

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

Date: 2013-01-15

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) Proposal**
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.

Working Draft

Contents

Page

Foreword	v
Introduction.....	vi
1 Scope.....	1
2 Conformance	1
2.1.1 Core	1
2.1.2 Lifecycle.....	1
2.1.3 Provenance.....	1
2.1.4 Locale.....	1
2.1.5 Full conformance	1
3 Normative references	2
4 Terms and definitions.....	2
5 Symbols and abbreviations	4
6 Address model	5
6.1 Introduction.....	5
7 Requirements class: Core.....	9
7.1 Dependencies	9
7.2 Core requirement 1: Classes.....	9
7.3 Core requirement 2: Associations.....	10
7.4 Core requirement 3: Attributes	11
7.5 Codelists	14
8 Requirements class: Lifecycle	15
8.1 Dependencies	15
8.2 Lifecycle requirement 1: Attributes.....	15
8.3 Codelists	17
9 Requirements class: Provenance	17
9.1 Dependencies	17
9.2 Provenance requirement 1: Attributes.....	18
10 Requirements class: Locale	19
10.1 Dependencies	19
10.2 Locale requirement 1: Attributes.....	19
Annex A (normative) Abstract test suites	20
A.1 Introduction.....	20
A.2 Conformance class: Core	20
A.2.1 Introduction	20
A.2.2 Core test 1: Classes	20
A.2.4 Core test 2: Associations	20
A.2.6 Core test 3: Attributes.....	20
A.3 Conformance class: Lifecycle.....	21
A.3.1 Introduction	21
A.3.2 Lifecycle test 1: Attributes.....	21
A.4 Conformance class: Provenance.....	21
A.4.1 Introduction	21
A.4.2 Provenance test 1: Attributes.....	21
A.5 Conformance class: Locale.....	21
A.5.1 Introduction	21
A.5.2 Locale test 1: Attributes.....	22

Annex B (informative) Profiles and instance data	23
B.1 Profiles	23
B.1.1 Introduction	23
B.1.2 Steps followed to develop a profile.....	23
B.1.3 Profile diagrams	24
B.2 Minimal profile	25
B.2.1 Introduction	25
B.1.2 Diagrams	25
B.1.3 Instance data for the minimal profile	26
B.2 Addresses in cultural heritage information.....	27
B.2 INSPIRE profile.....	27
B.3 Japan	27
B.4 South Africa	28
B.1.3 Attribute mapping from a minimal address to the minimal profile	30
B.1.4 Attribute mapping from the minimal profile to a minimal address	31
B.5 South Korea	32
B.5.1 Profile	Error! Bookmark not defined.
B.5.2 Instance data	Error! Bookmark not defined.
B.6 US FGDC	32
B.6 United Kingdom.....	32
B.7 UPU S42	32
Annex C (informative) Examples: Life cycle information.....	33
C.1 Life cycle stages of an address component	33
C.2 Life cycle stages of an address	35
C.3 Life cycle requirement 1: Life cycle stage.....	35
C.4 Life cycle requirement 2: Address component.....	35
C.5 Life cycle requirement 3: Address	37
C.6 Life cycle requirement 4: Addressable object	39
Annex D (informative) Examples: Identical addresses and address aliases	40
Annex E (informative) Ontology for the ISO 19160-1 conceptual model.....	41
F.1 ISO 19160-1 conceptual model represented in OWL (RDF XML).....	41
F.2 ISO 19160-1 conceptual model represented in OWL (Turtle).....	41
Annex F (informative) Examples: ReferenceObject in a profile	42
F.1 Introduction	42
Bibliography	43

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.

A wide variety of address standards and/or specifications is 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 are well integrated into various operational processes and in some cases legally enforced. At the same time, some countries are rationalising their addressing systems 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.

There are very 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 intention to develop a number of related addressing standards.

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. Life cycle, 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 two classes of requirements and conformance. Requirements and conformance URLs in this document are defined relative to <http://www.iso211.org/is/ISO19160/2012/02/>.

2□□□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□3□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□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□3□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□3□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:____¹, *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 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.

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

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

EXAMPLE 2 In Japan, a block (gaiku) is a parent addressable object.

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 Introduction

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.

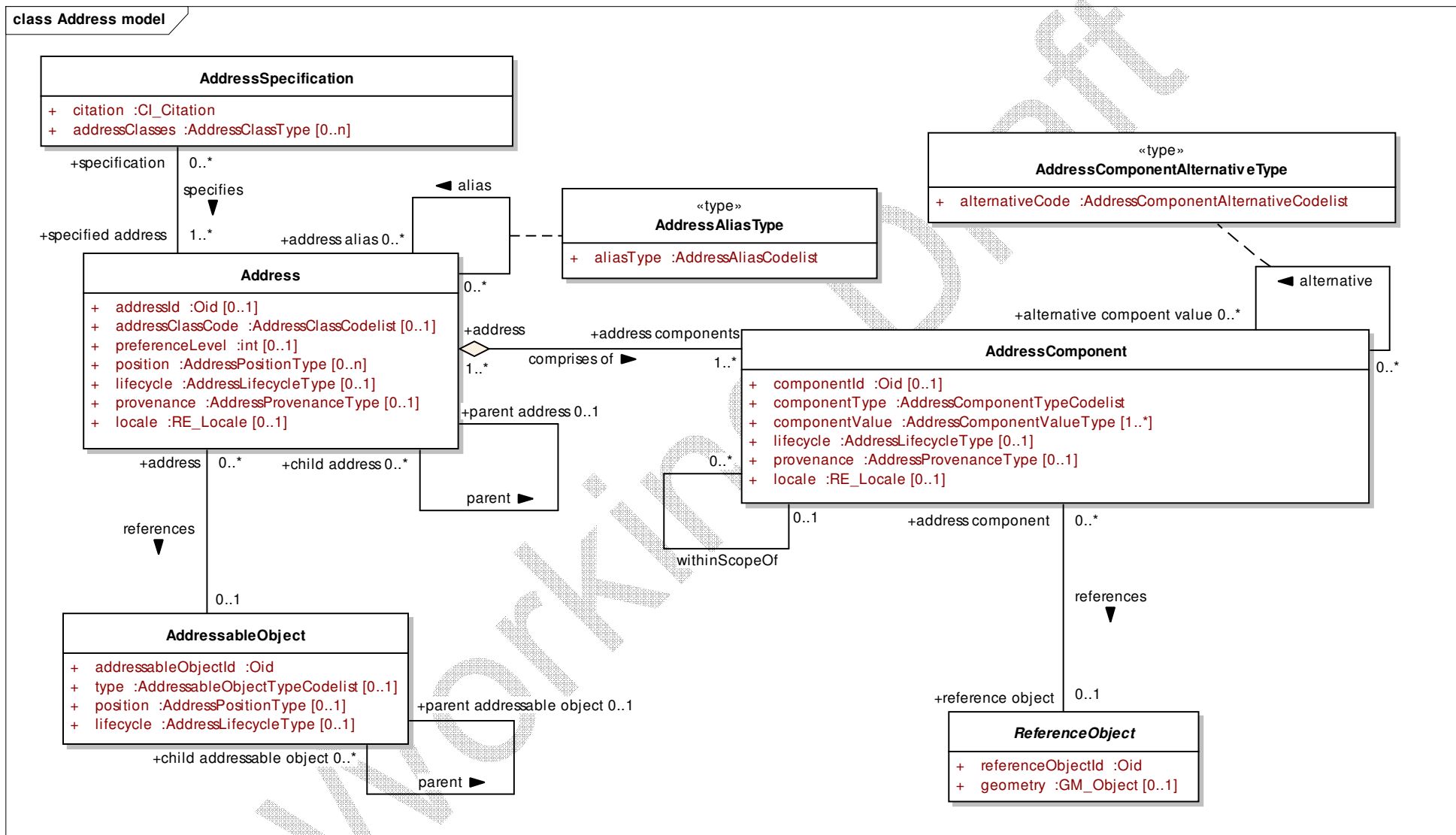


Figure 1 – Address model

Figures 2-4 show the types and codelists defined for the Core, Lifecycle and Provenance requirements classes respectively.



Figure 2 – Types and codelists 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 addressLine, streetName, placeName and postcode are examples of codes for the AddressComponentType.

EXAMPLE 3 streetAddress, landmarkAddress and informalAddress are examples of codes for the AddressClassCodelist.

EXAMPLE 4 centroid, streetFront and approximated are examples of codes for the AddressPositionCodelist.

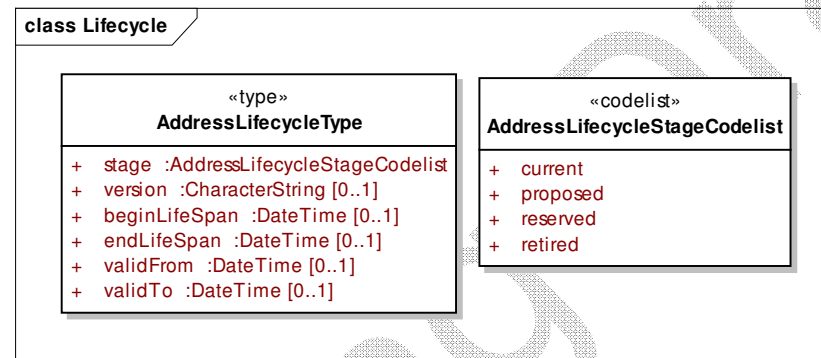


Figure 3 – Types and codelists in the Lifecycle requirements class

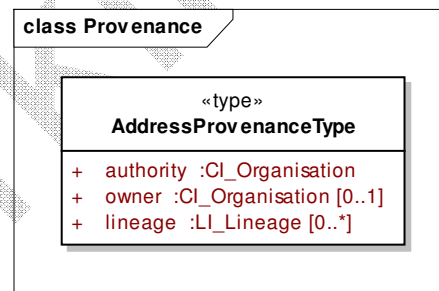


Figure 4 – Types and codelists in the Provenance requirements class

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 component types are thoroughfare name, place name, addressee and postcode.

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

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, i.e. address components with alternative component values. There are different types of alternative values, e.g. values in different languages or values used at different lifecycle stages.

NOTE Different names of address components (e.g. town names in different languages) to identify the same addressable object are alternative component values (i.e. NOT the same address component).

NOTE A spelling correction to the name of an address component does NOT create a new 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, a place name in two different languages (e.g. Köln and Cologne) are language alternatives of a place name.

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

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

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

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 component types, which are valid 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	Oid	See ISO 19152:2013
addressClassCode	Code of the address class to which the address belongs.	O	1	Class	<<codelist>> AddressClassCodelist
preferenceLevel	Indicates the level of preference of the address to use from a set of aliases. 1 indicates highest preference. EXAMPLE 1 A building on a street corner could be referenced by two addresses. One of them has the preferenceLevel set to 1. EXAMPLE 2	O	1	integer	Integer > 0

	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 19152:2013
componentType	The type of the component.	M	1	Class	<<codelist>> AddressComponentTypeCodelist
componentValue	Value of the component.	M	1	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 19152:2013
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 spatial object.	M	1	Class	Oid See ISO/DIS 19152:2013

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	<<codelist>> AddressClassCodelist
componentList	A list of valid address component types.	M	N	Class	<<codelist>> AddressComponentType

Table 8 – AddressPositionType attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
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	0	Class	<<codelist>> AddressPositionCodelist

Table 9 – AddressComponentValueType attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
componentValue	Value of the component.	M	1	any	
componentValueDataType	Data type of the componentValue attribute.	M	1	Class	<<codelist>> AddressComponentValueDataTypeCodelist
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

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 – AddressComponentAlternativeTypeCodelist 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.

AddressComponentValueDataTypeCodelist → add table for this codelist ...

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 15 – 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
version	Information to specify the version of the data object representing the address.	O	1	CharacterString	Free text
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	O	1	Class	DateTime Character encoding of a DateTime shall follow ISO 8601. This

	dataset.				class is documented in full in ISO/TS 19103:2005.
--	----------	--	--	--	---

8.3 Codelists

Table 16 – 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 17 – 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 18 – 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 19 – 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 shall have one or more metadata attributes specified in ISO 19115:2013. (This was proposed in Jeddah, does it need more discussion? If all agree, remove Table 18)

Table 20 – AddressProvenanceType attributes

Name	Definition	Mandatory/ Conditional/ Optional	Max Occur	Data Type	Domain
authority	Authority that assigns the address.	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 21 – Dependencies of the Locale requirements class

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

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 22 – Address 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.	M	1	Class	RE_Locale See ISO 19135:2005

Table 23 – 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

Conformance class identifier: Core

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

Conformance class identifier: Lifecycle

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

Conformance class identifier: Provenance

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

Conformance class identifier: Locale

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

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 – 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.1.2 Diagrams

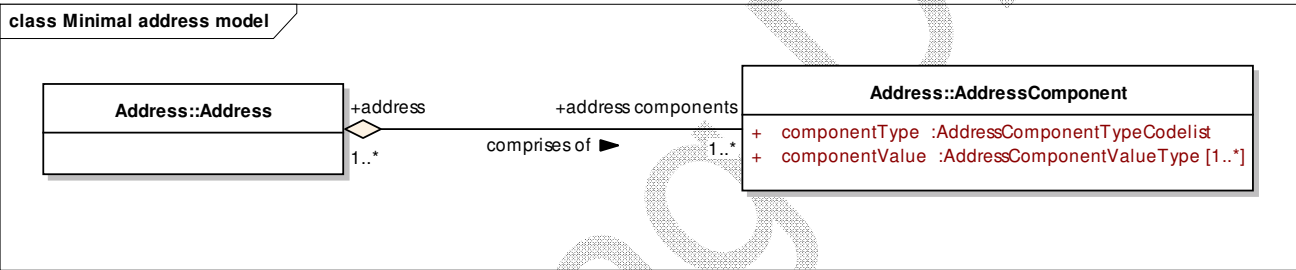


Figure B.2.1 Minimal profile: address model

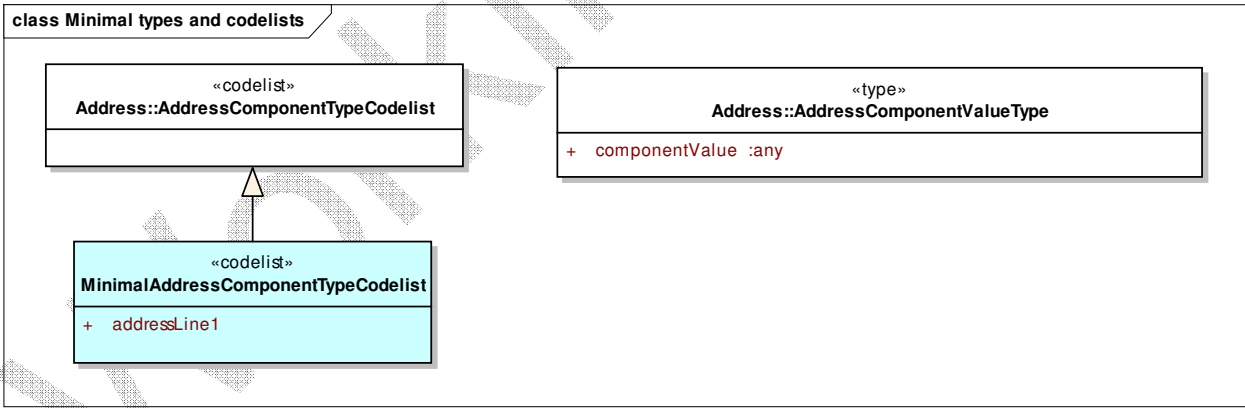


Figure B.2.2 Minimal profile: types and codelists

Table B.1 any type for the AddressComponentTypeCodelist

AddressComponentTypeCodelist value	Corresponding type
addressLine	CharacterString

B.1.3 Instance data for the minimal profile

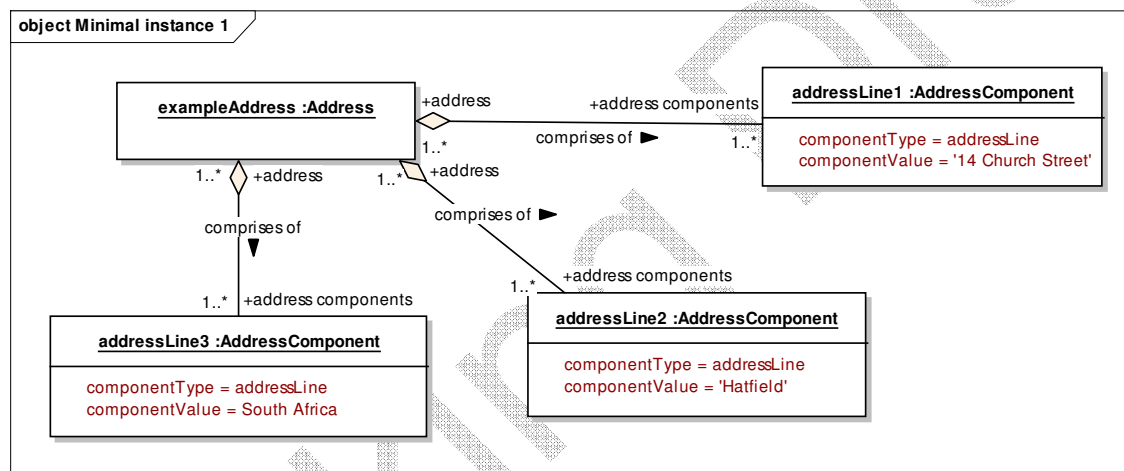


Figure B.2.3 Minimal profile: instance 1

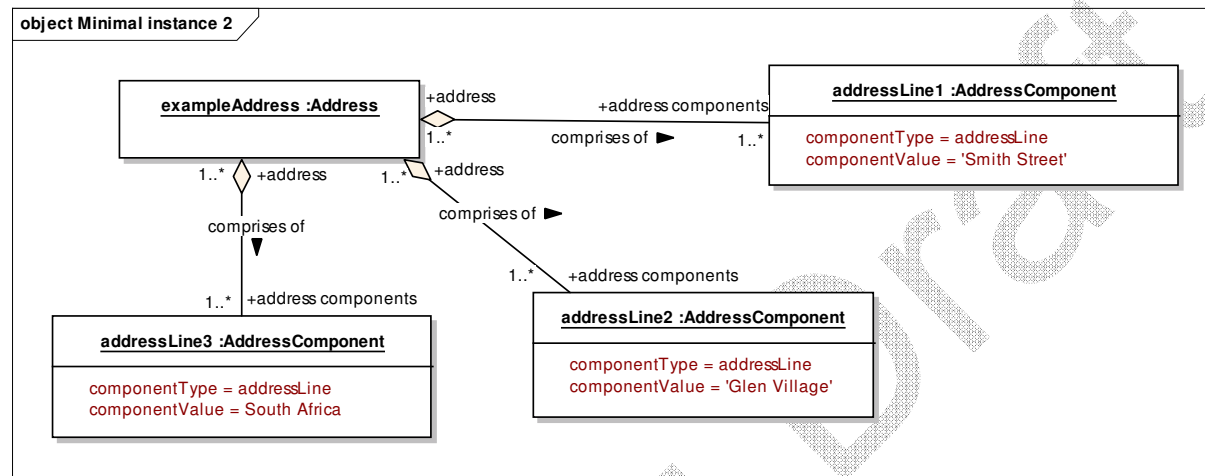


Figure B.2.4 Minimal profile: instance 2

B.2 Addresses in cultural heritage information

Under development

B.2 INSPIRE profile

Under development

B.3 Japan

Under development

Below is the original table with the Japanese examples, which need to be converted into a UML profile and instance data.

B.4 South Africa

B.4.1 Introduction

The South African profile has *Full* conformance with ISO 19160-1.

The South African profile is based on SANS 1883-1:2009, *Geographic information – Part 1: Data format of addresses*.

(Incomplete profile at the moment...)

B.4.2 Diagrams

<image>

Figure B.4.1 South African profile: address model

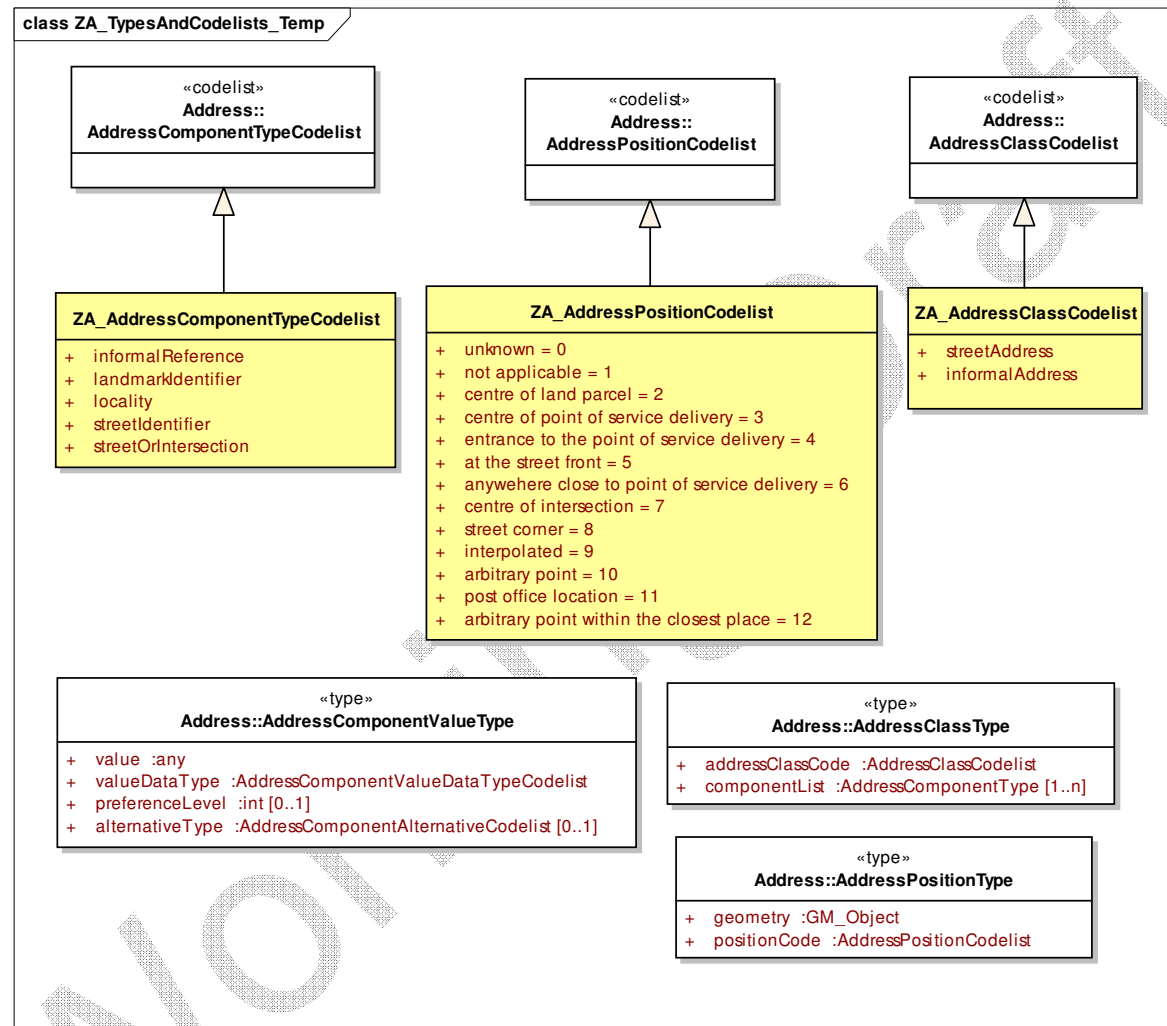


Figure B.4.2 South Africa profile: types and codelists

Table B.4.1 any type for the AddressComponentTypeCodelist

AddressComponentTypeCodelist value	Corresponding type
informalReference	CharacterString
intersectionIdentifier	CharacterString
landmarkIdentifier	CharacterString
locality	CharacterString
streetIdentifier	CharacterString

B.4.3 Attribute mapping from a SANS 1883 address to the South African profile

Table B.4.2 Address mapping from SANS 1883-1:2009 to the South African profile

South African profile attribute	SANS 1883-1:2009 attribute and data type		Mapping description
addressId.localId	Address.addressID	Alphanumeric	
addressId.namespace			
preferenceLevel	No equivalent attribute		
position.geometry	<Korean name>	???	
position.positionCode			

Table B.4.3 AddressComponent mapping from SANS 1883-1:2009 to the South African profile

South African profile attribute	South African attribute and data type		Mapping description
componentId			
componentType			
componentValue.value			

B.1.4 Attribute mapping from the South African profile to a SANS 1883:2009 address

Table B.2.6 Address mapping from the South African profile to SANS 1883-1:2009

South African profile attribute	SANS 1883-1:2009 attribute and data type		Mapping description
addressId.localId	Address.AddressID	Alphanumeric	Convert Address.addressId.localId from CharacterString to integer. If it is not a number...
addressId.namespace	No equivalent attribute		Discard
preferenceLevel	No equivalent attribute		Discard
position.geometry			
position.positionCode			

Table B.2.7 AddressComponent mapping from the South African profile to SANS 1883-1:2009

XYZ profile attribute	XYZ attribute and date type		Mapping description
componentId			
componentType			

B.4.5 Instance data for the South African profile

<image>

Figure B.2.3 South African profile: instance of a street address

<image>

Figure B.2.4 South African profile: instance of an informal address

B.5 South Korea

Under development

B.6 US FGDC

Under development

B.6 United Kingdom

Under development???

B.7 UPU S42

Under development

Working Draft

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

Annex F (informative)

Examples: ReferenceObject in a profile

F.1 Introduction

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.

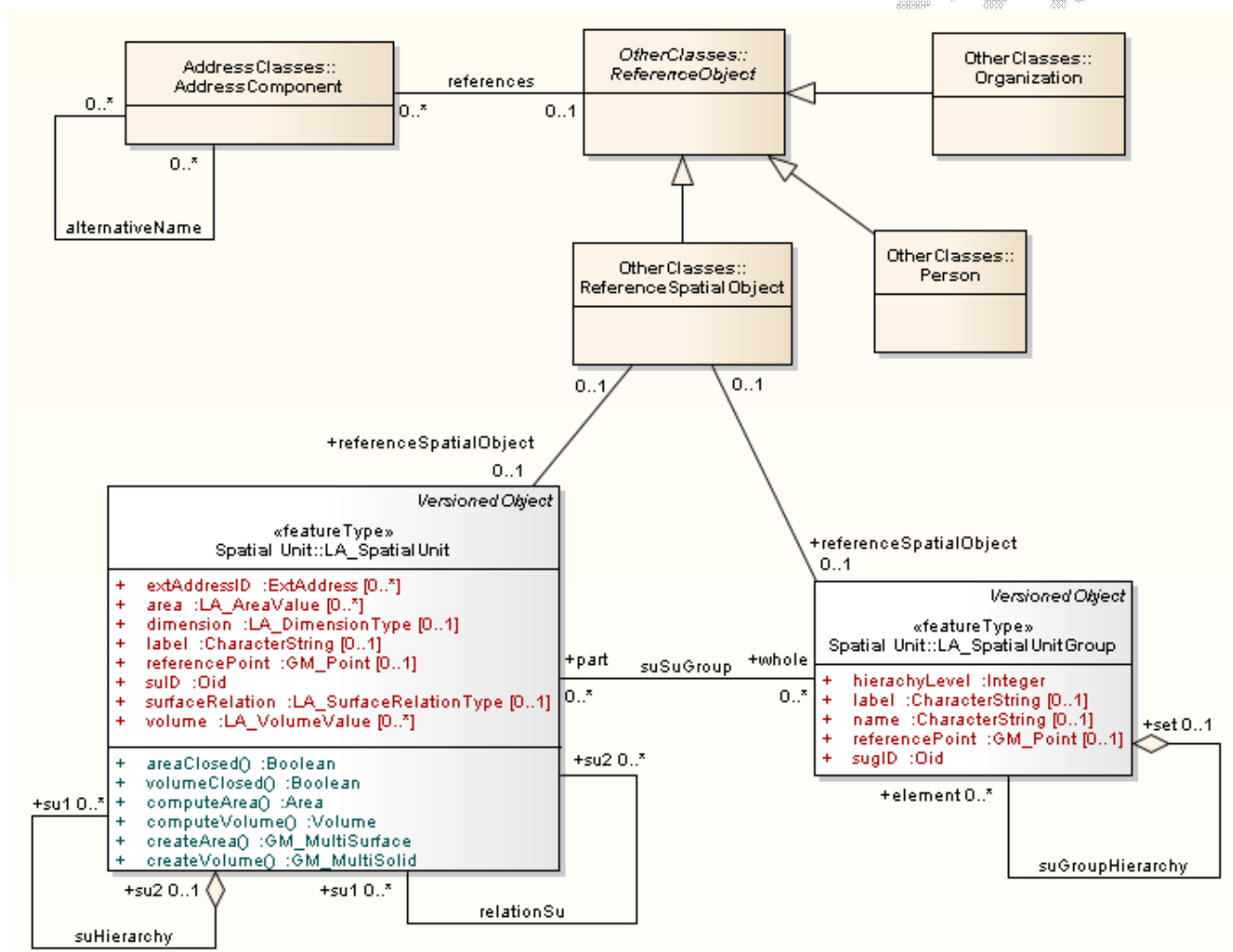


Figure F.1.1 Associating external data via ReferenceObject with AddressComponent

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.