BS ISO 19160-1:2015

BSI Standards Publication

Addressing

Part 1: Conceptual model

bsi.

...making excellence a habit.™

National foreword

This British Standard is the UK implementation of ISO 19160-1:2015.

The UK participation in its preparation was entrusted to Technical Committee IST/36, Geographic information.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Published by BSI Standards Limited 2015

ISBN 978 0 580 82534 7

ICS 35.240.70

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 December 2015.

Amendments/corrigenda issued since publication

Date	Text affected

BS ISO 19160-1:2015

INTERNATIONAL STANDARD

ISO 19160-1

First edition
2015-12-15

Addressing —

Part 1: Conceptual model

Adressage —

Partie 1: Modèle conceptuel

Reference number
ISO 19160-1:2015(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/TC 211, *Geographic information/Geomatics*.

ISO 19160 consists of the following parts, under the general title *Addressing*:

— *Part 1: Conceptual model*

The following parts are under preparation:

— *Part 4: International postal address components and template languages*

The following parts are planned:

— *Part 2: Good practices for address assignment schemes*

— *Part 3: Quality management for address data*

— *Part 5: Address rendering for purposes other than mail*

Introduction

Addresses are one of the most common ways to unambiguously determine an object for the purposes of identification and location. Addresses 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 location system (as opposed to a coordinate reference system) where 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 (activities involving addresses): for assigning addresses (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 for finding the address (citizens, delivery and emergency response service providers, etc.). Relevant stakeholders were identified during the preparatory work of the stage zero project on addressing and are now either involved or aware of the development of ISO 19160 addressing standards.

A variety of address standards and/or specifications are in use around the world. A number of these are described in the report of the preparatory work for this International Standard. These standards and specifications are well integrated into various operational processes and, in some cases, legally enforced. At the same time, some countries are rationalizing their addressing system or creating a new one. Addresses are also increasingly used to reference new geographic objects (e.g. road furniture) 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.

ISO 19112 was included in the investigation of existing standards and specifications during the preparatory work for this International Standard. ISO 19112 deals with geographic identifiers, which indirectly describe position in the real world in the form of a label or code (as opposed to directly or explicitly in the form of coordinates). The review summary concluded that the requirements for addressing standards are sufficiently different to the scope of ISO 19112. If necessary, a profile of this part of ISO 19160 could be developed to map relevant parts of ISO 19112 to this International Standard.

The preparatory work for this International Standard 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 part of ISO 19160 implements the first of these recommendations, the conceptual model. It aims to facilitate interoperability between address specifications, for example, in the cross-mapping of conceptual models between different address specifications.

Addressing —

Part 1:
Conceptual model

1 Scope

This part of ISO 19160 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).

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

2.1 General

This part of ISO 19160 defines six classes of requirements and conformance. Annex A specifies how conformance with these classes shall be tested. Refer to Annex B for guidelines on developing a profile conforming to this International Standard.

2.2 Model — Core

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

2.3 Model — Lifecycle

An Address, AddressComponent or AddressableObject class in the address model for which lifecycle conformance is claimed shall pass the requirements described in the abstract test suite in A.3.

2.4 Model — Provenance

An Address or AddressComponent class in the address model for which provenance conformance is claimed shall pass the requirements described in the abstract test suite in A.4.

2.5 Model — Locale

Any Address, AddressComponent or AddressComponentValue class in the address model for which locale conformance is claimed shall pass the requirements described in the abstract test suite in A.5.

2.6 Model — 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.

2.7 Address profile documentation

Any documentation for which conformance is claimed shall pass the requirements described in the abstract test suite in A.6.

NOTE Refer to Annex C for examples of address models documented in conformance to the address profile documentation conformance class.

3 Normative references

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

ISO 8601, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO 19103:2015, *Geographic information — Conceptual schema language*

ISO 19107:2003, *Geographic information — Spatial schema*

ISO 19115-1:2014, *Geographic information — Metadata — Part 1: Fundamentals*

ISO 19135-1: 2015, *Geographic information — Procedures for item registration — Part 1: Fundamentals*

ISO 19152:2012, *Geographic information — Land Administration Domain Model (LADM)*

4 Terms and definitions

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

4.1
address
structured information that allows the unambiguous determination of an object for purposes of identification and location

EXAMPLE 1 Address where the object is a business: *611 Fifth Avenue, New York NY 10022*.

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

EXAMPLE 3 Address where the object is a land parcel for a building: *San 4–5, Munjae-ro, Songpa-gu, Seoul, 13144, South Korea*.

EXAMPLE 4 Address where the object is a building group, such as a school or large apartment area: *228-dong 404-ho, 26 Kyunghee-daero, Dongdaemun-gu, Seoul 130–701, South Korea*.

Note 1 to entry: The object is identifiable in the real world, i.e. electronic and virtual addresses are excluded.

Note 2 to entry: “Identification” refers to the fact that the structured information in the address unambiguously determines the object, i.e. it helps the human to identify the object. In other words, “identification” here does not refer to unique identifiers in a database or dataset.

Note 3 to entry: There can be many addresses for an object, but at any moment (or lifecycle stage), an address unambiguously determines a single object (see Annex D for examples).

Note 4 to entry: Two addresses from two different *address classes* (4.4) (i.e. they have different sets of components) for the same addressable object are two different addresses (refer to Annex E for more examples).

Note 5 to entry: Two addresses for the same addressable object and from the same address class, but in two different languages are two different addresses (refer to Annex E for more examples).

Note 6 to entry: In addition to the addressable object, there may be a multitude of people, organizations, addressees or other objects associated with an address. These are external to the address model (refer to Annex C and Annex F for examples).

4.2
addressable object
object that may be assigned an *address* (4.1)

4.3
address alias
one of a set of *addresses* (4.1) unambiguously determining the same *addressable object* (4.2)

4.4
address class
description of a set of *addresses* (4.1) that share the same *address components* (4.5), operations, methods, relationships, and semantics

EXAMPLE 1 "25 Blue Avenue Hatfield 0028" and "384 Green Street Motherville 2093" are from the same address class.

EXAMPLE 2 "PO Box 765 Goodwood 33948" and "PO Box 567 Grayville 98373" are from the same address class.

4.5
address component
constituent part of the *address* (4.1)

Note 1 to entry: An address component may reference another object such as a *spatial object* (4.17) (e.g. an administrative boundary or a land parcel) or a non-spatial object (e.g. an organization or a person).

Note 2 to entry: An address component may have one or more alternative values, e.g. alternatives in different languages or abbreviated alternatives.

4.6
addressing
activities involving *addresses* (4.1)

4.7
address position
position representing the *address* (4.1)

Note 1 to entry: An address may be represented by more than one position, e.g. different entrances to a building.

4.8
address reference system
defined set of *address components* (4.5) and the rules for their combination into *addresses* (4.1)

4.9
child address
address (4.1) defined relative to a *parent address* (4.13)

4.10
child addressable object
addressable object (4.2) 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).

EXAMPLE 3 A building within a complex of buildings. In Korea, a *dong* (wing or section of a building) within a group of buildings.

4.11
lineage
provenance (4.16), source(s) and production process(es) used in producing a resource

[SOURCE: ISO 19115-1:2014, 4.9]

4.12
locale
definition of the subset of a user's environment that depends on language and cultural conventions

Note 1 to entry: In computing, a locale is a set of parameters that defines the user's language, country and any special variant preferences that the user wants to see in their user interface. Usually, a locale identifier consists of at least a language identifier and a region identifier.

[SOURCE: ISO/IEC IEEE 9945:2009, 3.211, modified — The notes given in ISO/IEC IEEE 9945:2009 for this entry have been omitted. Note 1 to entry has been added.]

4.13
parent address
address (4.1) of a *parent addressable object* (4.14)

Note 1 to entry: Addresses of the *child addressable objects* (4.9) fully inherit the *address components* (4.5) of a parent address.

4.14
parent addressable object
addressable object (4.2) 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).

EXAMPLE 3 A complex of many buildings. In Korea, a group of buildings with many *dong* (wings or sections of a building).

4.15
profile
set of one or more base standards or subsets of base standards, and, where applicable, the identification of chosen clauses, classes, options and parameters of those base standards, that are necessary for accomplishing a particular function

[SOURCE: ISO 19106:2004, 4.5]

4.16
provenance
organization or individual that created, accumulated, maintained and used records

Note 1 to entry: Provenance information includes

— the source or origin of the record,

— all changes to the record, and

— all organizations or individuals who have had custody of the record since its creation.

[SOURCE: ISO 5127:2001, 4.1.1.10, modified – Note 1 to entry has been added.]

4.17
spatial object
object used for representing a spatial characteristic of a feature

[SOURCE: ISO 19107:2003, 4.69]

5 Symbols and abbreviated terms

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

UML Unified Modeling Language

6 Address model

6.1 General

The address model described in this this part of ISO 19160 serves as a tool to develop specific addressing models, such as a model to describe postal addresses or a model for addresses used in a particular city or country. Figures 1 to 3 provide an overview of the address model with increasing levels of detail.

The core of the address model is built on the notion that an address is made up of a set of one or more address components (see Figure 1). An address is structured information that allows the unambiguous determination of an object for the purposes of identification and location. Address component values form the constituent parts of the structured information. In a simple example, a number of address lines make up an address. In a more complex example, an address comprises more than one kind of address component such as a number, a thoroughfare name, a place name, and a postcode. While the structured information in an address allows one to identify and locate an object, the address is not a unique identifier for the object.

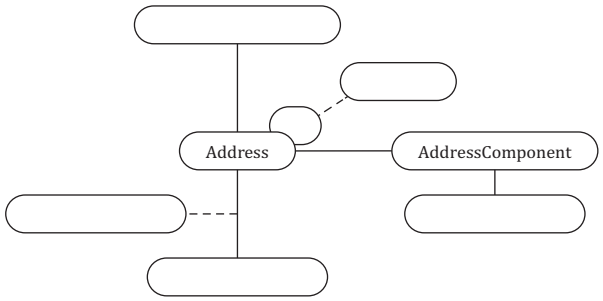

Figure 1 — Schematic overview of the address model showing only the core elements

The value of the address component is a label and sometimes also a reference to another object (ReferenceObject). For example, a place name may reference an object representing the boundary of the place, or an addressee may reference an object with information about the addressee, such as the client name and purchase history. The remaining elements in the address model allow an address to be associated with an object (AddressableObject) such as a building, a dwelling or a land parcel, and with metadata (AddressAlias, AddressedPeriod, AddressSpecifications). See Figure 2.

If more than one address unambiguously determines the same object, the addresses are referred to as address aliases. A typical example is a building on the corner of two streets with an entrance from each street and an address for each entrance. Other examples include colloquial variations of an address or addresses in multiple languages.

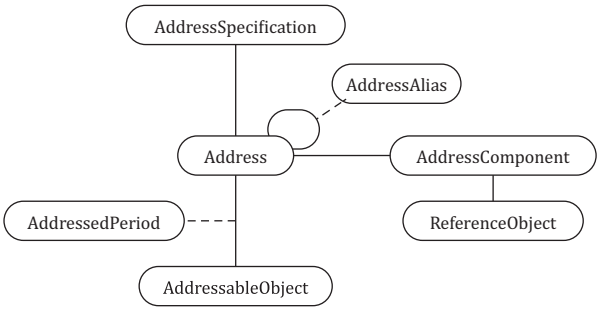

Figure 2 — Schematic overview of the address model showing all elements

Occasionally, an already established address is reassigned to a different object, e.g. in the case of sub-divisions or the construction of additional buildings on single premises. If necessary, AddressedPeriod allows for the representation of different periods during which an address was associated with a specific addressable object.

If applicable and available, metadata about the specification or document describing the address reference system (i.e. rules for combining address components into addresses) and/or addresses represented in the model is provided in the AddressSpecification class.

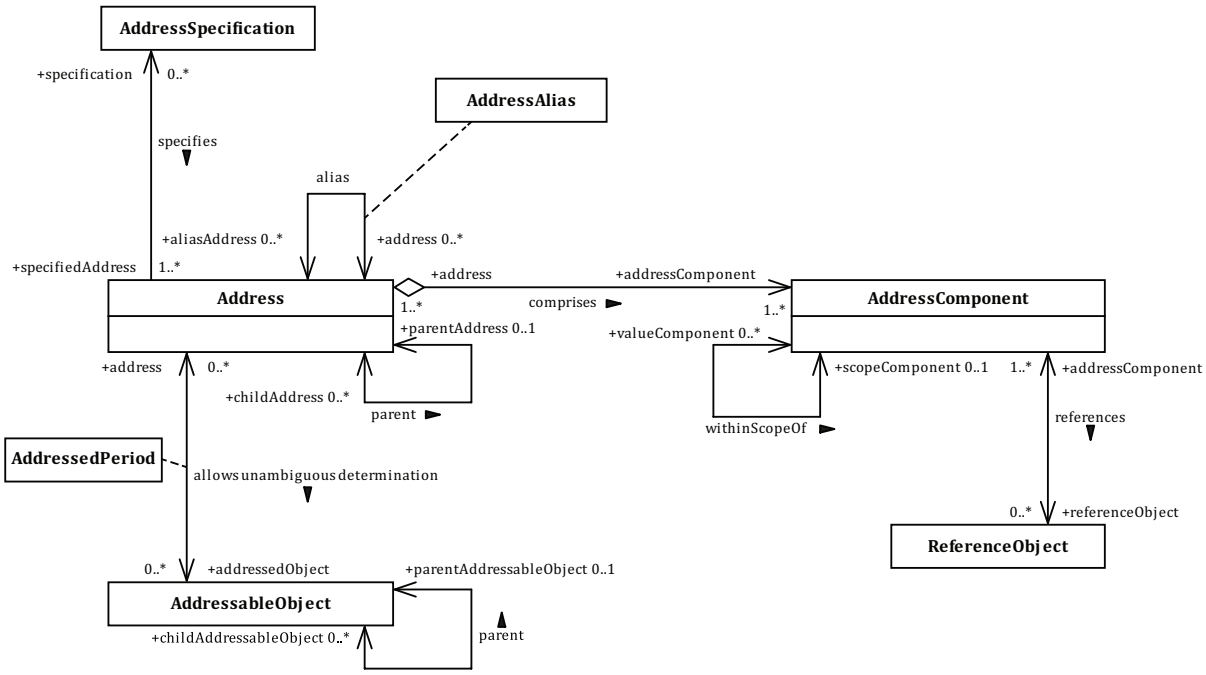

Figure 3 — Address model overview in UML

An address may have coordinates to specify its position. If an address is assigned to an object, the position of the address may be inferred from the addressed object. These are two very different ways of representing the position of an address, and it is, therefore, important that any address model conforming to this part of ISO 19160 clearly specifies how the position of an address is represented in the model.

Finally, an addressable object may have parent-child relationships with other addressable objects, e.g. a building is the parent addressable object of the apartments or offices within. An address may also have parent-child relationships with other addresses, e.g. the address of a building may be the parent address of the addresses for the apartments or offices within (see Figure 3).

6.2 Diagrams

Figure 4 provides an overview of the address model in UML.

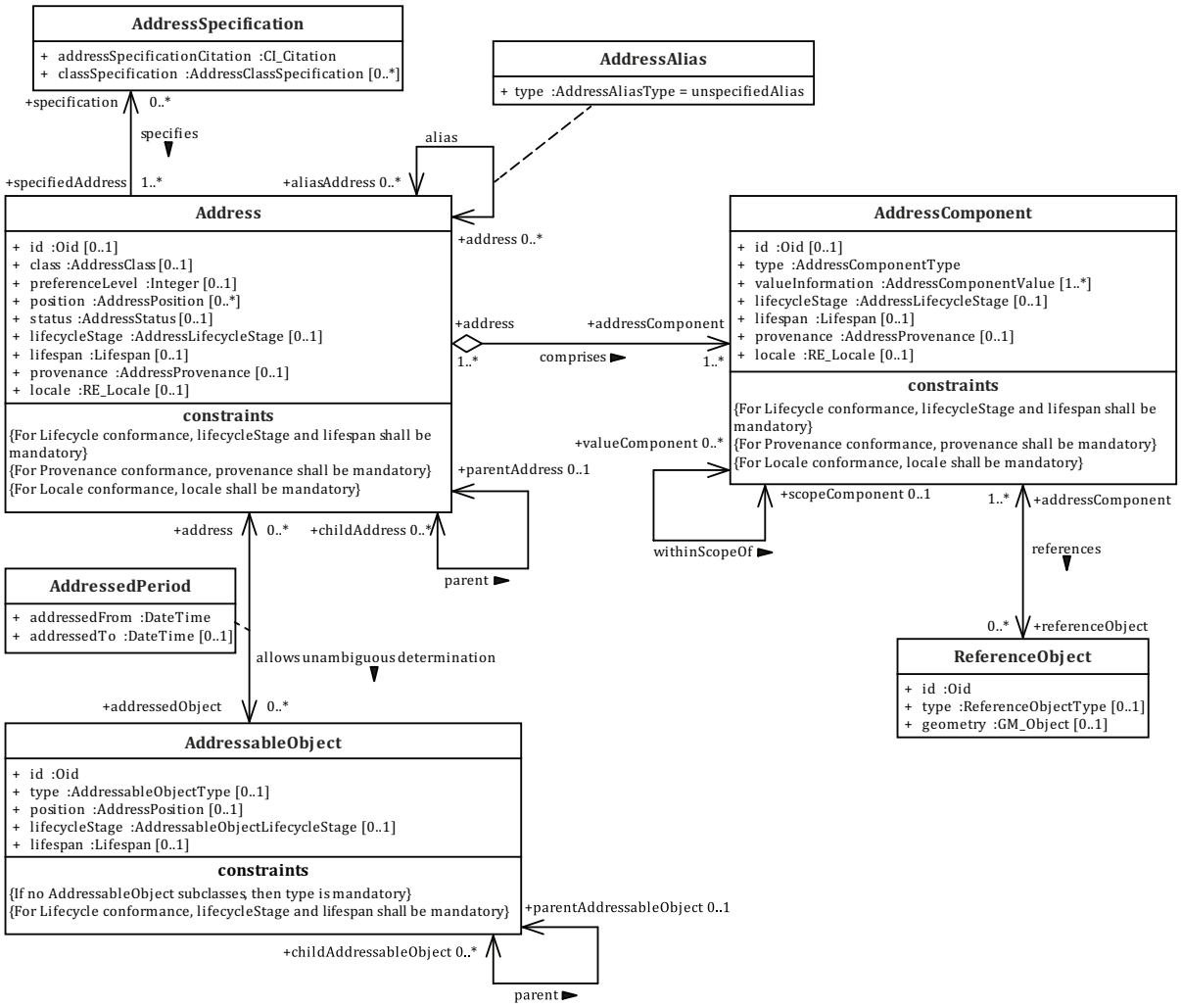

Figure 4 — Address model

Figure 5 shows the core types defined in the address model.

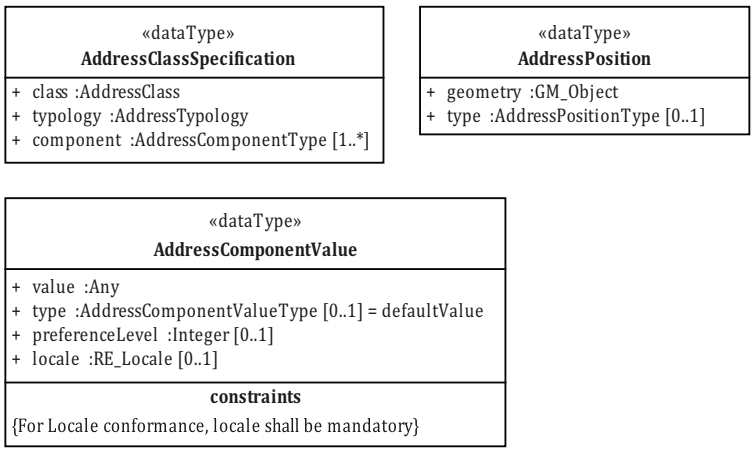

Figure 5 — Core types in the address model

Figure 6 shows the core codelists defined in the address model.

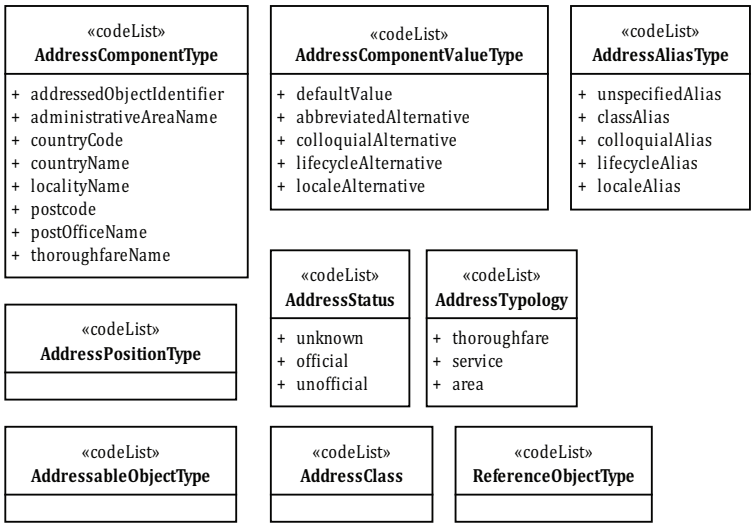

Figure 6 — Core codelists in the address model

NOTE There are too many possible values with little known overlap for the codelists AddressableObjectType, AddressClass, AddressPositionType and ReferenceObjectType. Therefore, these codelists are empty. Each address model has to specify codes, as required (see Annex C for possible codelist values in the sample profiles).

EXAMPLE 1 building, house, landParcel, landmark, apartment and complexOfBuildings are examples of codes for the AddressableObjectType codelist.

EXAMPLE 2 thoroughfareAddress, landmarkAddress and informalAddress are examples of codes for the AddressClass codelist.

EXAMPLE 3 centroid, streetFront and approximated are examples of codes for the AddressPositionType codelist.

EXAMPLE 4 street, administrativeArea, individual and organization are examples of codes for the ReferenceObjectType codelist.

Figure 7 shows the types and codelists in the address model related to lifecycle information.

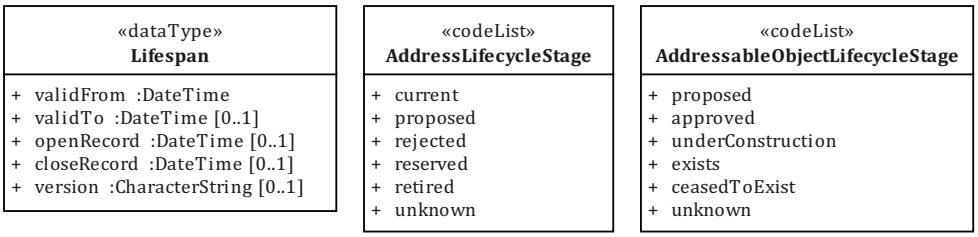

Figure 7 — Types and codelists in the address model for lifecycle information

Figure 8 shows the single type in the address model related to provenance information.

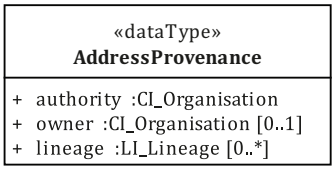

Figure 8 — Type in the address model for provenance information

6.3 Classes

6.3.1 General

The definitions of classes and their attributes are provided in 6.3.2. The name, definition, obligation or condition, maximum occurrence, data type, and domain of each attribute are provided. Some attribute domains are specified with a reference to a UML element, such as a datatype or codelist, in another International Standard. These UML elements can be found in the ISO/TC 211 Harmonized Model at www.isotc211.org.

6.3.2 Address

The Address class represents structured information that allows unambiguous determination of an object for the purposes of identification and location. It consists of a non-empty set of AddressComponents.

EXAMPLE An address such as "99 Lombardy Street, The Hills, 0039" consists of address number (99), thoroughfare name (Lombardy Street), place name (The Hills) and postcode (0039) components.

The attributes of the Address class are defined in Table 1.

Table 1 — Address attributes

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
id	Unique character string that identifies the address.	O	1	Class	<<datatype>> Oid, see ISO 19152
	NOTE id is a unique object identifier; not a primary key in a relational database.				
class	Code that specifies the address class to which the address belongs.	O	1	Class	<<codelist>> AddressClass
preferenceLevel	Indicates the ranking of the address in a set of address aliases. 1 indicates highest ranking.	O	1	Integer	<<interface>> Integer > 0
	EXAMPLE 1 A building on a street corner could be referenced by two addresses. One of them could have its preferenceLevel set to 1. EXAMPLE 2 In Switzerland, addresses containing German and French names, e.g. Biel (German) and Bienne (French), are different addresses with the same preference level.				
position	Geometry (coordinates) that represents the address location.	O	N	Class	<<dataType>> AddressPosition
	NOTE Good practice is to represent a generic position of the address (e.g. door, driveway, centroid) as opposed to a domain or purpose specific position, such as the emergency access or utility meter. Positions of the latter can be represented in a position attribute of an external class associated with the address or addressable object (see example in Figure C.22).				
status	Code that specifies the nature of the address assignment.	O	1	Class	<<codelist>> AddressStatus

Table 1 *(continued)*

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
lifecycleStage	Code that specifies the phase an address has reached in its lifecycle.	O	1	Class	<<codelist>> AddressLifecycle-Stage
	NOTE See Annex D for lifecycle examples of an address.				
lifespan	Information about the period over which an address exists.	O	1	Class	<<dataType>> Lifespan
	NOTE See Annex D for lifespan examples of an address.				
provenance	Information about the source or origin of the address such as the authority who assigned the address, the owner of the address data instance, and lineage of the address.	O	1	Class	<<dataType>> AddressProvenance
locale	A set or parameters that specify the cultural and linguistic environment.	O	1	Class	<<interface>> RE_Locale, see ISO 19135-1
addressedObject	An object that is unambiguously determined by the address.	O	N	Class	AddressableObject
	NOTE There is a single addressedObject at any given point in time.				
addressComponent	A component that is a constituent part of the address.	M	N	Class	AddressComponent
parentAddress	The address of an object that fully encloses the addressedObject.	O	1	Class	Address
childAddress	The address of an object that is fully enclosed by the addressedObject.	O	N	Class	Address
aliasAddress	An address that unambiguously determines the same addressable object as the address.	O	N	Class	Address
specification	A specification of the address and its constituent parts.	o	N	Class	AddressSpecifica-tion

6.3.3 AddressComponent

An AddressComponent is a constituent part of an address. A non-empty set of AddressComponents makes up an address.

EXAMPLE "The Hills" is a constituent part of the address "99 Lombardy Street, The Hills, 0039".

The attributes of the AddressComponent class are defined in Table 2.

Table 2 — AddressComponent attributes

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
id	Unique character string that identifies the address component.	O	1	Class	<<datatype>> Oid, see ISO 19152
	NOTE id is a unique object identifier; not a primary key in a relational database.				
type	Code that specifies the kind of address component.	M	1	Class	<<codelist>> AddressComponentType
	EXAMPLES Thoroughfare name, locality name, country name.				
valueInformation	Value of and information about one or more address component values.	M	N	Class	<<dataType>> AddressComponentValue
lifecycleStage	Code that specifies the phase an address component has reached in its lifecycle.	O	1	Class	<<codelist>> AddressLifecycleStage
	NOTE See Annex D for lifecycle examples of an address component.				
lifespan	Information about the period over which an address component exists.	O	1	Class	<<dataType>> Lifespan
	NOTE See Annex D for lifespan examples of an address component.				
provenance	Information about the source or origin of the address component such as the authority who assigned the address component value, the owner of the address component data instance, and the lineage of the address component.	O	1	Class	<<dataType>> AddressProvenance
locale	A set of parameters that specify the cultural and linguistic environment.	O	1	Class	<<interface>> RE_Locale, see ISO 19135-1
address	An address of which the address component is a constituent part.	M	N	Class	Address
referenceObject	The feature or non-spatial object to which the address component value refers.	O	N	Class	ReferenceObject
	EXAMPLE 1 A person or organization may be the referenceObject for an addressee. EXAMPLE 2 The place name boundary may be the referenceObject for a place name. EXAMPLE 3 A number of street centreline segments may be the referenceObjects for a thoroughfare name.				

Table 2 *(continued)*

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
scopeComponent	Identifies the superordinate address component.	O	1	Class	AddressComponent
	NOTE "within scope of" refers to the relationship between a superordinate address component and a subordinate address component in an address reference system. EXAMPLE 1 Thoroughfare names could be within scope of a place name. If thoroughfare names are unique within a place name, the place name is the scopeComponent of the thoroughfare name. EXAMPLE 2 Building names could be within scope of a thoroughfare name. If building names along a thoroughfare are unique, the thoroughfare name is the scopeComponent of the building name. EXAMPLE 3 Box numbers could be within scope of a post office. If each box number at a post office is unique, the post office is the scopeComponent of the box number.				
valueComponent	Identifies the subordinate address component	O	N	Class	AddressComponent
	NOTE "within scope of" refers to the relationship between a superordinate address component and a subordinate address component in an address reference system. EXAMPLE 1 Thoroughfare names could be within scope of a place name. If thoroughfare names are unique within a place name, the thoroughfare name is a valueComponent of the place name. EXAMPLE 2 Building names could be within scope of a thoroughfare name. If building names are unique along a thoroughfare name, the building name is a valueComponent of the thoroughfare name. EXAMPLE 3 Box numbers could be within scope of a post office. If each box number at a post office is unique, the box number is a valueComponent of the post office.				

6.3.4 AddressableObject

The Address allows unambiguous determination of an AddressableObject, i.e. an object that may be identified or located by an Address.

EXAMPLE 1 The following address allows the unambiguous determination of a person's residence: *99 Lombardy Street, The Hills, 0039, South Africa.*

EXAMPLE 2 The following address allows the unambiguous determination of a building: *Lombardy House, 809 Lombardy Street, The Hills, 0039, South Africa.*

EXAMPLE 3 The following address allows the unambiguous determination of a door (apartment) in a building: *Room 4–6, Lombardy House, 809 Lombardy Street, The Hills, 0039, South Africa.*

The attributes of the AddressableObject class are defined in Table 3.

Table 3 — AddressableObject attributes

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
id	Unique character string that identifies the addressable object.	M	1	Class	<<datatype>> Oid, see ISO 19152
	NOTE id is a unique object identifier; not a primary key in a relational database.				
type	Code that specifies the kind of object to which an address may be assigned.	C: Only if there are no subclasses of AddressableObject	1	Class	<<codelist>> AddressableObjectType
	EXAMPLES A person's residence, a building, a door (apartment) in a building.				
position	Geometry (coordinates) representing the addressable object.	O	1	Class	<<dataType>> AddressPosition
	NOTE Good practice is to represent a generic position of the addressable object (e.g. door, driveway, centroid), as opposed to a domain or purpose specific position such as the position of an emergency access or utility meter. Positions of the latter can be represented in a position attribute of an external class associated with the address or addressable object (see Example in C.4).				
lifecycleStage	Code that specifies the phase an addressable object has reached in its lifecycle.	O	1	Class	<<codelist>> AddressableObjectLifecycleStage
	NOTE See Annex D for lifecycle examples of an addressable object.				
lifespan	Information about the period over which an addressable object exists.	O	1	Class	<<type>> Lifespan
	NOTE See Annex D for lifespan examples of an addressable object.				
address	The address that unambiguously determines the addressable object.	O	N	Class	Address
parentAddressableObject	An addressable object that fully encloses the addressable object.	O	1	Class	AddressableObject
childAddressableObject	An addressable object that is fully enclosed by the addressable object.	O	N	Class	AddressableObject

6.3.5 ReferenceObject

The ReferenceObject is the object to which the address component value may refer.

EXAMPLE 1 A person or organization may be the ReferenceObject for an addressee.

EXAMPLE 2 The place name boundary may be the ReferenceObject for a place name.

EXAMPLE 3 A number of street centreline links may be the ReferenceObjects for a thoroughfare name.

The attributes of the ReferenceObject class are defined in Table 4.

Table 4 — ReferenceObject attributes

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
id	Unique character string that identifies the reference object.	M	1	Class	<<datatype>> Oid, see ISO 19152
	NOTE id is a unique object identifier, not a primary key in a relational database.				
type	Code that specifies the kind of reference object.	O	1	Class	<<codelist>> ReferenceObjectType
	EXAMPLES Thoroughfare name, locality name, country name.				
geometry	Geometry (coordinates) representing the reference object.	O	1	Class	<<type>> GM_Object. See ISO 19107
addressComponent	The address component whose value references the reference object.	M	N	Class	AddressComponent

6.3.6 AddressSpecification

The AddressSpecification is a specification of addresses and their constituent parts (components).

EXAMPLE SANS 1883–1[16] is a specification for addresses in use in South Africa.

NOTE The information in classSpecification can be used for address data validation and quality checks.

The attributes of the AddressSpecification class are defined in Table 5.

Table 5 — AddressSpecification attributes

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
addressSpecificationCitation	Reference to a specification or document that contains a definition of the address classes.	M	1	Class	CI_Citation, see ISO 19115-1
	EXAMPLES Addressing standard, technical specification, undocumented specification.				
classSpecification	Information about one or more address classes such as typology and valid component types.	O	N	Class	<<dataType>> AddressClassSpecification
specifiedAddress	An address that conforms to the rules of the address specification.	O	N	Class	Address

6.4 Types

6.4.1 General

The definitions of types and their attributes are provided in 6.4. The name, definition, obligation or condition, maximum occurrence, data type and domain of each attribute are provided.

6.4.2 AddressClassSpecification

The AddressClassSpecification type represents information about the address class. The attributes of AddressClassSpecification are defined in Table 6.

NOTE The attributes of AddressClassSpecification can be used for address data validation and quality checks.

Table 6 — AddressClassSpecification attributes

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
class	Code that specifies the address class to which the address belongs.	M	1	Class	<<codelist>> AddressClass
typology	Code that specifies the type of address class.	M	1	Class	<<codelist>> AddressTypology
	EXAMPLES Thoroughfare, service, area.				
component	The set of one or more address component types that may be considered together in various combinations to constitute an address of this class.	M	N	Class	<<codelist >> AddressComponentType

6.4.3 AddressPosition

AddressPosition represents information about the representative position of an address. The attributes of AddressPosition are defined in Table 7.

Table 7 — AddressPosition attributes

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
geometry	Geometry (coordinates) representing the address position.	M	1	Class	<<type>> GM_Object, see ISO 19107
type	Code that specifies how the geometry shall be interpreted.	O	1	Class	<<codelist>> AddressPositionType
	EXAMPLES Centroid, streetFront, approximated.				

6.4.4 AddressComponentValue

AddressComponentValue represents information about the value of an address component. The attributes of AddressComponentValue are defined in Table 8.

Table 8 — AddressComponentValue attributes

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
value	One or more instances of the value of the address component.	M	1	any	<<interface>> Any, see ISO 19103
type	Code that specifies the kind of value, i.e. whether it is a variation of the default address component value, and if so, what the variation is.	O	1	Class	<<codelist>> AddressComponentValu-eType
	EXAMPLES Variations could be abbreviations, colloquial values or values in alternate languages.				
preferenceLevel	Indicates the ranking of the address component in a set of alternatives. 1 indicates highest ranking.	O	1	Integer	<<interface>> Integer > 0
	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 could be set to 1. EXAMPLE 2 In Switzerland, Biel (German) and Bienne (French) are different address components with the same preference level.				
locale	A set of parameters that specify the cultural and linguistic environment.	O	1	Class	<<interface>> RE_Locale, see ISO 19135-1

6.4.5 AddressAlias

The AddressAlias type represents information about an address alias. The attributes of AddressAlias are defined in Table 9.

Table 9 — AddressAlias attributes

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
type	Code that specifies which justification for an alias applies to the address at the destination end of the association. The default value is unspecifiedAlias.	M	1	Class	<<codelist>> AddressAliasType
	EXAMPLES An address alias could be from a different address class, a colloquial version of the address, or at a different lifecycle stage.				

6.4.6 AddressedPeriod

The AddressedPeriod type represents the period during which the address was associated with the addressed object. The attributes of AddressedPeriod are defined in Table 10.

Table 10 — AddressedPeriod attributes

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
addressedFrom	Date and time from which the addressed object is unambiguously determined by the address.	M	1	Class	<<interface>> DateTime Character encoding of a DateTime shall follow ISO 8601. This class is documented in full in ISO 19103.
addressedTo	Date and time when the addressed object ceased to be unambiguously determined by the address.	O	1	Class	<<interface>> DateTime Character encoding of a DateTime shall follow ISO 8601. This class is documented in full in ISO 19103.

6.4.7 Lifespan

The Lifespan type represents information to describe the lifespan of an address, address component or addressable object. The attributes of Lifespan are defined in Table 11.

Table 11 — Lifespan attributes

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
validFrom	Date and time from which the object is valid in the physical world.	M	1	Class	<<interface>> DateTime Character encoding of a DateTime shall follow ISO 8601. This class is documented in full in ISO 19103.
validTo	Date and time when this object ceased to be valid in the physical world.	O	1	Class	<<interface>> DateTime Character encoding of a DateTime shall follow ISO 8601. This class is documented in full in ISO 19103.
openRecord	Specifies the date and time at which this version of the data object was inserted into the dataset.	O	1	Class	<<interface>> DateTime Character encoding of a DateTime shall follow ISO 8601. This class is documented in full in ISO 19103.
closeRecord	Specifies the date and time at which this version of the data object was superseded by another version or retired in the dataset.	O	1	Class	<<interface>> DateTime Character encoding of a DateTime shall follow ISO 8601. This class is documented in full in ISO 19103.
version	Unique identifier of this variant of the address record.	O	1	CharacterString	Free text
	NOTE The version is incremented with any change to the data object.				

6.4.8 AddressProvenance

The AddressProvenance type represents provenance information, e.g. the source or origin of the record, changes to the record, and organizations or individuals that have had custody of the record since its creation. The attributes of AddressProvenance are defined in Table 12. Classes derived from AddressProvenance may have one or more additional metadata attributes specified in ISO 19115-1 and/or Dublin Core (ISO 15836).

Table 12 — AddressProvenance attributes

Name	Definition	Mandatory/ conditional/ optional	Max occur	Data type	Domain
authority	The body that assigns the address or address component value.	M	1	Class	CI_Organisation, see ISO 19115-1
owner	Organization that maintains the data.	O	1	Class	CI_Organisation, see ISO 19115-1
lineage	Provenance, source(s), and production process(es) used in producing an address or address component.	O	N	Class	LI_Lineage, see ISO 19115-1

6.5 Codelists

6.5.1 General

Tables 13 to 18 provide definitions for individual codelist values in the address model.

NOTE Each profile may specify additional codelist values and/or include only a subset of the codelist values defined here.

6.5.2 AddressAliasType

AddressAliasType describes the alias association between two addresses. The values defined in the address model for the AddressAliasType codelist are defined in Table 13.

Table 13 — AddressAliasType values

Name	Definition
unspecifiedAlias	The type of address alias is not specified.
classAlias	The address alias is from a different address class.
colloquialAlias	The address alias is a colloquial version of the address.
lifecycleAlias	The address alias has a different lifecycle stage.
localeAlias	The address alias is in a different locale.

6.5.3 AddressComponentType

AddressComponentType contains values to represent commonly used address component types. These values are defined in Table 14.

Table 14 — AddressComponentType values

Name	Definition
addressedObjectIdentifier	Identifier of the addressed object, e.g. building name or address number
administrativeAreaName	Name of an administrative area
countryCode	ISO 3166-1 code for the country, territory, or area of geopolitical interest
countryName	Name of a country
localityName	Name of a locality
postcode	Code used for the sorting of mail [SOURCE: UPU S42]
postOfficeName	Name of a post office
thoroughfareName	Name of a thoroughfare

6.5.4 AddressComponentValueType

AddressComponentValueType specifies the type of address component value. The values defined in the address model for the AddressComponentValueType codelist are defined in Table 15.

Table 15 — AddressComponentValueType values

Name	Definition
defaultValue	The default component value (i.e. the one that is not an alternative).
abbreviatedAlternative	The alternative component value is an abbreviation.
colloquialAlternative	The alternative component value is a colloquial alternative for the component value.
lifecycleAlternative	The alternative component value was used in a different lifecycle stage.
localeAlternative	The alternative component value is in a different locale.

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

EXAMPLE 1 "Gordon Rd" is an abbreviated alternative for "Gordon Road" (refer to Annex E for more examples).

EXAMPLE 2 "Cologne" is a language alternative for "Köln" in Germany (refer to Annex E for more examples).

EXAMPLE 3 "Jozi", "Joburg" or "Egoli" are colloquial alternatives for "Johannesburg" in South Africa (refer to Annex E for more examples).

6.5.5 AddressLifecycleStage

AddressLifecycleStage represents the different lifecycle stages of an Address or AddressComponent. The values defined in the address model for the AddressLifecycleStage codelist are defined in Table 16.

Table 16 — AddressLifecycleStage values

Name	Definition
current	The address or address component is currently in use.
proposed	The address or address component has been proposed, i.e. the relevant authority has initiated approval procedures for the use of the address or address component.
rejected	The address or address component was proposed but rejected.
reserved	The address or address component has been reserved for future use.
retired	The address or address component was in use at some stage, but not anymore.
unknown	The lifecycle stage of the address or address component is unknown.

6.5.6 AddressableObjectLifecycleStage

AddressableObjectLifecycleStage represents the different lifecycle stages of an AddressableObject. The values defined in the address model for the AddressableObjectLifecycleStage codelist are defined in Table 17.

Table 17 — AddressableObjectLifecycleStage values

Name	Definition
proposed	The establishment or construction of the addressable object has been proposed, i.e. the relevant authority has initiated approval procedures.
approved	The establishment or construction of the addressable object has been approved, i.e. the relevant authority has given approval for the establishment or construction of the addressable object.
underConstruction	The establishment or construction of the addressable object is in progress.
exists	The addressable object exists.
ceasedToExist	The addressable object does not exist anymore (e.g. it has been demolished).
unknown	The lifecycle stage of the addressable object is unknown.

6.5.7 AddressStatus

AddressStatus contains values to specify the status of an address. The values defined in the address model for the AddressStatus codelist are defined in Table 18.

Table 18 — AddressStatus values

Name	Definition
unknown	The status of the address is unknown.
official	An official addressing authority assigned the address.
unofficial	The address was not assigned by an official addressing authority.

6.5.8 AddressTypology

AddressTypology contains values to specify the type of an address class. The values defined in the address model for the AddressTypology codelist are defined in Table 19.

Table 19 — AddressTypology values

Name	Definition
thoroughfare	The address class is based on navigable access features such as streets or canals.
service	The address class is based on delivery or collection services such as a group of post boxes at the same location or poste restante.
area	The address class is based on a division of land or water into demarcated areas such as neighbourhoods, precincts, or cadastral features.

EXAMPLE 1 An international trading corporation has customers in 140 countries and maintains contact and delivery addresses for them in a customer file. Rather than attempt to understand the individual classes used by each jurisdiction, the corporation utilizes the typology concept to gain a general understanding of how the jurisdiction addresses are structured.

EXAMPLE 2 When numerous profiles are available, the typology concept may be used to assert general rules, e.g. for the development of software tools.

7 Requirements

7.1 Requirements class: Core

7.1.1 Dependencies

Table 20 shows the target type and dependencies of the Core requirements class.

Table 20 — Dependencies of the Core requirements class

Requirements class identifier	Core
Target type	Conceptual model
Dependency	GM_Object in ISO 19107
Dependency	CI_Citation in ISO 19115-1
Dependency	LI_Lineage in ISO 19115-1
Dependency	CI_Organisation in 19115–1
Dependency	Oid in ISO 19152 Oid is reused for its general purpose of having a namespace attribute in addition to an identifier attribute.
Dependency	DateTime from ISO 8601 and ISO 19103
Dependency	CharacterString from ISO 19103
Dependency	RE_Locale in ISO 19135-1

7.1.2 Core requirement 1: Classes

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

EXAMPLE 1 Examples of address components are street name, place name, addressee and postcode.

The address model may include the AddressableObject, AddressSpecification, AddressAlias and AddressedPeriod classes or classes derived from them.

EXAMPLE 2 Examples of addressable objects are building, house, landmark and apartment.

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

EXAMPLE 3 The polygon representing an administrative area.

EXAMPLE 4 The polygon representing the footprint of a building or the point representing the building.

EXAMPLE 5 The polygon representing a land parcel.

EXAMPLE 6 Information about the person or organization who is the mail recipient (address component).

NOTE See Annex F for an example of how external data can be associated with an address component through a reference object.

7.1.3 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 allow the unambiguous determination an addressable object. An addressable object may be determined unambiguously by addresses. Multiple addresses that allow for the unambiguous determination of the same addressable object are address aliases.

EXAMPLE 1 Different addresses for a building on a street corner are an example of more than one address unambiguously determining the same addressable object (refer to Annex E for more examples).

An address component may be within the scope of a second address component, e.g. when the address component values are assigned according to rules ensuring that they are unambiguous within the second address component.

EXAMPLE 2 A street name assigned to be unambiguous within a suburb (the street name is not necessarily unambiguous within the city).

EXAMPLE 3 An address number assigned to be unambiguous within a thoroughfare.

An address component may reference reference objects. A reference object shall be referenced by one or more address components.

EXAMPLE 4 A thoroughfare name (derived from AddressComponent) referencing a street centre line.

EXAMPLE 5 A thoroughfare name (derived from AddressComponent) referencing street centre line segments.

EXAMPLE 6 A mail recipient (derived from AddressComponent) referencing information about the mail recipient, e.g. age, gender, education level.

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

EXAMPLE 7 The building is the parent addressable object of the individual apartments (child addressable objects) inside the building.

EXAMPLE 8 In Japan, the *gaiku* (block) is the parent addressable object of the individual *jukyo bango* (residence numbers) (child addressable objects) inside the *gaiku* (block).

EXAMPLE 9 In Korea, a group of buildings are the parent addressable object of the individual *dong* (buildings) (child addressable objects).

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

EXAMPLE 10 The address of the building is the parent address of the addresses of individual apartments inside the building.

EXAMPLE 11 In Japan, the address of the *gaiku* (block) is the parent address of the individual *jukyo bango* (residence numbers) addresses in the *gaiku* (block).

EXAMPLE 12 In Korea, the address of a group of buildings is the parent address of the address of an individual *dong* (buildings) in this group.

An address may be specified according to one or more address specification. An address specification may include a set of address class specifications which specify the typology and the set of valid component types for the class.

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

EXAMPLE 13 SANS 1883.[16]

EXAMPLE 14 Korea Road Name Address Act of 2011.

EXAMPLE 15 D2.8.1.5 INSPIRE.[10]

EXAMPLE 16 UPU S42.[17]

7.1.4 Core requirement 3: Attributes

Classes in an address model shall include mandatory, conditional, and optional attributes as specified in Tables 1 to 12.

7.2 Requirements class: Lifecycle

7.2.1 Dependencies

Table 21 shows the target type and dependencies of the Lifecycle requirements class.

Table 21 — Dependencies of the Lifecycle requirements class

Requirements class identifier	Lifecycle
Target type	Class in an address model
Dependency	DateTime from ISO 8601 and ISO 19103
Dependency	CharacterString from ISO 19103

7.2.2 Lifecycle requirement 1: Lifecycle attributes

The lifecycleStage and lifespan attributes (e.g. in the Address, AddressComponent or AddressableObject class) shall be mandatory.

7.2.3 Lifecycle requirement 2: Unique identifier

The unique identifier (e.g. Address.id, AddressComponent.id and AddressableObject.id) shall be mandatory.

7.2.4 Lifecycle requirement 3: Version increments

The version attribute in the lifespan attribute (e.g. in the Address, AddressComponent or AddressableObject class) shall be incremented with any change to the data object.

7.3 Requirements class: Provenance

7.3.1 Dependencies

Table 22 shows the target type and dependencies of the Provenance requirements class.

Table 22 — Dependencies of the Provenance requirements class

Requirements class identifier	Provenance
Target type	Class in an address model
Dependency	CI_Organisation in ISO 19115-1
Dependency	LI_Linage in ISO 19115-1

7.3.2 Provenance requirement 1: Provenance attribute

The provenance attribute (e.g. in the Address, AddressComponent and AddressableObject class) shall be mandatory.

7.4 Requirements class: Locale

7.4.1 Dependencies

Table 23 shows the target type and dependencies of the Locale requirements class.

Table 23 — Dependencies of the Locale requirements class

Requirements class identifier	Locale
Target type	Class in an address model
Dependency	RE_Locale in ISO 19135-1

7.4.2 Locale requirement 1: Locale attribute

The locale attribute (e.g. in the Address, AddressComponent and AddressComponentValue class) shall be mandatory.

7.5 Requirements class: Address profile documentation

7.5.1 Dependencies

Table 24 shows the target type and dependencies of the Address profile documentation requirements class.

Table 24 — Dependencies of the Address profile documentation requirements class

Requirements class identifier	Locale
Target type	Documentation of an ISO 19160-1 profile
Dependency	None

7.5.2 Requirements and recommendations

The address profile documentation shall contain the following:

a) the name of the developer (e.g. standards body or authority name) of the profile and its contact details;

b) the specification for which the profile is developed. This can be a citation for a specification, standard or legal act, or a description of the addresses that are represented in the profile;

c) the names of conformance class(es) to which the profile conforms;

d) background information about addresses represented in the profile such as the purpose for which addresses are assigned and the authority that assigns the addresses;

e) for optional classes (AddressSpecification, AddressableObject, ReferenceObject, AddressAlias, AddressedPeriod) and optional associations in the address model, explicitly state whether they are mandatory, optional, conditional, out of scope or prohibited in the profile. A constraint on the relevant class shall be used to specify whether an optional attribute is mandatory, optional, conditional, out-of-scope, or prohibited in the profile;

f) the following elements of the profile model in at least one diagram:

   1) each mandatory class (Address, AddressComponent), conditional class and optional class in the profile;

   2) each profile-specific class, type or codelist derived from a class, type or codelist in the address model;

3) each mandatory, optional, conditional and profile-specific association in the profile;

4) each mandatory, optional, conditional and profile-specific attribute in the profile;

g) a matrix to indicate which address components are mandatory, conditional or optional for each address class defined in the AddressClass codelist;

h) the *Any* type of AddressComponentValue.value for each value in the AddressComponentType codelist shall be indicated in the profile, either as constraints in the specialized AddressComponent class or as a table indicating the *Any* type of AddressComponentValue.value for each value in the AddressComponentType;

i) an explanation of where and how the position of an address and/or addressable object is represented in the profile;

j) diagrams of instance data (examples) for a few sample addresses.

The following shall be included in the documentation if locale information is included in the profile:

— a statement to specify which classes in the profile model include locale information.

The following should be included in the documentation:

— a bi-directional mapping between each attribute in the profile and the specification.

Diagrams in the documentation should be prepared as follows:

— to distinguish between base classes and profile-specific classes, it is recommended that a transparent background be used for (base) classes from the address model and a shaded fill colour background for the profile-specific classes;

— 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 Figures 1 to 4 of this part of ISO 19160.

The following may be included in the documentation:

— a table with the definitions of profile-specific codelist values.

NOTE Sample profiles conforming to these requirements and recommendations are included in Annex C.

Annex A
(normative)

Abstract test suites

A.1 General

The abstract test suites for the conformance classes defined by this part of ISO 19160 are presented in A.2 to A.5.

A.2 Conformance class: Core

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

Table A.1 — Core test 1: Classes

Test purpose	Check that the model contains the classes as specified.
Test method	Inspect the model
Reference	7.1.2
Test type	Basic

Table A.2 — Core test 2: Associations

Test purpose	Check that the model contains the associations as specified.
Test method	Inspect the model
Reference	7.1.3
Test type	Basic

Table A.3 — Core test 3: Attributes

Test purpose	For each class and type in the model, check that the model appropriately includes the mandatory, optional, and conditional attributes.
Test method	Inspect the model
Reference	7.1.4
Test type	Basic

A.3 Conformance class: Lifecycle

Tables A.4 to A.6 contain the details of the tests for the Lifecycle conformance class.

Table A.4 — Lifecycle test 1: Lifecycle attributes

Test purpose	Check that the lifecycleStage and lifespan attributes are mandatory.
Test method	Inspect the class in the model
Reference	7.2.2
Test type	Basic

Table A.5 — Lifecycle test 2: Unique identifier

Test purpose	Check that the id attribute is mandatory.
Test method	Inspect the class in the model
Reference	7.2.3
Test type	Basic

Table A.6 — Lifecycle test 3: Version increments

Test purpose	Check that the version attribute is incremented whenever there is a change to the relevant data object.
Test method	Inspect the data
Reference	7.2.4
Test type	Basic

A.4 Conformance class: Provenance

Table A.7 contains the details of the test for the Provenance conformance class.

Table A.7 — Provenance test 1: Provenance attribute

Test purpose	Check that the provenance attribute is mandatory.
Test method	Inspect the class in the model
Reference	7.3.2
Test type	Basic

A.5 Conformance class: Locale

Table A.8 contains the details of the test for the Locale conformance class.

Table A.8 — Locale test 1: Locale attribute

Test purpose	Check that the locale attribute is mandatory.
Test method	Inspect the class in the model
Reference	7.4.2
Test type	Basic

A.6 Conformance class: Address profile documentation

Table A.9 contains the details of the test for the Address profile documentation conformance class.

Table A.9 — Address profile documentation test

Test purpose	Check that the documentation meets the specified requirements.
Test method	Inspect the documentation
Reference	7.5.2
Test type	Basic

Annex B
(informative)

Guidelines for developing a profile

B.1 General

B.2 and B.3 are provided as guidelines for developers of profiles. The steps followed to develop a profile of ISO 19160-1 are described in B.2. In B.3, the steps to develop the UML for the profile's address model are described. Profiles may be published on the ISO website at http://standards.iso.org/iso/19160/-1/ (refer to the instructions in B.4).

B.2 Steps to develop a profile

a) Decide to which conformance classes the profile shall conform.

   — All profiles shall conform to the core conformance class.

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

b) Identify the classes to be included.

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

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

   — If required, add or derive profile-specific classes. In a Geography Markup Language (GML)-compatible profile, subclasses should have the featureType stereotype and attributes and constraints have to be copied to these sub-classes.

c) 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 (by specifying a constraint): mandatory, conditional, out-of-scope or prohibited.

   — If required, add profile-specific associations.

d) Specify codelist values.

   — Codelist values may be specified for the AddressClass.

   — Codelist values may be specified for AddressableObjectType, AddressPositionType and ReferenceObjectType. These values need only to be specified if the relevant attributes are included in the model.

   — If required, add profile-specific values to any of the other codelists, i.e. AddressComponentType, AddressComponentValueType, AddressStatus, AddressAliasType, AddressLifecycleStage and AddressableObjectLifecycleStage. Profile-specific values need to be added if the definitions of existing codelist values do not meet the requirements of the profile.

e) 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 (by specifying a constraint): mandatory, conditional, out-of-scope or prohibited.
- If required, add profile-specific attributes.
- Specify what the *Any* type is for each in AddressComponentType codelist, either as a table or as constraint in the specialized AddressComponent classes.
- If applicable, describe how each attribute in the profile is mapped to the relevant standard/specification and back.

f) If AddressClass codelist values are specified in the profile, specify the address classes of the profile.

- A matrix to indicate valid address components for each address class defined in the AddressClass codelist shall be included.

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

h) Prepare documentation according to the Address profile documentation conformance class (see 2.7).

B.3 Steps to develop a UML model

a) Start with a copy of the ISO/TC 211 Harmonized (UML) Model. Information about the model is available on the Harmonized Model Maintenance Group (HMMG) page on the www.isotc211.org website.

b) Create a new package for the profile.

c) Add shortcuts to this part of ISO 19160 classes required for the profile to the package (e.g. by dragging them to the package in Enterprise Architect).

d) Create specializations of this part of ISO 19160 address model classes, as required for the profile.

e) For each specialized class

- add additional attributes and associations to other classes as required for the profile,
- define constraints to identify elements not applicable in the profile, and
- define constraints to describe any special cases in profile elements deviating from the ISO 19160-1 element.

f) Create diagrams of the address model, types, code lists and any other specializations in the profile according to the Address profile documentation conformance class (see 2.6).

B.4 Steps to upload a profile on the ISO website

a) Send the profile documentation (PDF) and UML model (e.g. XMI files) to the ISO/TC 211 Secretariat.

b) The ISO 19160 project team will review the profile and completely inappropriate profiles will be returned and not published. Testing conformance to the Address profile documentation conformance class is the responsibility of the profile developer and will not be done by the project team.

c) The ISO/TC 211 Secretariat creates an online folder for the profile in http://standards.iso.org/iso/19160/-1/, uploads the relevant files into the folder and updates the readme.txt.

d) The ISO/TC 211 Secretariat notifies the submitter that its profile has been uploaded.

Annex C
(informative)

Sample profiles

C.1 General

Two sample profiles are included in C.2 and C.3. They illustrate what a profile is and how a profile shall be documented. While these profiles are ready to use and conform to this part of ISO 19160's Core conformance class, data prepared according to these profiles does not necessarily conform to this part of ISO 19160 (data conformance is not specified). In C.4, a number of diagrams from other profiles are included to provide further examples of profiles.

C.2 Example 1: Minimal address profile

C.2.1 Profile developer

Name: ISO 19160-1 project team

Contact details: ISO/TC 211 Secretariat, see www.isotc211.org for contact details.

C.2.2 Specification

The minimal address profile represents minimal addresses, i.e. addresses as they are used colloquially, without any additional information, such as identifiers, coordinates or metadata. This example is included to illustrate that even though the address model allows complexity, it can also be used to represent addresses comprising of address lines (character strings) only.

C.2.3 Conformance

The minimal address profile conforms to the Core conformance class.

C.2.4 Profile model

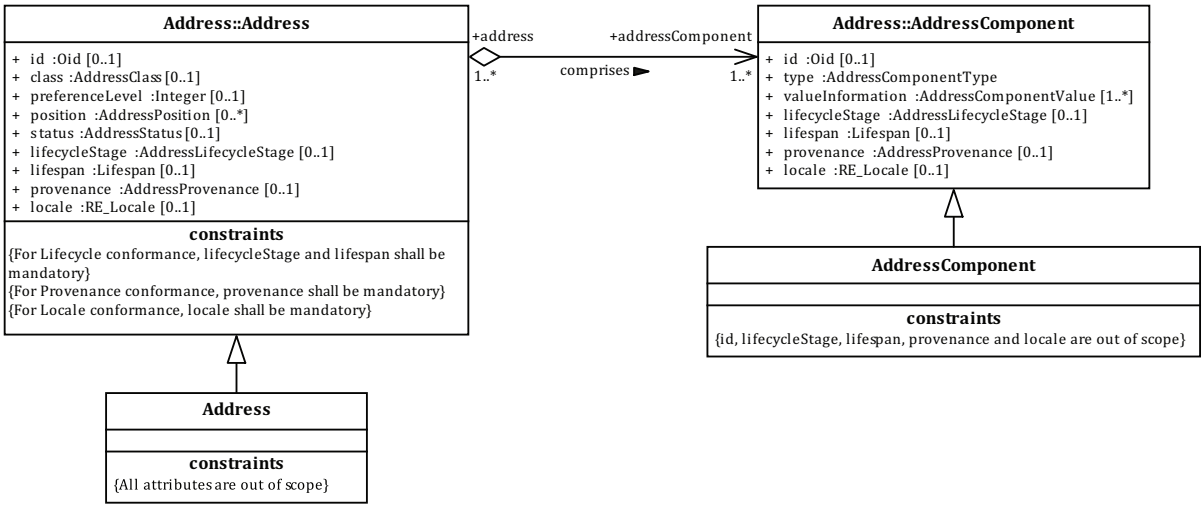

Figure C.1 — Minimal address profile: Address model

Address and AddressComponent are the only classes in the minimal address model and the association between these two classes is the only association in the model. All other classes and associations in the address model are out of scope in this profile (see Figure C.1).

The codeslists and types for the minimal address model are illustrated in Figure C.2. AddressComponent has two attributes: type and valueInformation, all other attributes are out of scope. The value attribute is the only attribute in AddressComponentValue included in the minimal address model, all other attributes are out of scope.

addressLine is the only valid component for this address class. The values of an addressLine component are of type CharacterString.

Thus, an address in the minimal address model consists of one or more addressLine components, each with a CharacterString value. See also the instance diagrams in C.2.5 (Figures C.3 to C.10).

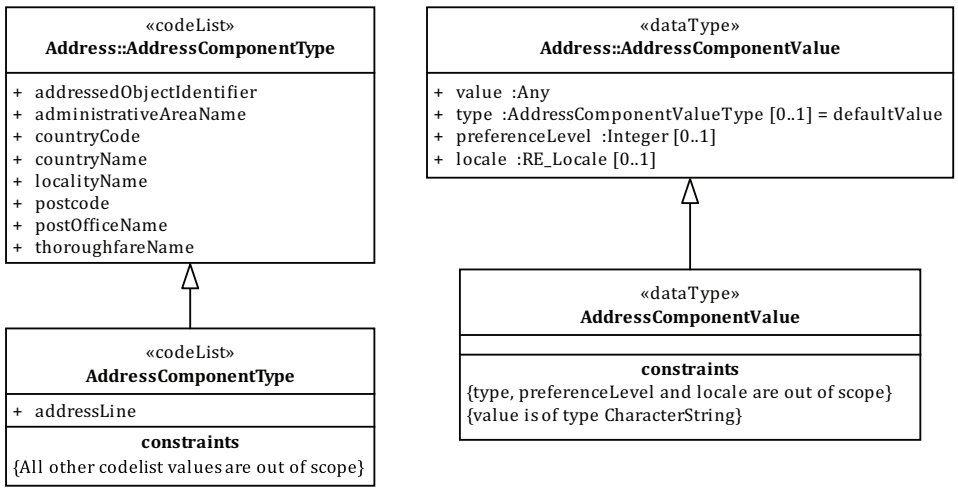

Figure C.2 — Minimal address profile: Codelists and types

Table C.1 — Minimal address profile: Any type for the AddressComponentType values

AddressComponentType value	Corresponding type
addressLine	CharacterString

Table C.2 — Minimal address profile: Matrix of valid components for AddressClass values

AddressComponentType value	minimalAddress
addressLine	M

C.2.5 Instance data

The diagrams in Figures C.3 to C.10 show instances of addresses represented in the minimal address profils.

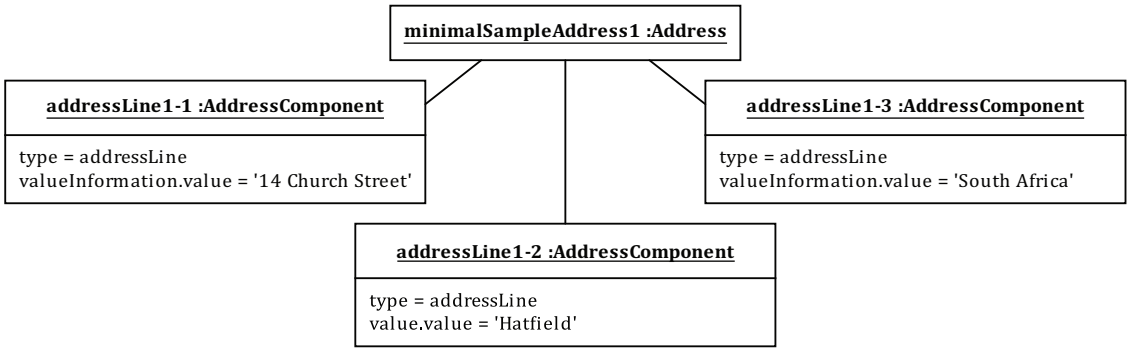

Figure C.3 — Minimal address profile: Instance 1 — South African street address

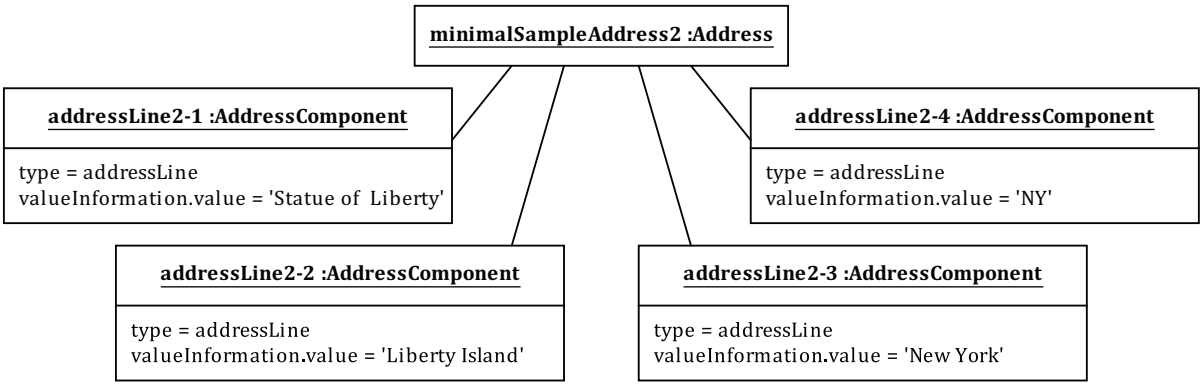

Figure C.4 — Minimal address profile: Instance 2 — US landmark address

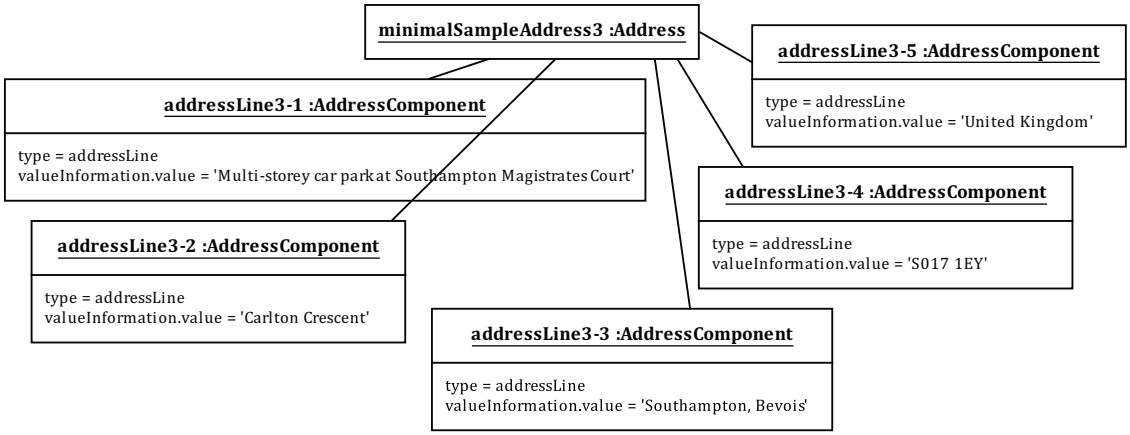

Figure C.5 — Minimal address profile: Instance 3 — UK address

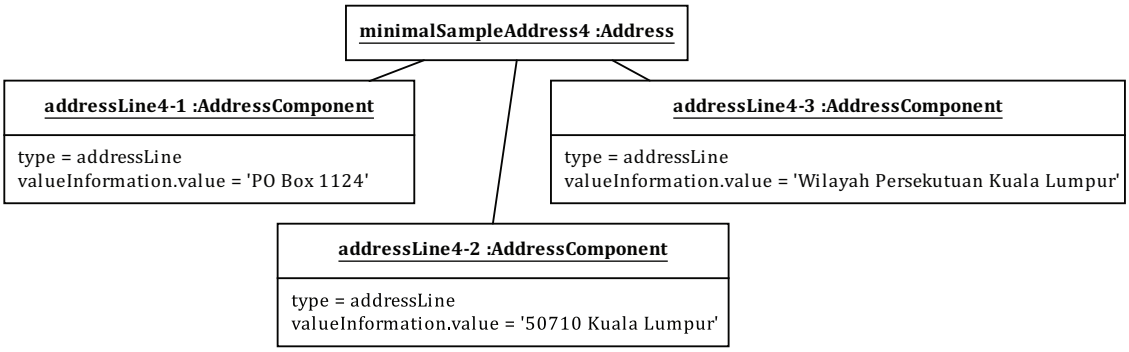

Figure C.6 — Minimal address profile: Instance 4 — Malaysian address

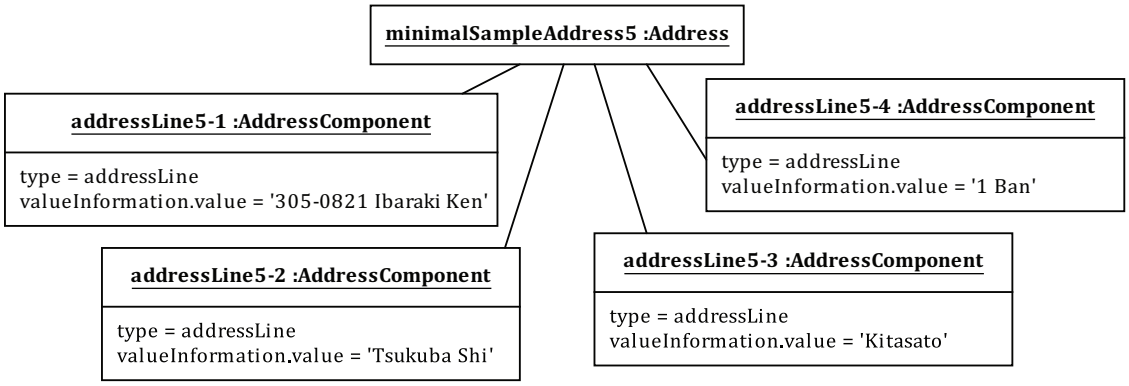

Figure C.7 — Minimal address profile: Instance 5 — Japanese address

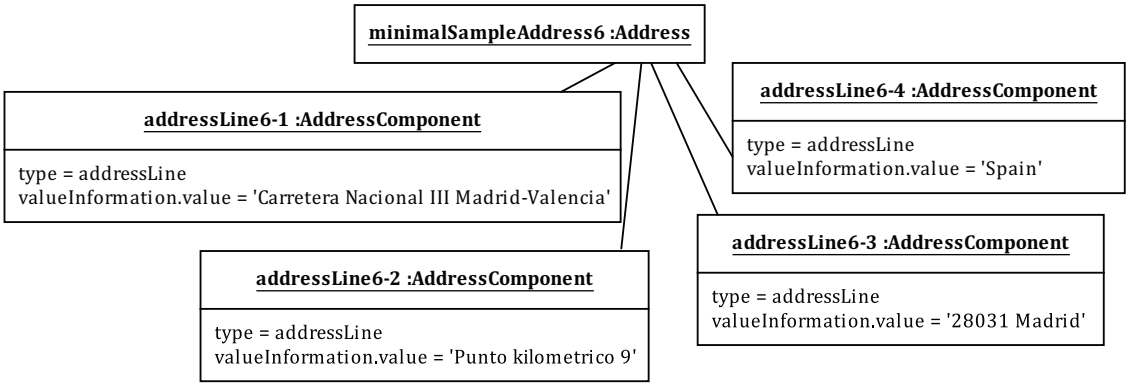

Figure C.8 — Minimal address profile: Instance 6 — Spanish address

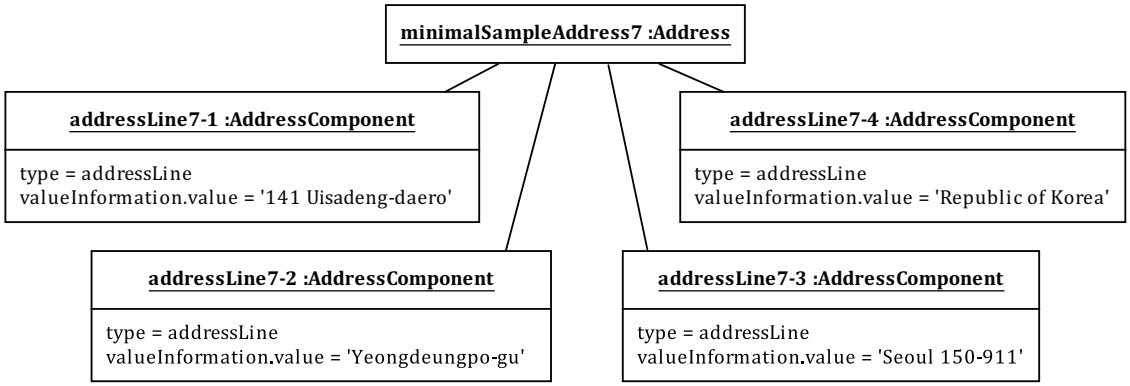

Figure C.9 — Minimal address profile: Instance 7 — Korean address

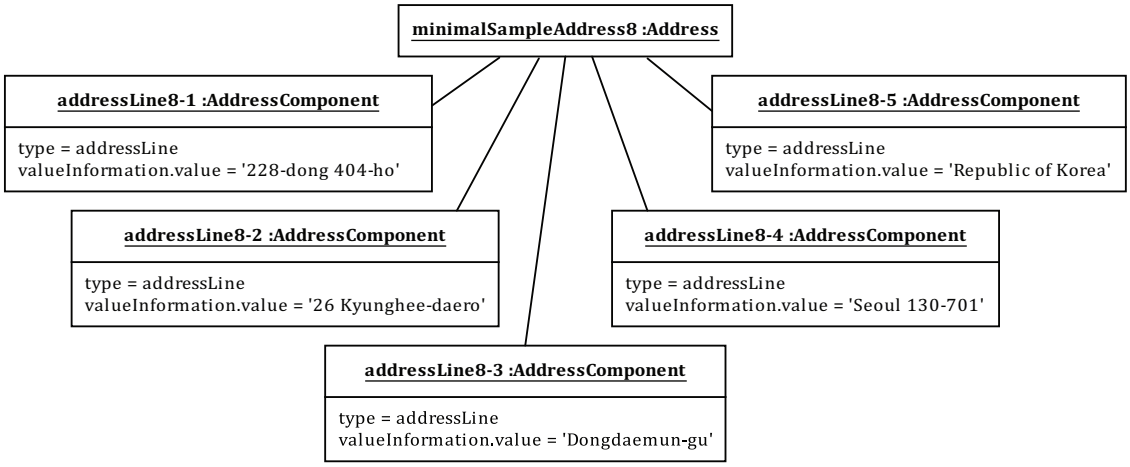

Figure C.10 — Minimal address profile: Instance 8 — Korean address for a group of buildings

C.3 Example 2: Sample address profile

C.3.1 Profile developer

Name: ISO 19160-1 project team

Contact details: ISO/TC 211 Secretariat, see www.isotc211.org for contact details.

C.3.2 Address specification

The sample address profile describes two kinds of addresses, street addresses and box addresses. It is included here as an example of this part of ISO 19160 profile and its documentation. The specification below describes imaginary addresses for illustrative purposes only.

A street address consists of an address number, thoroughfare name, place name, post code, and an optional country name such as “99 Lombardy Street, The Hills, 0039, South Africa”. The following rules apply.

— The address number is an integer between 1 and 10,000.

— The country name is optional, but all other parts of the address are mandatory.

Street addresses are assigned to residential dwellings, commercial buildings or land parcels. Local authorities are responsible for assigning street addresses which they use mainly for service delivery, but the general public of course also uses the addresses for orientation. Each local authority maintains

a building register of every building in its jurisdiction; each of the buildings has a unique identifier in the register. Each land parcel has a nationally unique land parcel identifier which is recorded in the country's cadastral information system.

A box address consists of a box number, post office name, post code, and an optional country name such as "PO Box 345, Orlando, 2020, South Africa". The following rules apply.

— The box number is an integer between 1 and 100,000.

— The box number is within scope of the post office name.

— Post office names are within scope of a country.

— The country name is optional, but all other parts of the address are mandatory.

Box addresses are assigned by a post office to post boxes attached to the post office building. Individuals or organizations can rent a box and have their mail delivered to the box.

C.3.3 Conformance

The sample address profile conforms to the Core conformance class.

C.3.4 Profile model

Position is represented in the addressable objects only. Figure C.11 shows the address model for the sample address profile, while Figure C.12 and Figure C.13, respectively, show the types and codelists in this profile. The value type for the different AddressComponentTypes is illustrated in Table C.3. Table C.4 lists valid component types by address class. Tables C.5 to C.7 show the mapping of address classes, address component types and addressable object types between sample addresses and profile.

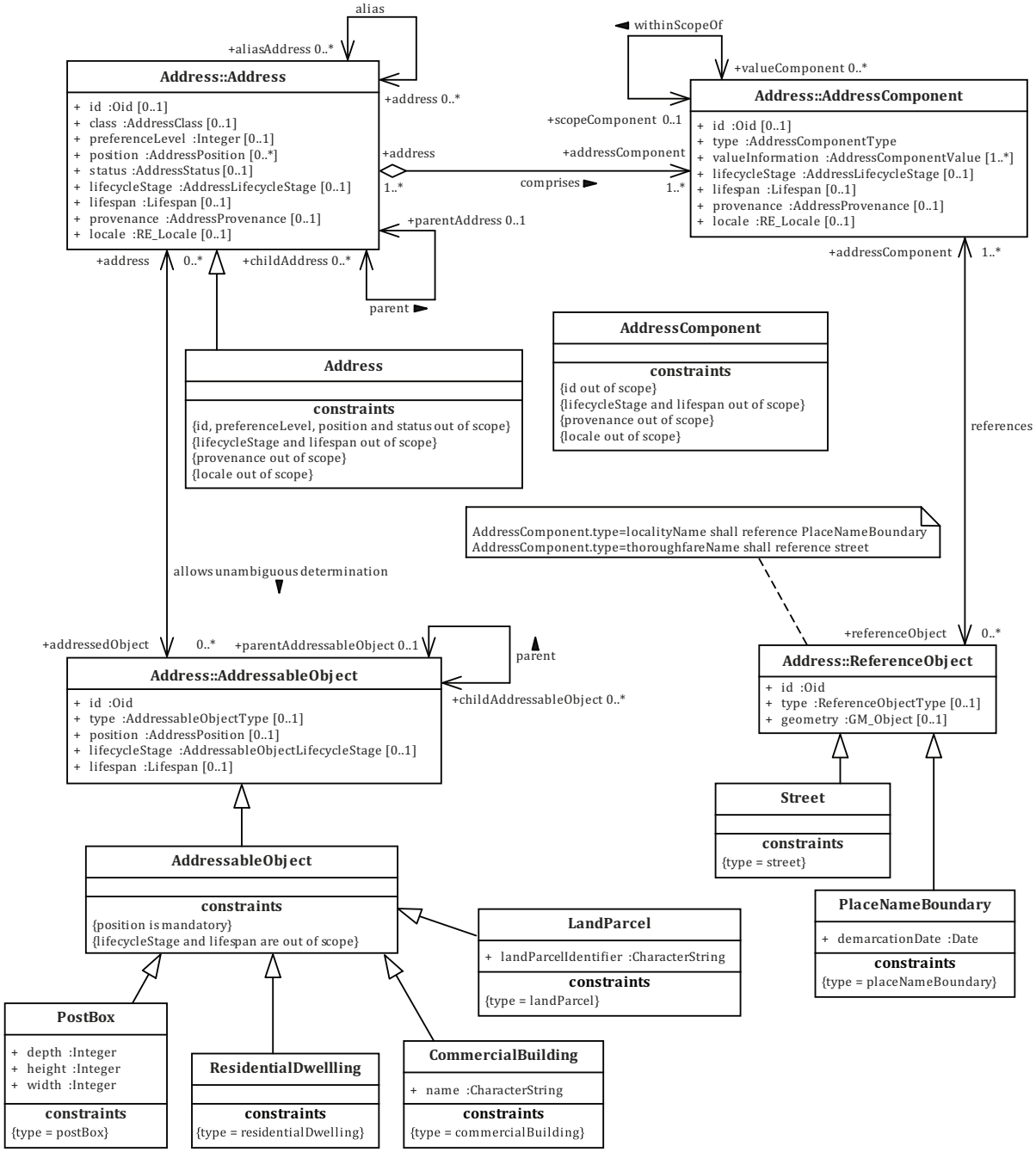

Figure C.11 — Sample address profile: Address model

Table C.3 — Sample address profile: Any type for the different AddressComponentType values

AddressComponentType value	Corresponding type
addressedObjectIdentifer	Integer
thoroughfareName	ThoroughfareNameValue
localityName	CharacterString
postOfficeName	CharacterString
postCode	CharacterString
countryName	CharacterString

Table C.4 — Sample address profile: Valid components types for different AddressClass values

addressComponentType	addressClass	
	streetAddress	boxAddress
addressedObjectIdentifer	M	M
thoroughfareName	M	
localityName	M	
postOfficeName		M
postCode	O	M
countryName	O	O

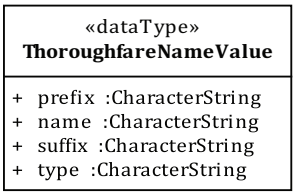

Figure C.12 — Sample address profile: Types

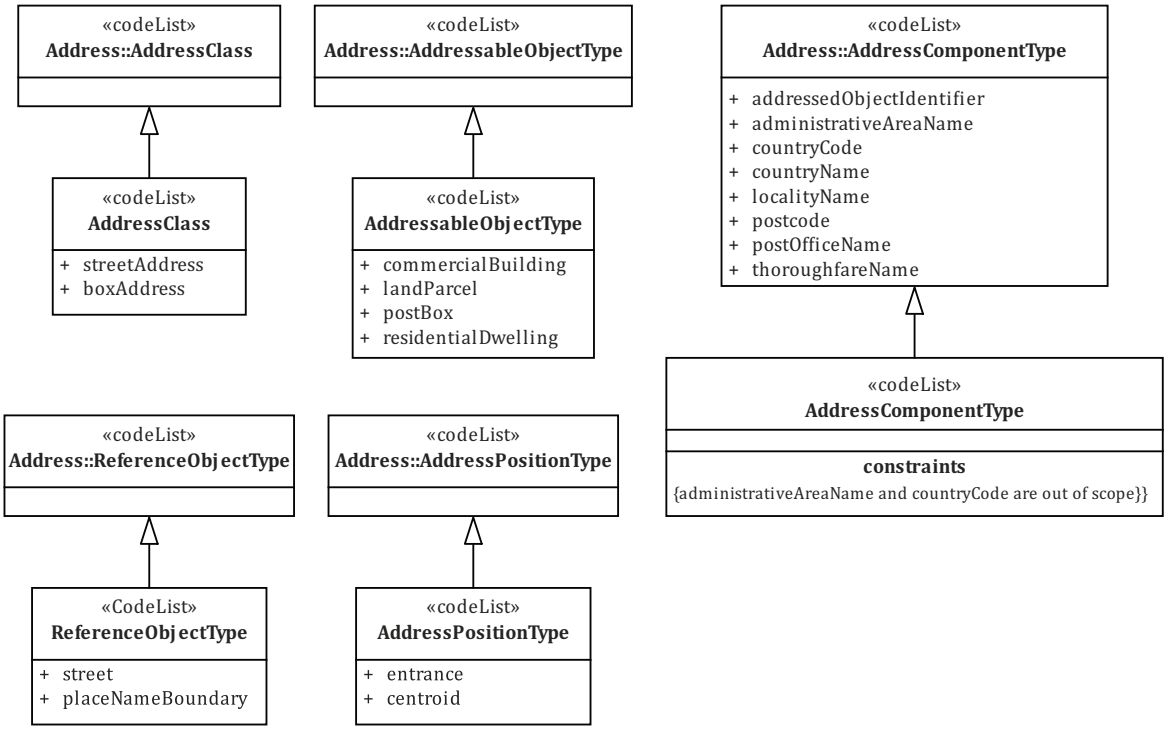

Figure C.13 — Sample address profile: Codelists

Table C.5 — AddressClass codelist mapping

Sample address	Profile
Street address	streetAddress
Box address	boxAddress

Table C.6 — AddressComponentType codelist mapping

Sample address	Profile
Address number	addressedObjectIdentifer
Box number	addressedObjectIdentifier
Thoroughfare name	thoroughfareName
Place name	localityName
Post office name	postOfficeName
Post code	postCode
Country name	countryName

Table C.7 — AddressableObjectType codelist mapping

Sample address	Profile
Commercial building	commercialBuilding
Land parcel	landParcel
Post box in the post office building	postBox
Residential dwelling	residentialDwelling

C.3.5 Mapping from sample addresses to the sample address profile

Tables C.8 to C.10 show how attributes of the Address, AddressComponent and AddressableObject classes are mapped from sample address to the profile.

Table C.8 — Address attributes: From sample address to the profile

Sample address		Profile attribute	Mapping description
No equivalent	n/a	class	Refer to Table C.6 for the mapping between the names of different sample addresses and the AddressClass codelist values.

Table C.9 — AddressComponent attributes: from sample address to the profile

Sample address		Profile attribute	Mapping description
No equivalent	n/a	type	Refer to Table C.7 for the mapping between the names of components in the sample address profile and AddressComponentType codelist values.
No equivalent	n/a	valueInformation.value	Table C.3 shows the applicable data type for the value of different address component types. In the case of the thoroughfare name, it is split into its constituent parts, each of which is mapped to the appropri-ate attribute of ThroughfareNameValue. EXAMPLE 1 “Church Street” in a sample address is mapped to “Church” in valueInformation.value.name and ‘Street’ in valueInfromation.value.type. EXAMPLE 2 “Park Avenue West” in a sample address is mapped to “Park” in valueInformation.value.name and “Avenue” to valueInfromation.value.type and “West” in valueInformation.value.suffix.
No equivalent	n/a	value.type	n/a

Table C.10 — AddressableObject attributes: From sample address to the profile

Sample address		Profile attribute	Mapping description
No equivalent	n/a	id.localId	Create a unique identifier as follows: For a postBox, assign the box number. For a landParcel, assign the unique land parcel identifier from the cadastral information system. For a residentialDwelling and commercialBuilding, assign the identifier from the local authority's building register.
No equivalent	n/a	id.namespace	For a postBox, assign the concatenation of the two-character country code, '_' and the post office name, e.g. 'ZA_Orlando'. For a landParcel, assign 'Cadastre'. For a residentialDwelling and commercialBuilding, assign the name of the local authority.
No equivalent	n/a	position.crs	Assign the relevant coordinate reference system.
No equivalent	n/a	position.geometry	For a postBox, assign the position of the front entrance to the post office building. For a landParcel, assign the centroid of the land parcel. For a residentialDwelling and commercialBuilding, assign the position of the main front entrance.
No equivalent	n/a	position.type	For a landParcel, assign the AddressPositionType.centroid. For a postBox, residentialDwelling and commercialBuilding, assign AddressPositionType.entrance.

C.3.6 Mapping from sample address profile to sample addresses

Tables C.11 to C.13 show how attributes of the Address, AddressComponent and AddressableObject classes are mapped from profile to the sample address.

Table C.11 — Address attributes: From profile to sample address

Profile attribute	Sample address		Mapping description
class	No equivalent	n/a	Refer to Table C.5 for the mapping between the names of different sample addresses and the AddressClass codelist values.

Table C.12 — AddressComponent attributes: From profile to sample address

Profile attribute	Sample address		Mapping description
type	Component name	n/a	Refer to Table C.6 for the mapping between the names of components in the sample address profile and AddressComponentType values.
valueInformation.value	Respective value in the address	n/a	Table C.3 shows the applicable data type for the value of different address component types. In the case of the thoroughfare name, the individual attributes of valueInfromation.value (of type ThoroughfareNameValue) are concatenated into a string. EXAMPLE 1 "Church" in valueInformation.value.name and "Street" to valueInformation.value.type maps to "Church Street" in the sample address. EXAMPLE 2 "Park" in valueInformation.value.name and "Avenue" to valueInformation.value.type and "West" to valueInformation.value.suffix maps to "Park Avenue West" in the sample address.
value.type	No equivalent	n/a	n/a

Table C.13 — AddressableObject attributes: From profile to sample address

Profile attribute	Sample address		Mapping description
id.localId	No equivalent	n/a	n/a
id.namespace	No equivalent	n/a	n/a
position.crs	No equivalent	n/a	n/a
position.geometry	No equivalent	n/a	n/a
position.type	No equivalent	n/a	n/a

C.3.7 Instance data

Figures C.14 and C.15 show examples of sample address instances according to the sample address profile.

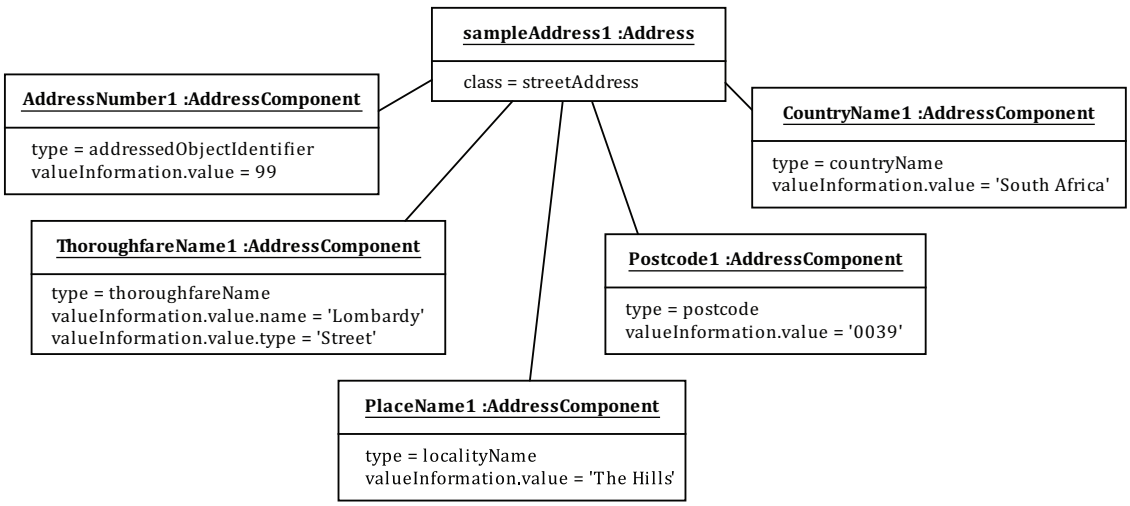

Figure C.14 — Sample address profile: Instance 1 — Street address

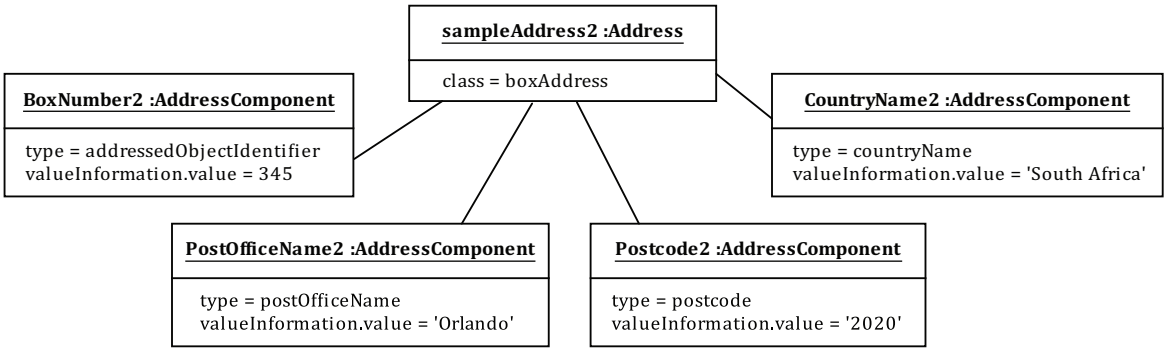

Figure C.15 — Sample address profile: Instance 2 — Box address

Figures C.16 and C.17 show examples of instances of sample address and addressable objects according to the sample address profile.

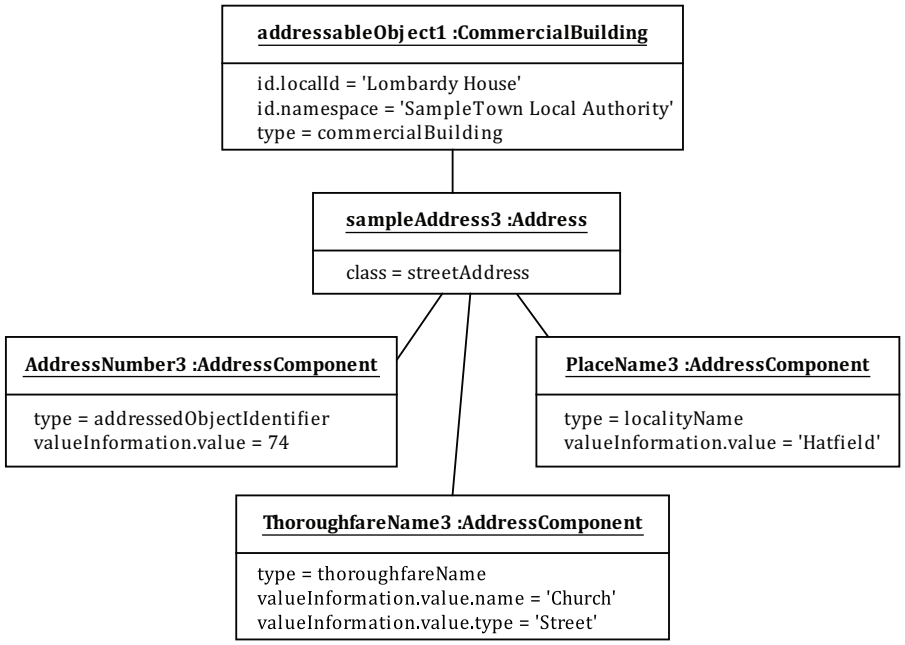

Figure C.16 — Sample address profile: Instance 3 — Address references a commercial building

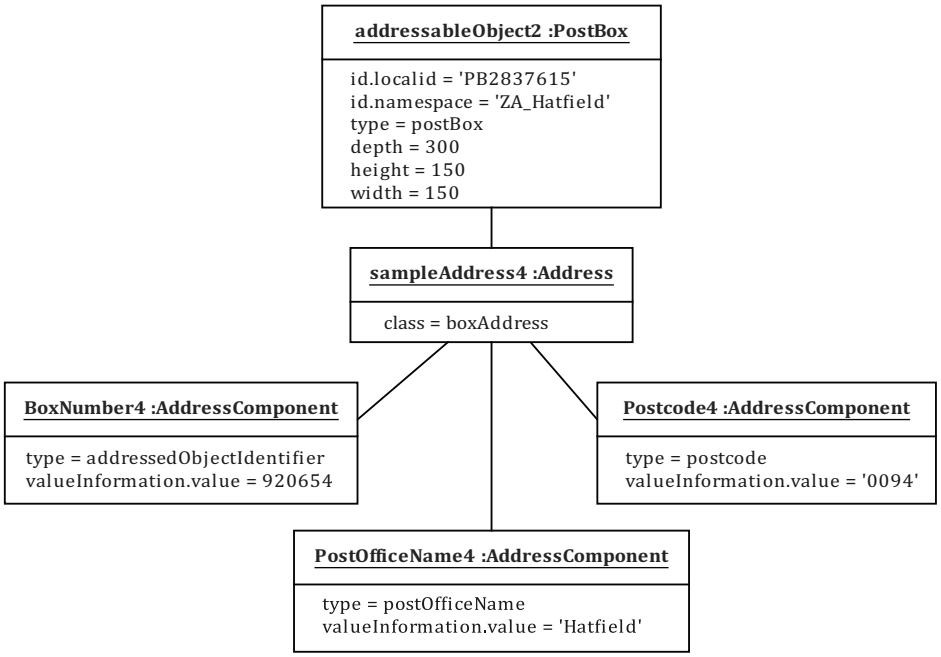

Figure C.17 — Sample address profile: Instance 4 — Address references a post box

C.3.8 Linking sample addresses to external data

Figure C.18 illustrates how external data may be linked to sample addresses. LA_SpatialUnit is defined in ISO 19152 while the Employee and BusinessRegister classes are fictional.

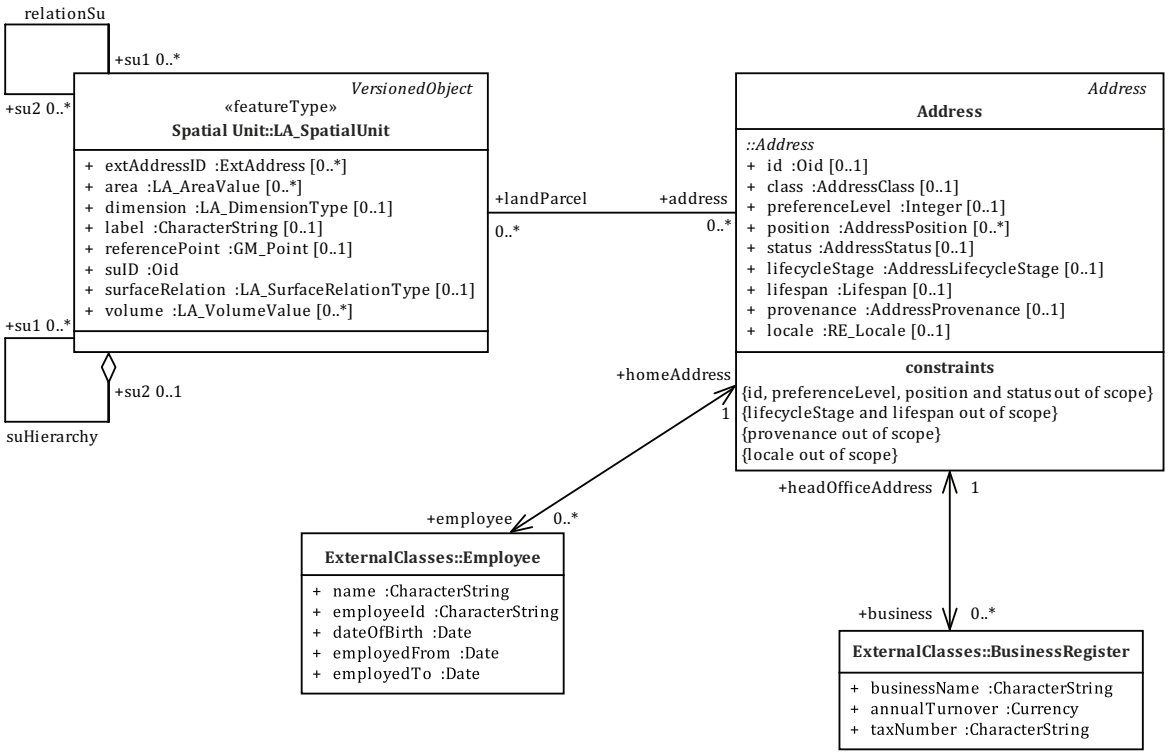

Figure C.18 — Linking sample addresses to external data

C.4 Diagrams from other profiles

Figure C.19 shows how address classes can be defined by modelling them as specializations of the address class. If this method is used, the class attribute in the specializations of address shall be mandatory and a constraint shall indicate the value for Address.class in each specialization. This method requires the address specializations to be aggregated from specialized AddressComponent classes illustrated in Figure C.20. If this method is used, the matrix, as described in the requirements for Address profile documentation in 7.5.2 and illustrated in Table C.4, reflects the associations between specializations of Address and the specialized AddressComponent classes in the model.

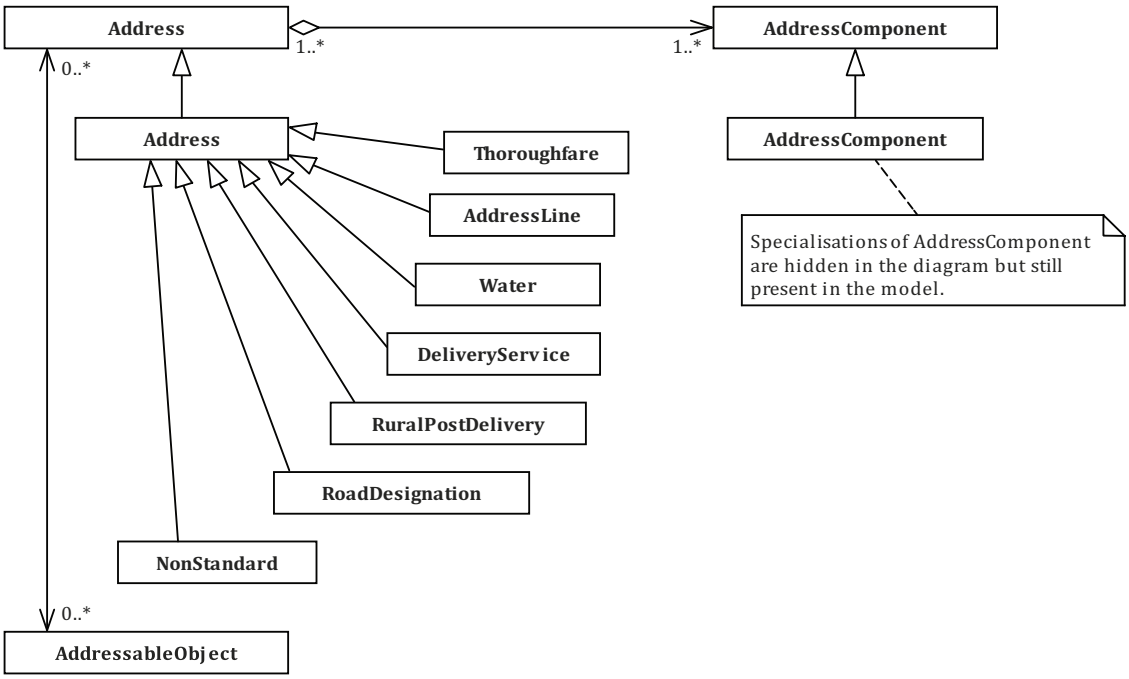

Figure C.19 — Address class specializations

Figure C.20, which is taken from a jurisdiction profile, shows how DeliveryService, a specialization of the Address class, is aggregated from a number of specializations of the AddressComponent class. The value of the type attribute in each of the AddressComponent specializations is specified as a constraint and the values (e.g. deliveryServiceName, deliveryServiceNumber, etc.) are taken from a profile specialization of the codelist AddressComponentType (not illustrated). The type of the value attribute is also specified as a constraint for each of the AddressComponent specializations and the types (e.g. CharacterString, Number, etc.) are taken from ISO 19103.

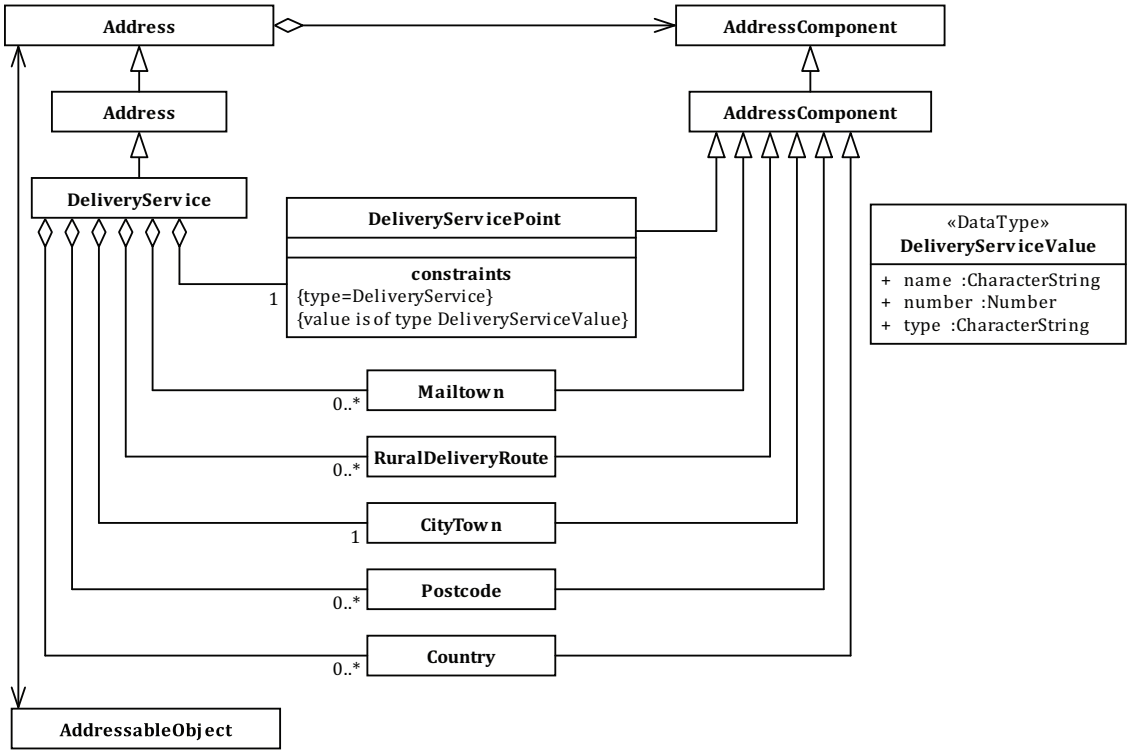

Figure C.20 — DeliveryService address aggregated from specializations of AddressComponent

The diagram in Figure C.21 illustrates how complex types for AddressComponentValue.value can be modelled. In the jurisdiction providing the diagram, many component types in the ThoroughfareAddress AddressClass have sub-components. For example, a Road has a mandatory name and optional designation, prefix, suffix and type elements. This is modelled as a <<type>> class, RoadValue, having these attributes which is cited by the value constraint in the Road specialization class.

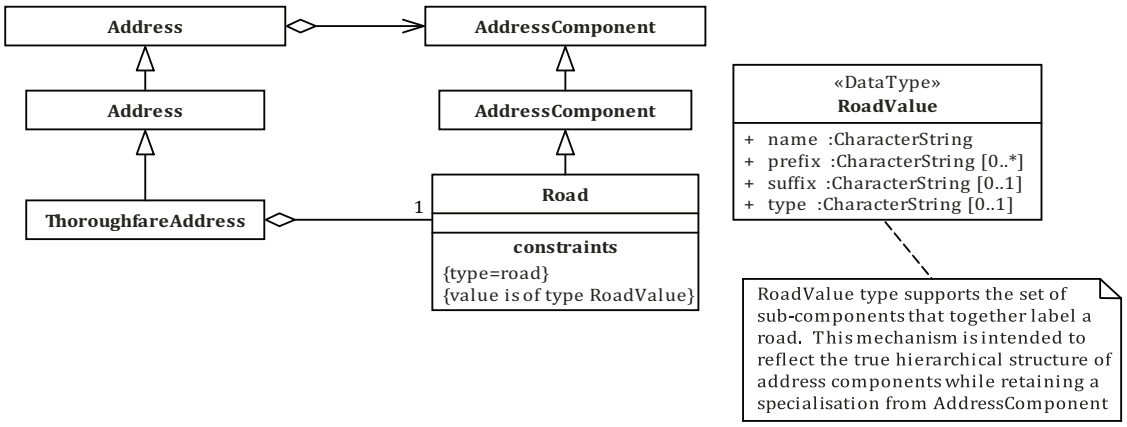

Figure C.21 — Complex type RoadValue for AddressComponentValue.value

Figure C.22 provides more examples of how complex types for AddressComponentValue.value can be modelled. In the jurisdiction providing the diagram, many component types in the ThoroughfareAddress AddressClass have sub-components. For example, a RoadNumber has a mandatory numeric attribute, optional alphanumeric attributes, and optional high and low range numbers. These attributes are modelled as a <<type>> class, RoadNumberValue cited by the value constraint in the RoadNumber class.

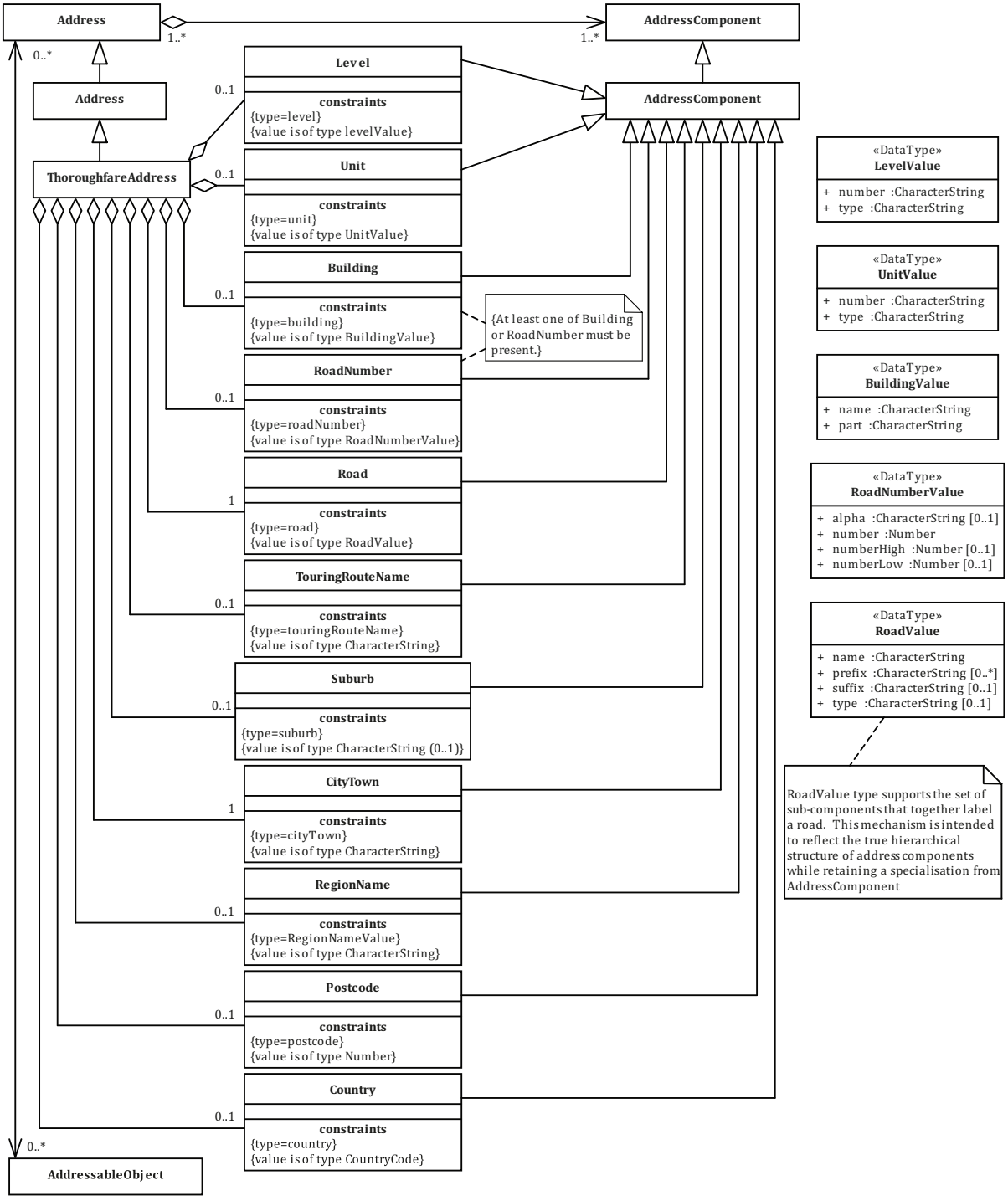

Figure C.22 — More complex types for AddressComponentValue.value

Figure C.23 shows how the position of an AddressableObject can be represented by a generic position as well as by one or more domain-specific positions. The diagram is from a profile this part of ISO 19160 which defines a positioning service for emergency response and utility agencies. The generic position is represented by the position attribute in the AddressableObject class. Domain-specific positions are represented by position attributes in the UtilityRegister and EmergencyAccessRegister classes. The latter each have a one-to-many association with the AddressableObject. The codelist AddressPositionType is specialized to define both generic access points, such as buildingCentroid, and domain-specific access points, such as serviceConnectionPoint (e.g. for a gas meter). A constraint on the AddressableObject restricts the allowed position types to the generic types, i.e. to buildingAccessPoint, buildingCentroid, and propertyCentroid. Constraints on UtilityRegister and EmergencyAccessRegister

restrict the allowed position types to the domain-specific types, i.e. serviceConnectionPoint and emergencyAccess respectively.

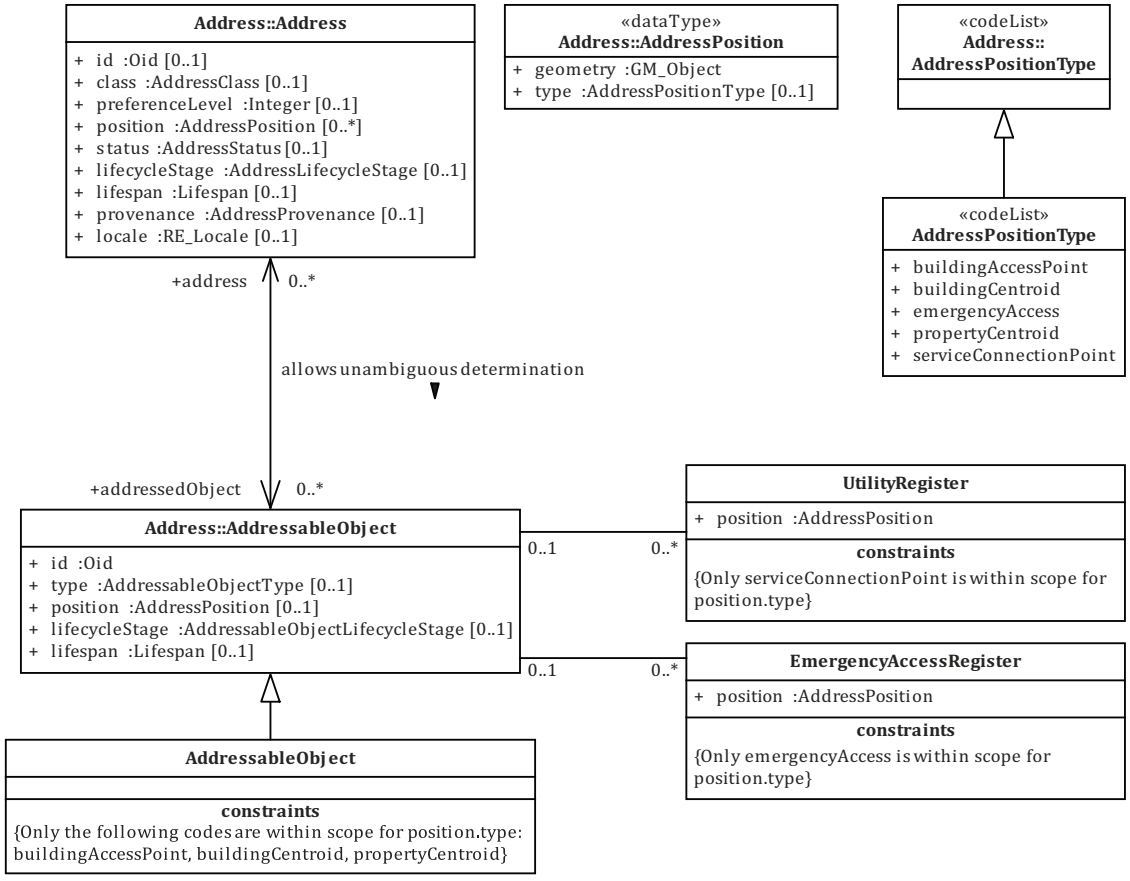

Figure C.23 — Generic and domain-specific positions

Annex D
(informative)

Examples: Lifecycle and lifespan of an address, address component and addressable object

D.1 General

Although the lifecycle of an address and its components sometimes mirror the lifecycle of the addressable object which it references, there are also many instances where an address or one of the components that make up an address may change in response to events unrelated to physical changes in the addressable object. Examples include the following:

— the municipal authority may propose an address for a property that has not yet been built;

— a new occupant may wish an existing building to be known by a new name;

— the municipal authority may change the name of a street;

— the postal operator may make a change to a postcode to reflect new delivery patterns;

— an error in the recording of an address component or attribute may need to be corrected.

The address model distinguishes between th following four sets of temporal attributes:

a) those relating to the period over which the representation of the address, address component or addressable object existed *in a dataset* represented by the beginLifespan, endLifespan and version attributes in the Lifespan type;

b) those relating to the period over which the address, address component or addressable object was valid/existed *in the physical world* represented by the validFrom and validTo attributes in the Lifespan type;

c) those reflecting the different stages through which an address, address component or addressable object proceeds in its lifecycle represented by the lifecycleStage attribute;

d) those reflecting the period over which an address was associated with an addressable object represented by the AddressedPeriod association class.

The content of this Annex illustrates how to use the attributes and how to maintain integrity between them. Each row in the tables of this Annex represents an instance in the dataset. The use of bold and italic indicates inserts of and updates to an instance, respectively. For validFrom and beginLifespan dates, it is assumed that the timestamp is 00:00:00. For validTo and endLifespan dates, it is assumed that the timestamp is 23:59:59.

Annex D is based on Annex C in D2.8.I.5 INSPIRE.[10]

D.2 Lifecycle of a thoroughfare address component

Event A (Table D.1):

2007-02-01: The addressing authority proposes a thoroughfare with the name “West Street”.

2007-02-03: The thoroughfare name is recorded in the dataset.

Table D.1 — AddressComponent attributes after Event A

id	type	value	lifecycle Stage	lifespan. validFrom	lifespan. validTo	lifespan. version	lifespan. beginLifespan	lifespan. endLifespan
12345	***thoroughfarename***	***West Street***	***proposed***	***2007–02–01***		***1***	***2007–02–03***	

Event B (Table D.2):

2007-07-01: The thoroughfare name is approved.

2007-07-11: This is now reflected in the dataset.

Table D.2 — AddressComponent attributes after Event B

id	type	value	lifecycle Stage	lifespan. validFrom	lifespan. validTo	lifespan. version	lifespan. beginLifespan	lifespan. endLifespan
12345	thoroughfarename	West Street	proposed	2007–02–01	***2007–06–30***	1	2007–02–03	***2007–07–10***
12345	***thoroughfarename***	***West Street***	***current***	***2007–07–01***		***2***	***2007–07–11***	

NOTE In a different scenario, the timestamp for the decision to take effect could be after the timestamp for recording it in the dataset.

Event C (Table D.3):

2009-02-13: The addressing authority decides to change the thoroughfare name to “Centre Street”. The new name shall take effect on 2009-03-01.

2009-02-15: The decision is recorded by updating the dataset.

Table D.3 — AddressComponent attributes after Event C

id	type	value	lifecycle Stage	lifespan. validFrom	lifespan. validTo	lifespan. version	lifespan. beginLifespan	lifespan. endLifespan
12345	thoroughfarename	West Street	proposed	2007–02–01	2007–06–30	1	2007–02–03	2007–07–10
12345	thoroughfarename	West Street	current	2007–07–01	***2009–02–28***	2	2007–07–11	***2009–02–14***
12345	***thoroughfarename***	***Centre Street***	***current***	***2009–03–01***		***3***	***2009–02–15***	

Event D (Table D.4):

2010-04-20: The addressing authority decides to change the spelling of the street from “Centre Street” to ‘Center Street’. The new name will take effect on 2010-05-01.

2010-04-25: The decision is recorded by updating the dataset.

Table D.4 — AddressComponent attributes after Event D

id	type	value	lifecycle Stage	lifespan. validFrom	lifespan. validTo	lifespan. version	lifespan. beginLifespan	lifespan. endLifespan
12345	thoroughfarename	West Street	proposed	2007–02–01	2007–06–30	1	2007–02–03	2007–07–10
12345	thoroughfarename	West Street	current	2007–07–01	2009–02–28	2	2007–07–11	2009–02–14
12345	thoroughfarename	Centre Street	current	2009–03–01	***2010–04–30***	3	2009–02–15	***2010–04–24***
12345	***thoroughfarename***	***Center Street***	***current***	***2010–05–01***		***4***	***2010–04–25***	

Event E (Table D.5):

2011-01-17: The addressing authority approves a construction project which will result in the existing "Center Street" being abandoned from 2011 to 02-01. From this date the name will be historic.

2011-01-18: The decision is recorded by updating the dataset.

Table D.5 — AddressComponent attributes after Event E

id	type	value	lifecycle Stage	lifespan. validFrom	lifespan. validTo	lifespan. version	lifespan. beginLifespan	lifespan. endLifespan
12345	thoroughfarename	West Street	proposed	2007–02–01	2007–06–30	1	2007–02–03	2007–07–10
12345	thoroughfarename	West Street	current	2007–07–01	2009–02–28	2	2007–07–11	2009–02–14
12345	thoroughfarename	Centre Street	current	2009–03–01	2010–04–30	3	2009–02–15	2010–04–24
12345	thoroughfarename	Center Street	current	2010–05–01	***2011–01–31***	4	2010–04–25	***2010–01–17***
12345	***thoroughfarename***	***Center Street***	***retired***	***2011–02–01***		***5***	***2010–01–18***	

D.3 Lifecycle of an address

Event A (Table D.6):

2012-06-01: The address authority proposes a new address for a planned subdivision of a property in "Mill Road". The proposal is published for consultation.

2012-06-03: The proposal is recorded in the dataset.

Table D.6 — Address attributes after Event A

id	class	addressCompo-nents	lifecycle Stage	lifespan. validFrom	lifespan. validTo	lifespan. version	lifespan. beginLifespan	lifespan. end-Lifespan
54321	***StreetAddress***	***114A*** ***Mill Road*** ***Hatfield***	***proposed***	***2012–06–01***		***1***	***2012–06–03***	

Event B (Table D.7):

2012-09-10: The authority approves the new address, but it is decided to rather use "114-1 Mill Road" for the address. The address will be official from 2012-10-01.

2012-09-12: The decision is recorded by updating the dataset.

Table D.7 — Address attributes after Event B

id	class	addressComponents	lifecycle Stage	lifespan. validFrom	lifespan. validTo	lifespan. version	lifespan. beginLifespan	lifespan. end-Lifespan
54321	StreetAddress	114A Mill Road Hatfield	proposed	2012–06–01	***2012–09–30***	1	2012–06–03	***2012–09–11***
54321	***StreetAddress***	***114–1*** ***Mill Road*** ***Hatfield***	***current***	***2012–10–01***		***2***	***2012–09–12***	

Event C (Table D.8):

2013-01-10: The property is merged together with the neighbouring property and it is decided that the address will no longer be valid from this date.

2013-01-15: The decision is recorded by updating the dataset.

Table D.8 — Address attributes after Event C

id	class	addressComponents	lifecycle Stage	lifespan. validFrom	lifespan. validTo	lifespan. version	lifespan. beginLifespan	lifespan. end-Lifespan
54321	StreetAddress	114A Mill Road Hatfield	proposed	2012–06–01	2012–19–30	1	2012–06–03	2012–09–11
54321	StreetAddress	114–1 Mill Road Hatfield	current	2012–10–01	***2013–01–09***	2	2012–09–12	***2013–01–14***
54321	***StreetAddress***	***114–1*** ***Mill Road*** ***Hatfield***	***retired***	***2013–01–10***		***3***	***2013–01–15***	

D.4 Lifecycle of an addressable object

Event A (Table D.9):

2012-06-01: The authority proposes the construction of a new office building to be called “Green Acres”. The proposal is published for consultation.

2012-06-03: The proposal is recorded in the dataset.

Table D.9 — Addressable object attributes after Event A

id	type	lifecycle Stage	lifespan. validFrom	lifespan. val-idTo	lifespan. version	lifespan. beginLifespan	lifespan. endLifespan
73746	building	proposed	2012–06–01		1	2012–06–03	

Event B (Table D.10):

2012-09-10: The authority approves the construction of the building.

2012-09-12: The decision is recorded by updating the dataset.

Table D.10 — Addressable object attributes after Event B

id	type	lifecycle Stage	lifespan. validFrom	lifespan. val-idTo	lifespan. version	lifespan. beginLifespan	lifespan. endLifespan
73746	building	proposed	2012–06–01	***2012–09–09***	1	2012–06–03	***2012–09–11***
73746	***building***	***current***	***2012–09–10***		***2***	***2012–09–12***	

Event C (Table D.11):

2013-01-10: The authority approves that the building is to be demolished on 2013-02-10.

2013-01-15: The decision is recorded by updating the dataset.

Table D.11 — Addressable object attributes after Event C

id	type	lifecycle Stage	lifespan. validFrom	lifespan. val-idTo	lifespan. version	lifespan. beginLifespan	lifespan. endLifespan
73746	building	proposed	2012–06–01	2012–09–09	1	2012–06–03	2012–09–11
73746	building	current	2012–09–10	***2013–02–09***	2	2012–09–12	***2013–01–14***
73746	***building***	***retired***	***2013–02–10***		***3***	***2013–01–15***	

D.5 Period of association between an address and an addressable object

Event A (Table D.12):

2000-06-01: The authority approves “44B Bakery Road” as the address for “The Milk House” building with object identifier 7762.

Table D.12 — AddressedPeriod after Event A

Address.id	AddressedPeriod.validFrom	AddresssedPeriod.validTo	AddressableObject.id
283746	2000–06–01		7762

Event B (Table D.13):

2012-09-10: The authority approves the destruction of “The Milk House” so that a new building may be erected in its place.

Table D.13 — AddressedPeriod after Event B

Address.id	AddressedPeriod.validFrom	AddresssedPeriod.validTo	AddressableObject.id
283746	2000–06–01	**2012–09–10**	7762

Event C (Table D.14):

2013-01-10: The authority approves “44B Bakery Road” as the address for the new building, “The Computer Shop” building with object identifier 8632.

Table D.14 — AddressedPeriod after Event C

Address.id	AddressedPeriod.validFrom	AddresssedPeriod.validTo	AddressableObject.id
283746	2000–06–01	2012–09–10	7762
283746	2013–01–10		8632

Annex E
(informative)

Examples: Address component alternatives and address aliases

E.1 General

E.2 and E.3 provide examples of address component value alternatives and address aliases, respectively.

E.2 Address component value alternatives

Tables E.1 to E.3 illustrate the attribute values of AddressComponentValue in different examples of alternative component values.

The example in Table E.1 shows the attributes of an address component and its abbreviated alternative. In this example, the abbreviated alternative has lower preference.

Table E.1 — Abbreviated alternative for a thoroughfare name

	value	type	preferenceLevel	locale.language
value[0]	Gordon Road	defaultValue	1	EN
value[1]	Gordon Rd	abbreviatedAlternative	2	EN

The example in Table E.2 shows the attributes of an address component and its colloquial alternative. In this example, the colloquial alternative has lower preference.

Table E.2 — Colloquial alternative for a locality name

	value	type	preferenceLevel	locale.language
value[0]	Pretorius Park Ext 5	defaultValue	1	EN
value[1]	Woodhill	colloquialAlternative	2	EN

The example in Table E.3 shows the attributes of an address component and its language alternative. In this example, the English (EN) and Afrikaans (AF) component values have the same (highest) preference level; the German alternative has a lower preference.

Table E.3 — Locale alternatives for a thoroughfare name

	value	type	preferenceLevel	locale.language
value[0]	Gordon Road	defaultValue	1	EN
value[1]	Gordonweg	localeAlternative	1	AF
value[2]	Gordonstrasse	localeAlternative	2	DE

E.3 Address aliases

Tables E.4 to E.6 illustrate the values of relevant attributes in the Address, AddressComponent, and AddressAlias classes for different sets of aliases. The example in Table E.4 shows two addresses referencing the same property on a street corner. One of the addresses has a higher preference level.

Table E.4 — Address aliases on a street corner (unspecifiedAlias)

Address				AddressComponent	AddressAlias
id	**class**	**preferenceLevel**	**locale.language**	**value**	**type**
11111	StreetAddress	1	EN	902 Gordon Road Queenswood	unspecifiedAlias
22222	StreetAddress	2	EN	36 Fry Street Queenswood	

The example in Table E.5 shows two addresses of different address classes referencing the same property. They have the same preference level.

Table E.5 — Address aliases from different classes (classAlias)

Address				AddressComponent	AddressAlias
id	**class**	**preferenceLevel**	**locale.language**	**value**	**type**
11111	StreetAddress	1	EN	902 Gordon Road Queenswood	classAlias
33333	PostalAddress	1	EN	902 Gordon Road Queenswood 9837	

The example in Table E.6 shows two addresses of the same address class in different languages referencing the same property. In the example, the addresses have the same preference level.

Table E.6 — Address aliases in different languages (localeAlias)

Address				AddressComponent	AddressAlias
id	**class**	**preferenceLevel**	**locale.language**	**value**	**type**
11111	StreetAddress	1	EN	902 Gordon Road Queenswood	localeAlias
44444	StreetAddress	1	AF	902 Gordonweg Queenswood	

Annex F
(informative)

Examples: External classes

F.1 General

This Annex includes a number of diagrams to illustrate how external (i.e. not included in this part of ISO 19160) classes may be linked to classes in this part of ISO 19160 address model.

F.2 Associations between addresses and external data

Figure F.1 shows how external data, such as employee, business and client information, may be associated with one or more address.

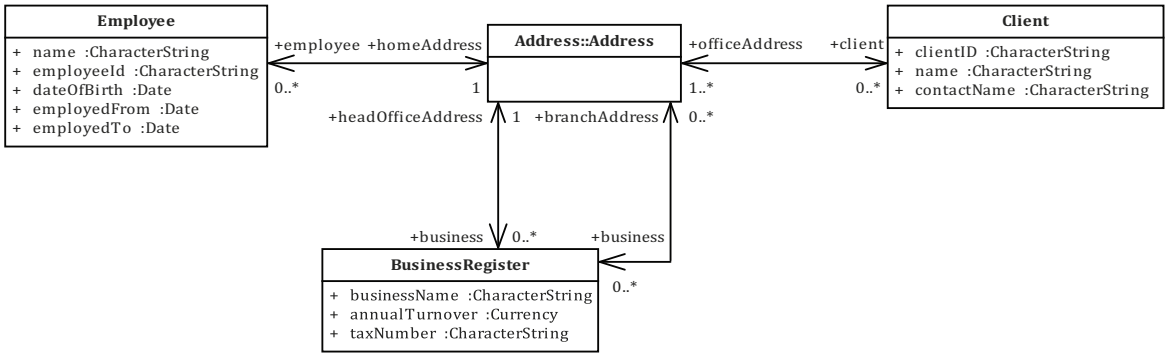

Figure F.1 — Associating external data with addresses

F.3 External classes deriving from ReferenceObject

Figures F.2 and F.3 show how external data can be associated with an address component in a profile of the address model.

In Figure F.2, ReferenceSpatialObject is derived from ReferenceObject which is associated with an AddressComponent. The external classes, LA_SpatialUnit and LA_SpatialUnitGroup (defined in ISO 19152), are associated to the ReferenceSpatialObject.

In Figure F.3, ReferenceSpatialObject, as well as the external classes, Organization and Person, are derived from ReferenceObject which is associated with an AddressComponent. The external classes, AdministrativeArea, and Thoroughfare are specializations of ReferenceSpatialObject. Figure F.3 also illustrates that these specializations could be associated to each other.

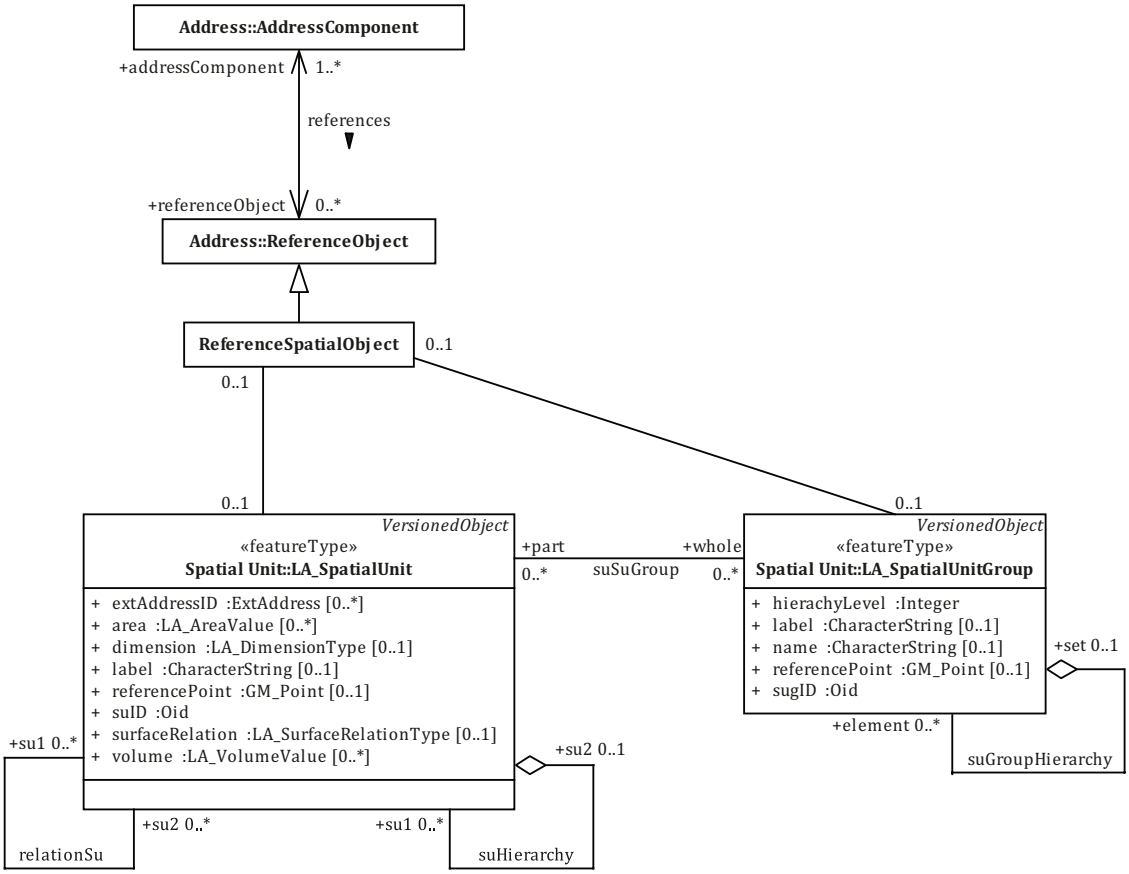

Figure F.2 — Associating external land administration classes via ReferenceSpatialObject with AddressComponent

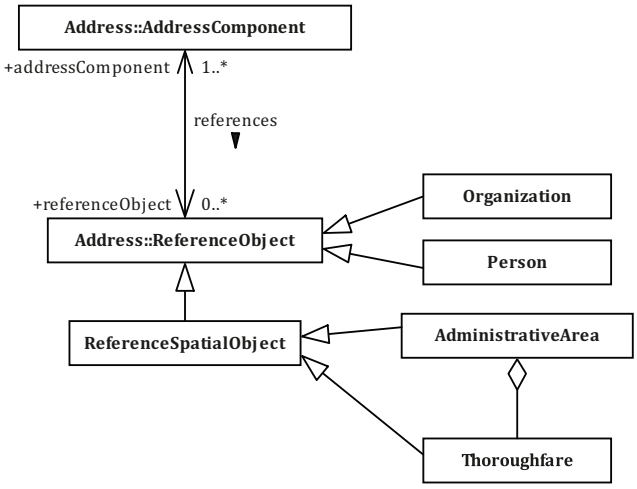

Figure F.3 — Deriving external classes from ReferenceObject and ReferenceSpatialObject

Bibliography

[1] ISO 3166-1:2013, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*

[2] ISO 5127:2001, *Information and documentation — Vocabulary*

[3] ISO 13527, *Space data and information transfer systems — XML formatted data unit (XFDU) structure and construction rules*

[4] ISO 15836, *Information and documentation — The Dublin Core metadata element set*

[5] ISO 21127, *Information and documentation — A reference ontology for the interchange of cultural heritage information*

[6] AS 4590:2006, *Interchange of client information, Standards Australia*

[7] AS/NZS 4819, *Rural and urban addressing, Standards Australia and Standards New Zealand*

[8] BS 7666-0, *Spatial datasets for geographical referencing — Part 0: General model for gazetteers and spatial referencing, British Standards Institute*

[9] Akeno K. 2008). Japanese address system, *ISO Workshop on address standards — Considering the issues related to an international address standard*, held under the auspices of WG7, Information Communities, of ISO/TC 211, Geographic information on 25 May 2008 Copenhagen, Denmark available at http://www.isotc211.org/Address/Copenhagen_Address_Workshop/workshop.htm, accessed 1 May 2013

[10] D2.8.1.5 *INSPIRE Data Specification on Addresses — Guidelines*, INSPIRE Thematic Working Group on. Addresses, 2010

[11] Japan (1962). *Address Indication Act*

[12] Japan (1899). *Real Property Registration Act*

[13] New Zealand Post. *Postal Address File Technical Guide.* Available at http://www.nzpost.co.nz/sites/default/files/uploads/shared/paftechguide.pdf, accessed 22 July 2013

[14] New Zealand Geospatial Office. 2011). *The Spatial Data Infrastructure Cookbook v.1.1.* Available at http://www.linz.govt.nz/geospatial-office/spatial-data-infrastructure/sdi-cookbook-v11-home, accessed 1 October 2013

[15] *Review summary of the ISO 19160 stage zero project.* ISO/TC 211, *Geographic information/Geomatics*, document N 3188

[16] SANS 1883-1, *Geographic information – Address, Part 1: Data format of addresses*, South African Bureau of Standards (SABS)

[17] UPU S42, *International postal address components and templates*, Universal Postal Union, Berne, Switzerland

[18] *US Thoroughfare, Landmark, and Postal Address Data Standard*, United States Federal Geographic Data Committee (US FGDC)

ICS 35.240.70

Price based on 57 pages