalizations where the profile differs from ISO 19110.

1) New abstract classes, *S100_FC_Item*, *S100_FC_NamedType,* and *s100_FC_ObjectType* are introduced.
2) A new class, S100_FC_InformationType is introduced.
3) New classes, S100_FC_FeatureBinding, S100_FC_InformationBinding and S100_FC_AttributeBinding are introduced.
4) A new class, S100_CD_AttributeConstraints is introduced.
5) The class FC_FeatureAttribute is specialized to be the abstract class S100_FC_Attribute.
6) New classes, S100_FC_SimpleAttribute and S100_FC_ComplexAttribute are introduced.
7) The classes FC_InheritanceRelation, FC_FeatureOperation FC_Binding, FC_Constraint and FC_BoundFeatureAttribute are not used.

Further reference or explanation of the above changes can be found in the following text where appropriate.

5-3 Normative References

ISO 19110:2005, Geographic Information – Methodology for feature cataloguing

5-4 Principal Requirements

5-4.1 Feature Catalogue

An S-100 based feature catalogue presents the abstraction of reality represented in one or more sets of geographic data as a defined classification of phenomena. The basic level of classification in the feature catalogue is the feature type. Features and attributes are bound in a feature catalogue. The definitions of features and attributes are drawn from a feature concept dictionary. A feature catalogue shall be available in electronic form (e.g. XML) for any set of geographic data that contains features. A feature catalogue may also comply with the specifications of this component of S-100 independently of any existing set of geographic data.

5-4.2 Information Elements

5-4.2.1 Introduction

The following clauses specify general and specific requirements for feature catalogue information elements. A feature catalogue generally consists of a list of named types, a list of properties for named types and the information on how both are linked together. Furthermore it contains a list of sources for its definitions. The model is primarily based on the ISO 19110 standard but there are both extensions and differences in this model.

There are two major extensions to the feature types: information types and complex attributes. To achieve a greater flexibility in modelling the data within a data set it is necessary to define complex structures of information. Both extensions allow the creation of those structures. Whereas complex attributes define complex characteristics for one named type, information types can be shared.

Unlike feature types, which are an abstraction of real world phenomena, information types are just shareable structured pieces of information. In a geographic data set they will be associated to feature types or to other information types. Both types: feature and information, have many common characteristics. This is accommodated by deriving both types from a common abstract base class: the named type.

Complex attributes are an aggregation of other attributes which may be either simple or complex.

The arrangement of content may be different depending on format e.g printed document, XML, hypertext etc.

5-4.2.2 Named Types

5-4.2.2.1 Common Characteristics

Feature and information types are inherited (see 5-4.2.2.2 below) from the abstract class *S100_FC_NamedType*. This class describes all common characteristics, for example, the name and the definition of the corresponding type. Furthermore a code has to be defined for the type. This code will later be used to identify an instance of a named type in a geographic data set. If the definition is taken from a feature concept dictionary that reference is also given.

Feature and information types can be derived from other feature or information types. This includes the possibility that some types are abstract, i.e., no instances of such types can be in a data set. Named types can be characterized by attributes and additional information may be available by information types that are related to them. The former is defined by attribute bindings whereas the latter is achieved by information bindings.

5-4.2.2.2 Inheritance

In data modelling, inheritance is a way to form new types using types that have already been defined. The new types, known as derived types (or sub-types), take over (or inherit) properties of the pre-existing types, which are referred to as base types (or super types). The derived types may define new additional properties. but also change existing properties, the latter is called overriding. This is used to assign unique property values to sub-types such as name and definition but overriding of characteristics such as bindings to attributes should be avoided by only including common characteristics in the super type. In the scope of a feature catalogue both feature and information types can be derived from other feature or information

types. But a feature type cannot be derived from an information type or vice versa. Attributes and associations defined for the super type will also belong to the sub type. The definition of the sub type is usually redefined. In the context of this standard inheritance will be always simple, i.e. each type cannot be derived from more than one super type.

EXAMPLE Cardinal and lateral buoys may be derived from an (abstract) type buoy. The super type already defines attributes like colour, shape, name, and associations to lights or top marks. The derived types add special information only valid for the specialized type like category of cardinal mark or category of lateral mark respectively.

Inheritance builds hierarchical structures which may become difficult to manage if they are too complex or not sufficiently mature. It is always good design practice to keep the depth of an inheritance tree as shallow as possible.

5-4.2.2.3 Feature Types

Feature types are the basic level of classification in the feature catalogue. In addition to the common characteristics they define a feature use type to categorize them. Feature types may be associated to other feature types through feature associations. This will be defined by feature bindings which specify the association as well as the role used for the relationship to the other feature type.

5-4.2.2.4 Information Types

Information types are complex pieces of information in a data set that can be shared between many other feature or information types. In regards to their structure, they can be also seen as feature types without a geometric property which have a structure similar to feature types and are categorized as a separate item type.

5-4.2.3 Properties

5-4.2.3.1 Common Characteristics

Properties for feature and information types are attributes and association roles although the latter only applies to feature types. The common characteristics include name, definition, remarks etc. A reference to a feature concept dictionary may be defined.

5-4.2.3.2 Attributes

Attributes carry the characteristics of feature and information types. Unlike information types they cannot be shared between different instances. i.e, an instance of an attribute belongs to one and only one feature or information type. In this standard there are two different kinds of attributes: simple and complex. Simple attributes carry the value itself, and complex attributes are aggregations of other attributes to achieve a complex and hierarchical data structure.

5-4.2.3.3 Simple Attributes

Simple attributes are designed to carry a value. In the feature catalogue the domain of the value shall be specified. All attribute values are data types. **Erreur ! Source du renvoi introuvable.** contains the full list of data types and their definitions. If the data type is an enumeration a list of ‘Listed Values’ will be defined.

Furthermore the value domain can be constrained by the following:

1) The length of the text
2) A format specification for structured text
3) A numeric range

Details are in 0.Appendix 5-A

5-4.2.3.4 Complex Attributes

Complex attributes are aggregations of other attributes that are either simple or complex. The aggregation is defined by means of attribute bindings.

5-4.2.3.5 Association Roles

An association role describes the nature of the relationship from one feature type to another feature type in a feature association. In this standard each association has exactly two roles. Either or both may be a default. The documentation of application schemas must specify the

rule used for default names. Different rules for default names may apply to different associations in the same application schema, but each role shall have an unambiguous name, be it an explicit role name or a default role name.

5-4.2.4 Feature Associations

Feature associations describe the relationships between feature types. Feature associations have a name, definition, remarks, code etc. Each association uses two roles that define the directed use of the relationship. Either or both of the roles may be a default as described in Part 3.

EXAMPLE 1 Master – Slave is an example of an association with two roles.

EXAMPLE 2 theAuthority – theContactDetails is an example of an association between classes Authority and ContactDetails which uses two default roles.

5-4.2.5 Bindings

5-4.2.5.1 Attribute Bindings

Attribute bindings are used to bind attributes to feature or information types. Additionally, they are used for defining the aggregation of attributes for a complex attribute. The binding specifies the target attribute and the Multiplicity of the attribute. The Multiplicity indicates how many instances of an attribute can be used. Bindings are used to define whether an attribute is mandatory (1..n) or optional (0..n). If the Multiplicity allows more than one instance of an attribute a Boolean flag indicates if the sequence of attributes has a meaning.

If the attribute is a simple attribute with a data type of Enumeration, a list of permitted values can be specified. An empty list indicates that all values defined for the attribute in the feature catalogue are valid.

5-4.2.5.2 Feature Bindings

The feature binding describes the association between two feature types. Both the feature association and the association role are specified together with the target feature type. Furthermore the Multiplicity and the role type are defined. The latter describes the nature of the role.

EXAMPLE The role ‘Lane’ used by a traffic separation scheme to associate its lane parts will have the role type Aggregation, whereas the role “Scheme” used from the lane part to the TSS has the role type Association

5-4.2.5.3 Information Bindings

Information bindings describe which information types can be associated to which feature or information types. In addition to the target information type the Multiplicity of this binding is also defined.

5-4.2.6 Definitions and source references

5-4.2.6.1 Definition sources

This is a list of source documents for the definitions used in the feature catalogue. They are given with their citation information. Usually the definitions will come from a feature concept dictionary but other sources are possible. It is also valid that a definition originates from the feature catalogue; in this case there will be no reference to a definition source.

5-4.2.6.2 Definition references

This information carries the link to the definition source. It points to a definition source and defines the place in that source by means of an identifier. In cases where the source is a feature concept dictionary maintained as a register this reference will be the item identifier.

5-4.2.7 Completeness

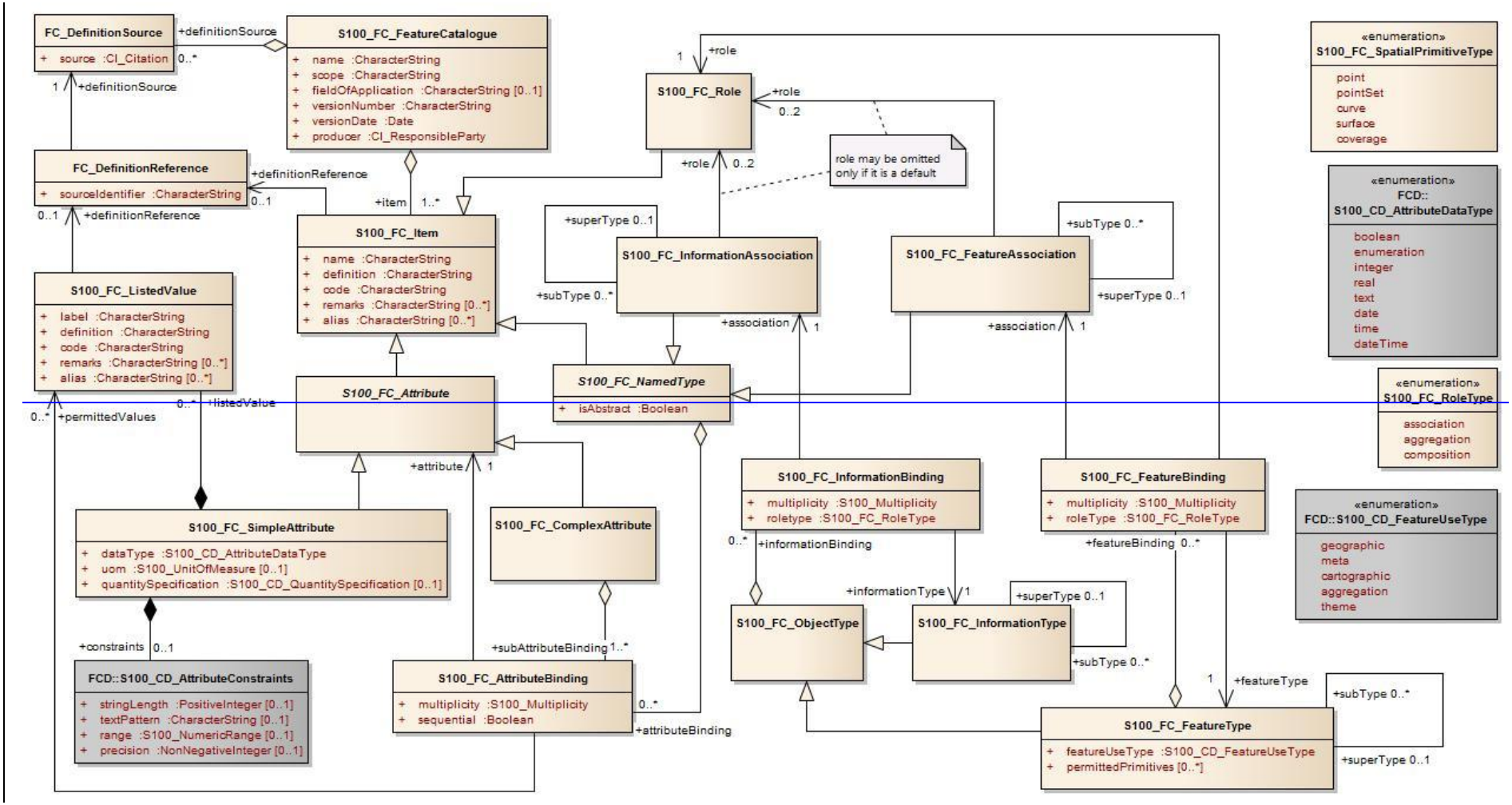
A template for the representation of feature classification information is specified in the following model (Appendix 5-1A). A feature catalogue prepared according to this template shall document all of the feature types and information types found in a given set of geographic data. The feature catalogue shall include identification information as specified. The feature catalogue shall include definitions and descriptions of all feature and information types contained in the data, including any feature attributes and feature associations

contained in the data that are associated with each feature type. To ensure predictability and comparability of feature catalogue content across different applications, it is recommended that the feature catalogue should include only the elements specified in the tables shown at Appendix 5-1A below.

Appendix 5-A (normative)

A-1 Feature Catalogue Model

This appendix presents the S-100 feature catalogue. Figure A.1 is the S-100 feature catalogue modelled in UML and Tables A.1-A.20 illustrate the structure of the Feature Catalogue in conformance to the model shown.


FC_DefinitionSource
+ source :CI_Citation
+definitionSource
0..*
1
+definitionSource
S100_FC_FeatureCatalogue
+ name :CharacterString
+ scope :CharacterString
+ fieldOfApplication :CharacterString [0..1]
+ versionNumber :CharacterString
+ versionDate :Date
+ producer :CI_ResponsibleParty
FC_DefinitionReference
+ sourceIdentifier :CharacterString
+definitionReference
0..1
0..1
+definitionReference
+item
1..*
S100_FC_Item
+ name :CharacterString
+ definition :CharacterString
+ code :CharacterString
+ remarks :CharacterString [0..*]
+ alias :CharacterString [0..*]
S100_FC_ListedValue
+ label :CharacterString
+ definition :CharacterString
+ code :CharacterString
+ remarks :CharacterString [0..*]
+ alias :CharacterString [0..*]
0..*
+permittedValues
0..*
+listedValue
S100_FC_Attribute
+attribute
1
S100_FC_SimpleAttribute
+ dataType :S100_CD_AttributeDataType
+ uom :S100_UnitOfMeasure [0..1]
+ quantitySpecification :S100_CD_QuantitySpecification [0..1]
+constraints
0..1
FCD::S100_CD_AttributeConstraints
+ stringLength :PositiveInteger [0..1]
+ textPattern :CharacterString [0..1]
+ range :S100_NumericRange [0..1]
+ precision :NonNegativeInteger [0..1]
S100_FC_ComplexAttribute
+subAttributeBinding
1..*
S100_FC_AttributeBinding
+ multiplicity :S100_Multiplicity
+ sequential :Boolean
0..*
+attributeBinding
S100_FC_Role
1
+role
+role
0..2
+role
0..2
role may be omitted only if it is a default
S100_FC_InformationAssociation
+superType 0..1
+subType 0..*
+association
1
S100_FC_FeatureAssociation
+subType 0..*
+superType 0..1
+association
1
S100_FC_NamedType
+ isAbstract :Boolean
S100_FC_InformationBinding
+ multiplicity :S100_Multiplicity
+ roletype :S100_FC_RoleType
0..*
+informationBinding
S100_FC_FeatureBinding
+ multiplicity :S100_Multiplicity
+ roleType :S100_FC_RoleType
+featureBinding
0..*
S100_FC_ObjectType
+informationType
1
S100_FC_InformationType
+superType 0..1
+subType 0..*
1
+featureType
S100_FC_FeatureType
+ featureUseType :S100_CD_FeatureUseType
+ permittedPrimitives [0..*]
+subType 0..*
+superType 0..1
«enumeration»
S100_FC_SpatialPrimitiveType
point
pointSet
curve
surface
coverage
«enumeration»
FCD::
S100_CD_AttributeDataType
boolean
enumeration
integer
real
text
date
time
dateTime
«enumeration»
S100_FC_RoleType
association
aggregation
composition
«enumeration»
FCD::S100_CD_FeatureUseType
geographic
meta
cartographic
aggregation
theme

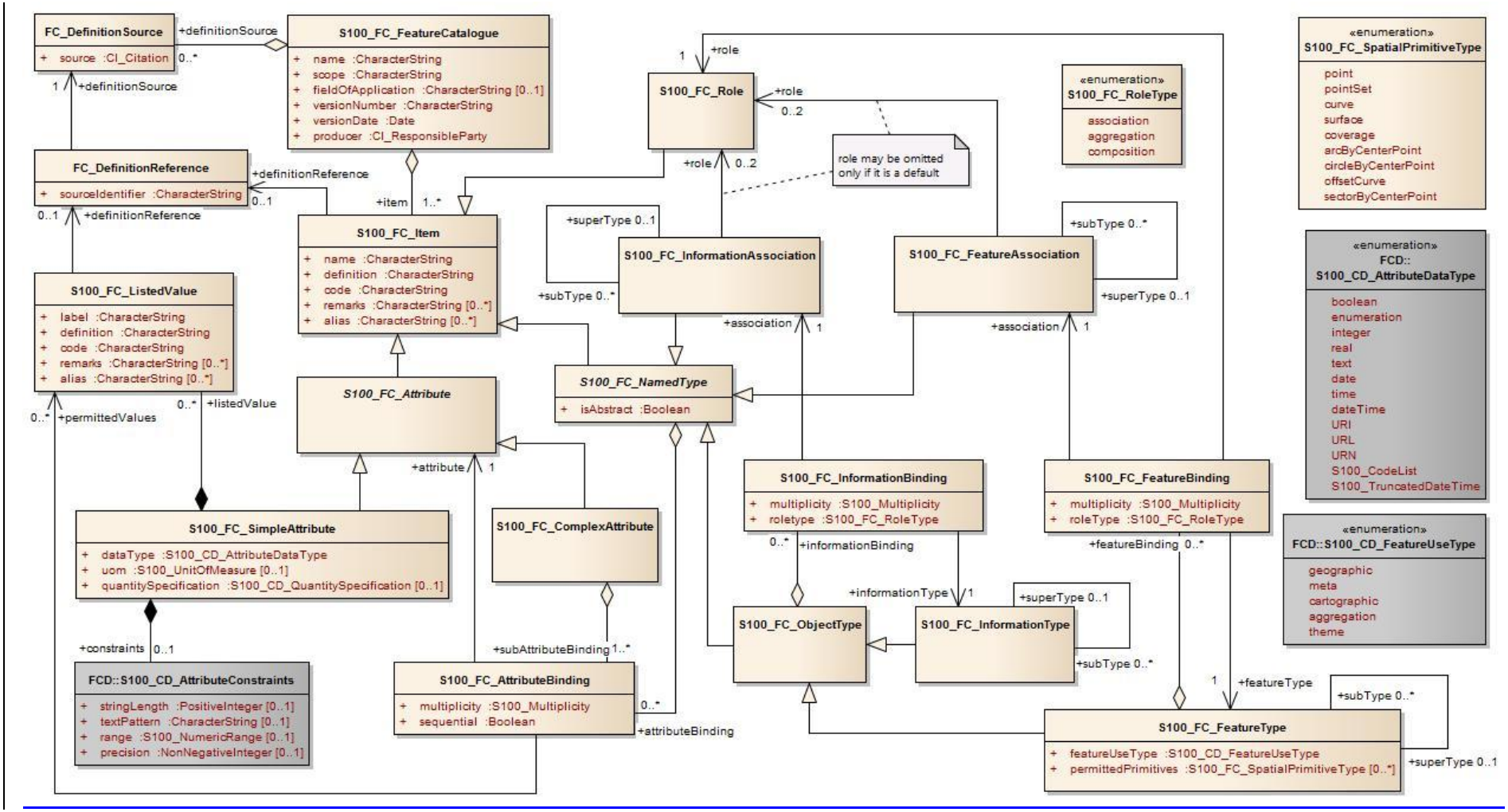


Figure A. 1. Feature Catalogue — UML Model

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_FeatureCatalogue	a feature catalogue contains its identification and contact information, and definition of some number of feature types with other information necessary for those definitions.	-	-	-
Attribute	name	name for this feature catalogue	1	CharacterString	
Attribute	scope	subject domain(s) of feature types defined in this feature catalogue	1	CharacterString	
Attribute	fieldOfApplication	description of kind(s) of use to which this feature catalogue may be put	0..1	CharacterString	
Attribute	versionNumber	version number of this feature catalogue, which may include both a major version number or letter and a sequence of minor release numbers or letters, such as “3.2.4a.” The format of this attribute may differ between cataloguing authorities.	1	CharacterString	
Attribute	versionDate	effective date of this feature catalogue	1	Date	
Attribute	producer	name, address, country, and telecommunications address of person or organization having primary responsibility for the intellectual content of this feature catalogue	1	CI_ResponsibleParty	
Role	item	list of items defined by this feature catalogue; items are feature types, information types, feature associations, information associations, attributes, and roles	1..*	S100_FC_Item	Aggregation
Role	definitionSource	list of sources of definitions of items and listed values that are defined by this feature catalogue. Usually those sources are feature data dictionaries.	0..*	FC_DefinitionSource	Aggregation

Table A.1 — S100_FC_FeatureCatalogue

Role Name	Name	Description	Mult	Type	Remarks
Class	FC_DefinitionSource	class that specifies the source of a definition.	-	-	-
Attribute	source	actual citation of the source, sufficient to identify the document and how to obtain it	1	CI_Citation	

Table A.2 — FC_DefinitionSource

Role Name	Name	Description	Mult	Type	Remarks
Class	FC_DefinitionReference	class that links a data instance to the source of its definition	-	-	-
Attribute	sourceIdentifier	information to locate the definition in the source document. The format of this information is specific to the structure of the source document.	1	CharacterString	
Role	definitionSource	the source of the definition	1	FC_DefinitionSource	

Table A.3 — FC_DefinitionReference

Role Name	Name	Description	Mult.	Type	Remarks
Class	S100_FC_Item	abstract base class that defines the common properties of all items in the feature catalogue; items are feature types, information types, feature associations, information associations, attributes and roles	-	-	abstract
Attribute	name	name of the item	1	CharacterString	
Attribute	definition	definition of the named type in a natural language	1	CharacterString	
Attribute	code	Code that uniquely identifies the named type within the feature catalogue.	1	CharacterString	
Attribute	remarks	further explanations about the item	0..*	CharacterString	
Attribute	alias	equivalent name(s) of this item	0..*	CharacterString	
Role	definitionReference	the link to the source of definition	0..1	FC_DefinitionReference	

Table A.4 — S100_FC_Item

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_NamedType	abstract base class that defines the common properties for feature types and information types	-	-	abstract class
Attribute	isAbstract	indicates if instances of this named type can exists in a geographic data set. Abstract types cannot be instantiated but serve as base classes for other (non-abstract) types.	1	Boolean	
Role	attributeBinding	list of bindings to attributes which describe the characteristic of this named type.	0..*	S100_FC_AttributeBinding	Aggregation

Table A.5 — S100_FC_NamedType

Role Name	Name	Description	Mult.	Type	Remarks
Class	S100_FC_ObjectType	abstract base class that defines the common properties for feature types and information types	-	-	abstract; derived from S100_FC_NamedType
Role	informationBinding	list of bindings to information types that can be associated to this object type by means of an information association	0..*	S100_FC_InformationBinding	aggregation

Table A.6 — S100_FC_ObjectType

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_InformationType	class that defines all properties of an information type	-	-	Derived from S100_FC_NamedType
Role	superType	indicates the information type from which an information type is derived. The sub type will inherit all properties from its super type: name, definition and code will usually be overridden by the sub type, although new properties may be added to the sub type.	0..1	S100_FC_InformationType	

Role	subType	indicates the information types which are derived from an information type.	0..*	S100_FC_InformationType	

Table A.7 — S100_FC_InformationType

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_FeatureType	class that defines all properties of a feature type	-	-	derived from S100_FC_NamedType
Attribute	featureUseType	the use type of this feature type	1	S100_FC_FeatureUseType	
Attribute	permittedPrimitives	the combination of 0 or more spatial primitives permitted for feature type.	0..*	S100_FC_SpatialPrimitiveType	
Role	featureBinding	list of bindings to feature types that can be related to this feature type by means of a feature association	0..*	S100_FC_FeatureBinding	Aggregation
Role	superType	indicates the feature type from which a feature type is derived. The sub type will inherit all properties from its super type: name, definition and code will usually be overridden by the sub type, although new properties may be added to the sub type.	0..1	S100_FC_FeatureType	
Role	subType	indicates the feature types which are derived from a feature type.	0..*	S100_FC_ FeatureType	

Table A.8 — S100_FC_FeatureType

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_InformationAssociation	An information association describes the relationship between an object (feature or information type) and an information type.	-	-	derived from S100_FC_NamedType Individual product specifications may restrict directionality.
Role	role	the role of the association.	0..2	S100_FC_Role	default role name if missing.

					Product specification can constrain further.
Role	superType	Indicates the information association from which an information association is derived. The sub type will inherit all properties from its super type: name, definition and code will usually be overridden by the sub type, although new properties may be added to the sub type.	0..1	S100_FC_InformationAssociation	
Role	subtype	Indicates the information associations which are derived from an information association.	0..*	S100_FC_InformationAssociation	

Table A.9 — S100_FC_InformationAssociation

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_FeatureAssociation	a feature association describes the relationship between two feature types. A feature association is bidirectional and has a separate role for each direction.	-	-	
Role	role	the role of the association	1	S100_FC_Role	
Role	superType	Indicates the feature association from which an feature association is derived. The sub type will inherit all properties from its super type: name, definition and code will usually be overridden by the sub type, although new properties may be added to the sub type.	0..1	S100_FC_FeatureAssociation	
Role	subType	Indicates the feature associations which are derived from a feature association.	0..*	S100_FC_FeatureAssociation	

Table A.10 — S100_FC_FeatureAssociation

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_Role	A role which can be used in a feature association or	-	-	derived from

		an information association			S100_FC_Item

Table A.11 — S100_FC_Role

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_Attribute	abstract base class for the two kinds of attributes: simple attributes and complex attributes. Attributes carry the characteristics of named types	-	-	abstract derived from S100_FC_Item

Table A.12 — S100_FC_Attribute

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_SimpleAttribute	attribute that carries a value	-	-	derived from S100_FC_Attribute
Attribute	dataType	the data type of this feature attribute	1	S100_CD_AttributeDataType	
Attribute	uom	unit of measure used for values of this feature attribute	0..1	S100_UnitOfMeasure	
Attribute	quantitySpecification	the physical quantity	0..1	S100_CD_QuantitySpecification	
Role	constraints	constraints which may apply to the attribute	0..1	S100_FC_AttributeConstraints	Composition
Role	listedValue	set of listed values for an enumerated attribute domain	0..*	S100_FC_ListedValue	Composition. Applies only if dataType is Enumeration

Table A.13 — S100_FC_SimpleAttribute

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_ComplexAttribute	a complex attribute consists of a list of sub-attributes which can be both simple and complex attributes	-	-	derived from S100_FC_Attribute

Role	subAttributeBinding	list of bindings to the sub-attributes	1..*	S100_FC_AttributeBinding	Aggregation

Table A.14 — S100_FC_ComplexAttribute

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_ListedValue	value of an enumerated attribute domain, including its codes and definition	-	-	
Attribute	label	descriptive label that uniquely identifies one value of the feature attribute	1	CharacterString	
Attribute	definition	definition of the listed value in a natural language.	1	CharacterString	
Attribute	code	Numeric code that uniquely identifies the listed value for the corresponding feature attribute.	1	PositiveInteger	
Attribute	remarks	further explanations about the listed value	0..*	CharacterString	
Attribute	alias	equivalent name(s) of this listed value	0..*	CharacterString	
Role	definitionReference	the link to the source of the definition	0..1	FC_DefinitionReference	

Table A.15 — S100_FC_ListedValue

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_AttributeBinding	class that is used to describe the specifics of how an attribute is bound to a particular named type or a complex attribute	-	-	
Attribute	multiplicity	multiplicity defining how many instances of the attribute can be part of the named type or complex attribute	1	S100_Multiplicity	
Attribute	sequential	describes if the sequence of the attributes is meaningful or not	1	Boolean	Applies only to attributes which may occur more than once.
Role	permittedValues	permissible values of the attribute	0..*	S100_FC_ListedValue	Applies only to attributes of data type enumeration.

Role	attribute	the attribute that is bound to the item or complex attribute	1	S100_FC_Attribute	

Table A. 16 — S100_FC_AttributeBinding

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_InformationBinding	class describing the use of a information type by a named type.	-	-	
Attribute	multiplicity	Multiplicity defining how many instances of the target information type can be linked to one instance of the named type	1	S100_Multiplicity	
Attribute	roleType	the nature of the association end	1	S100_FC_RoleType	
Role	association	the association used for the binding; defining also the role	1	S100_FC_InformationAssociation	
Role	informationType	the target information type	1	S100_FC_InformationType	

Table A. 17 — S100_FC_InformationBinding

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_FC_FeatureBinding	class describing the relationship from one feature type to another feature type by means of a feature association	-	-	
Attribute	multiplicity	Multiplicity defining how many instances of the target feature type can be linked to one instance of the source feature type	1	S100_Multiplicity	
Attribute	roleType	the nature of the association end	1	S100_FC_RoleType	
Role	featureType	the target feature type	1	S100_FC_FeatureType	
Role	role	the role used for the binding. It must be part of the association used for the binding and defines the end of the association.	1	S100_FC_AssociationRole	

Role	association	the association used for the binding.	1	S100_FC_FeatureAssociation	

Table A.18 — S100_FC_FeatureBinding

Role Name	Name	Description	Comments
Enumeration	S100_FC_RoleType	defines the type of a role	
Literal	association	an association is used to describe a relationship between two feature types that involves connections between their instances	
Literal	aggregation	an aggregation association is a relationship between two feature types, in which one of the feature types plays the role of a container and the other plays the role of a containee.	
Literal	composition	a composition association is a strong aggregation. In a composition association, if a container object is deleted then all of its containee objects are deleted as well. In other words containee objects cannot exist without the container object.	

Table A.19 — S100_FC_RoleType

Role Name	Name	Description	Remarks
Enumeration	S100_FC_SpatialPrimitiveType	specifies spatial primitives permitted for use with a feature instance	
Literal	point	point spatial primitive	**GM_Point**
Literal	pointSet	point set spatial primitive	**GM_MultiPoint**
Literal	curve	curve spatial primitive	**GM_Curve**
Literal	surface	surface spatial primitive	**GM_ Surface**
Literal	coverage	coverage spatial primitive	
Literal	arcByCenterPoint	primitive for arc described by centre, angles, and radius	**ArcByCenterPoint (ISO 19136)**
Literal	circleByCenterPoint	primitive for circle described by centre and radius	**CircleByCenterPoint (ISO 19136)**
Literal	offsetCurve	offset curve primitive	**GM_OffsetCurve**

Literal	sectorByCenterPoint	primitive for sector described by centre, angles, and radius	

Table A.20 — S100_FC_SpatialPrimitiveType