Mathias Jonas
BSH Hamburg

C&S/14/12B

ISO TC 211 work implications for S-52 App2

Preface of the purpose of discussion at C&S14 in Ottawa, May 2003.

TSMAD currently undergoes to discuss the application of ISO 191xx series to future S-57 Version 4.0. Although very little is known about this in detail on C&S side it seems that the interface between future S-57, version 4.0 and S-52, Appendix 2 may be affected to adapt.

The applying standard for this purpose of the named series seems to be ISO 19117 “Geographic information – Portrayal”. The scope of this standard says optimistically:

Definition of a schema describing the portrayal of geographic information in **a form understandable by humans** including the methodology for describing symbols and mapping of the schema to an application schema. This work does not include standardization of cartographic symbols, and their geometric and functional description.

Surprisingly (or not – as you feel) the general construction (see page 7 of the attached wording of the standard) is very similar to the PresLib construction. You can easily substitute

- Feature with object
- Portrayal catalogue with look-up table
- Portrayal rule with CSP
- External function with actual scale or actual position
- Portrayal specification with symbol libary.

However the description in detail shall be based on
- OCL - object constraint language
- UML - unified message language

and this havent been applied yet to an IHO document (at least to my knowledge).

Possible needs to take this for future revisions of Preslib into account, and if so, questions like
- when does it happen?
- who does it?
- who pays it?

shall be discussed in Ottawa.

ISO TC 211

Date: 2001-06-15

ISO/DIS 19117

ISO TC 211/WG 4

Secretariat: NSF

Geographic information — Portrayal

Information géographique — Présentation
Fehler! AutoText-Eintrag nicht definiert.

Document type: International Standard
Document subtype:
Document stage: (40) Enquiry
Document language: E

D:\daten\texte\IHO\colours&symbols\C&S 14 Ottawa\papers\C&S14-12B ISO 19117.doc STD Version 1.0

Contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

International Standard ISO 19117 was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics*.

Annex A forms a normative part of this International Standard. Annex B is for information only.

Introduction

This International Standard is an abstract document and is not intended for direct implementation. It gives general guidelines to the application developers about the mechanism that shall be used to portray the feature instances of a dataset. The portrayal mechanism described makes it possible to have general rules valid for the whole dataset, and at the same time rules valid for a specific value of a feature attribute only. Different computer graphics standards use different attributes to visualize geometric primitives. E.g. a line can be distinguished by thickness, width, colour, stippling, anti-aliasing etc. This International Standard therefore includes a mechanism for declaring portrayal attributes as part of the portrayal specification.

In some cases whole feature classes have to be referenced and portrayed in a specific way, e.g. as symbols on nautical charts. Several symbol standards exist, and without a portrayal standard the application would have to set up a separate interface to each of these standards. With this International Standard all the supported symbol standards may be handled in a uniform way.

Geographic information — Portrayal

1 Scope

Definition of a schema describing the portrayal of geographic information in a form understandable by humans including the methodology for describing symbols and mapping of the schema to an application schema. This work does not include standardization of cartographic symbols, and their geometric and functional description.

2 Conformance

Any schema describing the portrayal of geographic information claiming conformance with this International Standard shall pass all the requirements described in the abstract test suite, presented in annex A.

3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 19103:—[1] *Geographic information — Conceptual schema language.*
ISO 19107:—[1] *Geographic information — Spatial schema.*
ISO 19109:—[1] *Geographic information — Rules for application schema.*
ISO 19115:—[1] *Geographic information — Metadata.*

4 Terms and definitions

For the purpose of this International Standard, the following terms and definitions apply.

4.1
annotation
any marking on illustrative material for the purpose of clarification, such as numbers, letters, symbols, and signs

4.2
class
description of a set of objects that share the same attributes, operations, relationships, and semantics

NOTE A class may use a set of interfaces to specify collections of operations it provides to its environment. [Unified Modeling Language version 1.1]

4.3
curve
bounded, connected 1-dimensional geometric primitive, representing the continuous image of a line, and therefore fully realizable as a 1-parameter set of points

NOTE The boundary of a curve is the set of points at either end of the curve. The first point is called the start point, and the last is the end point.

[1] To be published.

4.4
dataset
identifiable collection of data

NOTE The principles which apply to datasets may also be applied to dataset series and reporting groups.

4.5
external function
function not part of the application schema

NOTE The electronic map in a car navigation system has to be displayed so that the up-direction of the map is always in the direction the car is moving. To be able to specify the rotation of the map, the current position of the car must be retrieved continuously from an external position device using an external function.

4.6
feature
abstraction of real world phenomena

4.7
feature attribute
characteristic of a feature

EXAMPLE 1 A feature attribute named ‘colour’ may have an attribute value ‘green’ which belongs to the data type ‘text’.

EXAMPLE 2 A feature attribute named ‘length’ may have an attribute value ’82.4’ which belongs to the data type ‘real’.

NOTE 1 A feature attribute has a name, a data type, and a value domain associated to it. A feature attribute for a feature instance also has an attribute value taken from the value domain.

NOTE 2 In a feature catalogue, a feature attribute may include a value domain but does not specify attribute values for feature instances.

4.8
feature portrayal rule set
collection of portrayal rules that apply to a feature instance

4.9
geographic information
information concerning phenomena implicitly or explicitly associated with a location relative to the Earth

4.10
geometric primitive
object representing a single, connected, homogenous element of geometry

4.11
instance
individual entity with its own identity and value [UML]
NOTE [Unified Modeling Language version 1.1]. A descriptor specifies the form and behaviour of a set of instances with similar properties.

4.12
metadata
data about data

4.13
point
0-dimensional geometric primitive

4.14
portrayal
presentation of information to humans

4.15
portrayal catalogue
collection of all defined portrayals

4.16
portrayal rule
rule that is applied to the feature to determine what portrayal specification to use

4.17
portrayal service
generic interface used to portray features

4.18
portrayal specification
collection of operations applied to the feature instance to portray it

4.19
spatial attribute
feature attribute describing the spatial representation of the feature by coordinates, mathematical functions and/or boundary topology relationships

4.20
surface
bounded, connected 2-dimensional geometric primitive, representing the continuous image of a region of a plane

4.21
type
class

5 Abbreviated terms

IDL	ISO Interface Definition Language
OCL	Object Constraint Language
UML	Unified Modeling Language
URL	Uniform Resource Locator (also called URI – Uniform Resource Identifier)

6 UML

6.1 Notations

The diagrams that appear in this International Standard are presented using the UML static structure diagram with the IDL basic type definitions and OCL as the conceptual schema language. The UML notations used in this International Standard are described in Figure 1.

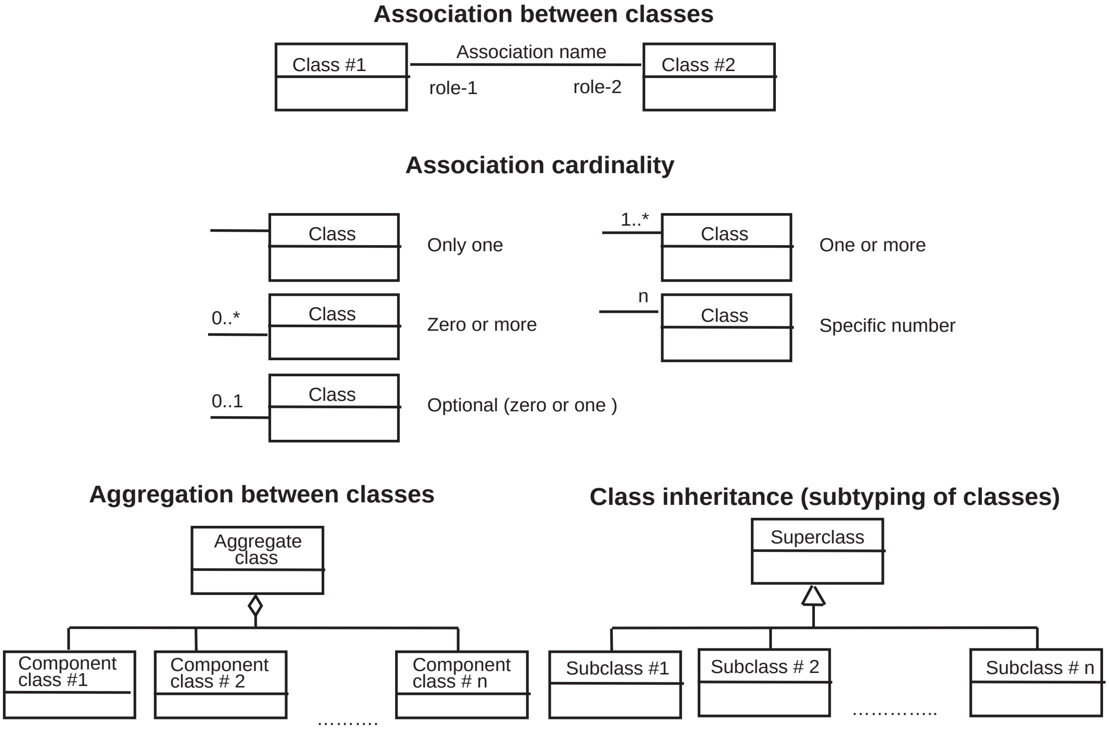

Figure 1 — The UML notations used in this International Standard

6.2 UML model stereotypes

A UML stereotype is an extension mechanism for existing UML concepts. It is a model element that is used to classify (or mark) other UML elements so that they in some respect behave as if they were instances of new virtual or pseudo metamodel classes whose form is based on existing base metamodel classes. Stereotypes augment the classification mechanisms on the basis of the built-in UML metamodel class hierarchy. Below are brief descriptions of the stereotypes used in this International Standard, for more detailed descriptions consult ISO/TS 19103.
In this International Standard the following stereotypes are used:

— Interface – definition of a set of operations that is supported by objects having this interface.

— Type – stereotyped class used for specification of a domain of instances (objects), together with the operations applicable to the objects. A type may have attributes and associations.

— MetaClass – a class whose instances are classes. Metaclasses are typically used in the construction of metamodels.

— Leaf – package that contains definitions, without any sub-packages.

7 The portrayal mechanism

7.1 Introduction

This International Standard defines a feature-centred rule based portrayal mechanism. Instances of features are portrayed based on rules, which make use of geometry and attribute information. The relationship between the feature instances, attributes and the underlying spatial geometry is specified in an application schema according to ISO 19109. Spatial geometry and associated topological relationships are defined in ISO 19107.

Portrayal information needs to be present to portray a dataset containing geographic data. The portrayal information is handled as portrayal specifications applied according to specific portrayal rules (see clause 7). The portrayal mechanism makes it possible to portray the same dataset in different ways without altering the dataset itself. The portrayal mechanism is illustrated by Figure 2.

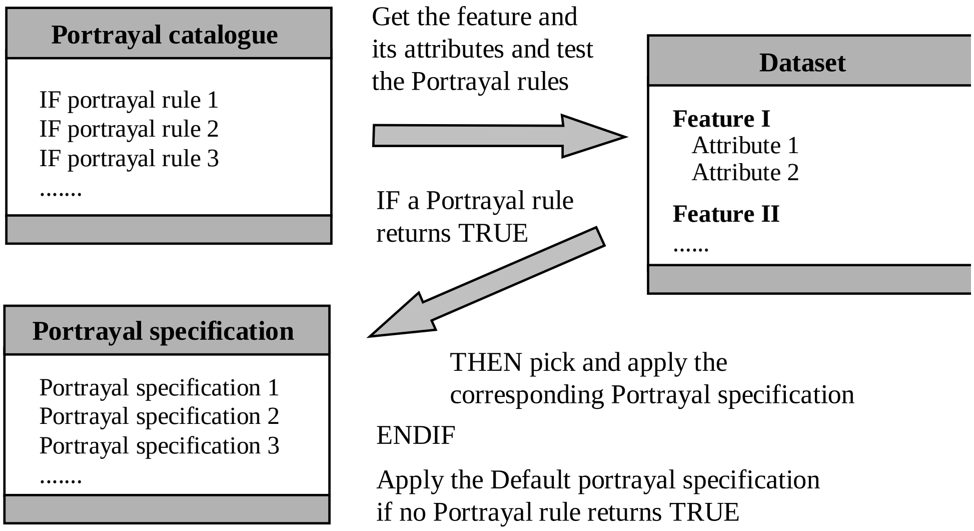
Portrayal catalogue
IF portrayal rule 1
IF portrayal rule 2
IF portrayal rule 3
....
Get the feature and its attributes and test the Portrayal rules
Dataset
Feature I
Attribute 1
Attribute 2
Feature II
......
IF a Portrayal rule returns TRUE
Portrayal specification
Portrayal specification 1
Portrayal specification 2
Portrayal specification 3
....
THEN pick and apply the corresponding Portrayal specification
ENDIF
Apply the Default portrayal specification if no Portrayal rule returns TRUE

Figure 2 — The portrayal mechanism without priority attributes

The portrayal specifications and portrayal rules shall not be part of the dataset. The portrayal rules shall be stored in a portrayal catalogue. The portrayal specifications shall be stored separately and referenced from the portrayal rules. The portrayal rules shall be specified for the feature class or feature instances they will be applied on. The portrayal specifications may be stored externally and referenced using a universal reference standard such as a network based URL.

Portrayal information may be specified either by sending a portrayal catalogue and portrayal specifications with the dataset, or by referencing to an existing portrayal catalogue and portrayal specifications from Metadata. The model in Figure 3 shows how the portrayal catalogue is referenced by the dataset metadata. Only the metadata reference is shown and not the contents of the portrayal catalogue (see ISO 19115).

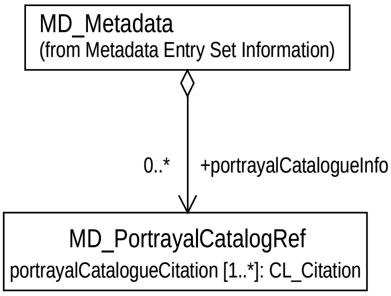
MD_Metadata
(from Metadata Entry Set Information)
0..*
+portrayalCatalogueInfo
MD_PortrayalCatalogRef
portrayalCatalogueCitation [1..*]: CL_Citation

Figure 3 — UML model of the portrayal part of the 19115

The portrayal rules shall be expressed using the OCL constraint language and UML action language. The portrayal rule mechanism may be used to handle portrayal issues that have to be solved as they happen, such as how to automatically place text on maps, and special representations of the feature instances according to e.g. time of day or scale. The value of external functions (see 8.3.5), such as time of day or scale, may be included in the portrayal rules.

The portrayal rules in the portrayal catalogue shall be tested on the attributes of the feature instances in the dataset. The portrayal rule shall be applied as a query statement that returns TRUE or FALSE. New portrayal rules shall be applied until one returns TRUE. The portrayal specification associated with that particular portrayal rule shall then be applied. If no portrayal rule returns TRUE then the default portrayal specification shall be used.

A portrayal service is used to portray a feature instance or instances. The portrayal service applies operations using the parameters defined in a portrayal specification (see 8.2).

7.2 Priority attribute

An optional priority attribute may be added to the portrayal rules. The attribute gives an integer value deciding in which order portrayal rules shall be applied if more than one returns TRUE for one feature instance. A portrayal rule with a high priority number takes precedence over one with a lower number. If two portrayal rules returning TRUE have the same priority value, then the application shall decide which one takes precedence. If priority attributes are used all the portrayal rules shall have a priority attribute.

7.3 Portray nothing

For a feature instance that is not to be portrayed, a portrayal rule shall return TRUE with associated portrayal specification that is empty when tested on the attributes of the feature instance (see 8.3.4). If no portrayal rule returns TRUE then the default portrayal specification shall be applied.

7.4 Default portrayal specification

The default portrayal specification shall be applied according to at least one of the spatial attributes of the feature instance. A default portrayal specification shall be present to ensure that no feature instance is left unportrayed by mistake, and the provider of the dataset shall specify its values. External functions shall not be used in the default portrayal specification.
If the application fails to portray the data for some reason the failure shall be handled by the application.

7.5 Annotation

The information that is to be portrayed shall be defined in an application schema. Typically there are two types of information included in a dataset. These are geospatial information and annotation. Annotation includes text, grids, legends and special features such as a compass rose.

7.6 Overview of portrayal

Portrayal is illustrated by Figure 4. The diagram is not part of the portrayal schema and not for implementation. It is intended as an explanatory aid only.

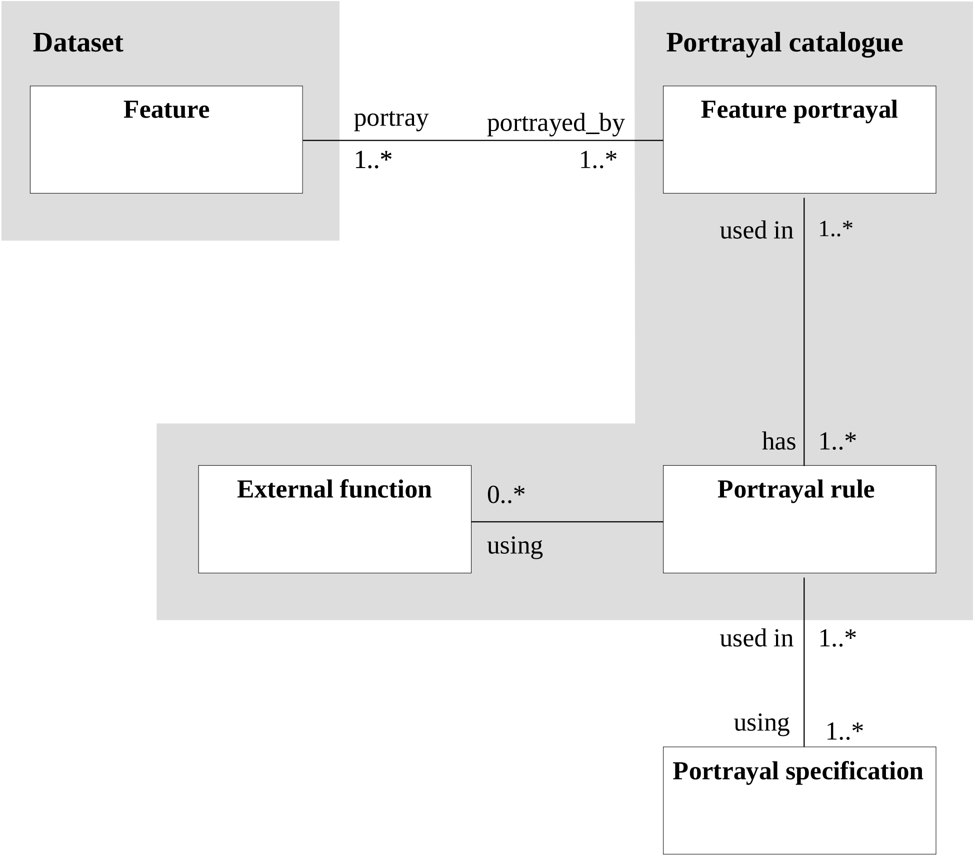

Figure 4 — Overview of portrayal (informative)

The portrayal catalogue consists of the feature portrayal, portrayal rule and external function, as shown in Figure 4. To produce different products, several portrayal catalogues may exist, portraying one or more datasets. Dataset is explained in ISO 19109. The portrayal catalogue relates to one portrayal specification, and one portrayal specification can be used in one or more portrayal catalogues. A portrayal rule consists of two parts, a query statement that can use one or more external functions, and one or more action statements.

EXAMPLE 1 A Dataset contains instances of the feature class Road. The feature class Road contains two attributes, classification and segment. The classification attribute is of string data type, and may have the value “country road” or “town road”. The segment attribute is of GM_Curve type and contains the spatial description of the road. The Portrayal Specification used is called N50_specification. The two Portrayal Rules in this example look like this (The “quotes” in this example are used to show the contents of a string):

IF (Road.classification EQ “country road”) THEN drawCurve (“N50_specification.Solid_red_line”, Road.segment)

IF (Road.classification EQ “town road”) THEN drawCurve (“N50_specification.Solid_yellow_line”, Road.segment)

In this example the THEN separates the query and action statements. The drawCurve is an action statement drawing an actual curve using geometry from Road.segment and colour, line width etc. information from N50_specification.Solid_red_line and N50_specification.Solid_yellow_line.

EXAMPLE 2 If the portrayal varies with the scale, an External Function is needed as part of the query statement. One of the Portrayal Rules then may look like this (The “quotes” in this example are used to show the contents of a string.):

IF (Road.classification EQ “country road” AND Scale (<=20000)) THEN drawCurve (“N50_specification.Solid_thin_red_line”, Road.segment)

Here Scale is a function that gets the display scale from the display device.

The portrayal rule shall refer to the appropriate attributes, functions and relationships defined in an application schema. The portrayal catalogue shall also list the external functions used, including the parameters and returned values.
EXAMPLE 3 In these cases external functions are necessary;

— The electronic map in a car navigation system has to be displayed so that the up-direction of the map is always in the direction the car is moving. To be able to specify the rotation of the map, the current position of the car must be retrieved continuously from an external position device using an external function.

— For electronic chart displays onboard a vessel some of the symbols are only valid for certain scale-intervals. To be able to turn the symbols on and off the system must be told what scale the map is displayed in by the display part of the chart system. A danger zone is defined spatially as a surface. Below a certain scale the danger zone is better displayed by a point symbol. An external function may be used to compute the centroid of the area and the coordinates of the centroid used to position the point symbol.

— An external function may be used to avoid visual conflicts between text and symbols placed on a map, or to handle the placement of text along curves.

8 Portrayal schema

8.1 Overview

The portrayal schema consists of three main parts:

1) The portrayal service, which defines the portrayal operations.

2) The portrayal catalogue package, which defines portrayal rules for the feature classes defined in an application schema.

3) The portrayal specification package, which defines the underlying parameters that are required by the portrayal service.

The portrayal catalogue and portrayal specification are described in separate packages, see Figure 5.

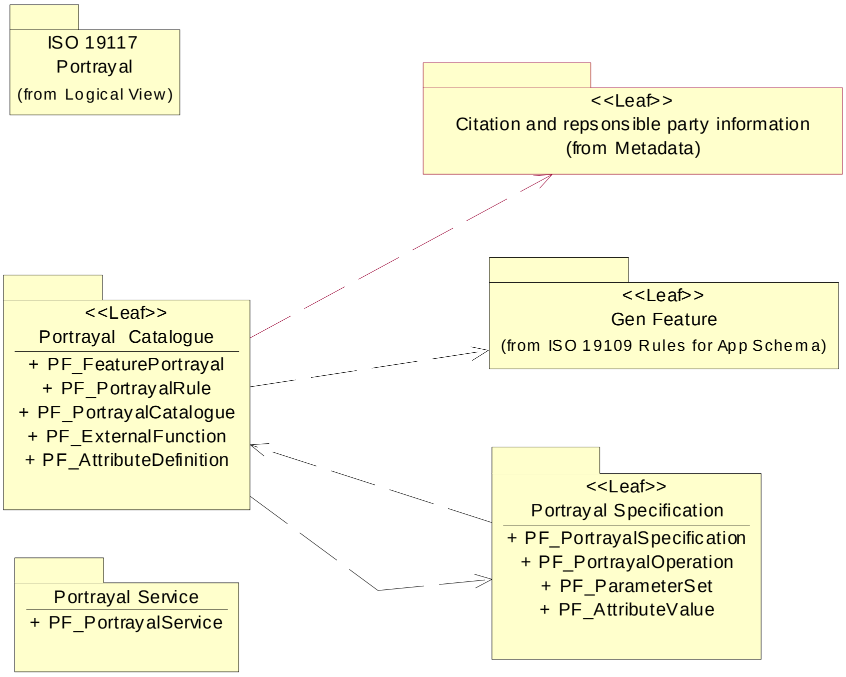

Figure 5 — Portrayal information and portrayal service

8.2 Portrayal service

The portrayal service is a service that is used to portray a feature instance or instances. Portrayal shall not be limited to visual rendering, but may include audio, tactile and other mediaPF_PortrayalService (see Figure 6) has one operation (portrayFeature). The operation refers to one or more feature instance(s), and a portrayalCatalogue.

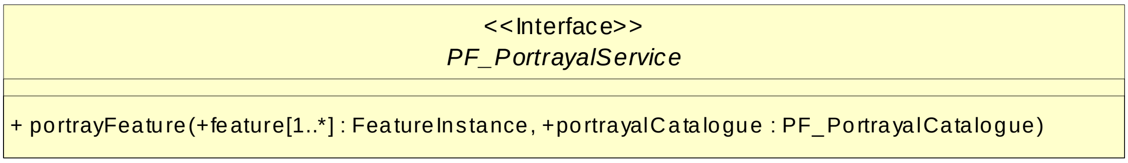
<<Interface>>
PF_PortrayalService
+ portrayFeature(+feature[1..*] : FeatureInstance, +portrayalCatalogue : PF_PortrayalCatalogue)

Figure 6 — Details of the portrayal service

8.3 Portrayal catalogue package

8.3.1 Overview

The portrayal catalogue package defines the following classes: PF_PortrayalCatalogue, PF_FeaturePortrayal, PF_PortrayalRule, PF_ExternalFunction and PF_AttributeDefinition.

The portrayal catalogue organizes a set of portrayal rules. The feature portrayal holds the portrayal rules associated with a particular feature class. A portrayal rule consists of a query statement and portrayal action. The portrayal action specifies which operations that should be invoked on the portrayal service. See Figure 7.

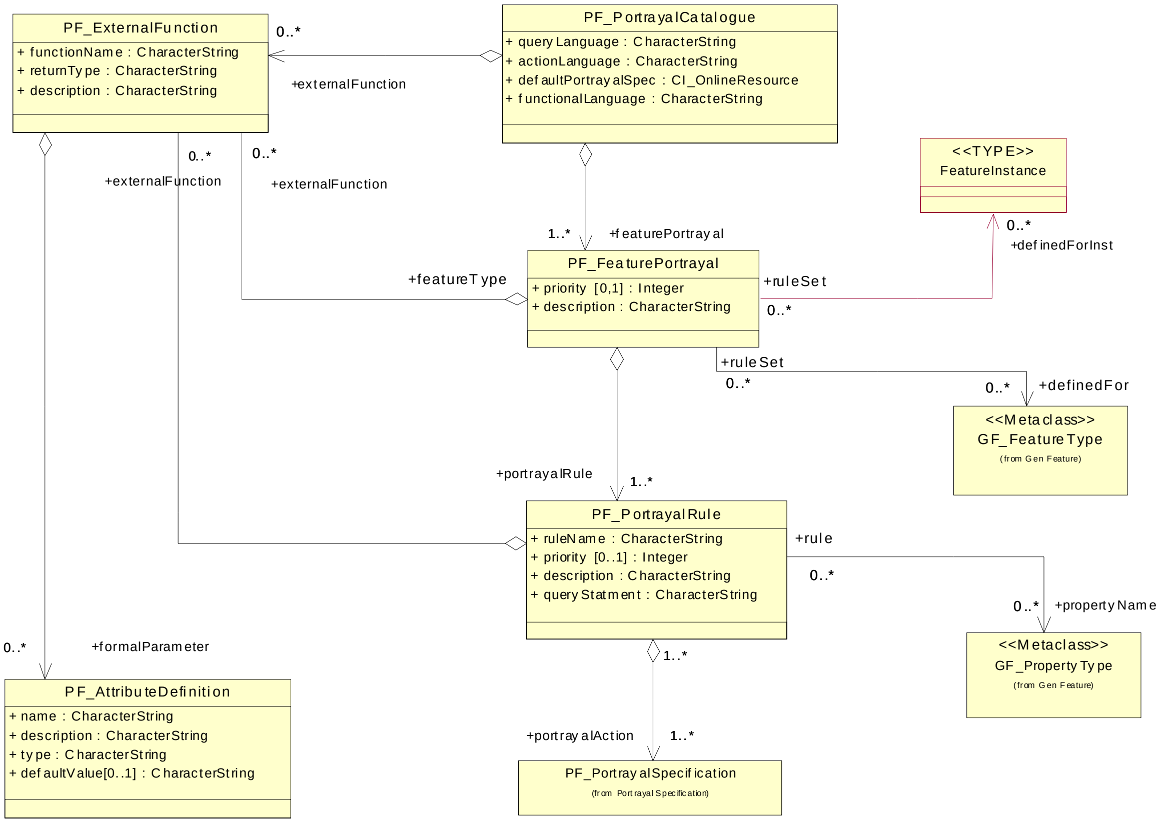
PF_ExternalFunction
+ functionName : CharacterString
+ returnType : CharacterString
+ description : CharacterString
0..*
+externalFunction
PF_PortrayalCatalogue
+ queryLanguage : CharacterString
+ actionLanguage : CharacterString
+ defaultPortrayalSpec : CI_OnlineResource
+ functionalLanguage : CharacterString
0..*
+externalFunction
0..*
+externalFunction
<<TYPE>>
FeatureInstance
1..*
+featurePortrayal
0..*
+definedForInst
+featureType
PF_FeaturePortrayal
+ priority [0,1] : Integer
+ description : CharacterString
+ruleSet
0..*
+ruleSet
0..*
0..*
+definedFor
<<Metaclass>>
GF_FeatureType
(from General Feature)
+portrayalRule
1..*
PF_PortrayalRule
+ ruleName : CharacterString
+ priority [0..1] : Integer
+ description : CharacterString
+ queryStatment : CharacterString
+rule
0..*
0..*
+propertyName
<<Metaclass>>
GF_PropertyType
(from General Feature)
0..*
+formalParameter
PF_AttributeDefinition
+ name : CharacterString
+ description : CharacterString
+ type : CharacterString
+ defaultValue[0..1] : CharacterString
1..*
+portrayalAction
1..*
PF_PortrayalSpecification
(from Portrayal Specification)

Figure 7 — Portrayal catalogue package

8.3.2 Portrayal catalogue

A portrayal catalogue class consists of a set of feature portrayal objects, many may exist for each feature type that may occur in the dataset. Each feature portrayal object has assigned a set of portrayal rules. The portrayal catalogue also has a relationship to the external function objects that may be used in the portrayal rules.
Class name: PF_PortrayalCatalogue
Attributes:

+queryLanguage : Specifies the notation for the queries performed.
+actionLanguage : Specifies the notation for the action performed.
+defaultPortrayalSpec: Identifies the default portrayal specification.
+functionalLanguage: Name of the functional language used.

Associations:

+featurePortrayal: A list of feature portrayal objects, many may exist for each feature type that may occur in the dataset.
+externalFunction: A list of external functions.

Constraints: None
OCL shall be used as the query language and UML as the action language.

8.3.3 Feature portrayal

PF_FeaturePortrayal object refers to a feature class through its feature name attribute PF_FeaturePortrayal object refers to a feature type through its feature name attribute. The feature type shall be defined in the feature catalogue and further specified in the application schema. For each feature class there can be a number of portrayal rules.
Class name: PF_FeaturePortrayal
Attributes:

+priority: Priority of portrayal.
+description: A textual description of the feature class and it's portrayal rules.

Associations:
+portrayalRule: A list of portrayal rule objects.
+definedFor: A reference to the feature class name of the feature classes that are specified in the application schema.
+externalFunction: A list of external functions.
+definedForInst A list of feature instances.

Constraints: None

8.3.4 Portrayal rule

PF_PortrayalRule describes one particular portrayal rule. It has a name, a textual description, a formal definition of rule statement and a portrayal action association. If the formal definition of the rule evaluates to true the corresponding portrayal action shall be invoked.
If no portrayal action is required no portrayal action shall be specified.
The portrayal rule is invoked by evaluating the expression in the +queryStatement attribute. If the expression evaluates to TRUE the +portrayalAction shall be performed. If the expression evaluates to FALSE the +portrayalAction shall not be performed.
Class name: PF_PortrayalRule
Attributes:
+ruleName: A name of the rule, this must be unique for the feature portrayal.
+priority: Priority between rules.
+description: A general description of what this portrayal rule does.
+queryStatement: A formal specification expressed in the stated query language which evaluates to true or false.

Associations:
+portrayalAction: A formal specification expressed in the stated action language which involves preprocessing statements and calls to portrayal operations.
+externalFunction: A list of external functions.
+propertyName: A list of property types.

Constraints: None
OCL shall be used in the +queryStatement attribute, and UML to specify the +portrayalAction.

8.3.5 External function

PF_ExternalFunctions are used to perform computations that sometimes are needed to evaluate the query statements and/or perform the portrayal actions of the portrayal rules. See 8.6 for examples.
As long as an external function is declared according to the PF_ExternalFunction class there are no limitations to the operations it can perform or the return types it can have. External functions shall be modeled as operations, as described in ISO 19109. External functions shall not be used in the default portrayal specification.
Class name: PF_ExternalFunction
Attributes:
+functionName: The name of the function. It shall contain no spaces and always start with a letter or underscore character.
+returnType: The return type of the function.
+description: A textual description of the functions behaviour and functionality.

Associations:
+formalParameters: A list of formal parameters that the external function takes.

Constraints: None
EXAMPLE How an external function is used.

In the portrayal catalogue below, we define the feature class that will use an external function by using a rule definition.

pr6: PortrayalRule

```
{
        rule_name="Moving Vehicle",
        priority="1",                          'display with priority one.
        description="Moving automobile used for displaying current location on map",
        query_statement=("TYPE=mvehicle and geometry=POINT"),
        portrayal_action=("Dynamics.Automobile(calcBearing(plat,plong, clat,clong))")
},
```

We then describe the external functions and its parameters needed to invoke the function.

```
External_Function_Sets=
ef2:PF_ExternalFunction
{
        +functionName="calcBearing",
        +returnType="real",
        +description="Determine the bearing based on previous position and new position and return a real number representing the bearing in degrees",
        formal_parameters=
        {
                ad21: AttributeDefinition
                (
                        name="prev_lat_position",
                        description="Previous Latitude in decimal degrees",
                        type="real"
                ),
                ad22: AttributeDefinition
                (
                        name="prev_long_position",
                        description="Previous Longitude in decimal degrees",
                        type="real"
                ),
                ad23: AttributeDefinition
                (
                        name="curr_lat_position",
                        description="Current Latitude in decimal degrees",
                        type="real"
                ),
                ad24: AttributeDefinition
                (
                        name="curr_long_position",
                        description="Current Longitude in decimal degrees",
                        type="real"
                )
        }
}
```

When the rule returns TRUE, the portrayal action is executed and the calcBearing function is invoked.

8.3.6 Attribute definition

PF_AttributeDefinition defines an attribute and is used to define the formal parameters of external functions and the underlying rendering operations of the portrayal service. This is invoked by specifying the name, description, basic type and default value of the attributes.
Class name: PF_AttributeDefinition
Attributes:

+name: Defines the name of the attribute definition. Shall be a legal name.
+description: Describes the usage of this attribute definition.
+type: Identifies the data type of this attribute definition. Shall be of legal basic data type.
+defaultValue: An optional value of the attribute definition.

Associations: None
Constraints: None

8.4 Portrayal specification package

8.4.1 Overview

The PF_PortrayalSpecification package defines the following classes: PF_PortrayalSpecification, PF_PortrayalOperation, PF_ParameterSet and PF_AttributeValue. It uses the PF_AttributeDefinition class defined in the PF_PortrayalCatalogue package. The classes are shown in Figure 8.

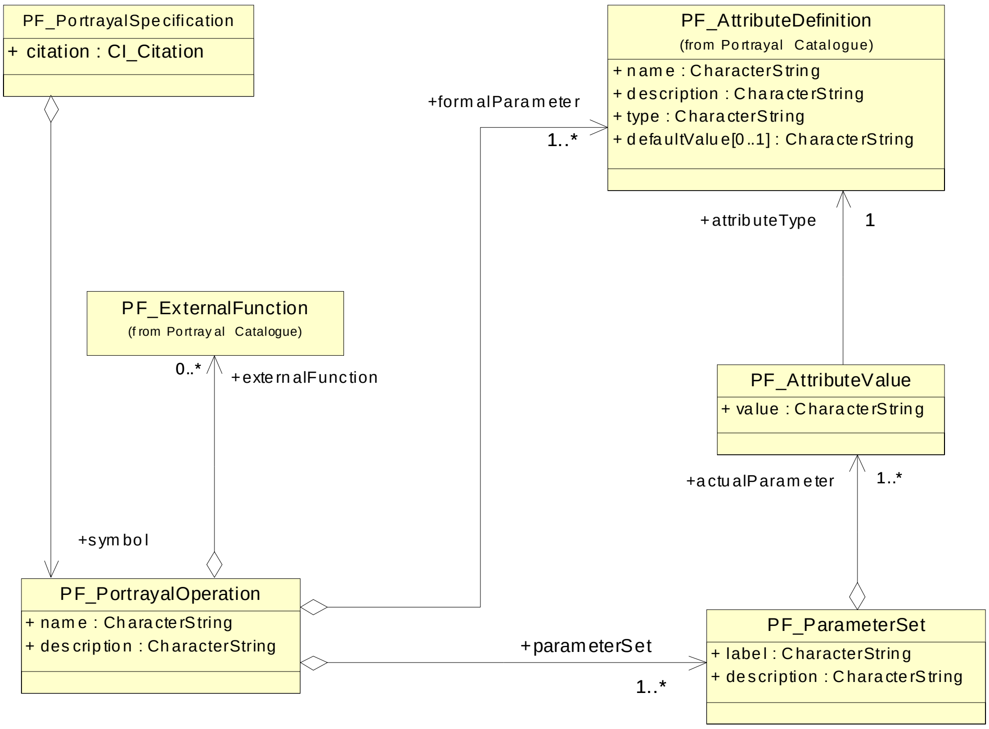

Figure 8 — Portrayal specification package

The PF_PortrayalSpecification is the root object of a portrayal specification, and it provides the header information about the portrayal specification. It also consists of the operations supported by the portrayal specification.
The PF_PortrayalOperation holds the name and a description of the portrayal operation; +portrayGeometry. Further on it contains two important aggregations. The first aggregation contains the formal parameters associated to the operation and is a list of attribute definitions. The second aggregation contains a list of pre-defined parameter sets.
The PF_ParameterSet is responsible for holding a list of actual parameter values and is identified by a label and a description. An instance of a PF_ParameterSet shall only contain actual parameters that are defined in the formal parameter list of its corresponding PF_PortrayalOperation.
The PF_AttributeValue is responsible for storing an actual parameter value. It has an association to its attribute type.

8.4.2 Portrayal specification

PF_PortrayalSpecification holds the instances of the portrayal specification, one for each PF_PortrayalOperation. The spatial types are defined by ISO 19107.
Class name: PF_PortrayalSpecification

Attributes:

+citation:	Citation information.

Associations:

+symbol: Association to a portrayal operation specification.

Constraints: None

8.4.3 Portrayal operation

PF_PortrayalOperation holds the details for a particular portrayal operation. It declares a set of formal parameters that are needed when invoking the underlying rendering functions. It has associations to the formal parameter values for the operation.

There shall be one instance of the portrayal specification class for each operation defined by the portrayal service.

Class name: PF_PortrayalOperation

Attributes:

+name: The name of the operation that is further specified.

+description: A description of the portrayal operations that are given by this object.

Associations:

+formalParameter: A list of the formal parameters that is needed to proper invoke the rendering functions of the underlying portrayal service operation.

+parameterSet: A list of parameter sets used by the operations.

+externalFunction: A list of functions not part of the application schema, used to perform computations that sometimes are needed to evaluate the query statements and / or perform the portrayal actions of the portrayal rules

Constraints:

1) All necessary parameters used by the rendering operation shall be defined by the +formalParameters association.

8.4.4 Parameter set

The PF_ParameterSet holds a list of actual parameter values. It can be thought of as a pre-defined operation. The actual values shall correspond to the formal parameters defined by PF_PortrayalOperation. The PF_ParameterSet has a label that is referred from the portrayal catalogue. It also has a textual description.

PF_ParameterSet objects are used to predefine specific portrayal operations. The label could for example be "thick_red_line" where the actual parameter values are colour=RED, thickness=5, brush=SOLID.

Class name: PF_ParameterSet

Attributes:

+label: A short label used to identify the portrayal specification data from the portrayal catalogue.

+description: A textual description of the role and purpose of this predefined operation.

Associations:

+actualParameter: A list of attribute value objects that hold the values of the actual parameters.

Constraints:

1) The elements of the +actualParameters association shall correspond to the list of +formalParameters defined by the PF_PortrayalOperation. But the order of the elements is not significant.

2) The label shall be unique within a PF_PortrayalOperation.

8.4.5 Attribute value

PF_AttributeValue holds attribute values encoded into a string type. An attribute value always shall be associated with an attribute definition.
Class name: PF_AttributeValue
Attributes:

+value: A value of the data type identified by attributeType.

Associations:

+attributeType: The attribute type of the attribute value.

Constraints: None

8.4.6 Attribute definition

PF_AttributeDefinition is defined in the portrayal catalogue package (see 8.3).

8.5 Complex symbols

External functions (see 8.3.5) can be used to portray geometry that is computed out of other geometry, like point symbols along lines or within surfaces, or to draw complex line types along computed parallels to curves.

EXAMPLE In the portrayal catalogue, we define the feature class that will use an external function by using a rule definition.

```
pr6: PortrayalRule
{
        rule_name="Small city",
        priority="1",                       'display with priority one.
        description="A city that is too small to display as an area",
        query_statement=("FACC_Code=AL020 and geometry=AREA and calcSize(geometry) < 50" ),
        portrayal_action=(" VMAP Level 1.small_city(calcCentroid(geometry))")
},
```

We then describe the external functions and its parameters needed to correctly operate.

```
External_Function_Sets=
Ef1:PF_ExternalFunction
{
        +functionName="calcSize",
        +returnType="real",
        +description="Determine the size of a given surface and return a real number representing the square kilometers",
        formal_parameters=
        {
                ad21: AttributeDefinition
                (
                        name="geometry",
                        description="Area geometry of surface",
                        type="geometry"
                )

        }
},
ef2:PF_ExternalFunction
{
        +functionName="calcCentroid",
        +returnType="real",
        +description="Determine the centroid of a surface return a coordinate value for portrayal and change the geometry type
to point",
        formal_parameters=
        {
                ad21: AttributeDefinition
                (
                        name="geometry",
                        description="geometry of surface",
                        type="geometry"
                )
        }
}
```

In the portrayal specification, the label small city is invoked by the rule and it is passed the new geometry from the external function.

```
ps2: ParameterSet
{
        label=" small_city ", description=" a small city or built up area",
        actual_parameters=
        {
        AttributeValue ( value="http://www.iso.org/vmap1v1/jog/symbols", attribute_type="ad1" ),
                AttributeValue ( value="builtup_area.cgm", attribute_type="ad2" ),
                AttributeValue ( value="color", attribute_type="ad3"),
                AttributeValue ( value="beige", attribute_type="ad4")
        }
},
```

8.6 Portrayal of textual data

Textual data may be handled as an attribute of a feature or by an annotation application schema. The geometric primitives to which a feature makes reference determines how the text may be displayed.

NOTE If the geometric primitive referred to is of type point, then the text can only be related to this point. If the geometric primitive is of type curve the text can be related to the curve (e.g. follow the curve) or to one or more points on the curve. If the geometric primitive is of type surface the text can be related to the surface (e.g. fill the surface), its boundary, or one or more points on a curve defining its boundary.

EXAMPLE The following example (see Figure 9) shows a string of text following a curve. The point at the start of the line defines the start of the string of text. The shape of the line defines the path the text follows.

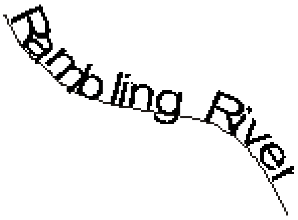

Figure 9 — Example of text following a curve

The character string that defines the textual content can be stored as an attribute of the feature class or it can be associated by reference or generated by rule. For example, a feature class “City” might have the city name stored as an attribute. The text string can also be generated by rule. For example, a sounding depth measurement could be represented as a real number attribute, but be portrayed as text.

8.7 Default portrayal specification

Below is an example of a default portrayal specification.

EXAMPLE Feature classes are defined by the application schema and referenced in the portrayal catalogue.

```
Feature_Portrayal_Sets=
{
        fp1: FeaturePortrayal
        {
                feature_name="AP030",    'curve
                priority="",
                description="Transportation - An open way maintained for vehicular use"   'curve
        },
        fp2: FeaturePortrayal
        {
                feature_name="AL015",    'point
                priority="",
                description="A relatively permanent structure, roofed and usually walled and designed for some particular use"
        },
….
```

The rules necessary to validate the feature instances and select a portrayal specification are defined.

```
Portrayal_Rule_Sets=
{
        pr1: PortrayalRule
        {
                rule_name="Prov_Hwy",
                priority="",
                description="Provincial Highway",
                query_statement=("FACC_CODE='AP030' and exs=28 and rst=1 and rtt=14 and geometry=CURVE"),
                'exs=28 identifies the state or condition of the highway to be operational
                'rst=1 identifies the physical surface composition of the road to be Hard/Paved
                'rtt=14 identifies the intended use of the route to be Primary route
                portrayal_action=("VMAP Level 1.Prov Highway")
                ' format is Portrayal specification name.Parameter Set(external function)
        },
}
…..
```

This is the default part of the portrayal specification.

```
        pr9: PortrayalRule
        {
                rule_name="DefaultCurve",
                priority="",
                description="Any curve that does not have an associated rule will be captured",
                query_statement=("geometry=CURVE"),
                portrayal_action=(“VMAP Level 1.Default Curve")
        },
…..
```

8.8 Representation of symbols

A parameter set (see 8.4.4) shall be used to represent symbols. Symbols shall be referenced by the name of the symbol library they are part of, and the identifier the symbol library uses to identify the symbol within the library. If one or many attributes of the symbol are modified as part of the portrayal process (e.g. scale, rotation, colour), the attribute and attribute value shall be stated.

EXAMPLE The following example shows the specification of a symbol library and the parameter set. In this example only the symbol library and symbol reference is included.

```
ps1: PortrayalSpecification
(
        name="VMAP Level 1",  version="1.0 March 1996", producer="ISOCorp",
        point= o1: PF_PortrayalOperation
        (
                name="portrayPoint", description="Portrays a point on any display",
                formal_parameters=
                {
                        ad1: AttributeDefinition ( name="plib_id", description="Name of point library to use or URL", type="String"),
                        ad2: AttributeDefinition ( name="psym_id", description="Name or ID of Point symbol to use", type="String")

                        parameter_sets=
                        {
                                ps1: ParameterSet
                                (
                                        label='Military_Symbol", description="",
                                        actual_parameters=
                                        {
                                                AttributeValue ( value="http://www.iso.org/vmap1v1/jog/symbols", attribute_type="ad1" ),
                                                AttributeValue ( value="milsymb", attribute_type="ad2" )
                                        }
……
```

Annex ZA
(normative)

Abstract test suite

A.1 Portrayal schema

a) Test purpose: To verify conformance to the portrayal schema.

b) Test method: verify that the portrayal specifications and portrayal rules are not part of the dataset, and that the portrayal specifications are stored separately and referenced from the portrayal rules. The portrayal specifications may be stored externally and referenced using a universal reference standard like URL. Verify that the portrayal rules are stored in a portrayal catalogue, and that the portrayal rules are specified for the feature classes they will be applied on. Verify that the portrayal rules are expressed using the OCL query language.

c) Reference: ISO 19117, 6.1

d) Test type: basic test

A.2 Availability of portrayal information

e) Test purpose: Verify that portrayal information is present.

f) Test method: Verify that portrayal catalogue and portrayal specification are present, or referenced from appropriate metadata.

g) Reference: ISO 19117, 6.1

h) Test type: basic test

A.3 Priority attribute

i) Test purpose: To verify correct use of priority attributes.

j) Test method: Verify that all the portrayal rules have a priority attribute if priority attributes are used. Verify that If two portrayal rules returning TRUE have the same priority value, then the application shall decide which one takes precedence. Verify that all the portrayal rules have a priority attribute if priority attributes are used.

k) Reference: ISO 19117, 6.2

l) Test type: basic test

A.4 Default portrayal specification

m) Test purpose: To verify that a default portrayal specification is present.

n) Test method: Verify that a default portrayal specification is associated with the dataset.

o) Reference: ISO 19117, 6.4.

p) Test type: basic test

A.5 External function

q) Test purpose: To verify correct use of external functions.

r) Test method: Verify that the portrayal catalogue lists the external functions used, including the parameters and returned values.

s) Reference: ISO 19117, 7.3.5

t) Test type: basic test

Annex ZB
(informative)

Examples

A.6 Introduction

All examples are described in the same syntax. The syntax is developed for these examples. An instance of a class is represented according to the following grammar:

- object ::= object-identifier ”:” class-name ”(” elements ”)”
- elements ::= element (”,” element)*
- element-list ::= ”{” elements ”}”
- element ::= element-list | attribute-value | reference | object
- attribute ::= attribute-name ”=” \” character-encoded value \”
- reference ::= role-name ”=” \” object-identifier \”

A.7 Example 1

```
ps1: PortrayalSpecification
{
        name="N50",  version="1.2 1999-03-22", producer="IsoCorp",
        point= o1: Operation
        {
                name="portrayPoint", description="Portrays a point on any display",
                formal_parameters=
                {
                        ad1: AttributeDefinition ( name="colour", description="", type="RGB", default_value="#BLACK"),
                        ad2: AttributeDefinition ( name="width", description="", type="mm", default_value="1"),
                        ad3: AttributeDefinition ( name="token", description="", type="Symbol", default_value="#POINT"),
                        ad4: AttributeDefinition ( name="url", description="", type="HREF", default_value="")
                },
                parameter_sets=
                {
                        ps0: ParameterSet
                        {
                                label="DEFAULT", description="Default point parameter set",
                                actual_parameters=
                                {
                                        AttributeValue ( value="0 0 0", attribute_type="ad1" )
                                }
                        },
                        ps1: ParameterSet
                        {
                                label="White_Dot", description="",
                                actual_parameters=
                                {
                                        AttributeValue ( value="1.0 1.0 1.0", attribute_type="ad1" )
                }
            },
            ps2: ParameterSet
            {
                label="Large_Church_Symbol", description="",
                actual_parameters=
                {
                    AttributeValue ( value="0 0.2 0.7", attribute_type="ad1" ),
                    AttributeValue ( value="50", attribute_type="ad2" ),
                    AttributeValue ( value="church.gif", attribute_type="ad3" )
                }
            },
            ps3: ParameterSet
            {
                label="Small_Church_Symbol", description="",
                actual_parameters=
                {
                    AttributeValue ( value="0.2 0.2 0.2", attribute_type="ad1" ),
                    AttributeValue ( value="25", attribute_type="ad2" ),
                    AttributeValue ( value="church.gif", attribute_type="ad3" )
                }
            },
            ps4: ParameterSet
            {
                label="Red_Dot", description="",
                actual_parameters=
                {
                    AttributeValue ( value="0.9 0 0", attribute_type="ad1" ),
                    AttributeValue ( value="5", attribute_type="ad2" ),
                    AttributeValue ( value="dot.cgm", attribute_type="ad3" )
                }
            }
        }
    }


...


    curve= o2: Operation
    {
        name="portrayCurve", description="Portrays a curve on any display",
        formal_parameters=
        {
            ad5: AttributeDefinition ( name="Colour", description="", type="RGB", default_value="#BLACK"),
            ad6: AttributeDefinition ( name="Width", description="", type="mm", default_value="1")
        }
        parameter_sets=
        {
            ps21: ParameterSet
            {
                label="Red_01mm_line", description="",
                actual_parameters=
                {
                    AttributeValue ( value="0.0 0.9 0.9", attribute_type="ad5" )
                }
            },
            ps22: ParameterSet
            {
                label="Large_Church_Symbol", description="",
                actual_parameters=
                {
                    AttributeValue ( value="#BLUE", attribute_type="ad1" ),
                    AttributeValue ( value="50", attribute_type="ad2" ),
                    AttributeValue ( value="#CHURCH", attribute_type="ad3" )
                }
            }
        }
    }
}
```

A.8 Example 2

Figure B.1 is a small object diagram showing a subset of example 2. It only shows the objects at the top level. The root object is ps1 (PortrayalSpecification) and it has three operation objects. Each of the operations has a list of formal_parameters, which defines the attributes that are associated to the operation and a list of parameter sets that contains the pre-defined labels with actual values.

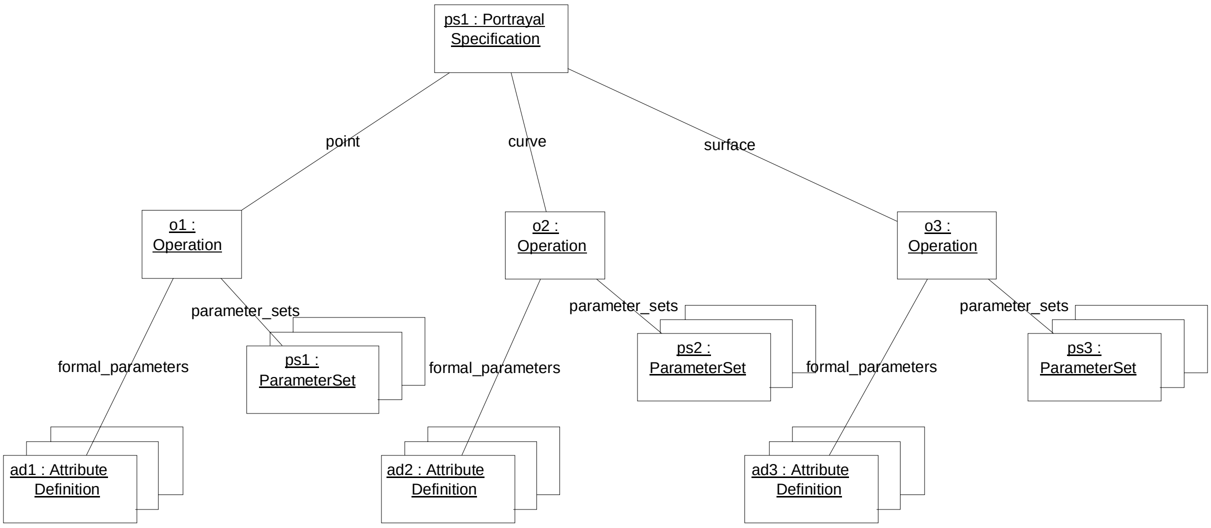

Figure B.1 — Object diagram showing a subset of example 2

```
ps1: PortrayalSpecification
'This is a sample portrayal specification for a VMAP Level 2 product.
'The parameters used here are to portray a ranger station, a generic building and a military symbol.
'
{
'       Section 1
'               Format:
'                       name - name of the portrayal specification
'                       version - version number and date
'                       producer - name, address, telephone number, and/or other contact information
'                                       to reach the producer of the Portrayal Specification

        name="VMAP Level 2",
        version="1.0 March 1996",
        producer="ISOCorp",

'       Section 2
'               Portrayal Operation
'                       Declares a set of formal parameters that are needed when invoking the underlying
'                               rendering functions.
'               Format:
```

```
'                       name - name of the operation that is further specified
'                       description - description of the portrayal operations that are given by this object
'                       formal_parameters - list of formal parameters that is needed to properly invoke the
'                               rendering functions of the underlying Portrayal Service operation
'                       ParameterSet
'                               It holds a list of pointers to Portrayal Specification Data objects that hold
'                                       actual parameter values for the operation
'                       Format:
'                               label - short label used to identify the Portrayal Specification Data from
'                                       the portrayal catalogue
'                               description - textual description of the role and purpose of this pre-defined
'                                       operation
'                               actual_parameters - list of attribute value objects that hold the values of
'                                       the actual parameters
'


        point= 01: Operation
        {
                name="portrayPoint", description="Portrays a point on any display",
                formal_parameters=
                {
                        ad1: AttributeDefinition ( name="plib_id", description="Name of point library to use or URL", type="String"),
                        ad2: AttributeDefinition ( name="psym_id", description="Name or ID of Point symbol to use", type="String")
                        parameter_sets=
                        {
                                ps1: ParameterSet
                                {
                                        label='Military_Symbol", description="",
                                        actual_parameters=
                                        {
                                                AttributeValue ( value="http://www.iso.org/vmaplv2/job/symbols", attribute_type="ad1" ),
                                                AttributeValue ( value="milsymb", attribute_type="ad2" )
                                        }
                                },
                                ps2: ParameterSet
                                {
                                        label="Ranger Station", description="",
                                        actual_parameters=
                                        {
                                                AttributeValue ( value="http://www.iso.org/vmaplv2/job/symbols", attribute_type="ad1" ),
                                                AttributeValue ( value="ranger.cgm", attribute_type="ad2" )
                                        }
                                },
                                ps3: ParameterSet
                                {
                                        label="Black Building", description="",
                                        actual_parameters=
                                        {
                                                AttributeValue ( value="http://www.iso.org/vmaplv2/job/symbols", attribute_type="ad1" ),
                                                AttributeValue ( value="building.cgm", attribute_type="ad2" )
                                        }
                                }
                        }
                }
        },
        curve= 02: Operation
        {
```

```
        name="portrayCurve", description="Portrays a curve on any display",
        formal_parameters=
        {
            ad21: AttributeDefinition ( name="ccolor", description="Curve Color in RGB", type="Sequence(Integers)"),
            ad22: AttributeDefinition ( name="cwidth", description="Curve Width in point size", type="Integer")
        },
        parameter_sets=
        {
        }
    },
    surface= 03: Operation
    {
        name="portraySurface", description="Portrays a surface on any display",
        formal_parameters=
        {
            ad31: AttributeDefinition ( name="scolor", description="Surface Color in RGB", type="Sequence(Integers)"),
            ad32: AttributeDefinition ( name="slib_id", description="Name of stipple pattern library to use or URL", type="String"),
            ad33: AttributeDefinition ( name="spattern_id", description="Name of stipple pattern to use", type="String")
        }
        parameter_sets=
        {
            ps31: ParameterSet
            {
                label="Solid Red Area", description="",
                actual_parameters=
                {
                    AttributeValue ( value="0.9 0 0", attribute_type="ad31" )
                }
            },
            ps32: ParameterSet
            {
                label="Hatch Red Area", description="",
                actual_parameters=
                {
                    AttributeValue ( value="0.9 0 0", attribute_type="ad31" ) ,
                    AttributeValue ( value="http://www.iso.org/hatch", attribute_type="ad32"),
                    AttributeValue ( value="crosshatch.cgm", attribute_type="ad33")
                }
            }
        }
    },
    text= 04: Operation
    {
        name="portrayText", description="Portrays a text on any display",
        formal_parameters=
        {
            ad41: AttributeDefinition ( name="tfont", description="Text Font", type="String"),
            ad42: AttributeDefinition ( name="tstyle", description="Text Style", type="String"),
            ad43: AttributeDefinition ( name="tsize", description="Text Size (pt)", type="String"),
            ad44: AttributeDefinition ( name="tjust", description="Text Justification", type="String"),
            ad45: AttributeDefinition ( name="tcolor", description="Text Color in RGB", type="Sequence(Integer)"),
            ad46: AttributeDefinition ( name="tdist", description="Distance from base (mm)", type="Integer"),
            ad47: AttributeDefinition ( name="tdir", description="Text Direction from base (degrees)", type="Integer")
         },
        parameter_sets=
        {
        }
    }
}
```

A.9 Example 3

A.9.1 Portrayal catalogue

This portrayal catalogue has examples of point, curve, and surfaces. It references two portrayal specifications. The first being the VMAP Level 1 specification and the second being the dynamics specification. Also included are sample definitions of external functions. Not all functions are referenced.

Given that we have this set of features:

Feature 1: Curve with attributes of :

- FACC Code AP030
- exs=28
- rst=1
- rtt=14

Feature 2: Point with attributes of::

- FACC Code AL015
- BFC=14

Feature 3: Surface with attributes of:

- FACC Code BH140
- Txt=”Red River”

Feature 4: Point with attributes of:

- TYPE=mvehicle

Using the portrayal catalogue and portrayal specification, we should end up with:

Feature 1: Rule_name=”Prov_Hwy” and the portrayal specification of “VMAP Level 1.Prov Highway”
Rendering would be a white line with red lines on either side

Feature 2: Rule_name=”Ranger_Station” and the portrayal specification of “VMAP Level 1.Ranger Station”
Rendering would be a gif symbol from the url http://www.iso.org/vmap1lv1/jog/symbols/ranger.cgm

Feature 3: Rule_Name=”Default river/stream” and the Portrayal Specification of “VMAP Level 1.Default river/stream”
Rendering would be Blue fill with outline and the txt Red River would be rendered with the feature

```
pc1: PortrayalCatalogue
'This is a complete example portrayal catalogue containing examples of surface, curve and points
'and the necessary information to use the sample portrayal specification for VMAP level 1.
{
'       Section 1
'               Format:
'                       name - name of the portrayal catalogue
'                       version - version number and date
'                       producer - name, address, telephone number, and/or other contact information
'                                       to reach the producer of the portrayal catalog
'                       query_language - specifies the notation for the queries performed
'                       action_language - specifies the notation for the action performed
        name="VMAP Level 1 Library 041",
        version="1.0 March 1996",
        producer="ISOCorp",
        query_language="SQL2",
        action_language="Language X"    'custom to GIS application XXXX

'       This portrayal catalogue has examples of point, curve, and surfaces.  It references two portrayal specifications.
'       The first being the VMAP Level 1 specification and the second being the dynamics specification.
'       Also included are sample definitions of external functions.  Not all functions are referenced

'       Section 2
'               Feature Portrayal Sets
'                       Refers to a feature class through its feature name attribute.
'                               The feature class shall be described in the application schema.
'                   For each feature class there can be a number of Portrayal rules.
'           Format:
'               feature_name - a reference to the feature class name of a feature class that
'                       is specified in the application schema.
'               priority - priority of portrayal.
'               description - a textual description of the feature class and its portrayal rules.

        Feature_Portrayal_Sets=
        {
            fp1: FeaturePortrayal
            {
                feature_name="AP030",       'curve
                priority="",
                description="Transportation - An open way maintained for vehicular use"   'curve
            },
            fp2: FeaturePortrayal
            {
                feature_name="AL015",       'point
                priority="",
                description="A relatively permanent structure, roofed and usually walled and designed for some particular use"
            },
            fp3: FeaturePortrayal
            {
                feature_name="GB005",       'point
                priority="",
                description="A defined area of land or water used for landing, take-off, and movement of aircraft including
associated buildings, runways and other facilities"
            },
            fp4: FeaturePortrayal
            {
                feature_name="BH140"        'surface
                priority="",
                description="A natural flowing watercourse"
            },
            fp5: FeaturePortrayal
            {
                feature_name="MVehicle"     'point
                    'example of an application specific feature instance being addressed by Portrayal
                priority="1",
                description="A motor vehicle being tracked by GPS"
            },
            fp6: FeaturePortrayal
            {
                feature_name="ZD040",       'point with text as attributes
                priority="1",
                description="A geographic place on the Earth, not normally appearing as a feature instance on a map, but
having a name that is required to be placed on a map"
            }
        }

'       Section 3
'           Portrayal Rule Sets
'               The formal definition of a rule with which to apply to a feature instance to evaluate to true or false
'           Format:
'               rule name - a name of the rule, this must be unique for the Feature Portrayal
'               priority - priority between rules
'               description - a general description of what this portrayal rule does
'               query_statement - a formal specification expressed in the stated query action language which evaluates to true
or false
```

```
'           portrayal_action - a formal specification expressed in the stated action language which involves preprocessing statements
‘           and calls to portrayal operations of the portrayal service

    Portrayal_Rule_Sets=
    {
        pr1: PortrayalRule
        {
            rule_name="Prov_Hwy",
            priority="",
            description="Provincial Highway",
            query_statement=("FACC_CODE='AP030' and exs=28 and rst=1 and rtt=14 and geometry=CURVE"),
                'exs=28 identifies the state or condition of the highway to be operational
                'rst=1 identifies the physical surface composition of the road to be Hard/Paved
                'rtt=14 identifies the intended use of the route to be Primary route
            portrayal_action=("VMAP Level 1.Prov Highway")
                ' format is Portrayal specification name.Parameter Set(external function)
        },
        pr2: PortrayalRule
        {
            rule_name="Ranger_Station",
            priority="",
            description="Building - Ranger Station",
            query_statement=("FACC_CODE='AL015' and BFC=14 and geometry=POINT"),
                'BFC=14 identifies the type or purpose of the building to be Ranger Station
            portrayal_action=("VMAP Level 1.Ranger Station")
        },
        pr3: PortrayalRule
        {
            rule_name="Perennial river/stream",
            priority="",
            description="",
            query_statement=("FACC_CODE='BH140' and hyc=8 and geometry=SURFACE"),
                'hyc=8 identifies the annual water content of the lake to being Perennial
            portrayal_action=("VMAP Level 1.Perennial Water")
        },
        pr4: PortrayalRule
        {
            rule_name="Non-Perennnial river/stream",
            priority="",
            description="Non-Perennial / Intermittent / Fluctuating river or stream.",
            query_statement=("FACC_CODE='BH140" and hyc=6 and geometry=SURFACE"),
                'hyc=6 identifies the annual water content of the lake to being Non-Perennial
            portrayal_action=("VMAP Level 1.NonPerennial Water")
        },
        pr5: PortrayalRule
        {
            rule_name="Default river/stream",
            priority="",
            description="River or stream that does not match any other criteria.",
            query_statement=("FACC_CODE='BH140' and geometry=SURFACE"),
            portrayal_action=("VMAP Level 1.Default river/stream(placeText(txt,font,color,size,steprate))")
        },
        pr6: PortrayalRule
        {
            rule_name="Moving Vehicle",
            priority="1",                       'display with priority one
            description="Moving automobile used for displaying current location on map.",
            query_statement=("TYPE=mvehicle and geometry=POINT"),
            portrayal_action=("Dynamics.Automobile(colour, symbol, calcBearing(plat,plong, clat,clong))")
        },
        pr7: PortrayalRule
        {
            rule_name="Named Location",
            priority="1",
            description="A geographic place on the Earth, not normally appearing as a feature instance on a map, but having a name that is required to be placed on a map."
            query_statement=("FACC_CODE=ZD040 and txt<>" and geometry=POINT"),
                'txt identifies the narritive or other description that is contained in the feature
            portrayal_action("VMAP Level 1.Text(placeText(txt,font,color,size))")
        },
        pr8: PortrayalRule
        {
            rule_name="DefaultPoint",
            priority="",
            description="Any feature instance of type point that does not have an associated rule will be captured.",
            query_statement=("geometry=POINT"),
            portrayal_action=(“VMAP Level 1.Default Point")
        },
        pr9: PortrayalRule
        {
            rule_name="DefaultCurve",
            priority="",
            description="Any feature instance of type curve that does not have an associated rule will be captured.",
            query_statement=("geometry=CURVE"),
            portrayal_action=(“VMAP Level 1.Default Curve")
        },
        pr10: PortrayalRule
        {
            rule_name="DefaultSurface",
            priority="",
            description="Any feature instance of type surface that does not have an associated rule will be captured.",
            query_statement=("geometry=SURFACE"),
            portrayal_action=(“VMAP Level 1.Default Surface")
        }
    }

'   Section 4
'       External Function Sets
'           External functions are used to perform computations that sometimes are needed
'           to evaluate the query statements and/or perform the portrayal actions of the portrayal rules.
'       Format:
'           +functionName - name of the function.
'           +returnType - return type of the function
'           +description - textual description of the functions behavior and functionality
'           formal_parameters - a list of formal parameters that the external function takes

    External_Function_Sets=
    {
        ef1:PF_ExternalFunction
        {
            +functionName="calc_daynight",
            +returnType="Integer",
            +description="Calculate whether is is daytime or nighttime to determine what type of display to use return 1 for day -1 for night",
            formal_parameters=
            {
                ad11: AttributeDefinition (name="time_of_day", description="Time of day taken from system clock",type="real")
            }
```

```
                },
                ef2:PF_ExternalFunction
                {
                        +functionName="calcBearing",
                        +returnType="real",
                        +description="Determine the bearing based on previous position and new position and return a real number
representing the bearing in degrees",
                        formal_parameters=
                        {
                                ad21: AttributeDefinition (name="prev_lat_position", description="Previous Latitude in decimal degrees",type="real"),
                                ad22: AttributeDefinition (name="prev_long_position", description="Previous Longitude in decimal
degrees",type="real"),
                                ad23: AttributeDefinition (name="curr_lat_position", description="Current Latitude in decimal degrees",type="real"),
                                ad24: AttributeDefinition (name="curr_long_position", description="Current Longitude in decimal
degrees",type="real")
                        }
                },
                ef3:PF_externalFunction
                {
                        +functionName="placeText",
                        +returnType="sequence(real)",
                                'return
                        +description="extracts the text from a feature instance and returns a location for placement"
                        formal_parameters=
                        {
                                ad31: AttributeDefinition (name="attributes", description="Attributes that need to be passed for the
rendering of the text",type="Sequence(String)")
                        }
                }
        }
}
```

A.9.2 Portrayal specification – VMAP Level 1

```
ps1: PortrayalSpecification
'This is a sample portrayal specification for a VMAP Level 1 product.
'The parameters used here are used by the portrayal catalogue for VMAP level 1 library 041.
{
'       Section 1
'               Format:
'                       name - name of the portrayal specification
'                       version - version number and date
'                       producer - name, address, telephone number, and/or other contact information
'                                       to reach the producer of the Portrayal Specification

        name="VMAP Level 1",
        version="1.0 March 1996",
        producer="ISOCorp",

'       Section 2
'               Portrayal Operation
'                       Declares a set of formal parameters that are needed when invoking the underlying
'                               rendering functions.
'               Format:
'                       name - name of the operation that is further specified
'                       description - description of the portrayal operations that are given by this object
'                       formal parameters - list of formal parameters that is needed to properly invoke the
'                               rendering functions of the underlying Portrayl Service operation
```

```
            parameter_sets - list of parameter sets used by the operations

    point= 01: Operation
    {
        name="portrayPoint", description="Portrays a point on any display",
        formal_parameters=
        {
            ad1: AttributeDefinition ( name="plib_id", description="Name of point library to use or URL", type="String"),
            ad2: AttributeDefinition ( name="psym_id", description="Name or ID of Point symbol to use", type="String"),
            ad3: AttributeDefinition ( name="pattr_id",  description="The attribute of the symbol to modify",type="String"),
            ad4: AttrbiuteDefinition ( name="pattr_value", description="Value to modify", type="String")
        }
    },
    curve= 02: Operation
    {
        name="portrayCurve", description="Portrays a curve on any display",
        formal_parameters=
        {
            ad21: AttributeDefinition ( name="ccolor", description="Curve Color in RGB", type="Sequence(Integers)"),
            ad22: AttributeDefinition ( name="cwidth", description="Curve Width in point size", type="Integer"),
            ad23: AttributeDefinition ( name="cxoffset", description="X offset in point size", type="real"),
            ad24: AttributeDefinition ( name="cyoffset", description="Y offset in point size", type="real"),
            ad25: AttributeDefinition ( name="linetype", description="line type", type="string")
        },
    },
    surface= 03: Operation
    {
        name="portraySurface", description="Portrays a surface on any display",
        formal_parameters=
        {
            ad31: AttributeDefinition ( name="scolor", description="Surface Color in RGB", type="Sequence(Integers)"),
            ad32: AttributeDefinition ( name="slib_id", description="Name of stipple pattern library to use or URL",
type="String"),
            ad33: AttributeDefinition ( name="spattern_id", description="Name of stipple pattern to use", type="String")
        }
    },

'   Section 3
'       ParameterSet
'           It holds a list of pointers to Portrayal Specification Data objects that hold
'               actual parameter values for the operation.
'       Format:
'           label - short label used to identify the Portrayal Specification Data from the
'               portrayal catalogue
'           description - textual description of the role and purpose of this pre-defined operation
'           actual_parameters - list of attribute value objects that hold the values of the
'               actual parameters
'
'   This section assigns values for representation of sample point, curve, surface and text.
'   It also shows that one feature instance might be represented using multiple geometry's such as a surface
'   that has a curve outline. Also, points could be rendered with multiple point symbols to represent
'   a complex features.

    parameter_sets=
    {
        ps1: ParameterSet
        {
            label="Ranger Station", description="",
            actual_parameters=
            {
                    AttributeValue ( value="http://www.iso.org/vmap1v1/jog/symbols", attribute_type="ad1" ),
                    AttributeValue ( value="ranger.cgm", attribute_type="ad2" )
            }
        },
        ps2: ParameterSet
        {
            label="Black Building", description="",
            actual_parameters=
            {
                    AttributeValue ( value="http://www.iso.org/vmap1v1/jog/symbols", attribute_type="ad1" ),
                    AttributeValue ( value="building.cgm", attribute_type="ad2" ),
                    AttributeValue ( value="color", attribute_type="ad3"),
                    AttributeValue ( value="black", attribute_type="ad4")
            }
        },
        ps21: ParameterSet
        {
            label="Prov_Highway", description="Provincial Highway - white line with red on either side.",
            actual_parameters=
            {
                    AttributeValue ( value="255 255 255", attribute_type="ad21"),
                    AttributeValue ( value="2", attribute_type="ad22"),
                    AttributeValue ( value="255 0 0", attribute_type="ad21"),
                    AttributeValue ( value="1", attribute_type="ad22"),
                    AttributeValue ( value="1", attribute_type="ad23"),
                    AttributeValue ( value="255 0 0", attribute_type="ad21"),
                    AttributeValue ( value="2", attribute_type="ad22"),
                    AttributeValue ( value="-1", attribute_type="ad23")
            }
        },
        ps31: ParameterSet
        {
            label="Solid Red Area", description="",
            actual_parameters=
            {
                    AttributeValue ( value="255 0 0", attribute_type="ad31" )
            }
        },
        ps32: ParameterSet
        {
            label="Hatch Red Area", description="",
            actual_parameters=
            {
                    AttributeValue ( value="0.9 0 0", attribute_type="ad31" ) ,
                    AttributeValue ( value="http://www.iso.org/hatch", attribute_type="ad32"),
                    AttributeValue ( value="crosshatch.cgm", attribute_type="ad33")
            }
        },

        'Here we have a feature of type surface that is being portrayed with a surface and a curve.
        ps33: ParameterSet
        {
            label="Lakes", description="Solid blue area with outline in black",
            actual_parameters=
            {
                    AttributeValue ( value="0 0 255", attribute_type=ad31"),
                    AttributeValue ( value="0 0 0", attribute_type="ad21"),
                    AttributeValue ( value="1", attribute_type="ad22")
            }
        }
```

```
                ps34: ParameterSet
                {
                        label="Perennial Water", description="blue solid fill with black dash outline",
                        actual_parameters=
                        {
                                AttributeValue ( value="0 0 255", attribute_type="ad31"),
                                AttributeValue ( value="0 0 0", attribute_type="ad21"),
                                AttributeValue ( value="dash", attribute_type="ad25")
                        }
                }
                ps35: ParameterSet
                {
                        label="NonPerennial Water", description="blue stipple dot pattern fill with outline",
                        actual_parameters=
                        {
                                AttributeValue ( value="0 0 255", attribute_type="ad31"),
                                AttributeValue ( value="c:\vmappatterns", attribute_type="ad32"),
                                AttributeValue ( value="stippledot.cgm", attribute_type="ad33"),
                                AttributeValue ( value="0 0 255", attribute_type="ad21")
                        }
                }
                ps36: ParameterSet
                {
                        label="Default river/stream", description="blue fill with outline",
                        actual_parameters=
                        {
                                AttributeValue ( value="0 0 255", attribute_type="ad31"),
                                AttributeValue ( value="0 0 255", attribute_type="ad21")
                        }
                }
        }
}
```

A.9.3 Portrayal specification – Dynamic specification

```
ps1: PortrayalSpecification
'This is the portrayal specification for the VMAP Level 1 portrayal catalogue.
'It contains the dynamic objects that will be portrayed.
{
'       Section 1
'               Format:
'                       name - name of the portrayal specification
'                       version - version number and date
'                       producer - name, address, telephone number, and/or other contact information
'                                       to reach the producer of the Portrayal Specification
        name="Dynamics",
        version="1.1 1999-03-22",
        producer="ISOCorp",

'       Section 2
'               Portrayal Operation
'                       Declares a set of formal parameters htat are needed when invoking the underlying
'                               rendering functions.
'               Format:
'                       name - name of the operation that is further specified
'                       description - description of the portrayal operations that are given by this object
'                       formal parameters - list of formal parameters that is needed to properly invoke the
'                               rendering functions of the underlying Portrayl Service operation
'                       ParameterSet
'                               It holds a list of pointers to Portrayal Specification Data objects that hold
'                   actual parameter values for the operation
'           Format:
'               label - short label used to identify the Portrayal Specification Data from the portrayal catalogue
'               description - textual description of the role and purpose of this pre-defined operation
'               actual_parameters - list of attribute value objects that hold the values of the actual parameters

    point= 01: Operation
    {
        name="portrayPoint", description="Portrays a point on any display",
        formal_parameters=
        {
            ad1: AttributeDefinition ( name="colour", description="", type="RGB", default_value="#BLACK"),
            ad2: AttributeDefinition ( name="symbol", description="", type="Symbol", default_value="#POINT"),
            ad3: AttributeDefinition ( name="rotation", description="", type="Real", default_value="0")
        },
        parameter_sets=
        {
            ps0: ParameterSet
            {
                label="DEFAULT", description="Default point parameter set",
                actual_parameters=
                {
                    AttributeValue ( value="0 0 0", attribute_type="ad1" )
                }
            },
            ps1: ParameterSet
            {
                label="Automobile", description="Moving Automobile showing direction of movement",
                actual_parameters=
                {
                    AttributeValue ( value="0 0 0", attribute_type="ad1" ),
                    AttributeValue ( value=”auto.gif", attribute_type="ad2"),
                    AttributeValue ( value="bearing", attribute_type="ad3")
                }
            }
        }
    }
}
```

A.10 Example 4

A.10.1 UML model of the application schema

Figure B.2 contains the application schema applied for this example.

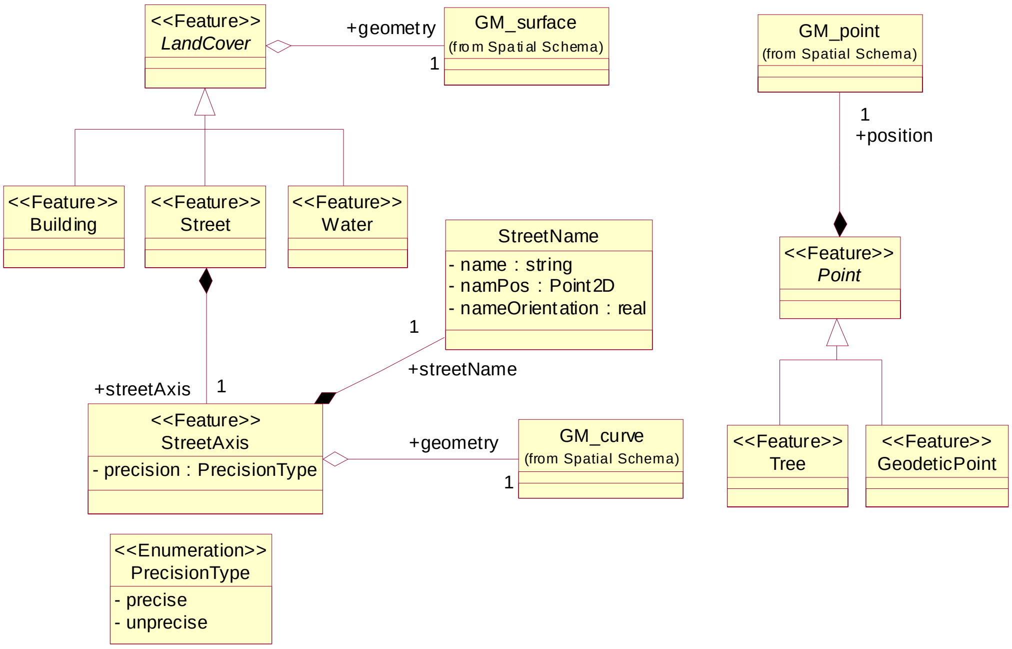

Figure B.2 — Example application schema

A.10.2 Portrayal catalogue

```
pc1: PortrayalCatalogue
'This is an example portrayal catalogue
{

  FeaturePortrayal_Sets=
  {
    fp1: FeaturePortrayal
    {
      featureName="StreetAxis"
      priority="100"
      description=""

      PortrayalRule_Sets=
     {
           'RULES FOR STREETAXIS
           pr1: PortrayalRule
           {
             ruleName="StreetPrecise"
             priority="100"
             description"Use a solid line for precise street"
             queryStatement=("FACC_CODE='StreetAxis' and StreetAxis.precision='precise'")
             portrayalAction=("ISO_STYLE.LINE_STYLE.Solid")
           }
           pr2: PortrayalRule
           {
             ruleName="StreetUnprecise"
             priority="100"
             description"Use a dotted line for unprecise street"
             queryStatement=("FACC_CODE='StreetAxis'
                                 and StreetAxis.precision='unprecise'
                                 and Scale <= 500 ") 'Scale is external function !???
             portrayalAction=("ISO_STYLE.LINE_STYLE.Dotted")
           }
      } ’end PortrayalRule_Sets
    } ’end FeaturePortrayal

      PortrayalRule_Sets=
     {
           fp1: FeaturePortrayal
          {
            featureName="Point"
            priority="110"
            description=""

           'RULES FOR POINT FEATURES
           pr3: PortrayalRule
           {
             ruleName="Tree"
             priority="110"
             description"Use Tree symbol"
             queryStatement=("FACC_CODE='Tree'")
             portrayalAction=("ISO_STYLE.SYMBOL_STYLE.Tree")
           }
           pr4: PortrayalRule
           {
             ruleName="GeodeticPoint"
             priority="110"
             description"Use GeodeticPoint symbol"
             queryStatement=("FACC_CODE='GeodeticPoint'")
             portrayalAction=("ISO_STYLE.SYMBOL_STYLE.GP")
           }
      } 'end PortrayalRule_Sets
  } 'end FeaturePortrayal_Sets
} 'end PortrayalCatalogue
```

A.10.3 Portrayal specification

```
ps1: PortrayalSpecification
'This is an example Portrayal Specification
{

  text= o1: PortrayalOperation
   {
     name="", description=""
     formalParameters=
     {
          ad1: AttributeDefinition (name="TXT", description="", type="String", defaultValue=""),
          ad2: AttributeDefinition (name="GEOMETRY", description="", type="C2",
                                   defaultValue=""),
          ad3: AttributeDefinition (name="ROT", description="", type="P_ANGLE",
                                   defaultValue="0.0"),
          ad4: AttributeDefinition (name="HALI", description="", type="HALIGNMENT",
                                   defaultValue="Center"),
          ad5: AttributeDefinition (name="VALI", description="", type="VALIGNMENT",
                                   defaultValue="Half"),
            ad6: AttributeDefinition (name="Style", description="additional parameters",
                                        type="P_TEXTID", defaultValue="")
   },

   parameterSets=
   {
   } 'end parameterSets
} 'end PortrayalOperation

symbol= o2: PortrayalOperation
{
   name="", description=""
   formalParameters=
   {
            ad7: AttributeDefinition (name="GEOMETRY", description="", type="C2",
                                        defaultValue=""),
            ad8: AttributeDefinition (name="ROT", description="", type="P_ANGLE",
                                        defaultValue="0.0"),
            ad9: AttributeDefinition (name="Style", description="additional parameters",
                                        type="P_SYMBID", defaultValue="")
   },

} 'end PortrayalOperation

curve= o3: PortrayalOperation
{
   name="", description=""
   formalParameters=
   {
            ad10: AttributeDefinition (name="GEOMETRY", description="", type="P_LIN",
                                        defaultValue=""),
            ad11: AttributeDefinition (name="Style", description="additional parameters",
                                        type="P_LINEID", defaultValue="")
   },

} 'end PortrayalOperation

surface= o4: PortrayalOperation
{
   name="", description=""
   formalParameters=
   {
            ad12: AttributeDefinition (name="GEOMETRY", description="", type="P_SURF",
                                        defaultValue=""),
            ad13: AttributeDefinition (name="HatchAng", description="additional parameters",
                                        type="P_ANGLE", defaultValue=""),
            ad14: AttributeDefinition (name="HatchOrg", description="", type="C2", defaultValue=""),
            ad15: AttributeDefinition (name="Style", description="", type="P_SURFID",
                                        defaultValue="")
   },

} 'end PortrayalOperation

textStyle= o5: PortrayalOperation
{
   name="", description=""
   formalParameters=
   {
            ad16: AttributeDefinition (name="Name", description="", type="P_TEXTID",
                                        defaultValue=""),
          ad17: AttributeDefinition (name="FontName", description="", type="String",
                                    defaultValue=""),
          ad18: AttributeDefinition (name="Color", description="", type="String", defaultValue="")
  },

  parameterSets=
  {
          ps0: ParameterSet
          {
            label="Arial", description="",
            actualParameters=
            {
                  AttributeValue( value="text1", attributeType="ad16")
           AttributeValue( value="Arial", attributeType="ad17")
         AttributeValue( value="black", attributeType="ad18")
            }
          }
  } 'end parameterSets
} 'end PortrayalOperation

symbolStyle= o6: PortrayalOperation
{
  name="", description=""
  formalParameters=
  {
          ad19: AttributeDefinition (name="Name", description="", type="P_SYMBID",
                                    defaultValue=""),
          ad20: AttributeDefinition (name="Geometry", description="", type="Bag(SYMBOL_GEOMETRY)",
                                    defaultValue=""),
          ad21: AttributeDefinition (name="Color", description="", type="String", defaultValue="")
  },

  parameterSets=
  {
          ps1: ParameterSet
          {
            label="Tree", description="",
            actualParameters=
            {
                  AttributeValue( value="Tree", attributeType="ad19")
           AttributeValue( value="<Polyline>...</Polyline>", attributeType="ad20")
                  AttributeValue( value="green", attributeType="ad21")
            }
          }

          ps2: ParameterSet
          {
            label="GP", description="",
            actualParameters=
            {
                  AttributeValue( value="GP", attributeType="ad19")
           AttributeValue( value="<Polyline>...</Polyline>", attributeType="ad20")
                  AttributeValue( value="red", attributeType="ad21")
            }
          }
  } 'end parameterSets
} 'end PortrayalOperation

lineStyle= o7: PortrayalOperation
{
  name="", description=""
  formalParameters=
  {
          ad22: AttributeDefinition (name="Name", description="", type="P_LINEID",
                                  defaultValue=""),
          ad23: AttributeDefinition (name="LineStyle", description="", type="Integer",
                                  defaultValue=""),
          ad24: AttributeDefinition (name="Color", description="", type="String", defaultValue="")
  },

  parameterSets=
  {
          ps3: ParameterSet
          {
            label="Solid", description="",
            actualParameters=
            {
                    AttributeValue( value="Solid", attributeType="ad22")
           AttributeValue( value="0", attributeType="ad23")
                    AttributeValue( value="black", attributeType="ad24")
            }
          }

          ps4: ParameterSet
          {
            label="Dotted", description="",
            actualParameters=
            {
                    AttributeValue( value="Dotted", attributeType="ad22")
         AttributeValue( value="1", attributeType="ad23")
                    AttributeValue( value="black", attributeType="ad24")
            }
          }

  } 'end parameterSets
} 'end PortrayalOperation

surfaceStyle= o8: PortrayalOperation
{
  name="", description=""
  formalParameters=
  {
          ad25: AttributeDefinition (name="Name", description="", type="P_SURFID",
                                  defaultValue=""),
          ad26: AttributeDefinition (name="BorderStyle", description="", type="Integer",
                                  defaultValue=""),
          ad27: AttributeDefinition (name="FillColor", description="", type="String", defaultValue="")
  },

  parameterSets=
  {
          ps5: ParameterSet
          {
            label="surf1", description="",
            actualParameters=
            {
                    AttributeValue( value="surf1", attributeType="ad25")
           AttributeValue( value="0", attributeType="ad26")
                    AttributeValue( value="gray", attributeType="ad27")
            }
            }
        } 'end parameterSets
    } 'end PortrayalOperation

} 'end PortrayalSpecification
```