

Reference number of working document: ISO/TC 211 N ?????

Date: 2013-02-20

Reference number of document: ISO/WD 19160-1

Committee identification: ISO/TC 211/WG 7

Secretariat: Norway

Addressing — Part 1: Conceptual model

Adressage —Partie 1: Modèle conceptuel

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: International standard

Document stage: (10) Propopsal

Document language: E

Copyright notice

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

*[Indicate :
the full address
telephone number
fax number
telex number
and electronic mail address*

as appropriate, of the Copyright Manager of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the draft has been prepared]

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Contents

Page

Foreword	vi
Introduction.....	vii
1 Scope.....	1
2 Conformance	1
2.1 Core	1
2.2 Lifecycle.....	1
2.3 Provenance.....	1
2.4 Locale.....	1
2.5 Full conformance	1
3 Normative references	2
4 Terms and definitions.....	2
5 Symbols and abbreviations	4
6 Address model	5
6.1 Overview.....	5
7 Requirements class: Core.....	10
7.1 Dependencies	10
7.2 Core requirement 1: Classes.....	10
7.3 Core requirement 2: Associations.....	11
7.4 Core requirement 3: Attributes	12
7.5 Codelists	15
8 Requirements class: Lifecycle	16
8.1 Dependencies	16
8.2 Lifecycle requirement 1: Attributes.....	16
8.3 Codelists	18
9 Requirements class: Provenance	18
9.1 Dependencies	18
9.2 Provenance requirement 1: Attributes.....	19
10 Requirements class: Locale	20
10.1 Dependencies	20
10.2 Locale requirement 1: Attributes	20
Annex A (normative) Abstract test suites	22
A.1 Introduction.....	22
A.2 Conformance class: Core	22
A.2.1 Introduction	22
A.2.2 Core test 1: Classes	22
A.2.4 Core test 2: Associations	22
A.2.6 Core test 3: Attributes.....	22
A.3 Conformance class: Lifecycle.....	23
A.3.1 Introduction	23
A.3.2 Lifecycle test 1: Attributes.....	23
A.4 Conformance class: Provenance.....	23
A.4.1 Introduction	23
A.4.2 Provenance test 1: Attributes.....	23
A.5 Conformance class: Locale.....	23
A.5.1 Introduction	23
A.5.2 Locale test 1: Attributes.....	23

Annex B (informative) Profiles and instance data	25
B.1 Profiles	25
B.1.1 Introduction	25
B.1.2 Steps followed to develop a profile.....	25
B.1.3 Profile diagrams	26
B.2 Minimal profile	27
B.2.1 Introduction	27
B.2.2 Diagrams	27
B.2.3 Instance data for the minimal profile	29
B.3 <name> profile	30
B.3.1 Introduction	30
B.3.2 Diagrams	30
B.3.3 Attribute mapping from <specification> to the <name> profile	31
B.3.4 Attribute mapping from the <name> profile to <specification>	31
B.3.5 Instance data for the <name> profile	32
B.4 Australian profile.....	33
B.4.1 Introduction	33
B.4.2 Diagrams	34
B.4.3 Attribute mapping from AS 4590:2006 to the Australian profile	37
B.4.4 Attribute mapping from the Australian profile to AS 4590:2006	41
B.4.5 Instance data for the Australian profile	47
B.5 Cultural heritage profile.....	49
B.6 INSPIRE profile	50
B.6.1 Introduction	50
B.6.2 Diagrams	51
B.6.3 Attribute mapping from the INSPIRE Addresses model to the INSPIRE profile	56
B.6.4 Attribute mapping from the INSPIRE profile to INSPIRE Addresses model	57
B.6.5 Instance data for the <name> profile	60
B.7 Japanese profile	60
B.8 New Zealand profile	61
B.8.1 Introduction	61
B.8.2 Diagrams	62
B.8.3 Provenance conformance	71
B.9 South African profile.....	72
B.9.1 Introduction	72
B.9.2 Diagrams	73
B.9.3 Attribute mapping from SANS 1883-1:2009 to the South African profile	77
B.9.4 Attribute mapping from the South African profile to SANS 1883-1:2009	78
B.9.5 Instance data for the South African profile	78
B.10 South Korean profile.....	79
B.10.1 Introduction	79
B.10.2 Diagrams	80
B.10.3 Attribute mapping from <specification> to the South Korean profile	84
B.10.4 Attribute mapping from the South Korean profile to <specification>	84
B.10.5 Instance data for the <name> profile	85
B.11 United Kingdom profile	89
B.12 UPU profile.....	91
B.11 US FGDC profile	98
B.13.1 Introduction	98
B.13.2 Diagrams	99
B.13.3 US FGDC Mapping Tables	104
Annex C (informative) Examples: Life cycle information.....	108
C.1 Life cycle stages of an address component	108
C.2 Life cycle stages of an address	110
C.3 Life cycle requirement 1: Life cycle stage	110
C.4 Life cycle requirement 2: Address component	110
C.5 Life cycle requirement 3: Address	112
C.6 Life cycle requirement 4: Addressable object	114
Annex D (informative) Examples: Identical addresses and address aliases	115

Annex E (informative) Ontology for the ISO 19160-1 conceptual model	116
F.1 ISO 19160-1 conceptual model represented in OWL (RDF XML)	116
F.2 ISO 19160-1 conceptual model represented in OWL (Turtle)	116
Annex F (informative) Examples: ReferenceObject in a profile	117
F.1 Introduction	117
Bibliography	118

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 2.

The main task of technical committees is to prepare International Standards. 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 document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 19160-1 was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics*.

Introduction

Addresses are one of the most common ways of referencing an object for purposes of identification and location. Address assignment schemes vary from country to country: in many Euro-centric countries reference to a road network in the address is common, while addresses in countries such as Japan and South Korea (though South Korea is moving away from this) comprise a hierarchy of administrative areas without reference to a thoroughfare. In the field of intelligent transport systems, an address can be considered as a simplified (as opposed to coordinates) location system, in which case points of interest and postcodes are addressing information applicable in car navigation. Addresses are used for a wide variety of purposes: postal delivery, emergency response, customer relationship management, land administration, utility planning and maintenance, to name a few.

There are many stakeholders involved in addressing: for assigning the address (local governments, postal operators, etc.); for using addresses in various ways (customer service providers and electronic business, local and national governments, utility service providers, election commissions, etc.); and finding the address (citizens, delivery and emergency response service providers, etc.). Stakeholders were identified during the preparatory work of the stage zero project and are now either involved or aware of the development of the ISO 19160 addressing standards.

A variety of address standards and/or specifications are in use around the world, described in the report of the preparatory work for this standard, the *Review summary of the ISO 19160 stage zero project* (2011). These standards and specifications are well integrated into various operational processes and in some cases legally enforced. At the same time, some countries are rationalising their addressing system or creating a new one. Addresses are increasingly used to reference new geographic objects, while they are also increasingly used in new technology such as in-vehicle navigation. The goal of this international standard is to facilitate interoperability between existing and future address specifications.

Revise this paragraph? (28/29 November teleconference)

This international standard aims to facilitate interoperability between address specifications, for example, in the cross-mapping of conceptual models between different address specifications; the conversion of address data from one specification to another; the geocoding of address data by matching address data (without coordinates) in one specification to address data with coordinates in another specification; and the linking of historic address data in one specification to current address data in another specification.

The preparatory work for this project is described in *Review summary of the ISO 19160 stage zero project* (2011) and recommended five projects with the following titles:

- *Addressing – Conceptual model*
- *Addressing – Good practices for address assignment schemes*
- *Addressing – Quality management for address data*
- *Addressing – International postal address components and templates*
- *Addressing – Address rendering for purposes other than mail*

This international standard implements the first of these recommendations.

Addressing — Part 1: Conceptual model

1 Scope

This International Standard defines a conceptual model for address information (address model), together with the terms and definitions that describe the concepts in the model. Lifecycle, metadata and address aliases are included in the conceptual model. The model is presented in the Unified Modeling Language (UML) and the Web Ontology Language (OWL).

The model provides a common representation of address information, independent of actual addressing implementations. It is not intended to replace conceptual models proposed in other specifications, but provides a means to cross-map between different conceptual models for address information and enables the conversion of address information between specifications.

The model provides a basis for developing address specifications by individual countries or communities.

2 Conformance

This International Standard defines five classes of requirements and conformance.

2.1.1 Core

Any address model for which Core conformance is claimed shall pass all the requirements described in the abstract test suite given in clause A.2.

2.1.2 Lifecycle

Any address model for which Lifecycle conformance is claimed shall pass all the requirements described in the abstract test suite given in clause A.3.

2.1.3 Provenance

Any address model for which Provenance conformance is claimed shall pass all the requirements described in the abstract test suite given in clause A.4.

2.1.4 Locale

Any address model for which Locale conformance is claimed shall pass all the requirements described in the abstract test suite given in clause A.5.

2.1.5 Full conformance

Any address model for which Full conformance is claimed shall pass all the requirements described in the abstract test suites specified for the Core, Lifecycle, Provenance and Locale conformance classes.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 19115-1:____¹, *Geographic information – Metadata – Part 1: Fundamentals*

ISO 19152:2012, *Geographic information – Land administration domain model (LADM)*

ISO 19501:2005, *Information technology -- Open Distributed Processing -- Unified Modeling Language (UML) Version 1.4.2*

ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1 address

structured information that uniquely references an object for purposes of identification and location

NOTE 1 The object is identifiable in the real world and persistent (i.e. virtual objects are excluded).

NOTE 2 There can be many addresses for an object, but one address references a single object.

NOTE 3 Two addresses based on two different address specifications (i.e. they have different sets of components) for the same addressable object are two different addresses. Refer to [Annex D](#) for more examples.

NOTE 4 If there are two addresses based on the same address specification but in two different languages, the model needs to specify whether these are two representations for the same address or two different addresses.

EXAMPLE 1 Address where the object that is referenced is a person's residence: *99 Lombardy Street, The Hills, 0039, South Africa*

EXAMPLE 2 Address where the object that is referenced is a building: *Lombardy House, 809 Lombardy Street, The Hills, 0039, South Africa*

EXAMPLE 3 Address where the object that is referenced is a door (apartment) in a building: *Room 4-6, Lombardy House, 809 Lombardy Street, The Hills, 0039, South Africa*

4.2 addressable object

object that may be referenced by an **address**

4.3 addressee

party who is the ultimate recipient of a delivery item

[adapted from UPU S42:2006]

NOTE 1 A delivery item may be a letter, parcel, couriered item or any other delivered goods.

¹ To be published

NOTE 2 A party is one or more natural or legal persons or organisations acting as a single recipient of a delivery [adapted from UPU S42:2006].

4.4

address alias

one of a set of **addresses** referencing the same **addressable object**

4.5

address class

addresses combined from the same list of **address components** according to the same **address reference system**

EXAMPLE 1 '25 Blue Avenue Hatfield 0028' and '384 Green Street Motherville 2093' belong to the same address class.

EXAMPLE 2 'PO Box 765 Goodwood 33948' and 'PO Box 567 Grayville 98373' belong to the same address class.

4.6

address component

constituent part of the **address**

NOTE 1 The address component may reference another object, such as a spatial object (e.g. an administrative boundary or a land parcel) or a non-spatial object, such as an organisation or a person.

NOTE 2 If there are two address components in two different languages, the model needs to specify whether these are two representations for the same address component or two different addresses components.

4.7

address position

position representing the **address**

4.8

address reference system

system that defines how address components are combined to form an **address**

4.9

child address

address defined relative to a **parent address**

4.10

child addressable object

addressable object that is addressed relative to another **addressable object**

EXAMPLE 1 An apartment within an apartment building

EXAMPLE 2 In Japan, a *jukyo bango* (residence number) within a *gaiku* (block).

4.11

locale

cultural and linguistic setting applicable to the interpretation of a character string

[ISO 19135:2005]

4.12

parent address

address of a **parent addressable object**

NOTE The address components of a parent address are fully inherited by addresses of its child addressable objects.

4.13

parent addressable object

addressable object that fully encloses one or more other **addressable objects**

EXAMPLE 1 An apartment building with many apartments within.

EXAMPLE 2 In Japan, a *gaiku* (block) with many *jukyo bango* (residence number).

4.14

spatial object

object used for representing a spatial characteristic of a feature

[ISO 19107:2003]

5 Symbols and abbreviations

OWL	Web Ontology Language
UML	Unified Modeling Language
URI	Uniform resource identifier
URL	Uniform resource locator
URN	Uniform resource name

6 Address model

6.1 Overview

Figure 1 gives an overview of the classes, attributes and associations (relationships) in the address model. The requirements classes corresponding to the different conformance class are presented in subsequent clauses.

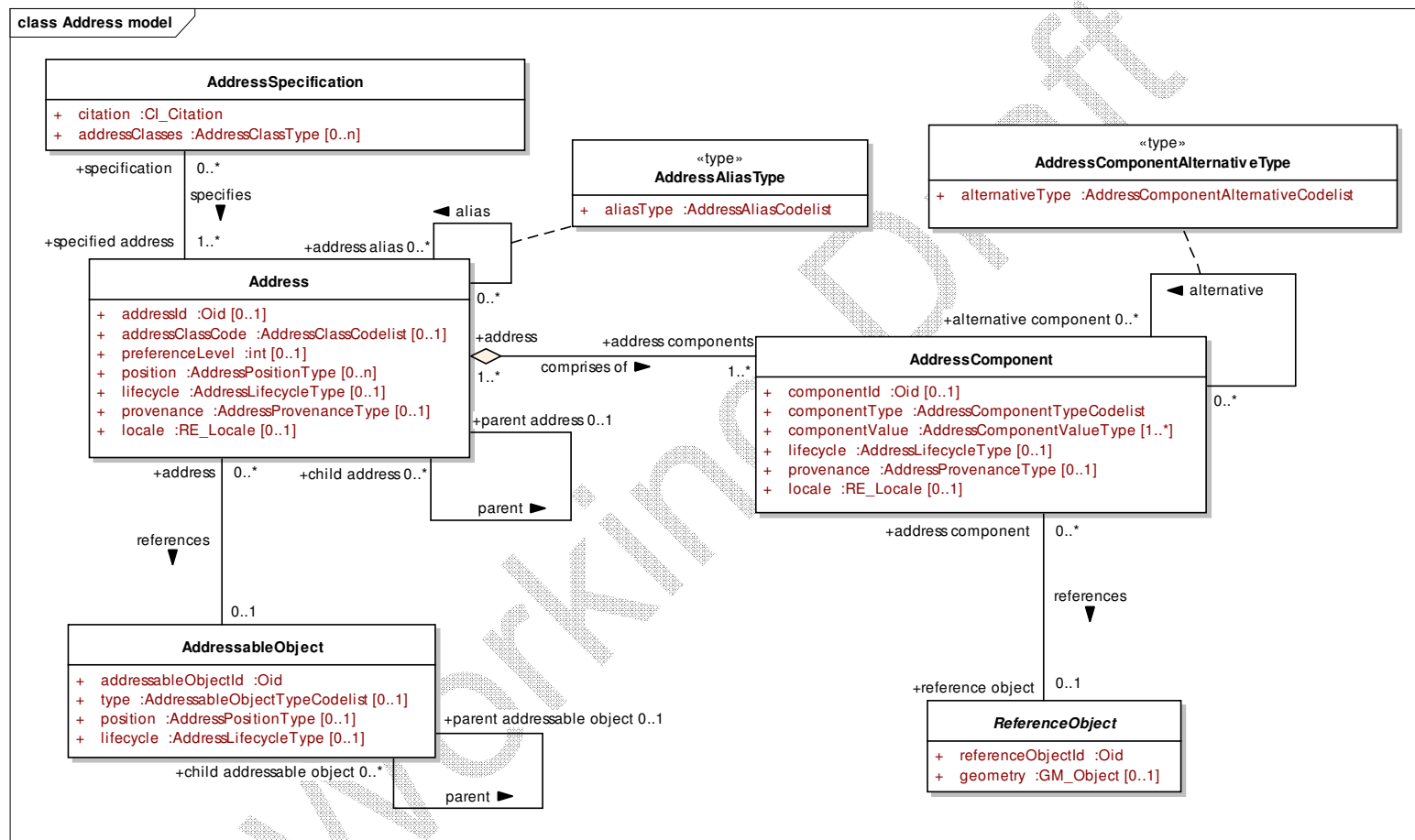


Figure 1 – Address model

Figures 2 shows the types defined in the Core requirements class.

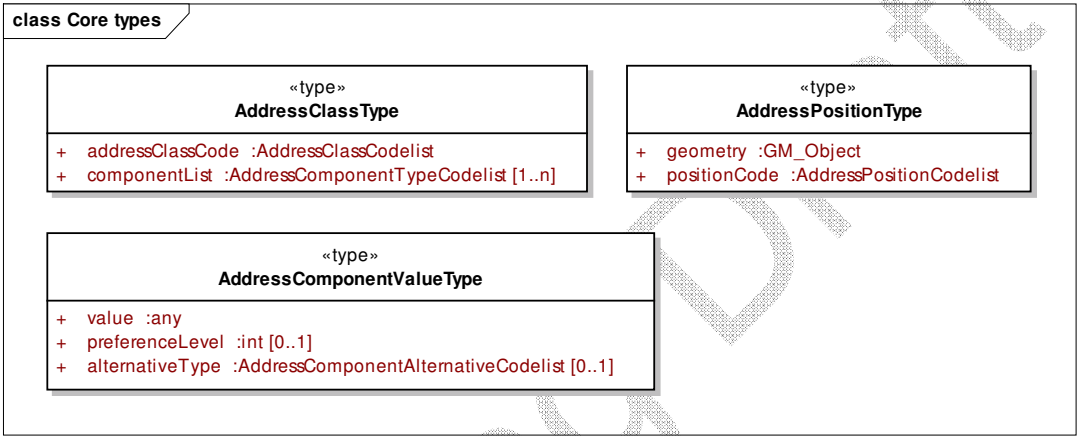


Figure 2 – Types defined in the Core requirements class

Figure 3 shows the types and codelists defined in the Core, Lifecycle and Provenance requirements classes respectively.

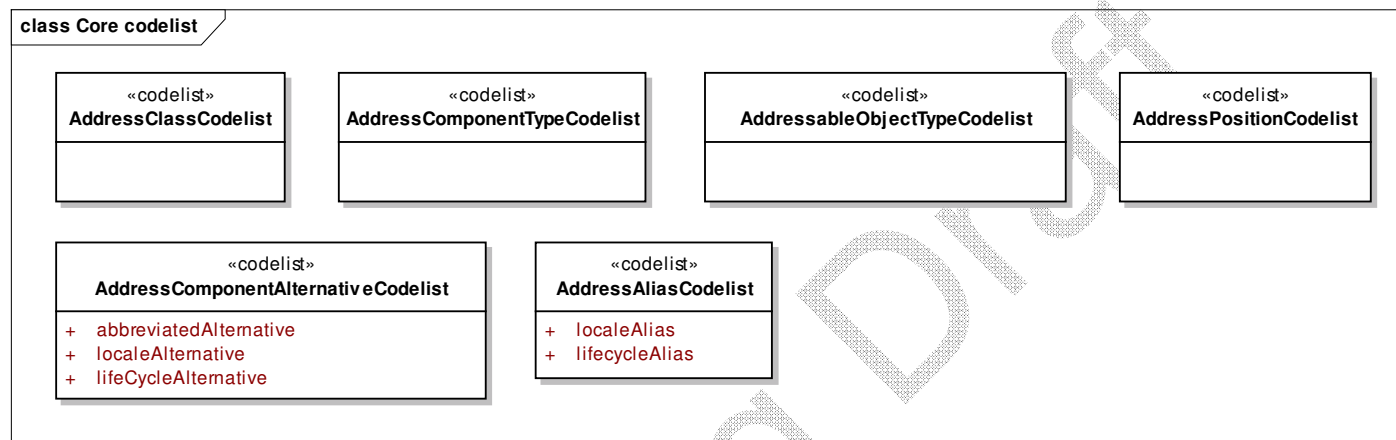


Figure 3 – Codelists defined in the Core requirements class

NOTE There are too many possible values with little known overlap for AddressableObjectTypeCodelist, AddressComponentTypeCodelist, AddressClassCodelist and AddressPositionCodelist, therefore these codelists are empty. Each model has to specify codes, as required.

- EXAMPLE 1 building, house, landmark, apartment are examples of codes for the AddressableObjectTypeCodelist.
- EXAMPLE 2 streetName, placeName, addressee and postcode are examples of codes for the AddressComponentTypeCodelist.
- EXAMPLE 3 thoroughfareAddress, landmarkAddress and informalAddress are examples of codes for the AddressClassCodelist.
- EXAMPLE 4 centroid, streetFront and approximated are examples of codes for the AddressPositionCodelist.

Figure 4 shows the types and codelists defined in the Lifecycle requirements class.

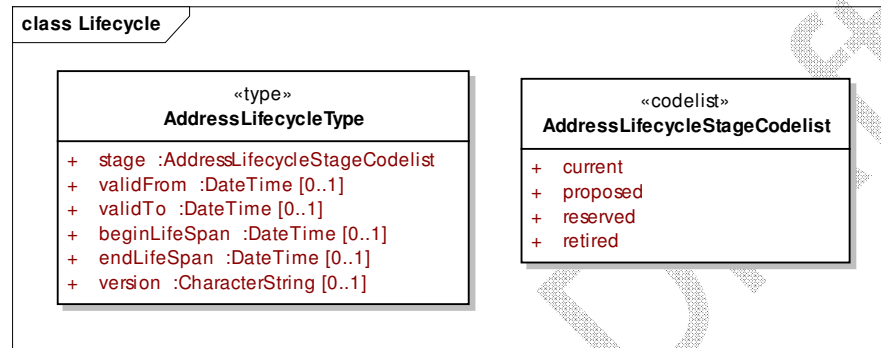


Figure 4 – Types and codelists defined in the Lifecycle requirements class

Figure 5 shows the only type defined in the Provenance requirements class.

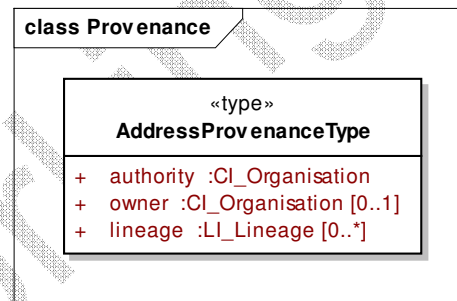


Figure 5 – AddressProvenanceType defined in the Provenance requirements class

7 Requirements class: Core

7.1 Dependencies

The table below shows the dependencies of the Core requirements class on other standards.

Table 1 – Dependencies of the Core requirements class

Requirements class identifier	Core
Target type	Conceptual model
Dependency	GM_Object in ISO 19107:2003
Dependency	CI_Citation and CI_Organisation in ISO 19115-1:2013
Dependency	Oid in ISO 19152:2013

7.2 Core requirement 1: Classes

The address model shall include the Address and AddressComponent class and may include one or more classes derived from it.

EXAMPLE Examples of address components are street name, place name, addressee and postcode.

The address model may include the AddressableObject and AddressSpecification class or classes derived from them.

EXAMPLE Examples of addressable objects are building, house, landmark and apartment.

The address model may include classes derived from the abstract class, ReferenceObject.

EXAMPLE 1 The polygon representing an administrative area.

EXAMPLE 2 The polygon representing the footprint of a building or the point representing the building.

EXAMPLE 3 The polygon representing a land parcel.

EXAMPLE 4 Information about the person or organisation who is the mail recipient.

7.3 Core requirement 2: Associations

An address shall comprise of one or more address components. An address component shall belong to one or more address.

An address may uniquely reference an addressable object. An addressable object may be referenced by addresses. Multiple addresses identifying the same addressable object are address aliases.

EXAMPLE Different access points to a building are an example of more than one address referencing the same addressable object. Refer to **Annex D** for more examples.

An address component may have alternative values. There are different types of alternative values, e.g. values in different languages or values used at different lifecycle stages.

NOTE The model has to specify whether different names of address components (e.g. town names in different languages) are alternative address components.

NOTE A spelling correction to the value (name) of an address component does NOT represent an alternative address component.

EXAMPLE For a thoroughfare name class derived from AddressComponent, a historic thoroughfare name could be a lifecycle alternative for the current thoroughfare name.

EXAMPLE For a place name class derived from AddressComponent, place names in two different languages (e.g. Köln and Cologne) are language alternatives of a place name address component.

An address component may be within the scope of another address component, i.e. the address component values are assigned according to rules ensuring they are unambiguous within the other address component.

EXAMPLE 1 A street name assigned to be unambiguous within a city (the street name is not necessarily unambiguous within the suburb).

EXAMPLE 2 An address number assigned to be unambiguous in a thoroughfare.

An address component may reference a reference object. A reference object may be referenced by address components.

EXAMPLE 1 For a thoroughfare name class derived from AddressComponent, the geometry of the street centre line could be the reference object.

EXAMPLE 2 For a mail recipient class derived from AddressComponent, information about the mail recipient, e.g. age, gender, education level, could be the reference object.

A child addressable object shall have one parent addressable object. A parent addressable object may have child addressable objects.

EXAMPLE 1 The building is the parent addressable object of the individual apartments (child addressable objects) inside the building.

EXAMPLE 2 In Japan, the *gaiku* (block) is the parent addressable object of the individual *jukyo bango* (residence numbers) (child addressable objects) inside the building.

A child address shall have one parent address. A parent address may have child addresses.

EXAMPLE 1 The address of the building is the parent address of the addresses of individual apartments inside the building.

EXAMPLE 2 In Japan, the address of the *gaiku* (block) is the parent address of the individual *jukyo bango* (residence numbers) addresses in the *gaiku* (block).

An address may be specified according to one or more address specification. An address specification may include a list of address classes. An address class specifies the list of valid component types for the class.

NOTE A legal act, a technical specification or an addressing standard may specify the address reference system.

EXAMPLE 1 SANS 1883:2009, *Geographic information – Addresses*.

EXAMPLE 2 Korea Road Name Address Act of 2011.

EXAMPLE 3 D2.8.1.5 *INSPIRE Data Specification on Addresses – Guidelines*, INSPIRE Thematic Working Group on *Addresses*

EXAMPLE 4 UPU S42:2006, International postal address components and templates

EXAMPLE 5 Japanese act...

7.4 Core requirement 3: Attributes

Classes in an address model conforming to the Core conformance class shall include mandatory, conditional and optional attributes as specified in the tables of this sub-clause. The name, definition, obligation or condition, maximum occurrence, data type and domain of each attribute are provided.

Table 2 – Address attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
addressId	Unique identifier of an address	O	1	Class	Oid See ISO 19152:2012
addressClassCode	Code of the address class to which the address belongs.	O	1	Class	<<odelist>> AddressClassCodeList
preferenceLevel	Indicates the level of preference of the address to use from a set of aliases. 1 indicates highest preference. EXAMPLE 1	O	1	integer	Integer > 0

	A building on a street corner could be referenced by two addresses. One of them has the preferenceLevel set to 1. EXAMPLE 2 In Switzerland, addresses with Biel (German) and Bienne (French) are different addresses with the same preference level.				
position	One or more position representing the address.	O	N	Class	AddressPositionType

Table 3 – AddressComponent attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
componentId	Uniquely identifies the address component.	O	1	Class	Oid See ISO/DIS 19152:2012
componentType	The type of the component.	M	1	Class	<<codelist>> AddressComponentTypeCodelist
componentValue	Value of the component.	M	N	Class	<<type>> AddressComponentValueType

Table 4 – AddressableObject attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
addressableObjectId	Unique identifier of the addressable object.	M	1	Class	Oid See ISO/DIS 19152:2012
type	The type of object that is addressed.	C : Only if there are no subclasses of AddressableObject	1	Class	<<codelist>> AddressableObjectTypeCodelist
position	Position of the addressable object.	O	1	Class	<<type>> AddressPositionType

Table 5 – ReferenceObject attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
referenceObjectId	Unique identifier of the reference object.	M	1	Class	OID See ISO/DIS 19152:2012
geometry	Geometry of the reference object.	O	1	Class	<<type>> GM_Object See ISO 19107:2003

Table 6 – AddressSpecification attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
reference	Citation for the specification.	M	1	Class	CI_Citation See ISO/DIS 19115-1:2013
addressClasses	A list of address classes included in the specification.	O	N	Class	<<type>> AddressClassType

Table 7 – AddressClassType attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
addressClassCode	Code for the address class.	M	1	Class	<<odelist>> AddressClassCodelist
componentList	A list of valid address component types.	M	N	Class	<<odelist> AddressComponentType

Table 8 – AddressPositionType attributes

Name	Definition	Mandatory/ Conditional/	Max Occur	Data Type	Domain
------	------------	----------------------------	--------------	--------------	--------

		Optional			
geometry	Geometry of the position.	M	1	Class	<<type>> GM_Object See ISO 19107:2003
positionCode	Code to identify the type of position. EXAMPLES centroid, streetFront, approximated	O	1	Class	<<codelist>> AddressPositionCodelist

Table 9 – AddressComponentValueType attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
value	Value of the component.	M	1	any	
preferenceLevel	Indicates the level of preference of the address component to use from a set of alternatives. 1 indicates highest preference. EXAMPLE 1 A building on a street corner could be referenced by two addresses, each with a different component value for the thoroughfare name address component. For one of them the preferenceLevel is set to 1. EXAMPLE 2 In Switzerland, Biel (German) and Bienne (French) are different address components with the same preference level.	O	1	integer	integer > 0
alternativeType	Indicate the type of alternative address component represented by this value.	O	1	Class	<<codelist>> AddressComponentAlternativeTypeCodelist

Add AddressAliasType and AddressComponentAlternativeType... (see address model)

7.5 Codelists

<add text>

Table 10 – AddressAliasCodelist values

Name	Definition
localeAlias	The association is between an address and its alias in a different locale.
lifeCycleAlias	The association is between an address and its alias at a different lifecycle stage.

Table 11 – AddressComponentAlternativeCodelist values

Name	Definition
abbreviatedAlternative	The association is between an address component and its abbreviation.
localeAlternative	The association is between an address component and its alternative in a different locale.
lifeCycleAlternative	The association is between an address component and its alternative at a different lifecycle stage.

8 Requirements class: Lifecycle

8.1 Dependencies

The table below shows the dependencies of the Lifecycle requirements class on other standards.

Table 12 – Dependencies of the Lifecycle requirements class

Requirements class identifier	Lifecycle
Target type	Conceptual model
Dependency	DateTime from ISO 8601:2004
Dependency	CharacterString from ISO/TS 19103:2005

8.2 Lifecycle requirement 1: Attributes

Classes in an address model conforming to the Lifecycle conformance class shall include mandatory, conditional and optional attributes as specified in the tables of this sub-clause. The name, definition, obligation or condition, maximum occurrence, data type and domain of each attribute are provided.

Table 13 – Address attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
lifecycle	Information about the life cycle of an address.	M	1	type	<<type>> AddressLifecycleType

Table 14 – AddressComponent attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
lifecycle	Information about the life cycle of an address component.	M	1	type	<<type>> AddressLifecycleType

Table 15 – AddressableObject attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
lifecycle	Information about the life cycle of an addressable object.	M	1	type	<<type>> AddressLifecycleType

Table 16 – AddressLifecycleType attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
stage	The stage in the life cycle of an addressable object.	M	1	Class	<<codelist> AddressLifecycleStageCodelist
validFrom	Date and time from which the object is valid.	O	1	Class	DateTime Character encoding of a DateTime shall follow ISO 8601.

					This class is documented in full in ISO/TS 19103:2005.
validTo	Date and time when this object ceases to be valid.	O	1	Class	DateTime Character encoding of a DateTime shall follow ISO 8601. This class is documented in full in ISO/TS 19103:2005.
beginLifespan	Specifies the date and time at which this version of the address was inserted or changed in the data set.	O	1	Class	DateTime Character encoding of a DateTime shall follow ISO 8601. This class is documented in full in ISO/TS 19103:2005.
endLifespan	Specifies the date and time at which this version of the address was superseded or retired in the dataset.	O	1	Class	DateTime Character encoding of a DateTime shall follow ISO 8601. This class is documented in full in ISO/TS 19103:2005.
version	Information to specify the version of the data object representing the address.	O	1	CharacterString	Free text

8.3 Codelists

Table 17 – AddressLifecycleStageCodelist values

Name	Definition
current	The address is currently in use.
proposed	The address is proposed for future use.
reserved	The address is reserved for future use.
retired	The address was in use at some stage, but not anymore.

9 Requirements class: Provenance

9.1 Dependencies

The table below shows the dependencies of the Provenance requirements class on other standards.

Table 18 – Dependencies of the Provenance requirements class

Requirements class identifier	Provenance
Target type	Conceptual model
Dependency	None

9.2 Provenance requirement 1: Attributes

Classes in an address model conforming to the Provenance conformance class shall include mandatory, conditional and optional attributes as specified in the tables of this sub-clause. The name, definition, obligation or condition, maximum occurrence, data type and domain of each attribute are provided.

Table 19 – Address attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
provenance	Information about the provenance of an address.	M	1	Class	<<type>> AddressProvenanceType

Table 20 – AddressComponent attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
provenance	Information about the provenance of an address component.	M	1	Class	<<type>> AddressProvenanceType

The AddressProvenanceType may have one or more additional metadata attributes specified in ISO 19115-1:2013 and/or Dublin Core ([reference](#)).

Table 21 – AddressProvenanceType attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
authority	Authority that assigns the address or address component values.	M	1	Class	CI_Organisation See ISO 19115-1:2013
owner	Organisation that maintains the address information.	O	1	Class	CI_Organisation See ISO 19115-1:2013
lineage	Lineage information about the address.	O	N	Class	LI_Lineage See ISO 19115-1:2013

10 Requirements class: Locale

10.1 Dependencies

The table below shows the dependencies of the Locale requirements class on other standards.

Table 22 – Dependencies of the Locale requirements class

Requirements class identifier	Locale
Target type	Conceptual model
Dependency	RE_Locale in ISO 19135:2005

Comment [SMC1]: Not in the revision (ISO 19135-1) anymore?

10.2 Locale requirement 1: Attributes

Classes in an address model conforming to the Locale conformance class shall include mandatory, conditional and optional attributes as specified in the tables of this sub-clause. The name, definition, obligation or condition, maximum occurrence, data type and domain of each attribute are provided.

Table 23 – Address attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
locale	Information about the locale of an address, i.e. the language, country information and character encoding for the proper interpretation of the content of character strings in the address.	M	1	Class	RE_Locale See ISO 19135:2005

Table 24 – AddressComponent attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
locale	Information about the locale of an address, i.e. specify language, country information and character encoding for the proper interpretation of the content of character strings in the address component.	M	1	Class	RE_Locale See ISO 19135:2005

Annex A

(normative)

Abstract test suites

A.1 Introduction

The abstract test suites for the conformance classes defined by this International Standard are presented in the clauses A.2 to A.5 below.

Annex needs to be updated to reflect latest changes

A.2 Conformance class: Core

A.2.1 Introduction

Tables A.1 to A.3 contain the details of the tests for the Core conformance class.

A.2.2 Core test 1: Classes

Table A.1 Core test 1: Classes

Test purpose	Check that the model contains the classes as specified.
Test method	Inspect the model
Reference	7.2
Test type	Basic

A.2.4 Core test 2: Associations

Table A.2 Core test 2: Associations

Test purpose	Check that the model contains the classes as specified.
Test method	Inspect the model
Reference	7.3
Test type	Basic

A.2.6 Core test 3: Attributes

Table A.3 Core test 3: Attributes

Test purpose	For each class and type in the model, check that the model includes the optional, mandatory and conditional attributes
Test method	Inspect the model
Reference	7.4

Test type	Basic
-----------	-------

A.3 Conformance class: Lifecycle

A.3.1 Introduction

Table A.4 contains the details of the test for the Lifecycle conformance class.

A.3.2 Lifecycle test 1: Attributes

Table A.4 Lifecycle test 1: Attributes

Test purpose	For each class and type in the model, check that the model includes the optional, mandatory and conditional attributes
Test method	Inspect the model
Reference	8.2
Test type	Basic

A.4 Conformance class: Provenance

A.4.1 Introduction

Table A.5 contains the details of the test for the Provenance conformance class.

A.4.2 Provenance test 1: Attributes

Table A.5 Provenance test 1: Attributes

Test purpose	For each class and type in the model, check that the model includes the optional, mandatory and conditional attributes
Test method	Inspect the model
Reference	9.2
Test type	Basic

A.5 Conformance class: Locale

A.5.1 Introduction

Table A.6 contains the details of the test for the Locale conformance class.

A.5.2 Locale test 1: Attributes

Table A.6 Provenance test 1: Attributes

Test purpose	For each class and type in the model, check that the model includes the optional, mandatory and conditional attributes
Test method	Inspect the model
Reference	10.2
Test type	Basic

Working Draft

Annex B (informative)

Profiles and instance data

B.1 Profiles

B.1.1 Introduction

This clause describes the steps followed to develop the profiles of ISO 19160-1 included in this annex. It also describes the information provided as specification of the profiles in this annex. **(Remove this clause in the final document?)**

B.1.2 Steps followed to develop a profile

1. Decide to which conformance classes the profile shall conform.

All profiles shall conform to the Core conformance class.

In addition, a profile may conform to one or more additional conformance classes (Lifecycle, Provenance and Locale).

2. Identify the classes to be included.

All mandatory classes (Address, AddressComponent) shall be included in the profile.

For each optional class (AddressSpecification, AddressableObject, ReferenceObject), decide whether the optional class remains optional in the profile, or whether it is one of the following in the profile: mandatory, conditional, out-of-scope or prohibited.

Add profile-specific classes.

3. Identify the associations to be included

All mandatory associations shall be included in the profile.

For each optional association, decide whether the optional association remains optional in the profile, or whether it is one of the following in the profile: mandatory, conditional, out-of-scope or prohibited.

Add profile-specific associations.

4. Identify the attributes to be included:

All mandatory attributes shall be included in the profile.

For each optional attribute, decide whether the optional attribute remains optional in the profile, or whether it is one of the following in the profile: mandatory, conditional, out-of-scope or prohibited.

Add profile-specific attributes.

Specify what the *any* type is for each value in AddressComponentValueType.

Describe how each attribute in the profile is mapped to the relevant standard/specification and back.

5. Create instance data (examples) for a number of addresses to verify the profile.

Temporary Enterprise Architect Note on how to create an instance on a diagram:

1. On the context menu of a class, choose Advanced > Convert to Instance. Click 'Yes' to confirm when prompted.

2. On the context menu of the instance, choose Properties and fill in the name of the instance.

3. On the context menu of the instance, choose Advanced > Set Run State (Ctrl+Shift+R). Select an attribute from the list, select an operator from the list, enter a value ...

B.1.3 Profile diagrams

In the profile diagrams, use a transparent background for ISO 19160-1 classes, and a shaded fill colour background for the profile-specific classes to distinguish between the two.

Where possible, diagrams should use a similar locational pattern (i.e. AddressSpecification at the left top, Address and AddressableObject underneath it, AddressComponent and ReferenceObject on the right hand side) to the one used for the address model in Figure 1.

B.2 Minimal profile

B.2.1 Introduction

The minimal profile conforms to the *Core* conformance class.

The minimal profile represents minimal addresses, i.e. addresses as they are used colloquially, without any additional information, such as identifiers, coordinates or metadata.

B.2.2 Diagrams

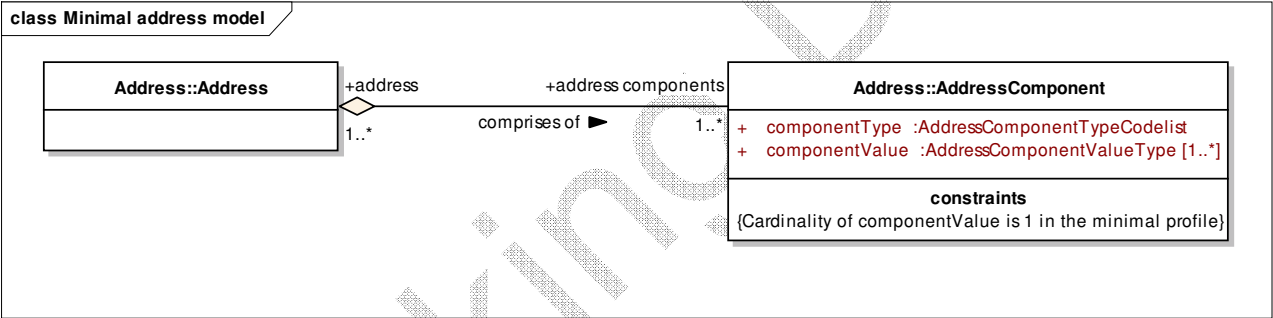


Figure B.2.1 Minimal profile: address model

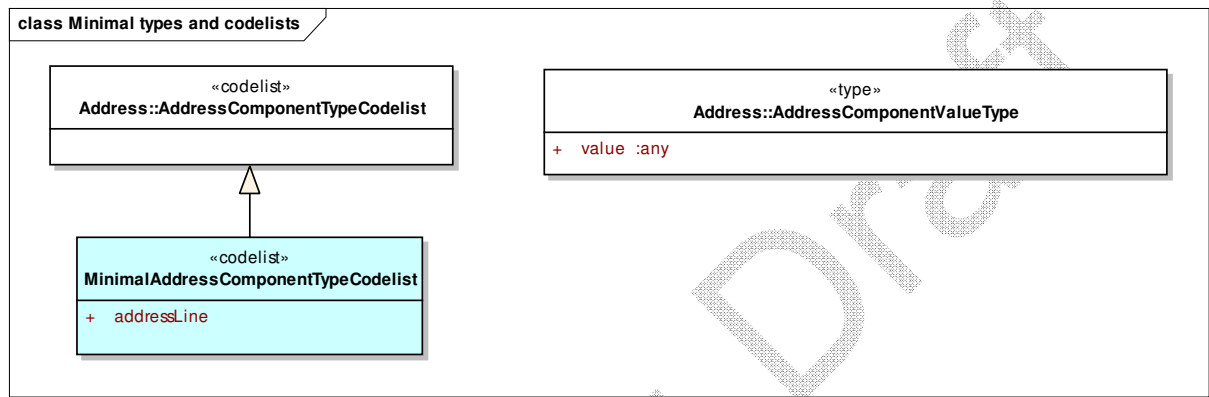


Figure B.2.2 Minimal profile: types and codelists

Table B.2.1 Minimal profile: any type for the AddressComponentTypeCodelist

AddressComponentTypeCodelist value	Corresponding type
addressLine	CharacterString

B.2.3 Instance data for the minimal profile

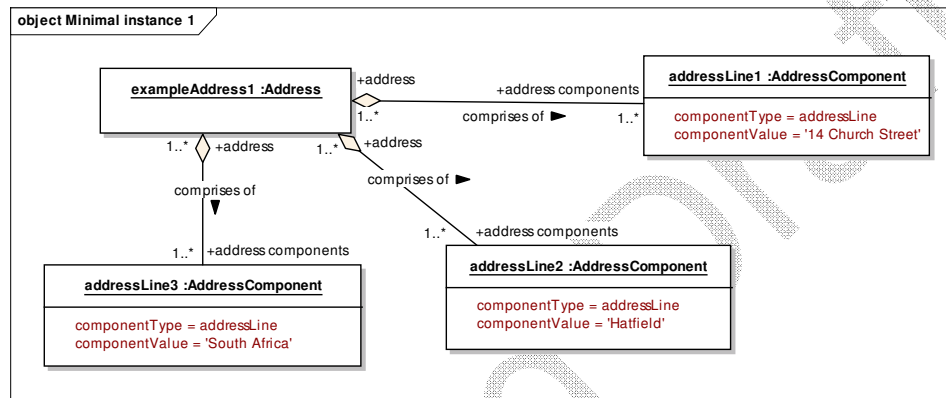


Figure B.2.3 Minimal profile: instance 1

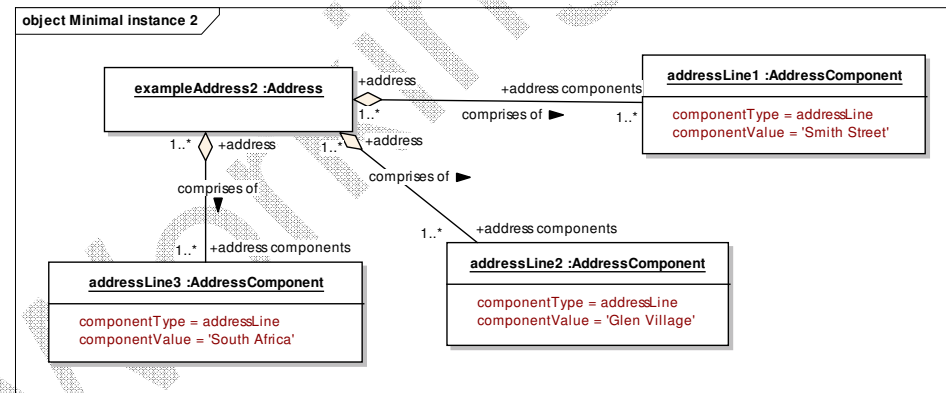


Figure B.2.4 Minimal profile: instance 2

B.3 <name> profile

B.3.1 Introduction

The <name> profile conforms to the <Core, Lifecycle, Provenance and Locale> conformance class.

The <name> profile represents addresses specified by <specification>.

B.3.2 Diagrams

<figure placeholder>

Figure B.3.1 <name> profile: address model

<figure placeholder>

Figure B.3.2 <name> profile: types

<figure placeholder>

Figure B.3.3 <name> profile: codelists

Table B.3.1 <name> profile: any type for the AddressComponentTypeCodelist

AddressComponentTypeCodelist value	Corresponding type

--	--

B.3.3 Attribute mapping from **<specification>** to the **<name>** profile

Table B.3.2 Address mapping from **<specification>** to the **<name>** profile

<specification> profile attribute and data type	<name> profile attribute	Mapping description

Table B.3.3 AddressComponent mapping from **<specification>** to the **<name>** profile

<specification> attribute and data type	<name> profile attribute	Mapping description

Add tables for additional classes as required...

B.3.4 Attribute mapping from the **<name>** profile to **<specification>**

Table B.3.4 Address mapping from the **<name>** profile to **<specification>**

<name> profile attribute	<specification> attribute and data type	Mapping description

<name> profile attribute	<specification> attribute and data type	Mapping description

Table B.3.5 AddressComponent mapping from the <name> profile to <specification>

<name> profile attribute	<specification> attribute and date type	Mapping description

Add tables for additional classes as required...

B.3.5 Instance data for the <name> profile

<figure placeholder>

Figure B.3.4 <name> profile: instance 1

<figure placeholder>

Figure B.3.5 <name> profile: instance 2

B.4 Australian profile

B.4.1 Introduction

The Australian profile conforms to the core conformance class.

The Australian profile represents addresses specified by *AS 4590:2006 Interchange of client information (incorporating Amendment No. 1)*.

B.4.2 Diagrams

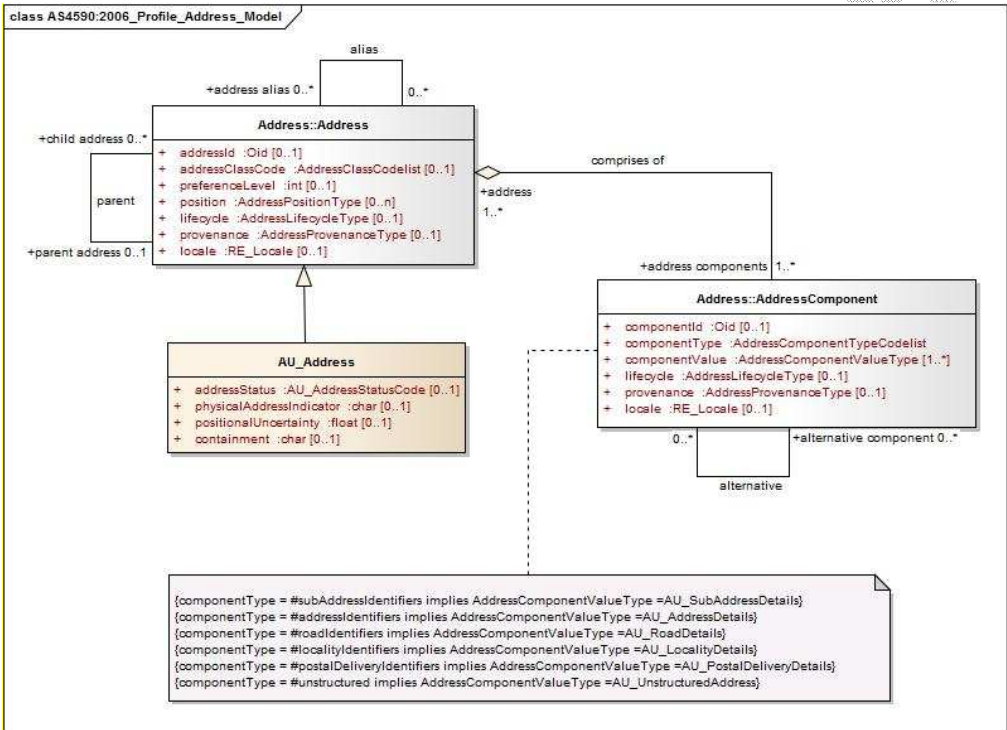


Figure B.4.1 Australian profile: address model

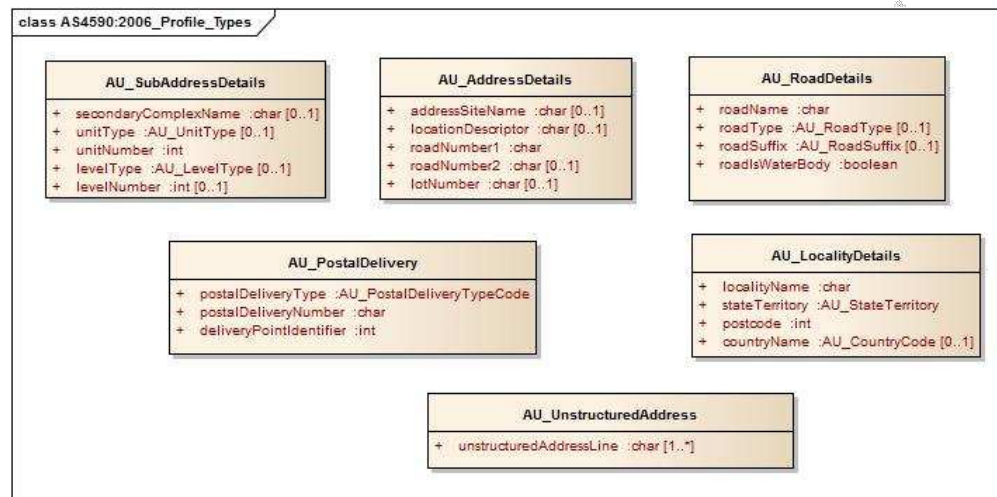


Figure B.4.2 Australian profile: types

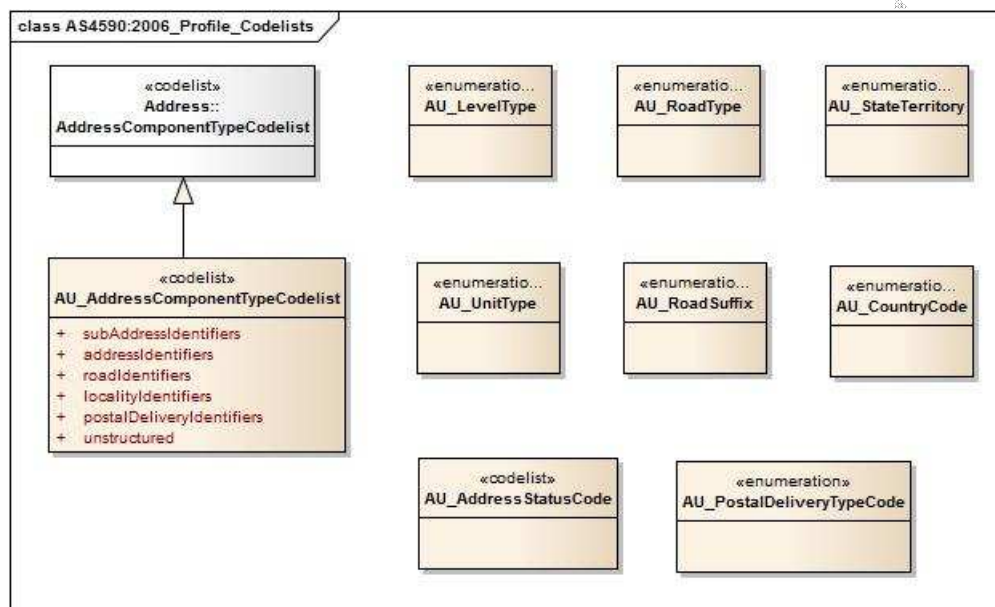


Figure B.4.3 Australian profile: codelists

Table B.4.1 Australian profile: any type for the AddressComponentTypeCodelist

AddressComponentTypeCodelist value	Corresponding type
subAddressIdentifiers	<<type>> AU_SubAddressDetails
addressIdentifiers	<<type>> AU_AddressDetails
roadIdentifiers	<<type>> AU_RoadDetails
localityIdentifiers	<<type>> AU_LocalityDetails
postalDeliveryIdentifiers	<<type>> AU_PostalDeliveryDetails
unstructured	<<type>> AU_UnstructuredAddress

B.4.3 Attribute mapping from AS 4590:2006 to the Australian profile

Note – Each of the AS 4590:2006 attributes is preceded by its clause reference in AS 4590:2006.

Table B.4.2 Mapping from AS 4590:2006 to the Australian profile

AS 4590:2006 attribute and data type		Australian profile attribute	Mapping description
5.1 Address usage code	codelist	N/A	No equivalent attribute
5.2 Physical address indicator	codelist	AU_Address.physicalAddressIndicator	

AS 4590:2006 attribute and data type		Australian profile attribute	Mapping description
5.3 Address status code	codelist	AU_Address.addressStatus	
5.4.1 Client address currency code	codelist	N/A	No equivalent attribute
5.4.2 Address start date	date	N/A	No equivalent attribute
5.4.3 Address start time	Time	N/A	No equivalent attribute
5.4.4 Address end date	date	N/A	No equivalent attribute
5.4.5 Address end time	Time	N/A	No equivalent attribute
5.5.1.1 Sub dwelling unit type code	codelist	AU_SubAddressDetails.unitType	
5.5.1.2 Sub dwelling unit number	character string	AU_SubAddressDetails.unitNumber	
5.5.2.1 Level type code	codelist	AU_SubAddressDetails.levelType	
5.5.2.2 Level number	character string	AU_SubAddressDetails.levelNumber	
5.5.3.1 Complex road number 1	character string	AU_AddressDetails.roadNumber1	
5.5.3.2 Complex road number 2	character string	AU_AddressDetails.roadNumber2	
5.5.4.1 Complex road name	character string	AU_RoadDetails.roadName	
5.5.4.2 Complex road type code	codelist	AU_RoadDetails.roadType	

AS 4590:2006 attribute and data type		Australian profile attribute	Mapping description
5.5.4.3 Complex road suffix code	codelist	AU_RoadDetails.roadSuffix	
5.5.4.4 Secondary complex name	character string	AU_SubAddressDetails.secondaryComplexName	
5.7 Address site name	character string	AU_AddressDetails.addressSiteName	
5.8.1 Lot number	character string	AU_AddressDetails.lotNumber	
5.8.2 Road number 1	character string	AU_AddressDetails.roadNumber1	
5.8.3 Road number 2	character string	AU_AddressDetails.roadNumber2	
5.9.1 Road name	character string	AU_RoadDetails.roadName	
5.9.2 Road type code	codelist	AU_RoadDetails.roadType	
5.9.3 Road suffix code	codelist	AU_RoadDetails.roadSuffix	
5.10.1 Postal delivery type code	codelist	AU_PostalDeliveryDetails.postalDeliveryType	
5.10.2 Postal delivery number	character string	AU_PostalDeliveryDetails.postalDeliveryNumber	
5.11 Locality Name	character string	AU_LocalityDetails.localityName	
5.12 State Territory Code	codelist	AU_LocalityDetails.stateTerritory	
5.13 Postcode	integer	AU_LocalityDetails.postcode	

AS 4590:2006 attribute and data type		Australian profile attribute	Mapping description
5.14.1 Unstructured address line	character string	AU_UnstructuredAddress.unstructuredAddressLine	
5.15 Country name code	codelist	AU_LocalityDetails.countryName	
5.16 Location descriptor	character string	AU_AddressDetails.locationDescriptor	
5.17 Delivery point identifier	integer	AU_PostalDeliveryDetails.deliveryPointIdentifier	
5.18.1 Geocode feature	character string	Address.position.positionCode	
5.18.2 Geocode geographic datum	character string	Address.position.geometry	
5.18.3 Geocode vertical datum	character string	Address.position.geometry	
5.18.4 Geocode latitude	real	Address.position.geometry	
5.18.5 Geocode longitude	real	Address.position.geometry	
5.18.6 Geocode height	real	Address.position.geometry	
5.18.7 Geocode positional uncertainty	real	AU_Address.positionalUncertainty	
5.18.8 Geocode containment	character string	AU_Address.containment	

B.4.4 Attribute mapping from the Australian profile to AS 4590:2006

Note – Each of the AS 4590:2006 attributes is preceded by its clause reference in AS 4590:2006.

Table B.4.3 Address mapping from the Australian profile to AS 4590:2006

Australian profile attribute (all in class Address)	AS 4590:2006 attribute and data type		Mapping description
addressID	No equivalent attribute		Out of scope
addressClassCode	No equivalent attribute		Out of scope
preferenceLevel	No equivalent attribute		Out of scope
position.geometry	5.18.2 Geocode geographic datum	character string	
position.geometry	5.18.3 Geocode vertical datum	character string	
position.geometry	5.18.4 Geocode latitude	real	
position.geometry	5.18.5 Geocode longitude	real	
position.geometry	5.18.6 Geocode height	real	
position.positionCode	5.18.1 Geocode feature	character string	
Lifecycle			Out of scope
Provenance			Out of scope
Locale			Out of scope

Table B.4.4 AU_Address mapping from the Australian profile to AS 4590:2006

Australian profile attribute (all in class AU_Address)	AS 4590:2006 attribute and data type		Mapping description
addressStatus	5.3 Address status code	codelist	
physicalAddressIndicator	5.2 Physical address indicator	codelist	
positionalUncertainty	5.18.7 Geocode positional uncertainty	real	
Containment	5.18.8 Geocode containment	character string	

Table B.4.5 AddressComponent mapping from the Australian profile to AS 4590:2006

Australian profile attribute (all in class AddressComponent)	AS 4590:2006 attribute and data type		Mapping description
componentID	No equivalent attribute		Out of scope
componentType	No equivalent attribute		
componentValue	No equivalent attribute		
Lifecycle			Out of scope
Provenance			Out of scope
Locale			Out of scope

Table B.4.6 AU_SubAddressDetails mapping from the Australian profile to AS 4590:2006

Australian profile attribute (all in class AU_SubAddressDetails)	AS 4590:2006 attribute and data type		Mapping description
secondaryComplexName	5.5.4.4 Secondary complex name	character string	
unitType	5.5.1.1 Sub dwelling unit type code	codelist	
unitNumber	5.5.1.2 Sub dwelling unit number	character string	
levelType	5.5.2.1 Level type code	codelist	
levelNumber	5.5.2.2 Level number	character string	

Table B.4.7 AU_AddressDetails mapping from the Australian profile to AS 4590:2006

Australian profile attribute (all in class AU_AddressDetails)	AS 4590:2006 attribute and data type		Mapping description
addressSiteName	5.7 Address site name	character string	
locationDescriptor	5.16 Location descriptor	character string	
roadNumber1	5.5.3.1 Complex road number 1 and 5.8.2 Road number 1	character string	
roadNumber2	5.5.3.2 Complex road number 2 and	character	

Australian profile attribute (all in class AU_AddressDetails)	AS 4590:2006 attribute and data type		Mapping description
	5.8.3 Road number 2	string	
lotNumber	5.8.1 Lot number	character string	

Table B.4.7 AU_RoadDetails mapping from the Australian profile to AS 4590:2006

Australian profile attribute (all in class AU_RoadDetails)	AS 4590:2006 attribute and data type		Mapping description
roadName	5.5.4.1 Complex road name and 5.9.1 Road name	character string	
roadType	5.5.4.2 Complex road type code and 5.9.2 Road type code	codelist	
roadSuffix	5.5.4.3 Complex road suffix code 5.9.3 Road suffix code	codelist	
roadIsWaterBody	No equivalent attribute		

Table B.4.8 AU_LocalityDetails mapping from the Australian profile to AS 4590:2006

Australian profile attribute (all in class AU_LocalityDetails)	AS 4590:2006 attribute and data type		Mapping description
localityName	5.11 Locality Name	character string	
stateTerritory	5.12 State Territory Code	codelist	
Postcode	5.13 Postcode	integer	
countryName	5.15 Country name code	codelist	

Table B.4.9 AU_PostalDeliveryDetails mapping from the Australian profile to AS 4590:2006

Australian profile attribute (all in class AU_PostalDelivery)	AS 4590:2006 attribute and data type		Mapping description
postalDeliveryType	5.10.1 Postal delivery type code	codelist	
postalDeliveryNumber	5.10.2 Postal delivery number	character string	
deliveryPointIdentifier	5.17 Delivery point identifier	integer	

Table B.4.10 AU_UnstructuredAddress mapping from the Australian profile to AS 4590:2006

Australian profile attribute (all in AU_UnstructuredAddress)	AS 4590:2006 attribute and data type	Mapping description
---	---	----------------------------

Australian profile attribute (all in AU_UnstructuredAddress)	AS 4590:2006 attribute and data type		Mapping description
unstructuredAddressLine	5.14.1 Unstructured address line	character string	

B.4.5 Instance data for the Australian profile

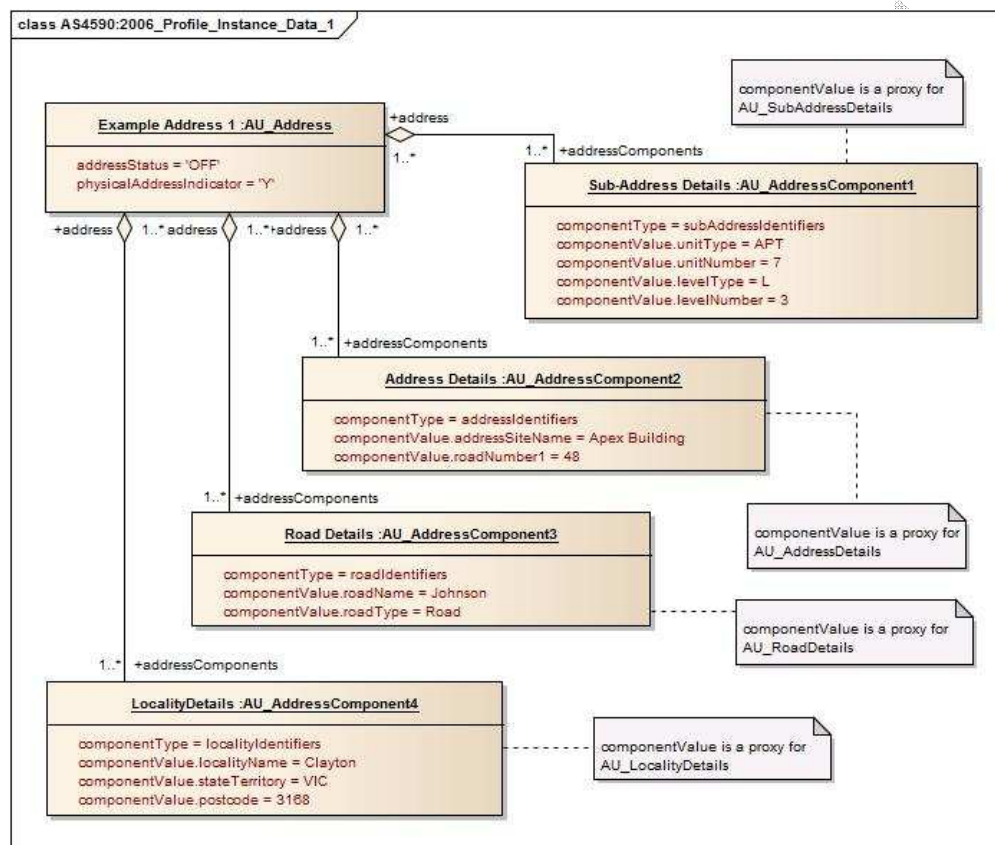


Figure B.4.4 Australian profile: instance 1

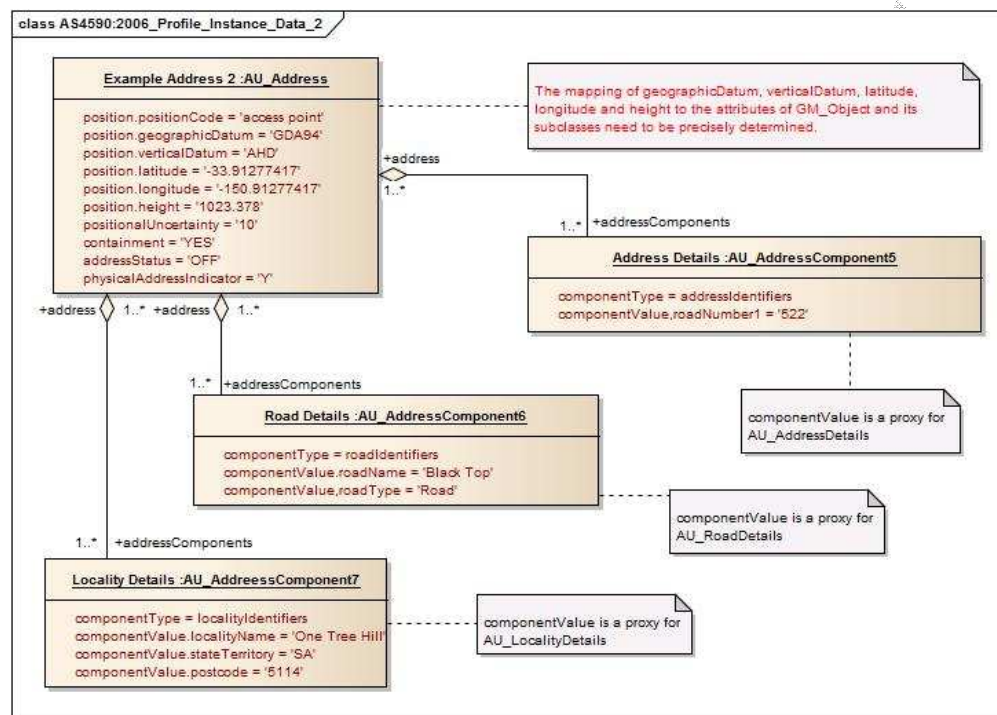


Figure B.4.5 Australian profile: instance 2

B.5 Cultural heritage profile

Under development

B.6 INSPIRE profile

B.6.1 Introduction

The INSPIRE profile conforms to the *Core [Lifecycle, Provenance and Locale]* conformance class.

Comment [ML2]: not sure what to put here.

The INSPIRE profile represents addresses specified by Commission Regulation (EU) No 1089/2010 of 23 November 2010 implementing Directive 2007/2/EC of the European Parliament and of the Council as regards interoperability of spatial data sets and services².

² OJ L 323, 8.12.2010, p. 11.

B.6.2 Diagrams

Working Draft

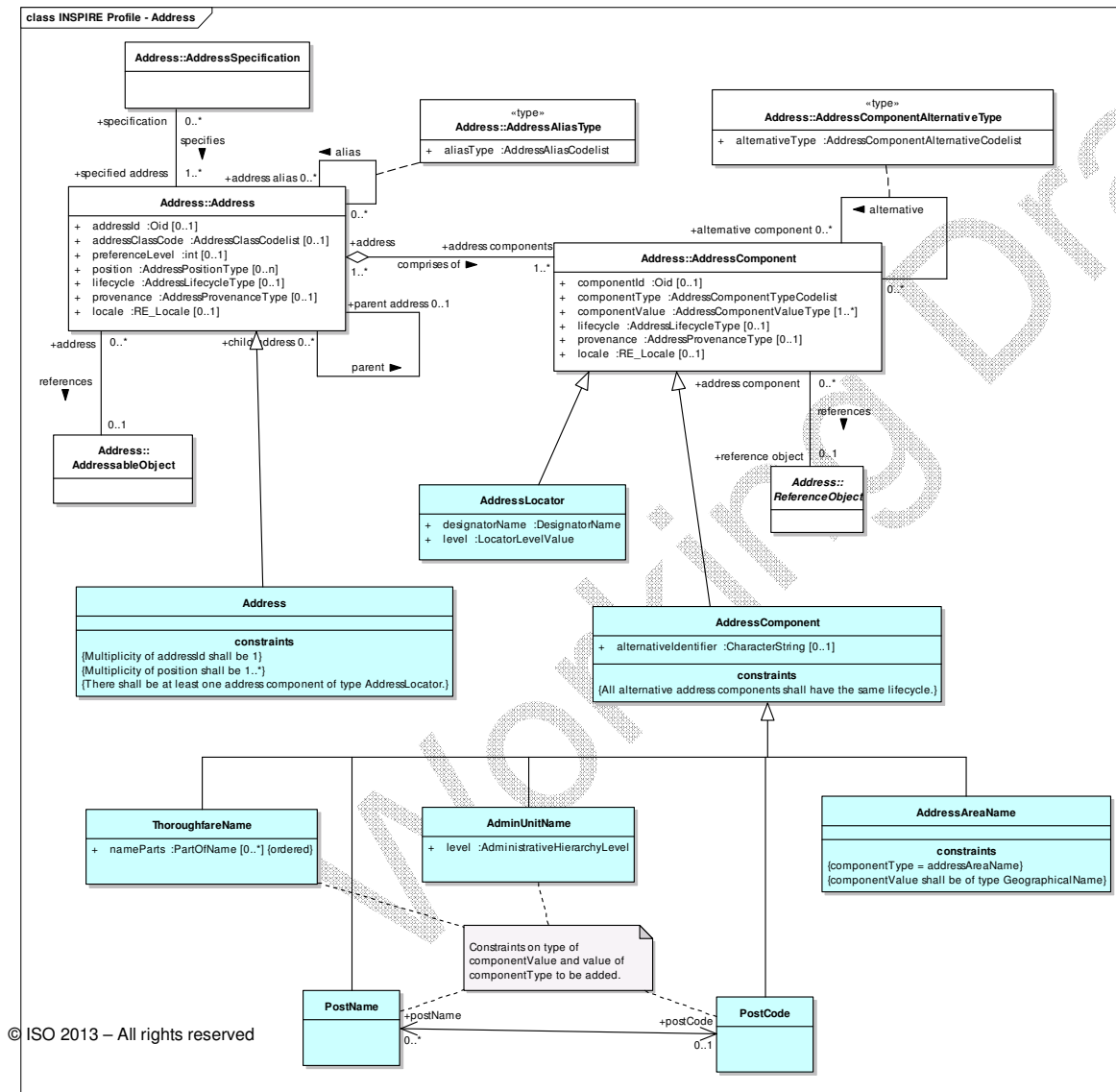


Figure B.6.1 INSPIRE profile: address model

Table B.6.1 INSPIRE profile: Additional constraints

INSPIRE profile type	Property	Constraint	Constraint (OCL)
Address	addressId	Multiplicity shall be 1.	inv: addressId->size() = 1
Address	position	Multiplicity shall be 1..*	inv: position->size()>0
Address	address components	There shall be at least one address component of type AddressLocator.	
AddressAreaName	componentType	componentType shall be addressAreaName.	inv: componentType = AddressComponentType::addressAreaName
AddressAreaName	componentValue	componentValue shall be of type GeographicalName.	inv: componentValue.ocllsTypeOf(GeographicalName)

Comment [ML3]: Other constraints for specific address components to be added.

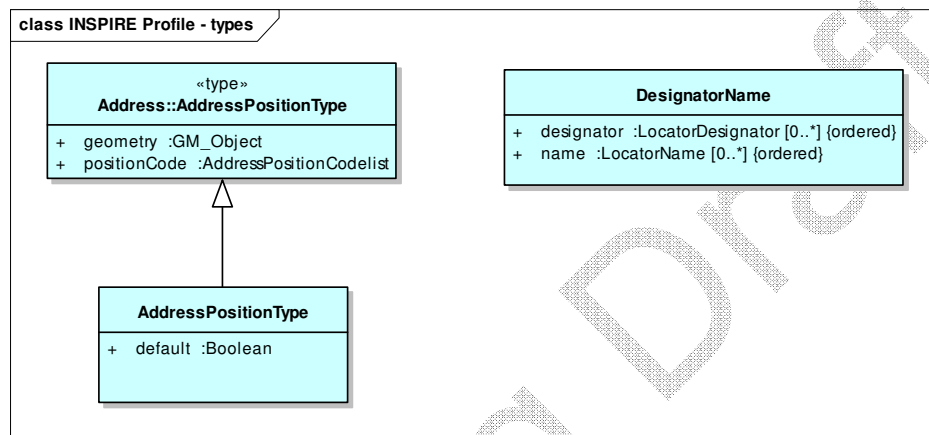


Figure B.6.2 INSPIRE profile: types

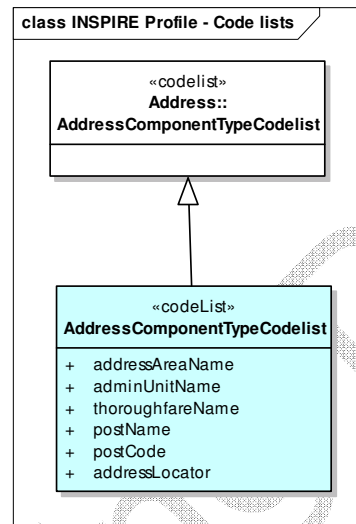


Figure B.6.3 INSPIRE profile: codelists

Table B.6.2 INSPIRE profile: any type for the AddressComponentTypeCodeList

AddressComponentType sub-type	AddressComponentTypeCodeList value	Corresponding type	Additional properties
AddressAreaName	addressAreaName	GeographicalName	;
ThoroughfareName	thoroughfareName	ThoroughfareNameValue	nameParts : PartOfName [0..*]
PostName	postName	GeographicalName	postCode : PostCode [0..1]
PostCode	postCode	CharacterString	postName : PostName [0..*]

AdminUnitName	adminUnitName	GeographicalName	level : AdministrativeHierarchyLevel
AddressLocator	addressLocator	DesignatorName	

B.6.3 Attribute mapping from the INSPIRE Addresses model to the INSPIRE profile

Comment [ML4]: We have not yet worked on the mappings from INSPIRE to the INSPIRE ISO profile.

Table B.6.3 Address mapping from INSPIRE Addresses model to the INSPIRE profile

INSPIRE Address model attribute and data type		INSPIRE profile type and attribute		Mapping description
inspireId	Identifier	Address	addressId	Map Identifier (INSPIRE) to Oid data type (from 19152).

Table B.6.4 AddressComponent mapping from INSPIRE Addresses model to the INSPIRE profile

INSPIRE Address model attribute and data type		INSPIRE profile type and attribute		Mapping description

Add tables for additional classes as required...

B.6.4 Attribute mapping from the INSPIRE profile to INSPIRE Addresses model

Table B.6.5 Address mapping from the INSPIRE profile to INSPIRE Addresses model

INSPIRE profile attribute	INSPIRE Addresses model type attribute and data type		Mapping description	
addressId	Address	inspireId	Identifier	Map Oid (from 19152) to Identifier (INSPIRE) data type.
lifecycle.stage	Address	status	StatusValue	See mapping AddressLifecycleStageCodeList → StatusValue
lifecycle.beginLifeSpan	Address	beginLifeSpanVersion	DateTime	if not present --> "void"
lifecycle.endLifeSpan	Address	endLifeSpanVersion	DateTime	
lifecycle.validFrom	Address	validFrom	DateTime	if not present --> "void"
lifecycle.validTo	Address	validTo	DateTime	
address components	Address	component	AddressComponent	Only map address components that are not of type AddressLocator. All objects in a set of INSPIRE profile AddressComponent objects that are related to each other through the <i>alternative</i> association shall be mapped to <i>one</i> INSPIRE AddressComponentObject.
address components	Address	locator	AddressLocator	Only map address components that are of type AddressLocator
parent address	Address	parentAddress	Address	

INSPIRE profile attribute	INSPIRE Addresses model type, attribute and data type		Mapping description
position	Address	position	GeographicPosition
preferenceLevel	no equivalence		See mapping AddressPositionType → GeographicPosition
lifecycle.version	no equivalence (could maybe be mapped to inspireId.version)		
provenance	no equivalence		
specification	no equivalence (could be fixed to certain value)		
locale	no equivalence		
address alias	no equivalence		
addressable object	no equivalence		

Comment [ML5]: Should all properties that cannot be mapped be explicitly be forbidden in the profile (e.g. by including constraints in the constraints table)?

Should they be included in the mapping table?

Table B.6.6 AddressComponent mapping from the INSPIRE profile to INSPIRE Addresses model

INSPIRE profile attribute	INSPIRE Addresses model type, attribute and data type		Mapping description
componentId	AddressComponent	inspireId	Identifier
componentType	Represented by classes in INSPIRE and therefore not mapped.		Map Oid (from 19152) to Identifier (INSPIRE) data type.
componentValue	Different attributes depending on the value of the componentType.		See mapping of specific address component classes.
lifecycle.stage	AddressComponent	status	StatusValue
lifecycle.beginLifeSpan	AddressComponent	beginLifeSpanVersion	DateTime

if not present --> "void"

INSPIRE profile attribute	INSPIRE Addresses model type, attribute and data type			Mapping description
lifecycle.endLifeSpan	AddressComponent	endLifeSpanVersion	DateTime	
lifecycle.validFrom	AddressComponent	validFrom	DateTime	if not present --> "void"
lifecycle.validTo	AddressComponent	validTo	DateTime	
withinScopeOf	AddressComponent	situatedWithin	AddressComponent	
alternativeIdentifier	AddressComponent	alternativeIdentifier	CharacterString	
lifecycle.version	no equivalence (could maybe be mapped to inspireId.version)			
provenance	no equivalence			
reference object	no equivalence			
locale	no equivalence			
alternative component	no equivalence			All objects in a set of INSPIRE profile AddressComponent objects that are related to each other through the <i>alternative</i> association shall be mapped to <i>one</i> INSPIRE AddressComponentObject.
reference object	no equivalence			

Table B.6.7 AddressAreaName mapping from the INSPIRE profile to INSPIRE Addresses model

INSPIRE profile attribute	INSPIRE Addresses model type, attribute and data type			Mapping description
componentValue	AddressAreaName	name.spelling.text	CharacterString	
locale	AddressAreaName	name.language		

Comment [ML6]: One table to be added per address component

INSPIRE profile attribute	INSPIRE Addresses model type, attribute and data type	Mapping description
componentType	Represented by classes in INSPIRE and therefore not mapped.	See mapping of specific address component classes.
componentValue	Different attributes depending on the value of the componentType.	See mapping of specific address component classes.

B.6.5 Instance data for the <name> profile

<figure placeholder>

Figure B.6.4 <name> profile: instance 1

<figure placeholder>

Figure B.6.5 <name> profile: instance 2

Comment [ML7]: Still to be done.

Rather than two examples of instances using the profile it would be interesting to show the same address instance using the profile as well as the other specification.

B.7 Japanese profile

Under development

B.8 New Zealand profile

B.8.1 Introduction

Conformance of the profile:

- Core, yes
- Life Cycle, probably, when we look at this.
- Provenance, yes
- Locale, likely not, NZ has two official languages but they are not used in place or street name management.

Documents consulted are (citations to follow)

(for types and codelists)

- The Australia/New Zealand Addressing Standard AUS/NZ 4819:2011
- The NZ Postal Address File Specification
- The Emergency Services and Administration Model
- The NZ CIQ xAL Profile

(for the Provenance codelist)

- The NZ Spatial Data Infrastructure Cookbook

B.8.2 Diagrams

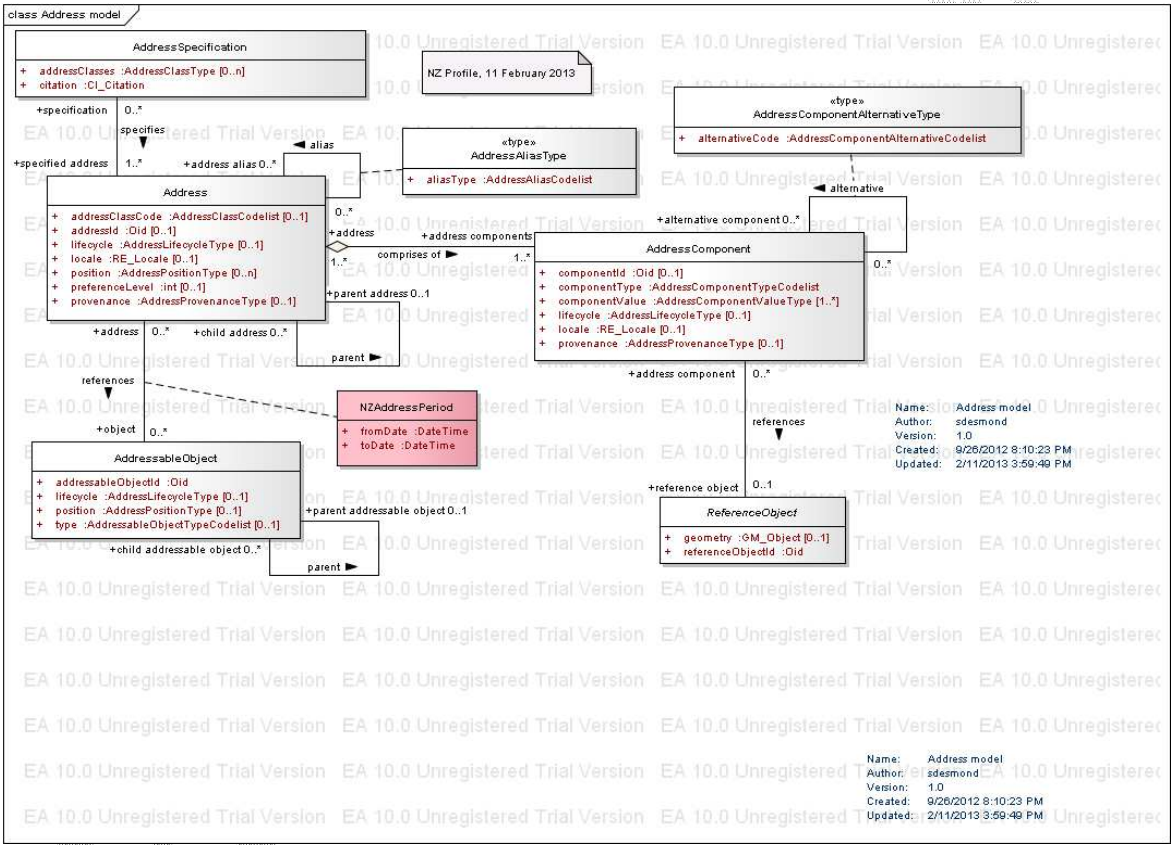


Figure B.8.1 New Zealand profile: Address model

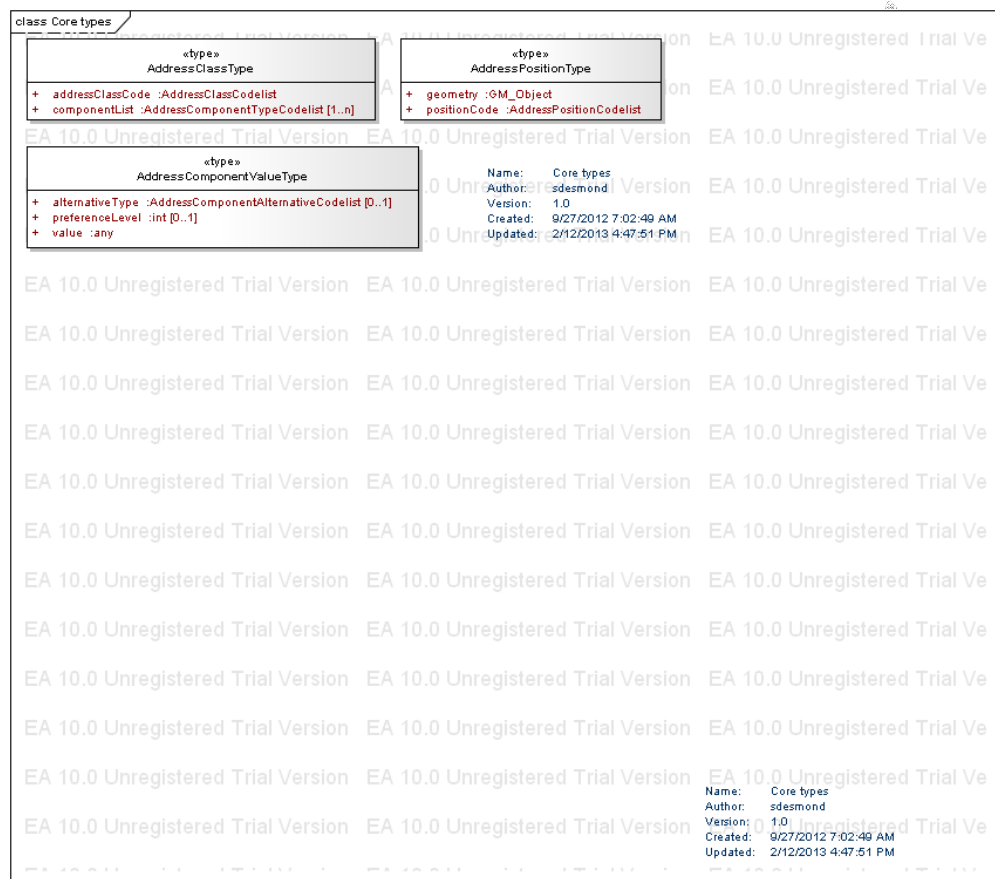


Figure B.8.2 New Zealand profile: Types

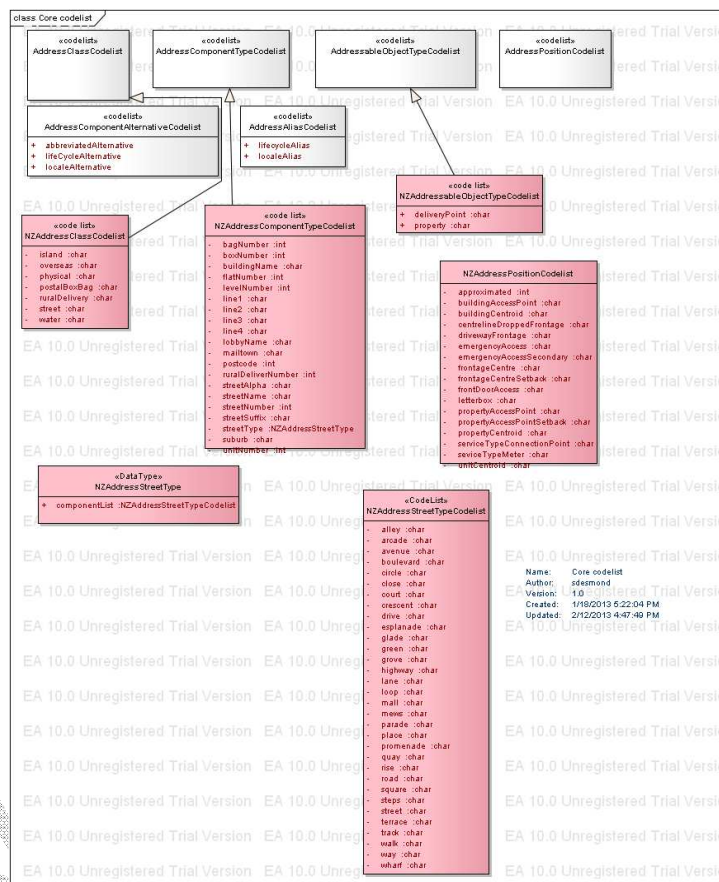


Figure B.8.3 New Zealand profile: Code lists

NZAddressClassCodelist

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	island	char							Addresses for sites on an island, numbered from a datum point in a clockwise direction
	overseas	char							A "catch all" category for overseas addresses which do not fit any NZ address classes.
	physical	char							Addresses used in place of street addresses, for instance huts in National Parks which are not in any way connected to a road.
	postalBoxBag	char							Addresses containing a PO Box or Private Bag, used to deliver mail to those boxes or bags.
	ruralDelivery	char							Addresses containing a Rural Delivery Number for postal delivery purposes.
	street	char							The physical street address, used to locate the addressable object on the surface of the globe, and also to deliver mail in an urban area.
	water	char							Address applied to those address sites only accessible by water.

NZAddressComponentTypeCodelist

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	bagNumber	int							Private Bag number
	boxNumber	int							The PO Box number
	buildingName	char							Holds the building property name for a recognised significant building
	flatNumber	int							The number of a flat located at an address, used in sub-addressing.

levelNumber	int							The level of the unit in a block of units (e.g. apartment block) used in sub-addressing.
line1	char							A "catch all" component used for overseas class addresses to allow for addresses of different formats. Included in recognition of pre-existing use, not to indicate any desire to promulgate use.
line2	char							A "catch all" component used for overseas class addresses to allow for addresses of different formats. Included in recognition of pre-existing use, not to indicate any desire to promulgate use.
line3	char							A "catch all" component used for overseas class addresses to allow for addresses of different formats. Included in recognition of pre-existing use, not to indicate any desire to promulgate use.
line4	char							A "catch all" component used for overseas class addresses to allow for addresses of different formats. Included in recognition of pre-existing use, not to indicate any desire to promulgate use.
lobbyName	char							The name of the New Zealand Post outlet or Agency outlet where the PO Box, Private Bag or Counter Delivery is physically located
mailtown	char							Holds the name of the Town/City/Mailtown associated with this delivery point.
postcode	int							Holds the four digit code defined by New Zealand Post, which is used to describe the locations linked to sorting zones for mail sorting purposes
ruralDeliverNumber	int							Rural Post – Rural Delivery route number
streetAlpha	char							Holds the alpha component of a street number
streetName	char							Holds the name of the street
streetNumber	int							Holds the base street number.
streetSuffix	char							An extension to streetName and streetType, allowing for additional descriptors such as East/West/North/South etc, used to differentiate between different sections of a thoroughfare separated by other thoroughfares.

streetType	NZAddressStreetType								Holds the type of a street (e.g. Terrace, Lane, Road).
suburb	char								Holds the name applied to a group of geographically associated streets
unitNumber	int								Holds the address elements that identifies a specific deliver sub dwelling delivery point within a building or building complex

NZAddressPositionCodelist

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	approximated	int							
	buildingAccessPoint	char							Point of access to the building.
	buildingCentroid	char							Point as centre of building and lying within its bounds (e.g. for U-shaped building).
	centrelineDroppedFrontage	char							A point on the road centre-line opposite the centre of the road frontage of an address site.
	drivewayFrontage	char							Centre of driveway on address site frontage.
	emergencyAccess	char							Specific building or property access point for emergency services.
	emergencyAccessSecondary	char							Specific building or property secondary access point for emergency services.
	frontageCentre	char							Point on the centre of the address site frontage.
	frontageCentreSetback	char							A point set back from the centre of the road frontage within an address site.
	frontDoorAccess	char							Front door of building
	letterbox	char							Place where mail is deposited.

propertyAccessPoint	char							Access point (centre of) at road frontage of the property.
propertyAccessPointSetback	char							A point set back from the (centre of the) access point at the road frontage of the property.
propertyCentroid	char							Point of centre of parcels making up a property and lying within its boundaries (e.g. for L-shaped property).
serviceTypeConnectionPoint	char							The utility connection point (e.g. box, or underground chamber). "Service type" to be specified (e.g. water, electricity, gas, telephone).
serviceTypeMeter	char							The utility meter (e.g. box, or underground chamber). "Service type" to be specified (e.g. water, electricity, gas, telephone).
unitCentroid	char							Point at centre of unit and lying within its bounds (e.g. for U-shaped unit).

NZAddressStreetTypeCodelist

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	alley	char							Usually narrow roadway in a city or town.
	arcade	char							Covered walkway with shops along the sides.
	avenue	char							Broad roadway, usually planted on each side with trees.
	boulevard	char							Wide roadway, well paved, usually ornamented with trees and grass plots
	circle	char							Roadway that generally forms a circle; or a short enclosed roadway bounded by a circle.
	close	char							Short enclosed roadway.
	court	char							Short enclosed roadway, usually surrounded by buildings.
	crescent	char							Wide main roadway without many cross-streets.

drive	char							Level roadway along the seaside, lake, or a river.
esplanade	char							Roadway usually in a valley of trees.
glade	char							Roadway usually in a valley of trees.
green	char							Roadway often leading to a grassed public recreation area.
grove	char							Roadway that features a group of trees standing together.
highway	char							Main thoroughfare between major destinations.
lane	char							Narrow roadway between walls, buildings or a narrow country roadway.
loop	char							Roadway that diverges from and rejoins the main thoroughfare.
mall	char							Wide walkway, usually with shops along the sides.
mews	char							Public roadway or promenade that has good pedestrian facilities along the side.
parade	char							Public roadway or promenade that has good pedestrian facilities along the side.
place	char							Short, sometimes narrow, enclosed roadway.
promenade	char							Wide flat walkway, usually along the water's edge.
quay	char							Roadway alongside or projecting into water.
rise	char							Roadway going to a higher place or position.
road	char							Open roadway primarily for vehicles.
square	char							Roadway which generally forms a square shape, or an area of roadway bounded by four sides.
steps	char							Walkway consisting mainly of steps.

	street	char							Public roadway in an urban area, especially where paved and with footpaths along one or both sides.
	terrace	char							Roadway on a hilly area that is mainly flat.
	track	char							Walkway in a natural setting.
	walk	char							Thoroughfare for pedestrians.
	way	char							Short enclosed roadway.
	wharf	char							A roadway on a wharf or pier.

NZAddressableObjectTypeCodelist

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	deliveryPoint	char							
	property	char							

Figure: 4

B.8.3 Provenance conformance

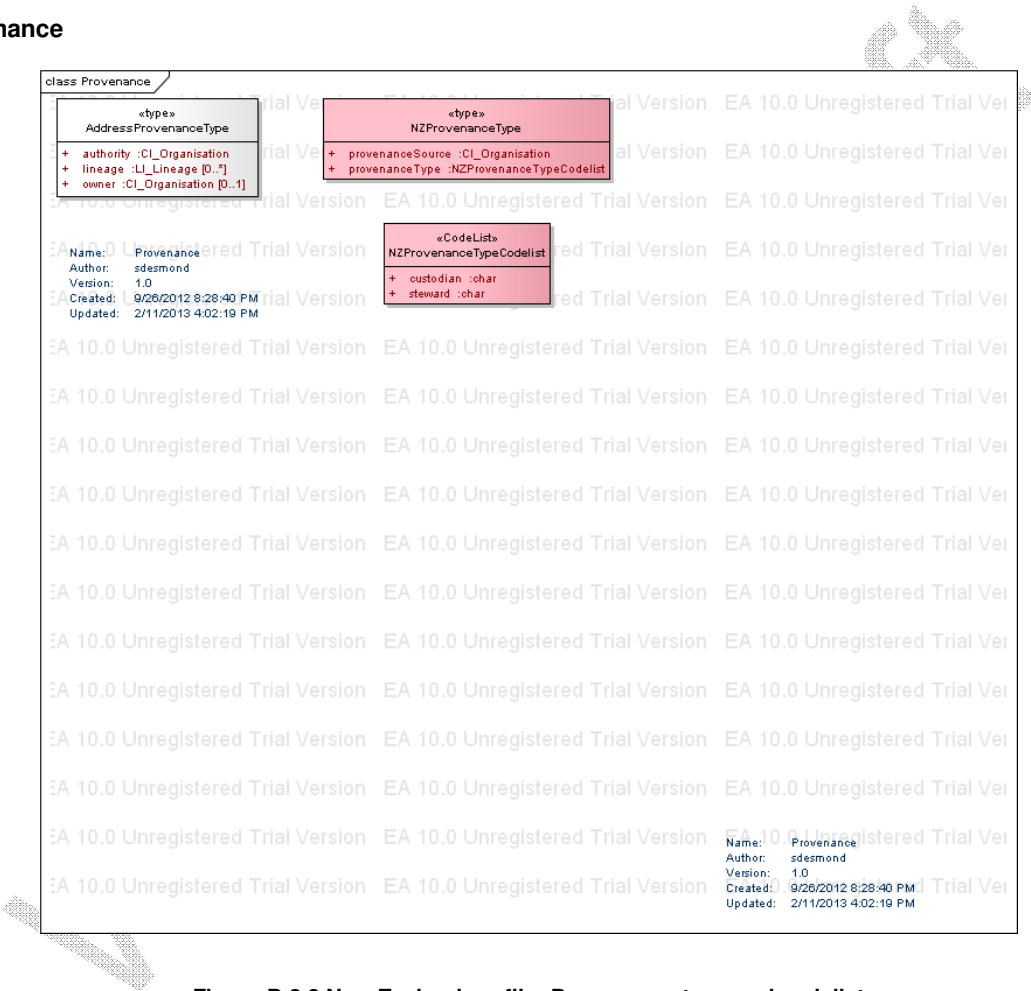


Figure B.8.3 New Zealand profile: Provenance type and codelist

NZProvenanceTypeCodeList

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	custodian	char							Responsible for ensuring appropriate care in the collection, storage, maintenance and supply of the information.
	steward	char							Responsible for ensuring that appropriate data management policies and standards are developed and maintained on behalf of the Crown. Stewardship covers all aspects of managing spatial information. This includes data architecture, quality, metadata, pricing, licensing, access and dissemination, bearing in mind any statutory responsibilities and government data management policies.

B.9 South African profile

B.9.1 Introduction

The South African profile conforms to the Core **<Lifecycle, Provenance and Locale – to be decided>** conformance class.

The South Africa profile represents addresses specified by SANS 1883-1:2009, Geographic information – Addresses, Part 1: Data format.

B.9.2 Diagrams

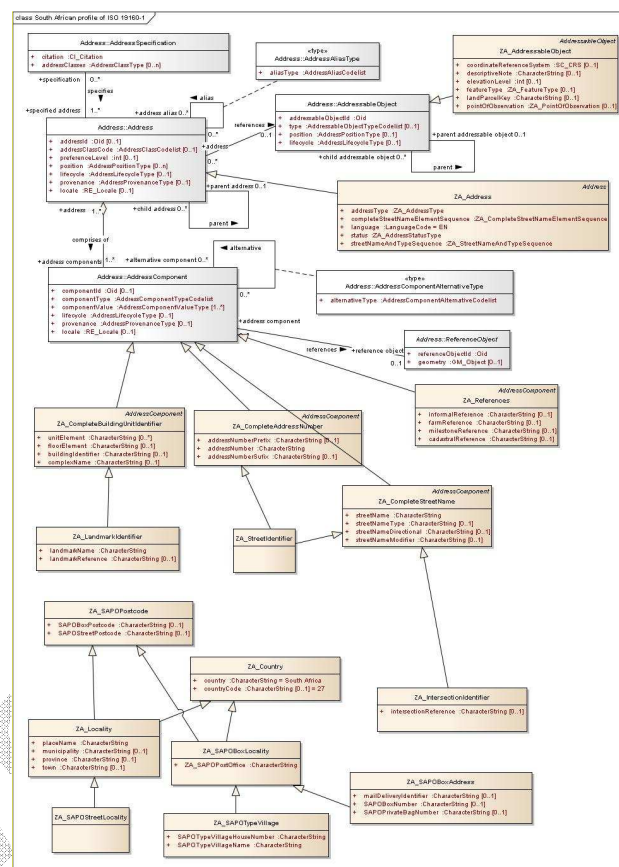


Figure B.9.1 South African profile: address model

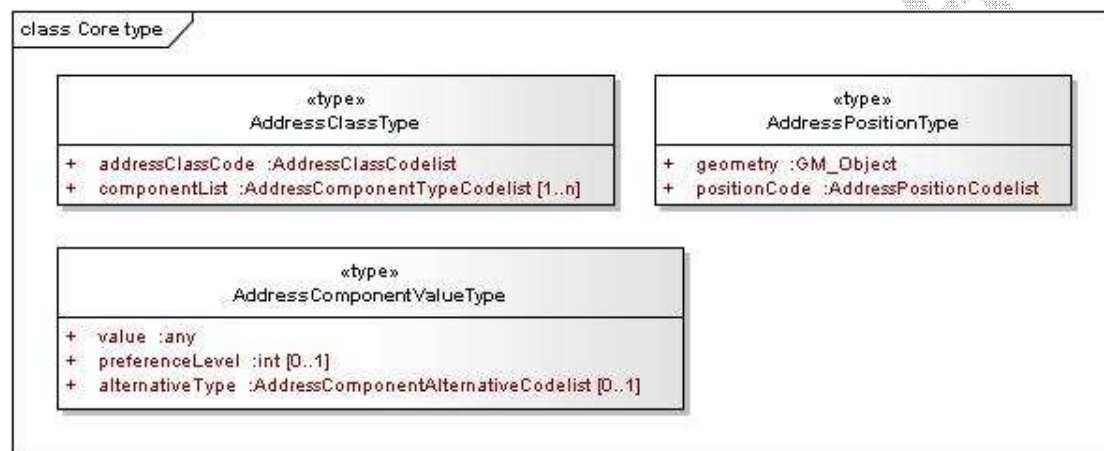


Figure B.9.2 South African profile: types

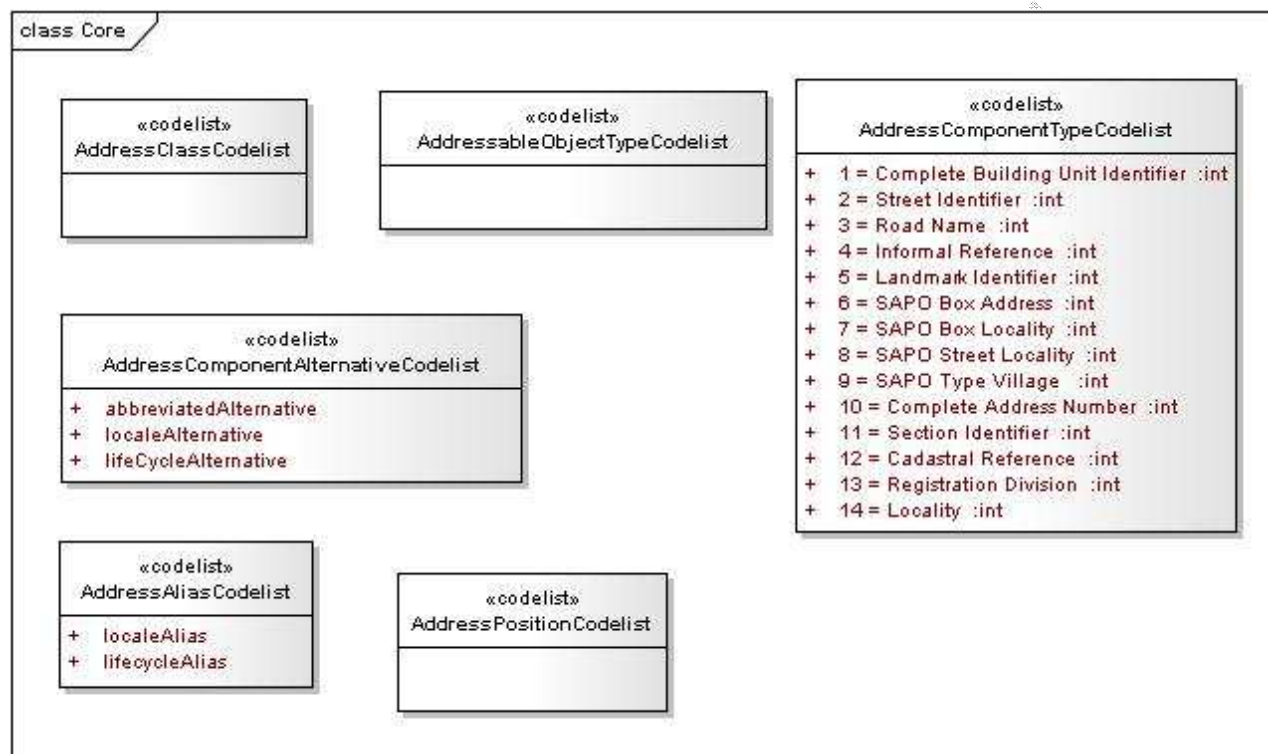


Figure B.9.3 South African profile: codelists

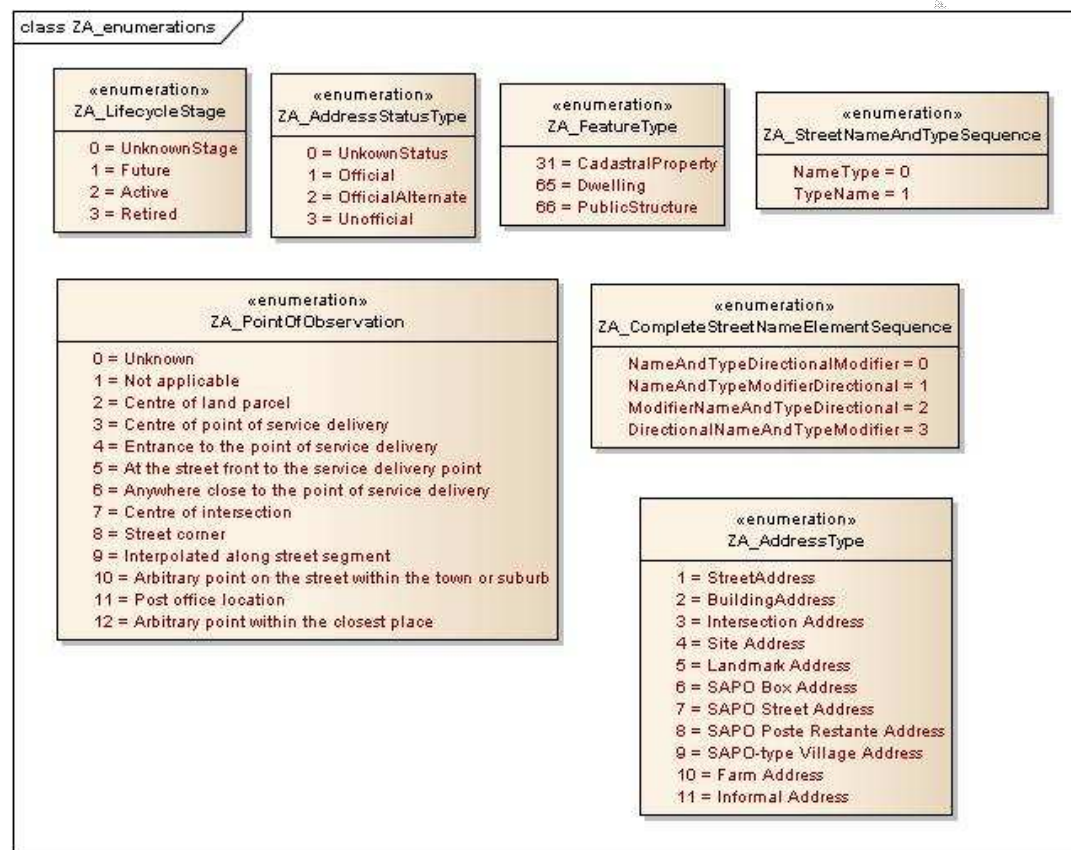


Figure B.9.3 South African profile: enumerations

Table B.9.1 South African profile: *any* type for the AddressComponentTypeCodelist

AddressComponentTypeCodelist value	Corresponding type
To be completed	

B.9.3 Attribute mapping from SANS 1883-1:2009 to the South African profile

Table B.9.2 Address mapping from <specification> to the <name> profile

<specification> profile attribute and data type	<name> profile attribute	Mapping description

Table B.9.3 AddressComponent mapping from <specification> to the <name> profile

<specification> attribute and data type	<name> profile attribute	Mapping description

B.9.4 Attribute mapping from the South African profile to SANS 1883-1:2009

Table B.9.4 Address mapping from the <name> profile to <specification>

<name> profile attribute	<specification> attribute and data type		Mapping description

Table B.9.5 AddressComponent mapping from the <name> profile to <specification>

<name> profile attribute	<specification> attribute and date type		Mapping description

B.9.5 Instance data for the South African profile

<figure placeholder>

Figure B.9.4 <name> profile: instance 1

<figure placeholder>

Figure B.9.5 <name> profile: instance 2

B.10 South Korean profile

B.10.1 Introduction

The South Korean profile conforms to the *<Core, Lifecycle, Provenance and Locale>* conformance class.

The South Korean profile represents addresses specified by *<specification>*.

B.10.2 Diagrams

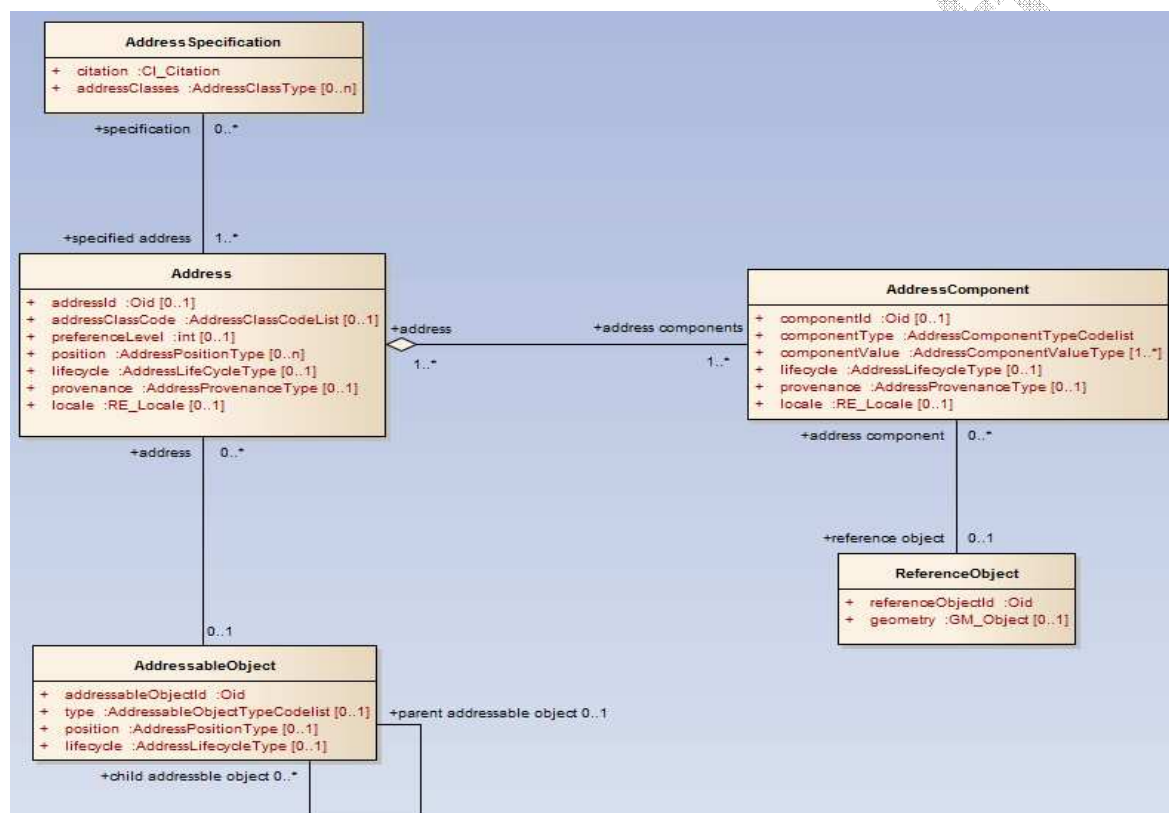


Figure B.10.1 South Korean profile: address model

<figure placeholder>

Figure B.10.2 <name> profile: types

Working Draft

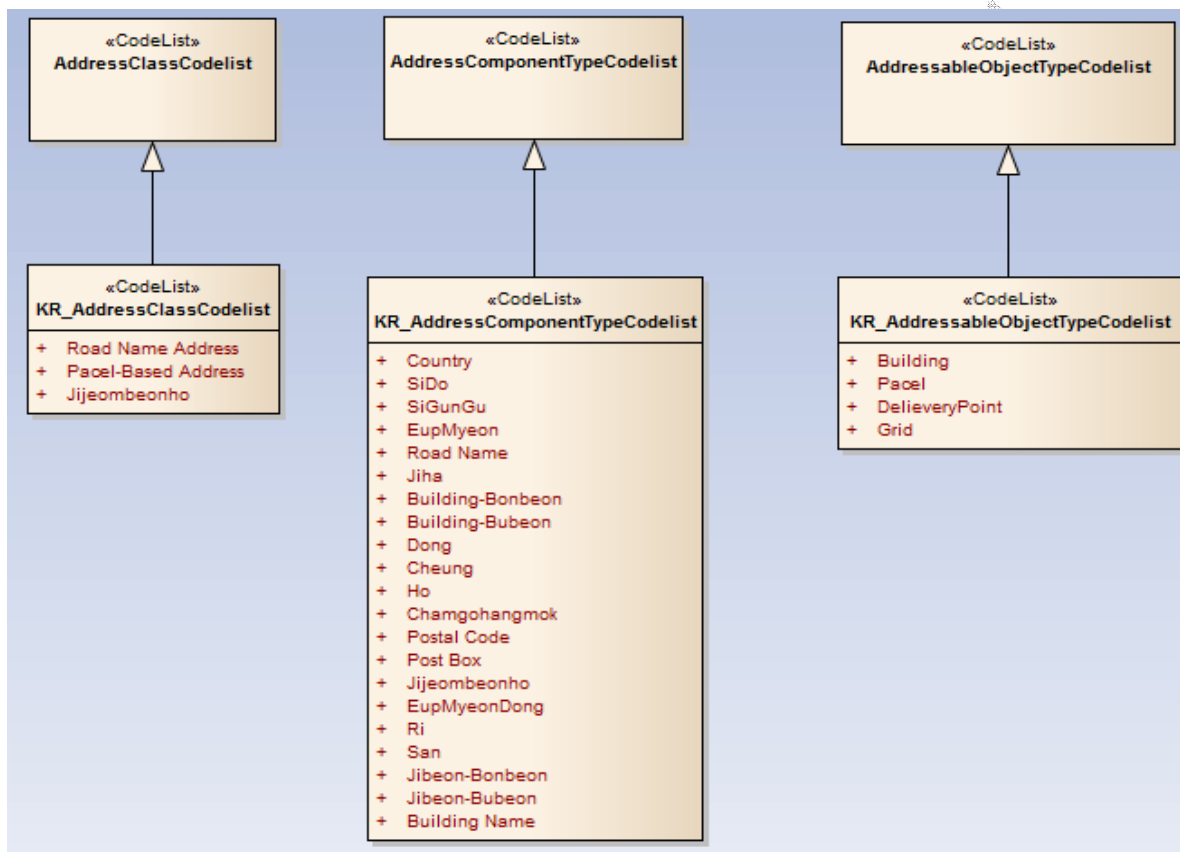


Figure B.10.3 South Korean profile: codelists

Table B.10.1 South Korean profile: *any* type for the AddressComponentTypeCodelist

AddressComponentTypeCodelist value	Corresponding type
Country	Character String
Sido	Character String
SiGunGu	Character String
EupMyeon	Character String
Road Name	Character String
Jiha	Boolean
Building-Bonbeon	Character String
Building-Bubeon	Character String
Dong	Character String
Cheung	Character String
Ho	Character String
Chamgohanmok	Character String
Postal Code0	Character String
Post Box	Character String
Jijeombeonho	Character String
EupMyeonDong	Character String
Ri	Character String
San	Boolean
Jibeon-Bonbeon	Character String
Jibeon-Bubeon	Character String
Building Name	Character String

B.10.3 Attribute mapping from <specification> to the South Korean profile

Table B.10.2 Address mapping from <specification> to the <name> profile

<specification> profile attribute and data type	<name> profile attribute	Mapping description

Table B.10.3 AddressComponent mapping from <specification> to the <name> profile

<specification> attribute and data type	<name> profile attribute	Mapping description

B.10.4 Attribute mapping from the South Korean profile to <specification>

Table B.10.4 Address mapping from the <name> profile to <specification>

<name> profile attribute	<specification> attribute and data type	Mapping description

Table B.10.5 AddressComponent mapping from the <name> profile to <specification>

<name> profile attribute	<specification> attribute and data type	Mapping description

B.10.5 Instance data for the <name> profile

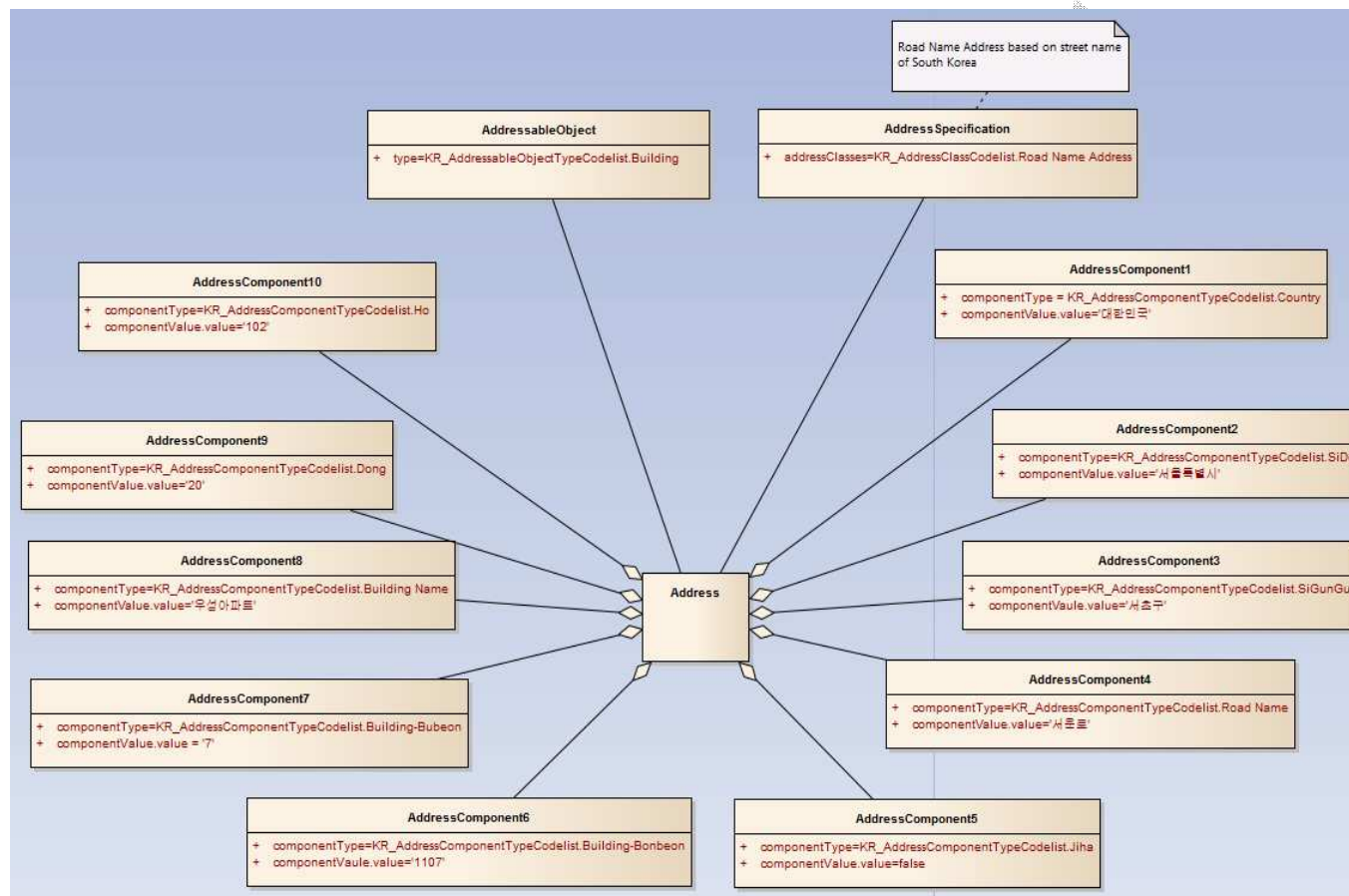


Figure B.10.4 South Korean profile: instance 1 (Road Name Address)

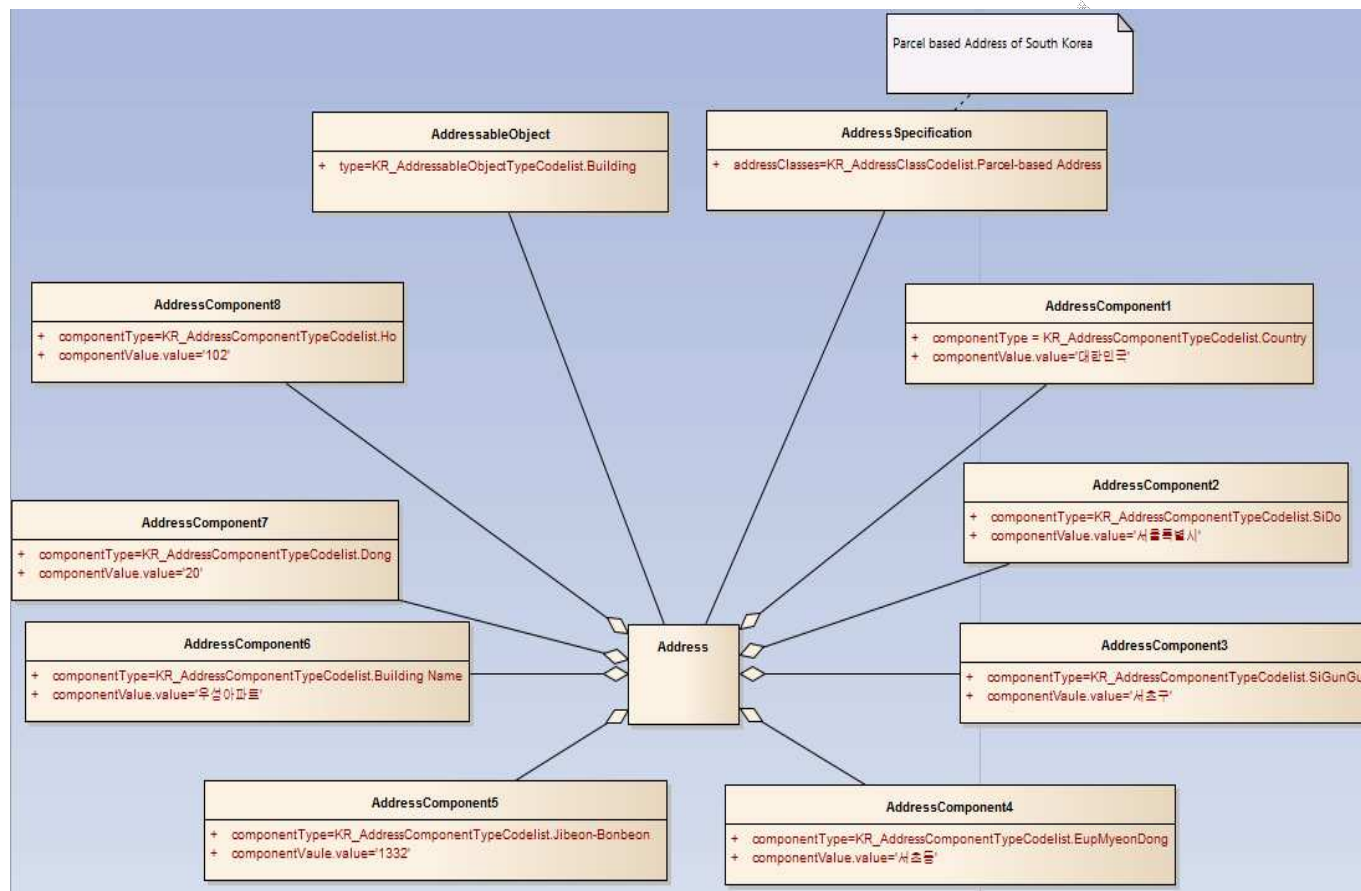


Figure B.10.5 South Korean profile: instance 2 (Parcel-based Address)

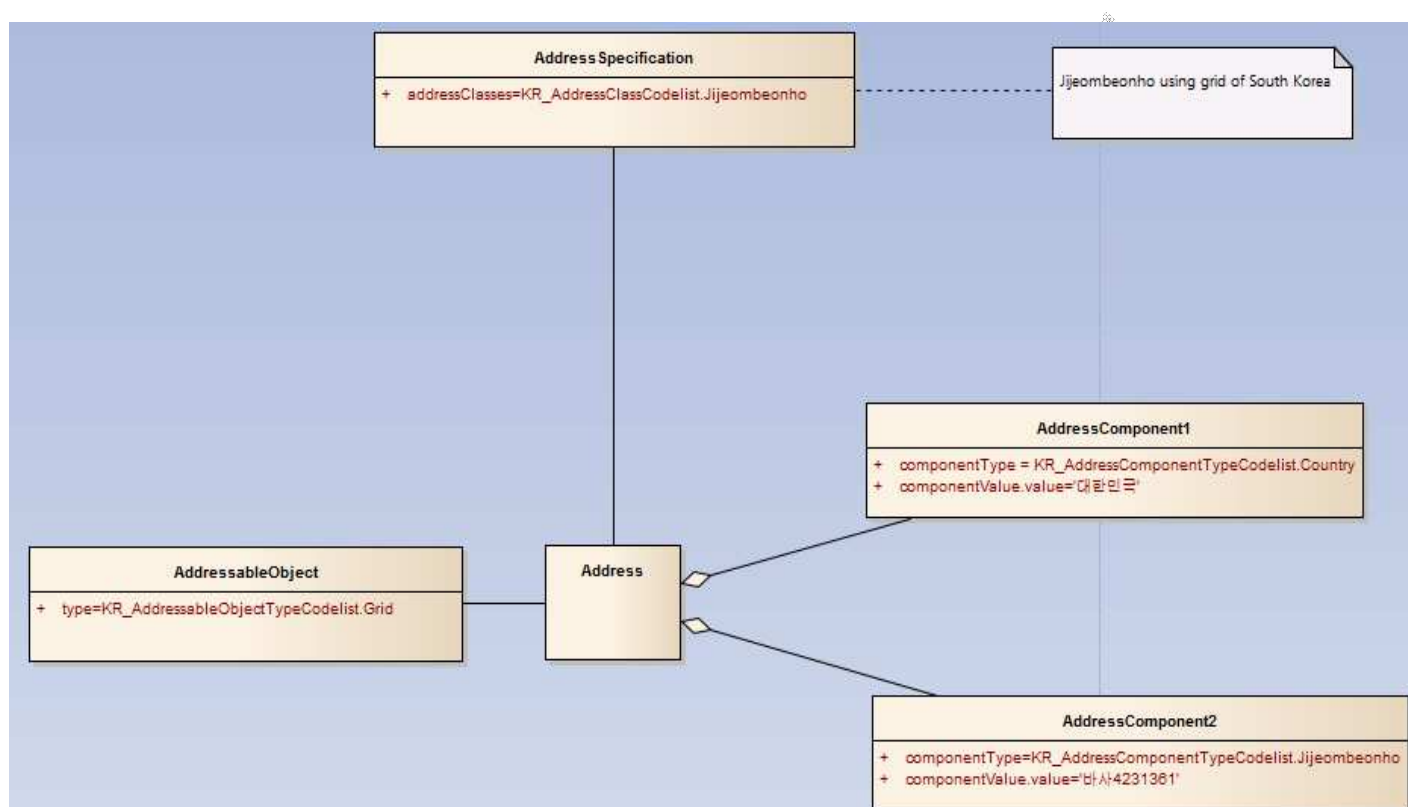
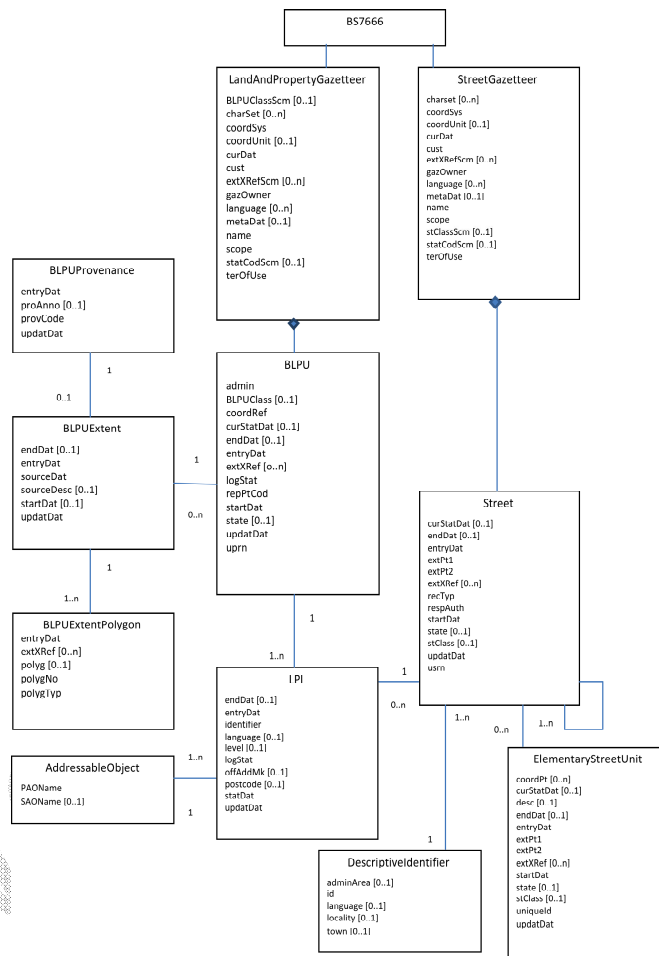


Figure B.10.6 South Korean profile: instance 3 (Jijeombeonho)

B.11 United Kingdom profile

Working Draft



B.12 UPU profile

B.12.1 Introduction

The UPU profile conforms to the <Core, **Locale**> conformance class. The UPU profile represents addresses specified by the UPU S42 (2012).

B.12.2 Diagrams

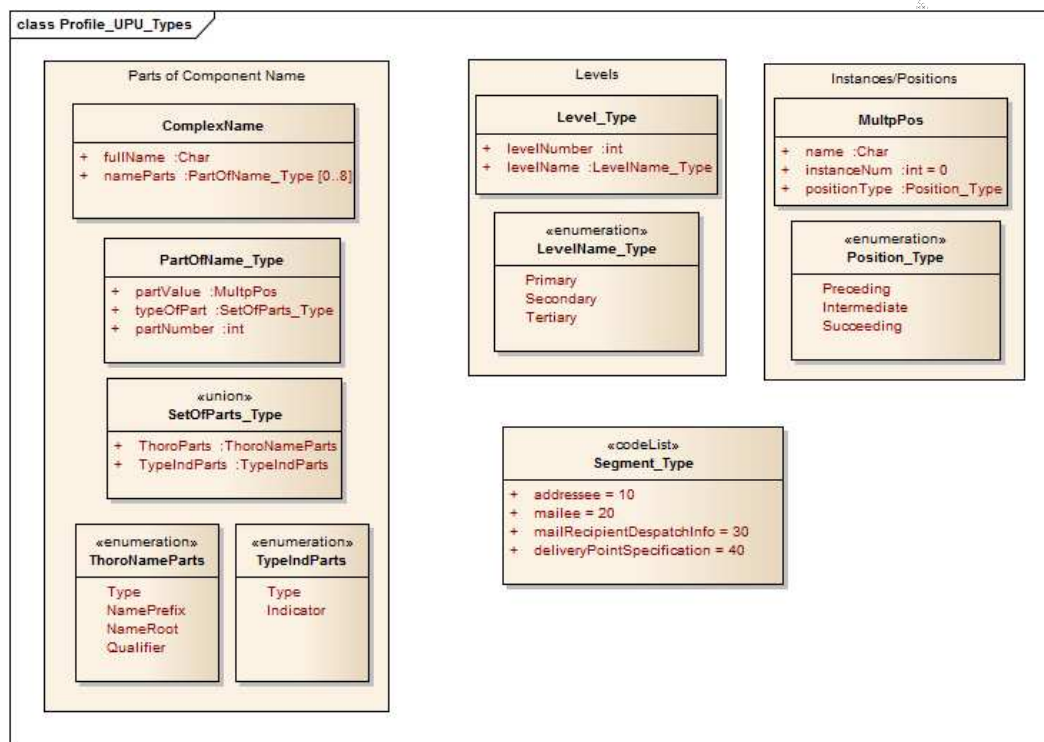


Figure B.12.2 UPU profile: types

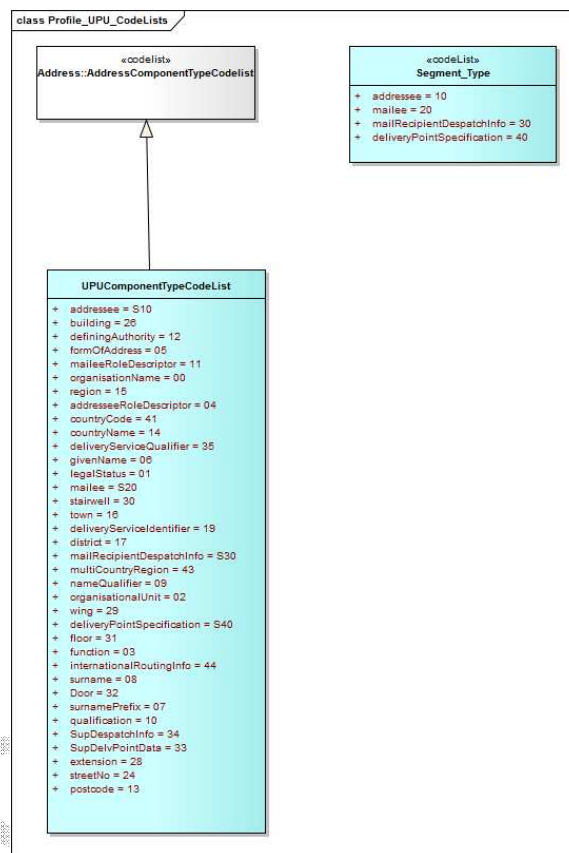


Figure B.12.3 UPU profile: codelists

Table B.12.1 UPU profile: *any* type for the AddressComponentTypeCodelist

AddressComponentTypeCodelist value	Corresponding type
PostalAddressComponent	ComplexName

B.12.3 Attribute mapping from the UPU S42 to the UPU profile

Table B.12.2 Address mapping from the UPU S42 to the UPU profile

the UPU S42 profile attribute and data type	UPU profile attribute	Mapping description

Table B.12.3 AddressComponent mapping from the UPU S42 to the UPU profile

the UPU S42 attribute and data type	UPU profile attribute	Mapping description

B.12.4 Attribute mapping from the UPU profile to the UPU S42

Table B.12.4 Address mapping from the UPU profile to the UPU S42

UPU profile attribute	UPU S42 attribute and data type		Mapping description
script (possible use of locale)	script	string	
language (possible use of locale)	language	string	

Table B.12.5 AddressComponent mapping from the UPU profile to the UPU S42

UPU profile attribute	UPU S42 attribute and date type	Mapping description
-----------------------	---------------------------------	---------------------

UPU profile attribute	UPU S42 attribute and date type		Mapping description

B.12.5 Instance data for the UPU profile

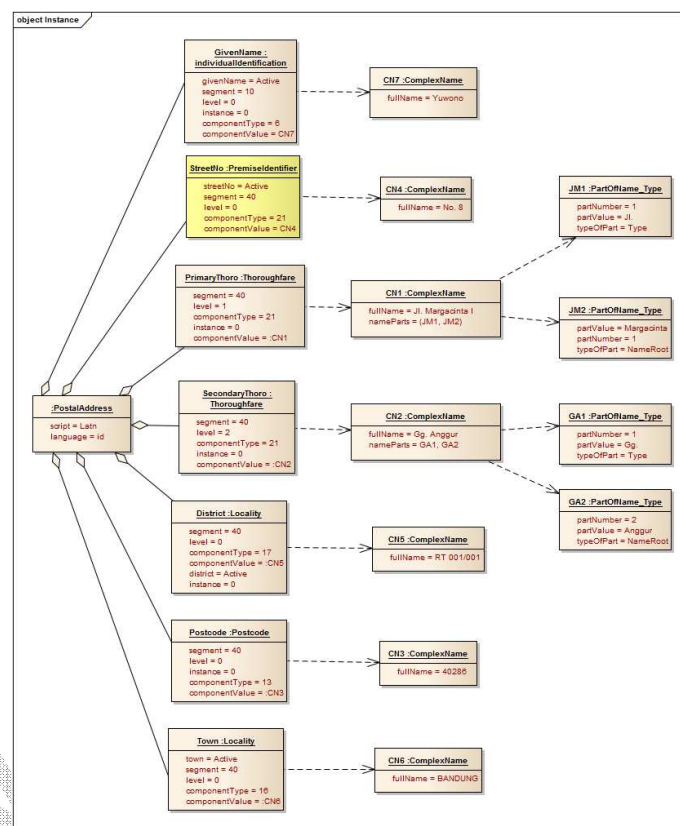


Figure B.12.4 UPU profile: instance 1

B.11 US FGDC profile

B.13.1 Introduction

The US FGDC profile conforms to the core, lifecycle, and provenance classes.

The US FGDC profile represents addresses specified by *FGDC-STD-016-2011: FGDC United States Thoroughfare, Landmark, and Postal Address Data Standard*.

B.13.2 Diagrams

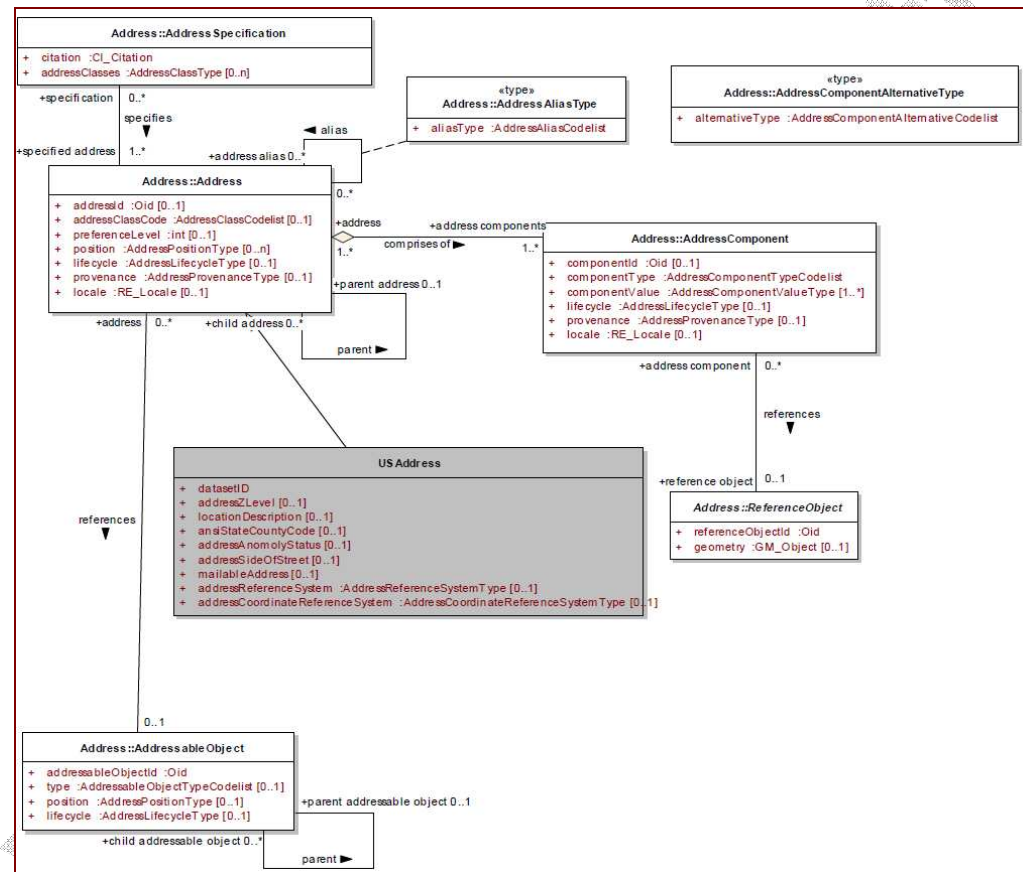


Figure B.13.1 US FGDC Profile: Address class

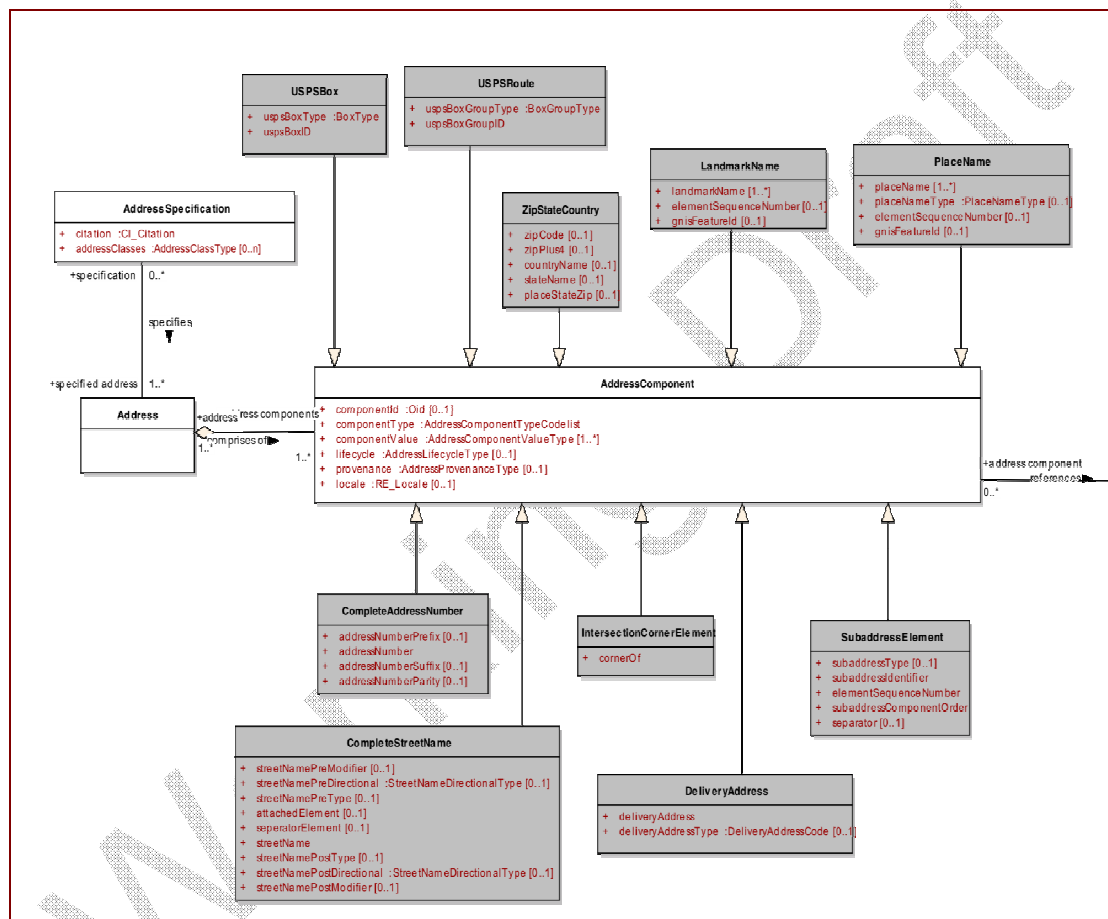


Figure B.13.2 US FGDC Profile: AddressComponent class

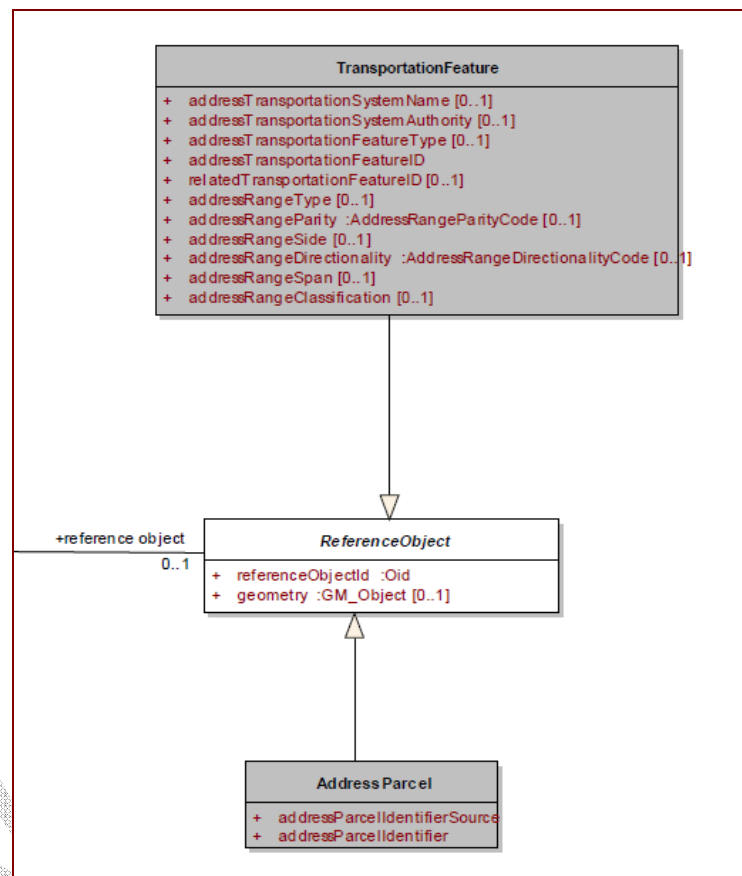


Figure B.13.3 US FGDC Profile: Reference Object Class

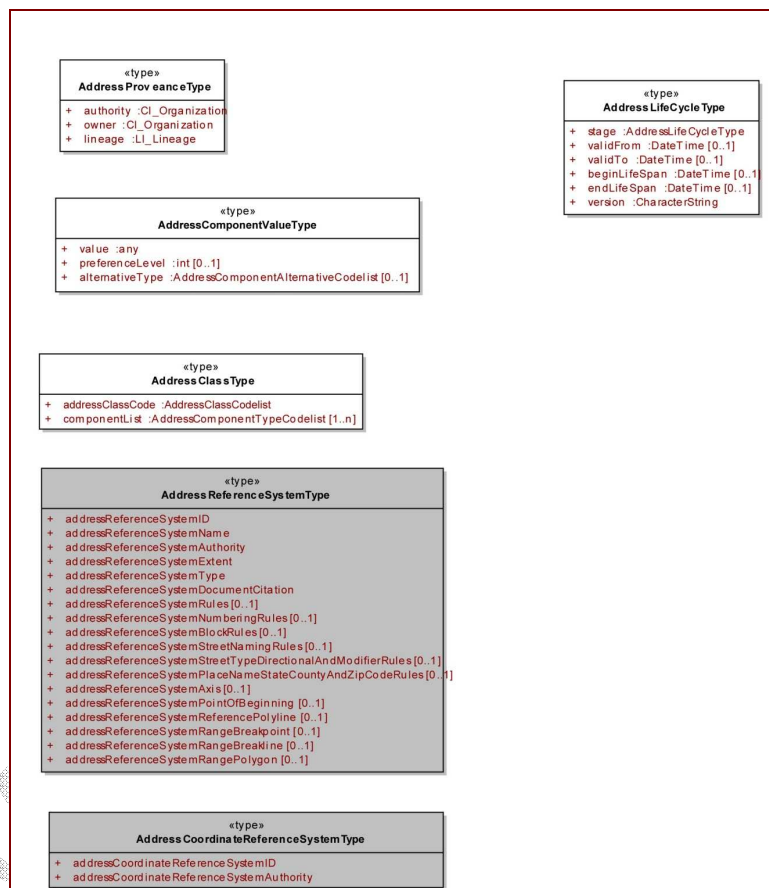


Figure B.13.4 US FGDC Profile: Types

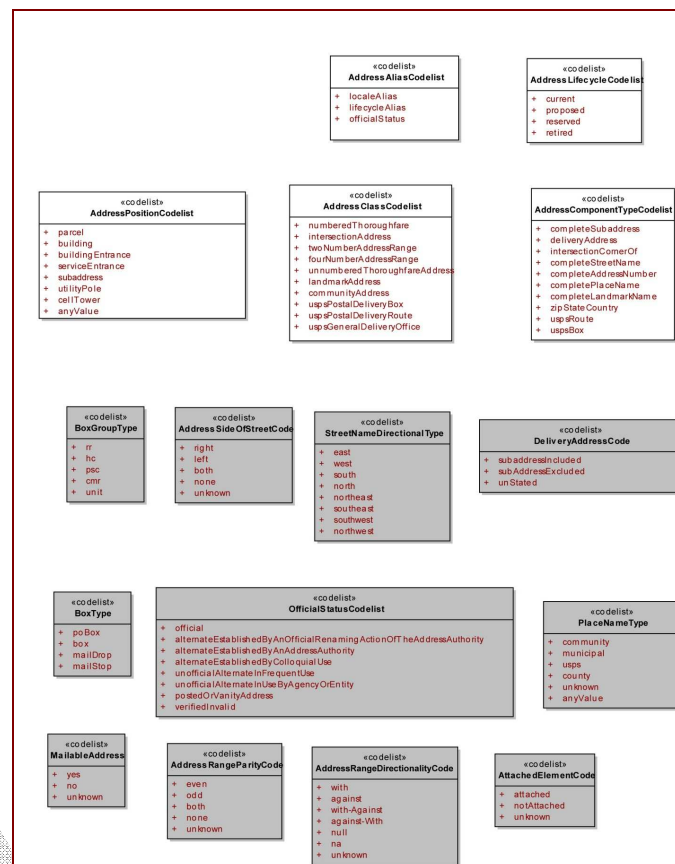


Figure B.13.5 US FGDC Profile: Codelists

B.13.3 US FGDC Mapping Tables

Table B.13.1 Address mapping from US Profile of ISO 19160-1 to US Address Standard

US Profile of ISO 19160-1 Attribute:	Class	US Address Standard Attribute and Data Type		Mapping description
addressID	Address	AddressID	characterString	Map OID to US FGDC AddressID_type
addressClassCode	Address	AddressClassification	characterString	Map from AddressClassCodeList to AddressClassification
preferenceLevel	Address	no equivalent attribute		
aliasType	Address	AddressRelationType	characterString	Map data from AddressAliasCodeType to AddressRelationType
position	Address			
geometry	Address	AddressXCoordinate, AddressYCoordinate	Real	
	Address	AddressLongitude, AddressLatitude	Real	
	Address	USNationalGridCoordinate	characterString	Described in FGDC-STD-011-2001
	Address	AddressElevation		
	Address	AddressCoordinateReferenceSystem	characterString	Combine Address CRS Authority and ID
	Address	AddressCoordinateReferenceSystemAuthority	characterString	
positionCode			Integer	
	Address	AddressCoordinateReferenceSystemID		
	Address	AddressFeatureType	characterString	Map data from AddressPositionCodeType to AddressFeatureType
lifecycle.stage	Address	AddressLifecycleStatus	characterString	Map data
lifecycle.version	Address	no equivalent attribute		
lifecycle.beginLifeSpan	Address	AddressStartDate	Date (YYYYMMDD)	Map data
lifecycle.endLifeSpan	Address	AddressEndDate	Date (YYYYMMDD)	Map data
lifecycle.validFrom	Address	no equivalent attribute		
lifecycle.validTo	Address	no equivalent attribute		
provenance.authority	Address	AddressAuthority	characterString	Map data
provenance.owner	Address	no equivalent attribute		
provenance.L1_Lineage	Address	AddressDirectSource	Text	
locale.name	Address	no equivalent attribute		
locale.language	Address	no equivalent attribute		
locale.country	Address	CountryName	characterString	Map data
locale.characterEncoding	Address	no equivalent attribute		
locale.citation	Address	no equivalent attribute		
parent.address	Address	no equivalent attribute (?)		
RelatedAddressID	Address	RelatedAddressID	characterString	Transfer
OfficialStatus	Address	OfficialStatus		Transfer
AddressAnomalyStatus	Address	AddressAnomalyStatus		Transfer
AddressSideofStreet	Address	AddressSideofStreet		Transfer
AddressLevel	Address	AddressLevel		Transfer
LocationDescription	Address	LocationDescription		Transfer
MailableAddress	Address	MailableAddress		Transfer
DataSetID	Address	DataSetID		Transfer
AddressReferenceSystemID	Address	AddressReferenceSystemID		Transfer
AddressReferenceSystemAuthority	Address	AddressReferenceSystemAuthority		Transfer

Table B.13.2 Component mapping from US Profile of ISO 19160-1 to US Address Standard

US Profile of ISO 19160-1 Attribute:	Class	US Address Standard Attribute and Data Type		Mapping description
componentId	Component	no equivalent attribute		
componentType	Component	Element Name		Map from AddressComponentType code list to US FGDC Element Name
componentValue	Component		Component values are defined in US Standard for each element type	
lifecycle.stage	Component	no equivalent attribute		
lifecycle.version	Component	no equivalent attribute		
lifecycle.beginLifeSpan	Component	no equivalent attribute		
lifecycle.endLifeSpan	Component	no equivalent attribute		
lifecycle.validFrom	Component	no equivalent attribute		
lifecycle.validTo	Component	no equivalent attribute		
provenance.authority	Component	no equivalent attribute		
provenance.owner	Component	no equivalent attribute		
provenance.l1_Lineage	Component	no equivalent attribute		
locale.name	Component	no equivalent attribute		
locale.language	Component	no equivalent attribute		
locale.country	Component	no equivalent attribute		
locale.characterEncoding	Component	no equivalent attribute		
locale.citation	Component	no equivalent attribute		
reference ReferenceObject	Component	AddressTransportationFeatureId		Key for relating feature component to transportation ReferenceObject

Table B.13.3 Addressable Object mapping from US Profile of ISO 19160-1 to US Address Standard

US Profile of ISO 19160-1 Attribute:	Class	US Address Standard Attribute and Data Type		Mapping description
addressableObjectID	Addressable Object	no equivalent attribute		
type	Addressable Object	no equivalent attribute		
position	Addressable Object	no equivalent attribute		
lifecycle.stage	Addressable Object	no equivalent attribute		
lifecycle.version	Addressable Object	no equivalent attribute		
lifecycle.beginLifeSpan	Addressable Object	no equivalent attribute		
lifecycle.endLifeSpan	Addressable Object	no equivalent attribute		
lifecycle.validFrom	Addressable Object	no equivalent attribute		
lifecycle.validTo	Addressable Object	no equivalent attribute		

Table B.13.4 Specification mapping from US Profile of ISO 19160-1 to US Address Standard

US Profile of ISO 19160-1 Attribute:	Class	US Address Standard Attribute and Data Type		Mapping description
citation	Specification		CL_Citation	
addressClasses	Specification	no equivalent attribute		

Table B.13.5 Reference Object (Transportation Feature) mapping from US Profile of ISO 19160-1 to US Address Standard – under development

US Profile of ISO 19160-1 Attribute:	Class	US Address Standard Attribute and Data Type		Mapping description
referenceObjectID	Reference Object	AddressTransportationSystemFeatureID	characterString	
geometry	Reference Object	geometry	GM_Object	
AddressTransportationSystemName	Reference Object	AddressTransportationSystemName		
AddressTransportationSystemAuthority	Reference Object	AddressTransportationSystemAuthority		
AddressTransportationSystemFeatureType	Reference Object	AddressTransportationSystemFeatureType		
RelatedTransportationSystemName	Reference Object	RelatedTransportationSystemName		
AddressRangeType	Reference Object	AddressRangeType		
AddressRangeParity	Reference Object	AddressRangeParity		
AddressRangeDirectionality	Reference Object	AddressRangeDirectionality		
AddressRangeSpan	Reference Object	AddressRangeSpan		

Table B.13.6 Reference Object (Cadastral) mapping from US Profile of ISO 19160-1 to US Address Standard – under development

US Profile of ISO 19160-1 Attribute:	Class	US Address Standard Attribute and Data Type		Mapping description
AddressParcelIdentifier				
AddressParcelIdentifierSource				

*Mappings from US Address Standard to US Profile need to be developed.

**US instance data needs to be developed.

Annex C (informative)

Examples: Life cycle information

Add explanation + examples to explain relationship between lifespan and lifecycle and validity

C.1 Life cycle stages of an address component

In 1999, the relevant authority includes the name, 'Smith Street' for a street in the plans of a town that will be constructed in future. See Table 8.

On 14 February 2000, the relevant authority activates the street name once it is built. See Table 9.

On 23 January 2005, the street is renamed to 'Mandela Street'. See Table 10.

On 20 February 2012, the street name is retired because the street is demolished during construction of a new centre. See Table 11.

Table 8 – Plan an AddressComponent

Name	Before	After
componentName	Empty	Smith Street
componentId	Empty	3847
lifeCycleStage	Empty	Planned
activeFrom	Empty	(Optional)
activeTo	Empty	(Optional)

Table 9 – Activate an AddressComponent

Name	Before	After
componentName	Smith Street	Smith Street
componentId	3847	3847
lifeCycleStage	Planned	Active
activeFrom	(Optional)	2000-02-14
activeTo	(Optional)	

Table 10 – Rename an AddressComponent

Name	Before	After	
componentName	Smith Street	Smith Street	Mandela Street
componentId	3847	3847	4938
lifeCycleStage	Active	Retired	Active
activeFrom	2000-02-14	2000-02-14	2005-01-23
activeTo		2005-01-23	

Table 11 – Retire an AddressComponent

Name	Before	After
componentName	Mandela Street	Mandela Street
componentId	4938	4938
lifeCycleStage	Active	Retired
activeFrom	2005-01-23	2005-01-23
activeTo		2012-02-20

C.2 Life cycle stages of an address

To be done

C.3 Life cycle requirement 1: Life cycle stage

The values in the LifeCycleStage shall have the following meaning:

- Planned: The address or address component are planned to be used at some point in the future.
- Active: The address or address component are actively in use.
- Retired: The address or address component are no longer in use.

C.4 Life cycle requirement 2: Address component

Tables 8-12 in this sub-clause illustrate how the componentName, componentId, lifeCycleStage, activeFrom and activeTo attributes shall be populated as the life cycle stage of an address component changes from *Planned* to *Active* to *Retired*. Instance data to further illustrate this requirements class is presented in the informative Annex C.

Table 8 – Plan an address component

Attribute	Before	After
componentName	<Empty>	<Name of the component>
componentId	<Empty>	<A new component identifier is assigned>
lifeCycleStage	<Empty>	Planned
activeFrom	<Empty>	<Empty>
activeTo	<Empty>	<Empty>

Table 9 – Activate an address component

Attribute	Before	After
-----------	--------	-------

componentName	<Name of the component>	<Name of the component>
componentId	<componentId>	<componentId>
lifeCycleStage	Planned	Active
activeFrom	<Empty>	<Date on which the address component is activated>
activeTo	<Empty>	<Empty>

Table 10 – Retire an address component

Attribute	Before	After
componentName	<Name of the component>	<Name of the component>
componentId	<componentId>	<componentId>
lifeCycleStage	Active	Retired
activeFrom	<Date on which the address component is activated>	<Date on which the address component is activated>
activeTo	<Empty>	<Date on which the address component is retired>

To be discussed/decided: show here that the lifeCycleStage of an address to which the retired component belongs, remains Active?

Table 11 – Rename an address component

Attribute	Before	After	
componentName	<Name of the component>	<Name of the component>	<New name of the component>
componentId	<componentId>	<componentId>	<A new component identifier is assigned>
lifeCycleStage	Active	Retired	Active
activeFrom	<Date on which the address component is activated>	<Date on which the address component is activated>	<Date on which the address component is renamed>
activeTo	<Empty>	<Date on which the address component is	<Empty>

		renamed>	
--	--	----------	--

To be discussed/decided: should the renamed address component have the same componentId as the old one? If yes, then the activeFrom and activeTo do not reflect when the old address (i.e. the specific combination of address components) was active, but rather that an address was for the addressable object was active for a certain period.

To be discussed/decided: show here that the lifeCycleStage of an address to which the component with the old name belongs, remains Active?

Table 12 – Spelling correction to an address component name

Attribute	Before	After
componentName	<Name of the component>	<Corrected name of the component>
componentId	<componentId>	<componentId>
lifeCycleStage	Active	Active
activeFrom	<Date on which the address component is activated>	<Date on which the address component is activated>
activeTo	<Empty>	<Empty>

C.5 Life cycle requirement 3: Address

Tables 13-16 in this sub-clause illustrate how the addressComponentSet, addressId, lifeCycleStage, activeFrom and activeTo attributes shall be populated as the life cycle stage of an address changes from *Planned* to *Active* to *Retired*. Instance data to further illustrate this requirements class is presented in the informative [Annex C](#).

Table 13 – Plan an address

Attribute	Before	After
addressComponentSet	<Empty>	<Set of components>
addressId	<Empty>	<A new address identifier is assigned>
lifeCycleStage	<Empty>	Planned
activeFrom	<Empty>	<Empty>
activeTo	<Empty>	<Empty>

Table 14 – Activate an address

Attribute	Before	After
addressComponenSet	<Set of components>	<Set of components >
addressId	<addressId>	<addressId>
lifeCycleStage	Planned	Active
activeFrom	<Empty>	<Date on which the address is activated>
avtiveTo	<Empty>	<Empty>

Table 15 – Retire an address

Attribute	Before	After
addressComponenSet	<Set of components>	<Set of components>
addressId	<addressId>	<addressId>
lifeCycleStage	Active	Retired
activeFrom	<Date on which the address is activated>	<Date on which the address is activated>
avtiveTo	<Empty>	<Date on which the address is retired>

Table 16 – Change an address

Attribute	Before	After	
addressComponenSet	<Set of components>	<Set of components>	<New set of components>
addressId	<addressId>	<addressId>	<A new address identifier is assigned>
lifeCycleStage	Active	Retired	Active
activeFrom	<Date on which the address component is activated>	<Date on which the address component is activated>	<Date on which the address component is renamed>
avtiveTo	<Empty>	<Date on which the	<Empty>

		address component is renamed>	
--	--	-------------------------------	--

To be discussed/decided: should the renamed address have the same addressId as the old one? If yes, then the activeFrom and activeTo do not reflect when the old address (i.e. the specific combination of address components) was active, but rather that an address was for the addressable object was active for a certain period.

C.6 Life cycle requirement 4: Addressable object

The tables in this sub-clause illustrate how the lifeCycleStage, existsFrom and existsTo attributes shall be populated as the life cycle stage of an addressable object changes from *Planned* to *Active* to *Retired*. Instance data to further illustrate this requirements class is presented in the informative Annex C.

To be discussed/decided: do we need this requirement? If yes, table needs to be added.

Annex D

(informative)

Examples: Identical addresses and address aliases

"The same" address can be expressed in various languages, using abbreviated or full names of its components, split to elements according to various specifications.

The same addressable object can have addresses using two different sets of components.

Annex E

(informative)

Ontology for the ISO 19160-1 conceptual model

To discuss/decide: Exact form of URIs for ontologies still under discussion in ISO 19150-2

To discuss/decide: The ISO/TC211 repository for ontologies is not yet available and the ontology files are therefore not available at the URI

F.1 ISO 19160-1 conceptual model represented in OWL (RDF XML)

F.2 ISO 19160-1 conceptual model represented in OWL (Turtle)

To be done

Examples: ReferenceObject in a profile

Figure F.1.1 shows how external data can be associated with an address component in a profile of the address model: classes are derived from the abstract class `ReferenceObject`, such as the `ReferenceSpatialObject`, `Organization` and `Person` classes in the diagram in Figure F.1.1. If classes for the external data already exist, for example, `LA_SpatialUnit` and `LA_SpatialUnitGroup`, an association to the `ReferenceSpatialObject` (or other class derived from `ReferenceObject`) is added.

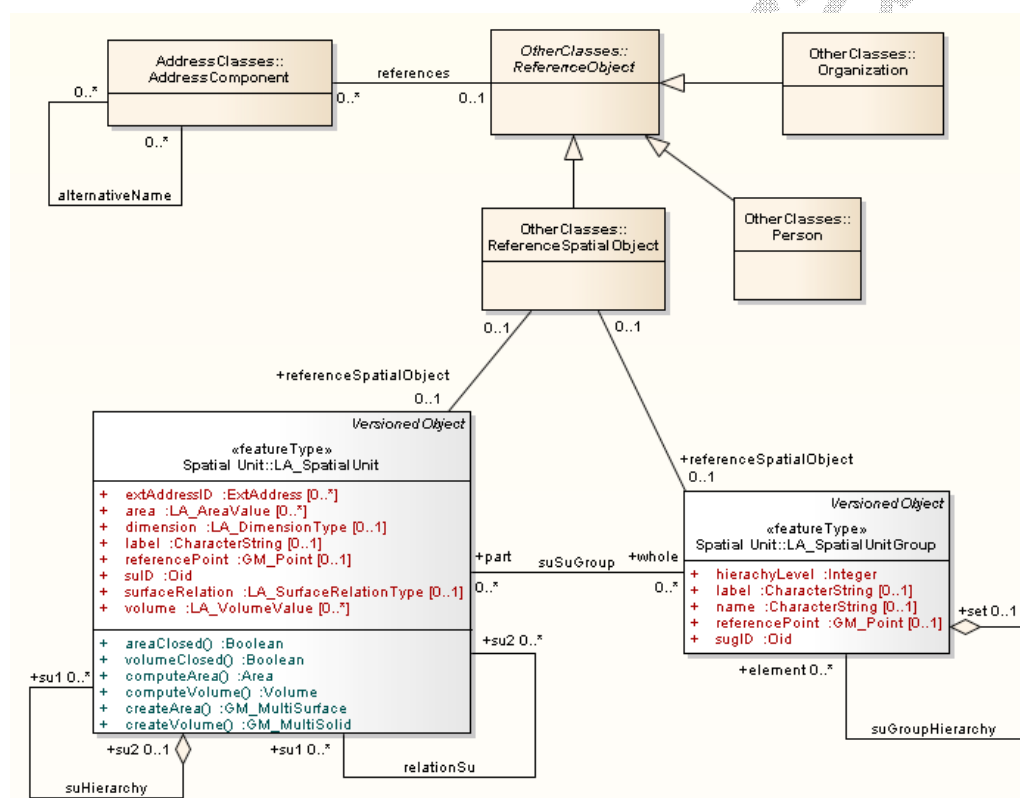


Figure F.1.1 Associating external data via ReferenceObject with AddressComponent

Bibliography

- [1] BS 7666....
- [2] D2.8.1.5 *INSPIRE Data Specification on Addresses – Guidelines*, INSPIRE Thematic Working Group on Addresses, 2010.
- [3] ISO 19103:2005, *Geographic information – Conceptual schema language*
- [4] ISO 19107:2003, *Geographic information – Spatial schema*
- [5] ISO 19111:2007, *Geographic information – Spatial referencing by coordinates*
- [6] ISO 19135:2005, *Geographic information – Procedures for item registration*
- [7] Japanese act
- [8] Korean standard + act
- [9] *Review summary of the ISO 19160 stage zero project*. ISO/TC 211, *Geographic information/Geomatics*, document n3188.
- [10] ISO 21127:2006, *Information and documentation -- A reference ontology for the interchange of cultural heritage information*.
- [11] SANS 1883:2009, *Geographic information – Addresses*, South African Bureau of Standards (SABS), Pretoria, South Africa.
- [12] UPU S42:2006, *International postal address components and templates*, Universal Postal Union, Berne, Switzerland.