INTERNATIONAL STANDARD

ISO/IEC 5087-1

First edition
2023-08

Information technology — City data model —

Part 1: Foundation level concepts

Reference number
ISO/IEC 5087-1:2023(E)

Documents
Document Sharing Hub
© ISO/IEC 2023

Full document details: ISO_IEC 5087-1_2023

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2023

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 5087-1_2023

Contents

Page

Foreword ... **v**
Introduction ... **vi**
1 Scope ... **1**
2 Normative references ... **1**
3 Terms and definitions ... **1**
4 Abbreviated terms and namespaces ... **3**
5 General ... **4**
5.1 Unique identifiers ... 4
5.2 Reference to existing patterns ... 5
6 Foundational ontologies ... **5**
6.1 General ... 5
6.2 Generic properties ... 5
6.2.1 General ... 5
6.2.2 Key Properties ... 6
6.3 Mereology pattern ... 6
6.3.1 General ... 6
6.3.2 Key classes and properties ... 6
6.3.3 Formalization ... 7
6.4 City Units Pattern ... 8
6.4.1 General ... 8
6.4.2 Key classes and properties ... 8
6.4.3 Formalization ... 9
6.5 Time Pattern ... 9
6.5.1 General ... 9
6.6 Change pattern ... 10
6.6.1 General ... 10
6.6.2 Key classes and properties ... 10
6.6.3 Formalization ... 17
6.7 Location pattern ... 17
6.7.1 General ... 17
6.7.2 Key classes and properties ... 18
6.7.3 Formalization ... 19
6.8 Activity pattern ... 20
6.8.1 General ... 20
6.8.2 Key classes and properties ... 20
6.8.3 Formalization ... 24
6.9 Recurring Event pattern ... 25
6.9.1 General ... 25
6.9.2 Key classes and properties ... 25
6.9.3 Formalization ... 28
6.10 Resource pattern ... 29
6.10.1 General ... 29
6.10.2 Key classes and properties ... 29
6.10.3 Formalization ... 33
6.11 Agent pattern ... 34
6.11.1 General ... 34
6.11.2 Key classes and properties ... 34
6.11.3 Formalization ... 34
6.12 Organization Structure pattern ... 35
6.12.1 General ... 35
6.12.2 Key classes and properties ... 35
6.12.3 Formalization ... 35

© ISO/IEC 2023 – All rights reserved

AVA Documents
Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

6.13 Agreement pattern ... 35
6.13.1 General ... 35
6.13.2 Key classes and properties ... 35
6.13.3 Formalization ... 37
6.14 Provenance pattern ... 38
6.14.1 General ... 38
6.14.2 Key classes and properties ... 38
6.14.3 Formalization ... 38
Annex A (informative) **Implementation alternatives for additional change semantics** ... **39**
Annex B (informative) **Relationship to existing standards** ... **41**
Annex C (informative) **Extended recurring event example** ... **47**
Annex D (informative) **Location of pattern implementations** ... **48**
Bibliography ... **49**

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

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 document 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 or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

A list of all parts in the ISO/IEC 5087 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

Introduction

The intended audience for this document includes municipal information systems departments, municipal software designers and developers, and organizations that design and develop software for municipalities.

Cities today face a challenge of how to integrate data from multiple, unrelated sources where the semantics of the data are imprecise, ambiguous and overlapping. This is especially true in a world where more and more data are being openly published by various organizations. A morass of data is increasingly becoming available to support city planning and operations activities. In order to be used effectively, it is necessary for the data to be unambiguously understood so that it can be correctly combined, avoiding data silos. Early successes in data "mash-ups" relied upon an independence assumption, where unrelated data sources were linked based solely on geospatial location, or a unique identifier for a person or organization. More sophisticated analytics projects that require the combination of datasets with overlapping semantics entail a significantly greater effort to transform data into something useable. It has become increasingly clear that integrating separate datasets for this sort of analysis requires an attention to the semantics of the underlying attributes and their values.

A common data model enables city software applications to share information, plan, coordinate and execute city tasks, and support decision making within and across city services, by providing a precise, unambiguous representation of information and knowledge commonly shared across city services. This requires a clear understanding of the terms used in defining the data, as well as how they relate to one another. This requirement goes beyond syntactic integration (e.g. common data types and protocols), it requires semantic integration: a consistent, shared understanding of the meaning of information.

To motivate the need for a standard city data model, consider the evolution of cities. Cities deliver physical and social services that have traditionally operated as silos. If during the process of becoming smarter, transportation, social services, utilities, etc. were to develop their own data models, the result would be smarter silos. To create truly smart cities, data needs to be shared across these silos. This can only be accomplished through the use of a common data model. For example, "Household" is a category of data that is commonly used by city services. Members of Households are the source of transportation, housing, education and recreation demand. This category represents who occupies a home, their age, their occupations, where they work, their abilities, etc. Though each city service can potentially gather and/or use different aspects of a Household, much of the data needs to be shared with each other.

Supporting this interoperability among city datasets is particularly challenging due to the diversity of the domain, the heterogeneity of its data sources, and data privacy concerns and regulations. The purpose of this document is to support the precise and unambiguous specification of city data using the technology of ontologies[1],[2] as implemented in the Semantic Web[3]. By doing so it will:

— enable the computer representation of precise definitions, thereby reducing the ambiguity of interpretation;

— remove the independence assumption, thereby allowing the world of Big Data, open source software, mobile apps, etc., to be applied for more sophisticated analysis;

— achieve semantic interoperability, namely the ability to access, understand, merge and use data available from datasets spread across the Semantic Web;

— enable the publishing of city data using Semantic Web and ontology standards, and

— enable the automated detection of city data inconsistency, and the root causes of variations.

With a clear semantics for the terminology, it is possible to perform consistency analysis, and thereby validate the correct use of the document.

Figure 1 identifies the three levels of the ISO/IEC 5087 series. The lowest level, defined in ISO/IEC 5087-1 (this document) provides the classes, properties and logical computational definitions for representing the concepts that are foundational to representing any data. The middle level, defined

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

in ISO/IEC 5087-2:—[1], will provide the classes, properties and logical computational definitions for representing concepts common to all cities and their services but not specific to any service. The top level provides the classes, properties and logical computational definitions for representing service domain specific concepts that are used by other services across the city. For example, ISO/IEC TS 5087-3:—[2], will define the transportation concepts. In the future, additional parts will be added to the ISO/IEC 5087 series covering further services such as education, water, sanitation, energy, etc.

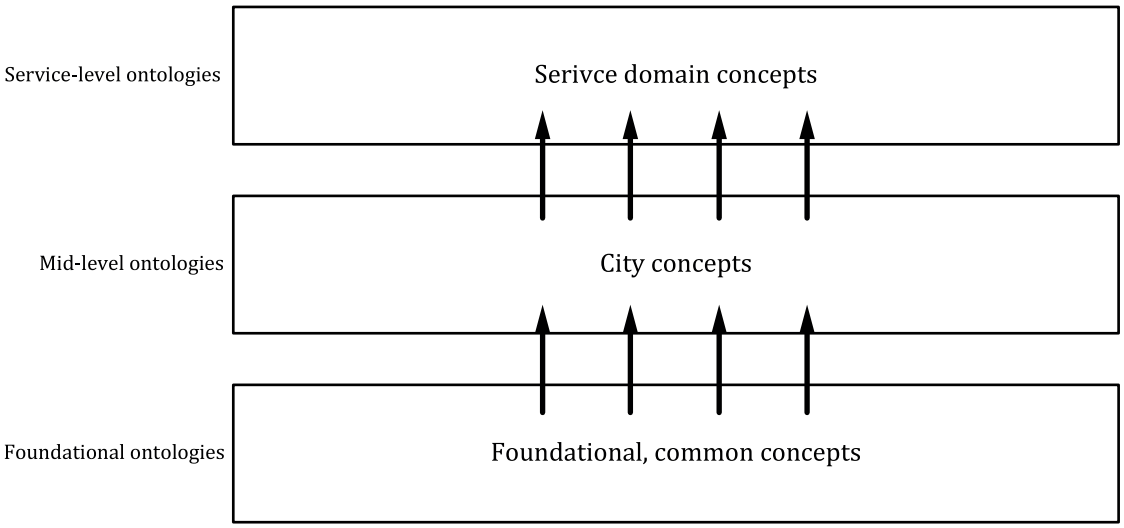

Figure 1 — Stratification of city data model

Figure 2 depicts example concepts for the three levels.

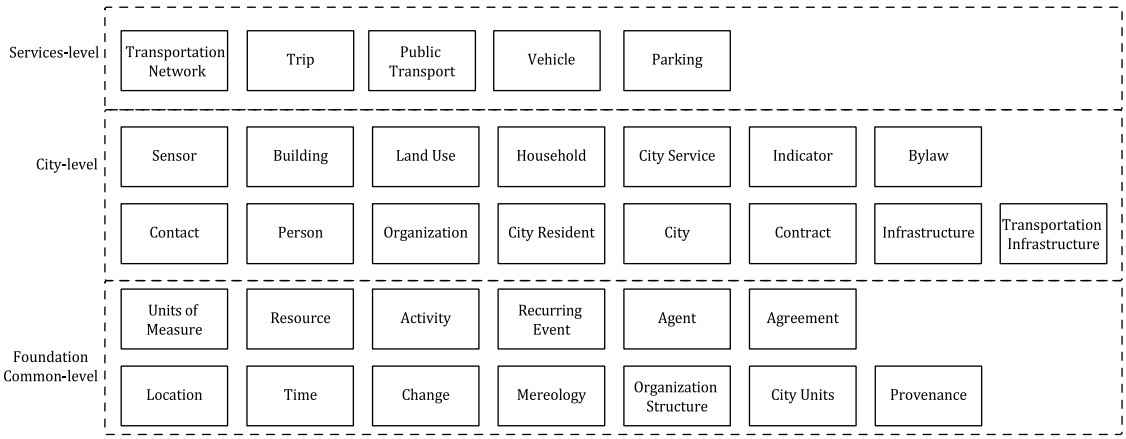

Figure 2 — Example concepts for each level

It is important to distinguish between the ISO/IEC 5087 series and the related, but distinct effort of ISO/IEC 30145-2. As specified in its Scope, ISO/IEC 30145-2:2020 "*specifies a generic knowledge management framework for a smart city, focusing on creating, capturing, sharing, using and managing smart city knowledge. It also gives the key practices which are required to be implemented to safeguard the use of knowledge, such as interoperability of heterogeneous data and governance of multi-sources services within a smart city.*" Figure 3 depicts the smart city knowledge management framework as described in ISO/IEC 30145-2. The smart city domain knowledge model includes a (cross-domain) core concept model and several domain knowledge models. This document defines the foundation level of the core concept model. ISO/IEC 5087-2 is intended to address some of the core concept model and cuts across

1) Under preparation. Stage at the time of publication: ISO/IEC DIS 5087-2:2023.

2) Under preparation.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

the domain knowledge models. There is a possibility that subsequent parts of the ISO/IEC 5087 series (not yet defined) will define knowledge models for the services of citizen livelihood, urban management and smart transportation illustrated in the Figure 3.

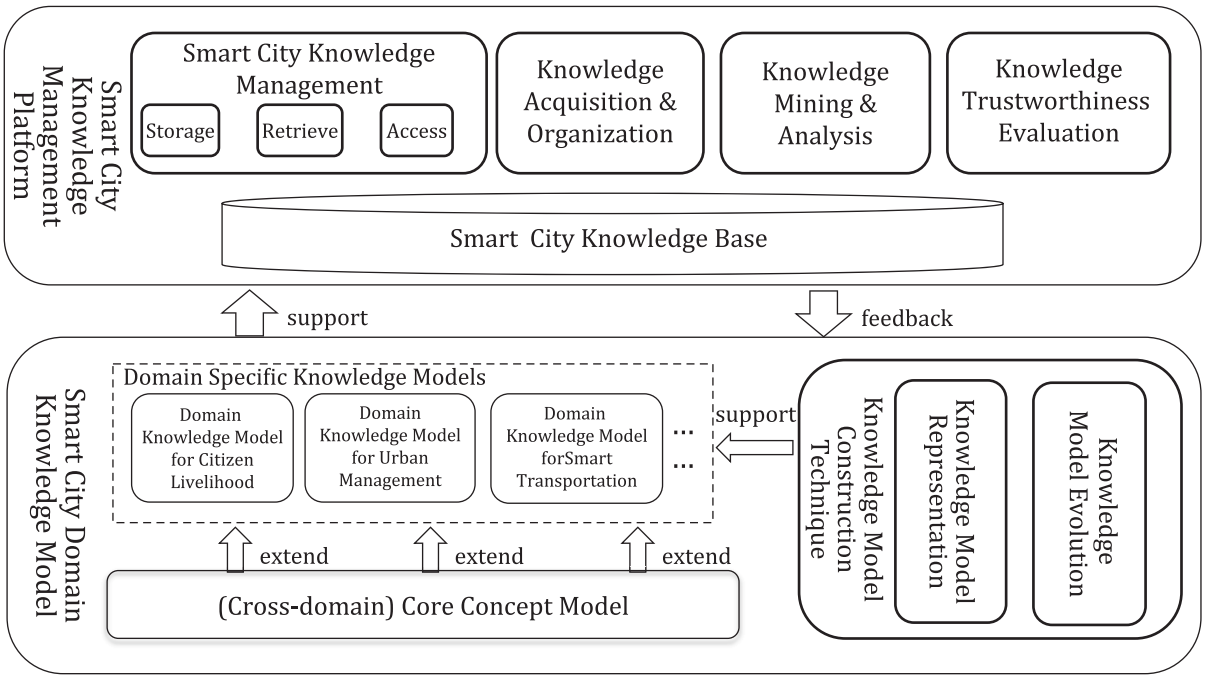

Figure 3 — The framework of smart city knowledge management from ISO/IEC 30145-2:2020

There are other existing standards that overlap conceptually with some of the terms presented in this document. The relationship between ISO/IEC 5087-1 and existing documents that address similar or related concepts is identified in Annex A.

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

Information technology — City data model —

Part 1: Foundation level concepts

1 Scope

This document is part of the ISO/IEC 5087 series, which specifies a common data model for cities. This document specifies the foundation level concepts.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 21972, *Information technology — Upper level ontology for smart city indicators*

OGC GeoSPARQL, A Geographic Query Language for RDF Data, OGC 11-052r4, Open Geospatial Consortium, 10 September 2012. https://www.ogc.org/standards/geosparql

The Ontology in OWL, W3C Candidate Recommendation 26 March 2020, https://www.w3.org/TR/owl-time/

PROV-O. The PROV Ontology, W3C Recommendation 30 April 2013, https://www.w3.org/TR/prov-o/

The Organization Ontology, W3C Recommendation 16 January 2016, https://www.w3.org/TR/vocaborg/

3 Terms and definitions

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

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
cardinality
number of elements in a set

[SOURCE: ISO/TS 21526:2019, 3.11]

3.2
description logic
DL
family of formal knowledge representation languages that are more expressive than propositional logic but less expressive than first-order logic

[SOURCE: ISO/IEC 21972:2020, 3.2]

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

3.3
manchester syntax
compact, human readable syntax for expressing Description Logic descriptions

[SOURCE: https://www.w3.org/TR/owl2-manchester-syntax/ (Copyright © 2012. World Wide Web Consortium. https://www.w3.org/Consortium/Legal/2023/doc-license).]

3.4
measure
value of the measurement (via the numerical_value property) which is linked to both Quantity and Unit_of_measure

[SOURCE: ISO/IEC 21972:2020, 3.4]

3.5
namespace
collection of names, identified by a URI reference, that are used in XML documents as element names and attribute names

Note 1 to entry: Names may also be identified by an IRI reference.

[SOURCE: ISO/IEC 21972:2020, 3.5, modified — Note 1 to entry has been added.]

3.6
ontology
formal representation of phenomena of a universe of discourse with an underlying vocabulary including definitions and axioms that make the intended meaning explicit and describe phenomena and their interrelationships

[SOURCE: ISO 19101-1:2014, 4.1.26]

3.7
ontology web language
ontology language for the Semantic Web with formally defined meaning

Note 1 to entry: OWL 2 ontologies provide classes, properties, individuals, and data values and are stored as Semantic Web documents.

[SOURCE: https://www.w3.org/TR/owl2-overview/ (Copyright © 2012. World Wide Web Consortium. https://www.w3.org/Consortium/Legal/2023/doc-license).]

3.8
quantity
property of a phenomenon, body, or substance, where the property has a magnitude that can be expressed by means of a number and a reference

Note 1 to entry: Quantities can appear as base quantities or derived quantities.

EXAMPLE 1 Length, mass, electric current (ISQ base quantities).

EXAMPLE 2 Plane angle, force, power (derived quantities).

[SOURCE: ISO 80000-1:2009, 3.1, modified — NOTEs 1 to 6 have been removed; new Note 1 to entry and two EXAMPLEs have been added.]

3.9
Semantic Web
W3C's vision of the Web of linked data

Note 1 to entry: Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write rules for handling data. The goal is to make data on the Web machine-readable and more precise.

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

[SOURCE: https://www.w3.org/standards/semanticweb/(Copyright © 2015. World Wide Web Consortium. https://www.w3.org/Consortium/Legal/2023/doc-license).]

3.10
unit_of_measure
definite magnitude of a quantity, defined and adopted by convention and/or by law

[SOURCE: ISO/IEC 21972:2020, 3.9]

4 Abbreviated terms and namespaces

DL	description logic
OWL	ontology web language
RDF	resource description framework
RDFS	resource description framework schema
IRI	international resource identifier

The following namespace prefixes are used in this document:

— activity: https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/Activity/

— agent: https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/Agent/

— agreement: https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/Agreement/

— change: https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/Change/

— cityunits: https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/CityUnits/

— genprop: https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/GenericProperties/

— geo: http://www.opengis.net/ont/geosparql#

— i72: http://ontology.eil.utoronto.ca/5087/2/iso21972/

— loc: https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/SpatialLoc/

— mereology: https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/Mereology/

— org: http://www.w3c.org/ns/org#

— org_s: https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/OrganizationStructure/

— owl: http://www.w3.org/2002/07/owl#

— partwhole: https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/Mereology/

— prov: http://www.w3.org/ns/prov-o#

— 5087prov: https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/Prov/

— rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#

— rdfs: http://www.w3.org/2000/01/rdf-schema#

— recurringevent: https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/RecurringEvent/

— resource: https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/Resource/

— time: http://www.w3.org/2006/time#

— xsd: http://www.w3.org/2001/XMLSchema#

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

The formalization of the classes in this document is specified using the following table format, which is a simplification of description logic (DL) where the first column identifies the class name, the second column its properties (a class is defined as the subclass of all of its properties) and the third column each property's range restriction. It shall be read as: The <Class> is a subClassOf the conjunction of the associated <property>s with their <value>s. Range restrictions are specified using the Manchester syntax. For example, Table 1 specifies that Agent is a subclass of the intersection of genprop:hasName exactly 1 xsd:string and resource:resourceOf only resource:TerminalResourceState and performs only activity:Activity.

Table 1 — Example class formalization

Class	Property	Value restriction
Agent	genprop:hasName	exactly 1 xsd:string
	resource:resourceOf	only resource:TerminalResourceState
	performs	only activity:Activity

The following value restrictions are used in this document:

— "min n": Specifies that the property has to have a minimum n values.

— "max n": Specifies that the property has to have a maximum n values.

— "exactly n": Specifies that the property has to have exactly n values.

— "only": Specifies that the values of the property can only be an instance/type of the class specified, e.g., a string, integer or another class such as Organization.

CamelCase is used for specifying classes, properties and instances. For example, "legalName" instead of "legal_name". The first letter of a class name is capitalized. The first letter of a property and instance name are not capitalized. An instance of a class shall satisfy the class's definition. The instance's properties and values shall satisfy the value restrictions of the class it is an instance of.

The formalization of the properties in this document is done similarly, using the following table format that allows for the identification of properties and their sub-properties, inverse properties, or other characteristics. It is to be read as: The <property> is <characteristic> of <value>, or simply the <property> is <characteristic> if no value is applicable. For example, in Table 2 hasPrivilege is a sub-property of the agentInvolvedIn property. Characteristics are specified using the Manchester syntax.

Table 2 — Example property formalization

Property	Characteristic	Value (if applicable)
hasPrivilege	rdfs:subPropertyOf	agentInvolvedIn
	Irreflexive	

In the case of DL definitions of classes where the simplified table representation is insufficient, the DL specification will be supplied.

The patterns defined in this document have also been implemented in OWL and made available online. The location of these encodings is identified in Annex D.

5 General

5.1 Unique identifiers

All classes, properties and instances of classes have a unique identifier that conforms to Linked Data/Semantic Web standards. The unique identifier is an IRI. When using ISO/IEC 5087-1 (this document) in an application, a class is identified by the IRI for the pattern of which it is a member, followed by the class name. In the Agent example in Clause 4, the Agent class's unique identifier would be:

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/Agent/Agent

Breaking the IRI down:

— "5087" identifies the series number

— "-1" identifies the part number

— "ed-1" indicates that the class is defined in edition 1 of the standard

— "en" indicates that the class is defined in a pattern implemented in English

— The first "Agent" identifies the Agent pattern

— The second "Agent" identifies the Agent class within the Agent pattern

The IRI can be shortened using the prefix's defined in Clause 4:

agent:Agent

where agent: is the prefix for the Agent pattern.

Properties are identified in the same manner. The IRIs of individuals created by an application of ISO/IEC 5087-1 would have IRIs unique to the application.

5.2 Reference to existing patterns

The practice of reusing and referencing existing standard vocabularies is an important practice in the context of the Semantic Web. It is important that existing standard patterns are included as normative references where appropriate, rather than duplicating and inserting the appropriate content as a pattern within this document. The use of shared vocabularies directly enables interoperability and shareability between implementations and avoids the additional work of attempting to map to these standards after the fact. Where possible, existing standardized ontologies have been included as normative references to specify the patterns below. In cases where an extension is required, this is done in such a way as to preserve the content of the normative reference in order to support interoperability and make the relationship transparent.

6 Foundational ontologies

6.1 General

Beyond the domain-specific subjects that are clearly identified in consideration of the requirements, there are fundamental concepts that are necessary to formulate an accurate definition of the domain. These concepts are defined in a series of foundational ontologies, so-named because they provide a reusable foundation for the development of other ontologies for city services. The clear definition and uncoupling of the foundational concepts make the fundamental commitments of the city data model clear and accessible to potential adopters. It also ensures interoperability and consistency in the representation of key concepts such as time and location. The city data model defines eleven foundational patterns to capture these concepts. A pattern is a set of concepts that are related by topic and inter-connected by properties, thereby forming a graph. A foundational pattern is a pattern composed of a set of foundational concepts. These are described in the following subclauses.

6.2 Generic properties

6.2.1 General

Most of the properties are identified and defined relative to a certain Class and in the context of a particular pattern in the following subclauses. However, there are certain exceptions where generic properties can be recognized as applicable to a wide range of classes with no common theme amongst

© ISO/IEC 2023 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

them. Such properties are defined separately as generic properties. This allows for the reference to these properties independent of any particular pattern. These generic properties are imported by all of the patterns defined in the ISO/IEC 5087 series.

6.2.2 Key Properties

The following generic properties have been identified:

— **hasName**: identifies the name of a certain object;

— **hasDescription**: specifies a description of a certain object;

— **hasIdentifier**: specifies an identifier for a certain object.

6.3 Mereology pattern

6.3.1 General

Notions of parthood are ubiquitous. While sometimes conflated, there are clear distinctions which can be made between different types of parthood. The Mereology pattern focuses on identifying these differences and making them explicit. The distinction between types of parthood may be best explained with the use of examples. An item may be *contained in* a car, but that does not make it a *component of* a car. For example, there may be a need to describe passengers or cargo being *contained in* a vehicle, but this relation needs to be distinguished from the parts and components that make up a vehicle. Similarly, the front of a car is intuitively a part of the car, but not a component of the car. While components of a vehicle may be defined, different city zone systems (wards, postal codes) are not components, but proper parts of larger areas.

6.3.2 Key classes and properties

They key properties are formalized in Table 3. The Mereology pattern identifies the following different types of parthood: proper-part-of and component-of. A more detailed analysis, presented in Reference [10] reveals clear, ontological distinctions between these relations (as well as a containment relation) that may formalized clearly with a set of first-order logic axioms. The different properties may be described as follows:

— **partOf**: specifies a part-whole relationship between objects

— **properPartOf**: specifies a part-whole relationship between objects where an object cannot be part of itself

— **componentOf**: specifies a part-whole relationship between objects where the part is defined based on actual boundaries. The parts are often also defined according to distinct functions. For example, a trunk is a componentOf a car.

— **immediateComponentOf**: specifies a componentOf relationship where the if x immediateComponentOf y, then there does not exist a z where x immediateComponentOf z immediateComponentOf y.

The aforementioned analysis (presented in Reference [10]) also identifies the expressive limitations of OWL, which prevent a complete representation of this semantics, and discussed the various possible approximations. It is important to consider what should be captured, and what distinctions should be made in the introduction of properties, in contrast with what is actually expressible in the logic. Since the required semantics cannot be completely captured in OWL, some trade-off(s) is required for any partial specification, (e.g. OWL only allows the specification of transitivity for simple object properties).

The difficulty with such an approximation is that the resulting theory defines a semantics for something else entirely. Inherently, some semantics are omitted, which can potentially not be required for one application but can potentially be important for another. For example, if transitivity is a key aspect of some required reasoning, then perhaps a parthood relation would be defined as transitive and some

© ISO/IEC 2023 – All rights reserved
AVA Documents Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

omissions would be made with respect to the formalization of other restrictions (e.g. cardinality) that should be applied to the parthood relation. Certainly, the use of approximations will be required in some cases, for example in order to support some desired reasoning problems. However, precisely which axiomatization is most suitable will vary between different usage scenarios. The Mereology pattern therefore omits a detailed, partial axiomatization in favour of an under-axiomatized specification of the key relations, in order to avoid prescribing one trade-off over another. This leaves the commitment open-ended and variable to suit individual applications' needs.

This ontology defines the general properties such that the commonality between domain-specific part-of relations may be captured, and more detailed semantics may be defined in extensions of the properties. This creates a means of indicating the intended semantics of a relation by identifying the type of parthood that it is intended to capture, while allowing for the specification of different partial approximations of the semantics (and possibly also specializations of this semantics), as required. For example, a notion of parthood arises in the description of a building and the units it is divided into. In this case, this relationship can be identified as a sort of hasComponent relation; a new property hasBuildingUnit can be identified then as a subPropertyOf hasComponent. The most suitable approximation of the component-of relation can then be defined for the hasBuildingUnit relation. The approximation chosen for one type of parthood relation does not constrain the choice of approximation for another.

Figure 4 illustrates the use of the properties defined in the Mereology pattern to serve as generic parthood properties. In this example, the hasComponent property is made more specific with the hasBuildingUnit subObjectProperty defined between Building and BuildingUnit classes.

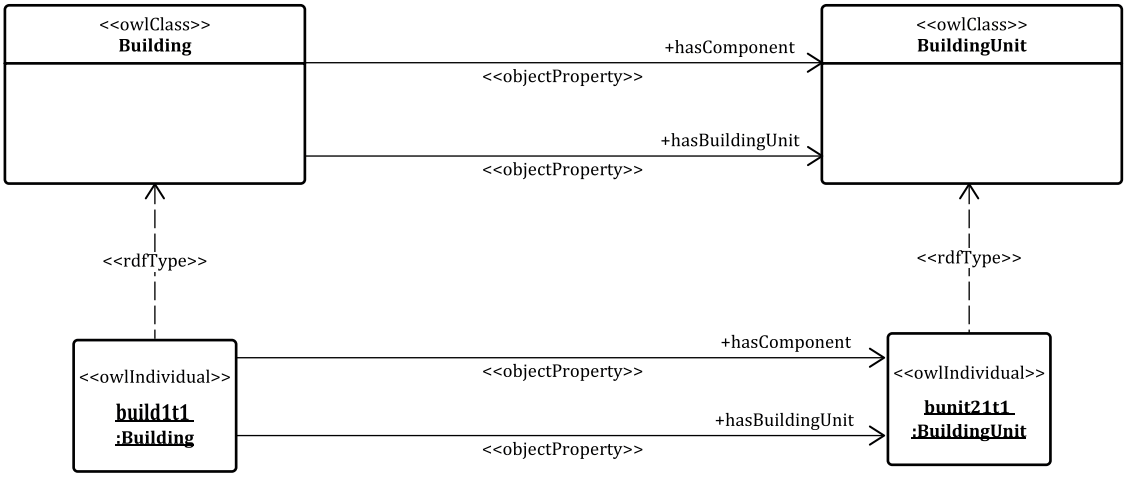

Figure 4 — Example use of the Mereology pattern

6.3.3 Formalization

Table 3 — Key properties in the Mereology pattern

Property	Characteristic	Value restriction (if applicable)
partOf		
properPartOf	inverseOf	hasProperPart
hasProperPart	inverseOf	properPartOf
componentOf	rdfs:subPropertyOf	properPartOf
	inverseOf	hasComponent
hasComponent	rdfs:subPropertyOf	hasProperPart
	inverseOf	componentOf
immediateComponentOf	rdfs:subPropertyOf	componentOf

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

6.4 City Units Pattern

6.4.1 General

Units of measure are an important concept due to the observational nature of city data collection. It is particularly important to capture the relationship between some quantity and the unit of measure it is described with. This allows for a representation in which the same individual quantity may be associated with several values, according to different units of measurement.

The representation of quantities and measures information shall conform to the ontology specified in ISO/IEC 21972. ISO/IEC 21972 is a standard that defines classes and properties required for a foundational representation of indicators, of which units are an integral part. It is included in its entirety with the prefix 'i72'.

6.4.2 Key classes and properties

The City Units pattern provides a structured vocabulary to describe, among other things, the different values (measures) that are associated to given quantities. This allows for the provision of greater detail regarding specific measurements that are defined in the ontology.

This pattern extends ISO/IEC 21972 with the classes and properties outlined in Table 4 and Table 5, respectively, to include the wider scope of quantities required for city data, such as those pertaining to physical descriptions.

Figure 5 illustrates the use of the City Units Pattern to capture numerical values and their associated units. This example shows the representation of a vehicle's speed. The speed is a single object ("veh123s1t1") that can be associated with multiple different values (e.g. 62 or 100) depending on the associated unit.

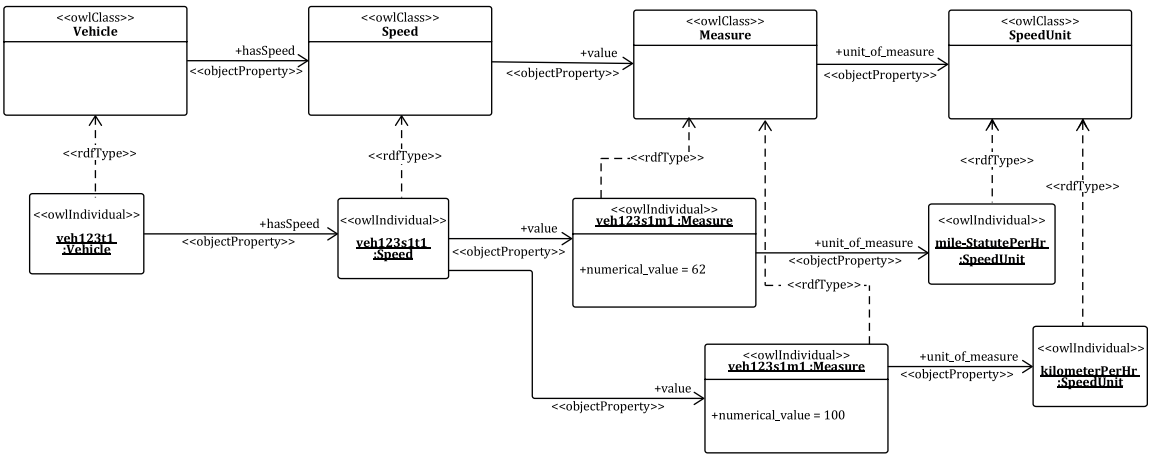

Figure 5 — Example use of the City Units pattern

Quantities, units, and/or measures that are defined using domain-specific concepts (e.g. vehicles, lanes) are defined by reusing and extending the City Units pattern in the relevant ontologies, such that the necessary concepts may be captured and the foundational ontology is not complicated with domain-specific concepts.

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

6.4.3 Formalization

Table 4 — Key classes in the City Units pattern

Class	Property	Value restriction
Area	rdfs:subClassOf	i72:Quantity
	i72:unit_of_measure	only AreaUnit
AreaUnit	rdfs:subClassOf	i72:Unit_of_measure
Length	rdfs:subClassOf	i72:Quantity
	i72:unit_of_measure	only LengthUnit
LengthUnit	rdfs:subClassOf	i72:Unit_of_measure
Speed	rdfs:subClassOf	i72:Quantity
	i72:unit_of_measure	only SpeedUnit
SpeedUnit	rdfs:subClassOf	i72:Unit_of_measure
ValueOfMoney	rdfs:subClassOf	i72:Quantity
	rdfs:subClassOf	i72:AmountOfMoney
	i72:value	only MonetaryValue
MonetaryValue	rdfs:subClassOf	i72:Measure
	hasRelativeYear	exactly 1 xsd:gYear
	i72:unit_of_measure	only i72:Amount_of_money_unit
Cardinality_unit_per_time	rdfs:subClassOf	i72:UnitDivision
	i72:hasNumerator	only i72:Cardinality_unit
	i72:hasDenominator	only TimeUnit
TimeUnit	rdfs:subClassOf	i72:Unit_of_measure
average	rdf:type	Function
Duration	rdfs:subClassOf	i72:Quantity
	i72:value	only i72:Measure and (i72:hasUnit only TimeUnit)
	rdfs:subClassOf	time:TemporalDuration
Ratio	rdfs:subClassOf	i72:Quantity
i72:RatioIndicator	rdfs:subClassOf	Ratio
Volume	rdfs:subClassOf	i72:Quantity
	i72:value	only i72:Measure and (i72:hasUnit only VolumeUnit)

Table 5 — Key properties in the City Units pattern

Property	Characteristic	Value (if applicable)
hasAggregateFunction	rdfs:domain	i72:Quantity
	rdfs:range	i72:function
aggregateOf	rdfs:domain	i72:Quantity
aggregateOver	rdfs:domain	i72:Quantity

6.5 Time Pattern

6.5.1 General

To define an ontology pattern for time, it is necessary to identify the objects of interest, i.e., which things will be described. In general, three approaches to a representation of time can be identified: point-based, interval-based, and mixed. In a point-based representation, the objects of interest are

© ISO/IEC 2023 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023

timepoints. The passing of time is described as an ordering over time points, and periods of time can be represented as a series of timepoints. In an interval-based representation the objects of interest are time intervals, whereas the mixed representation includes both timepoints and time intervals. Key to all of these representations is that there is an ordering that holds over these time objects. It is important to be able to describe whether one time object is before another; in the case of time intervals, it is also important to be able to describe other relationships, such as whether one interval is contained in or overlaps with another. The representation of time information shall conform to the ontology specified in the W3C Candidate Recommendation "Time Ontology in OWL". It specifies a mixed representation that includes time points and intervals. It is included in its entirety with the prefix 'time'.

6.6 Change pattern

6.6.1 General

Many of the concepts identified in the urban system ontologies are subject to change. For example, a Vehicle will have one location at one time, and another location at a later time; it can have only one passenger at one time, and four passengers at a later time. Similarly, many attributes of Persons, Households and even Transportation Networks are subject to change.

Change over time plays a role in many domains and thus the representation of change is by no means a new research topic. To this end, several approaches for capturing change in OWL have been proposed[11],[12]. Despite these solutions, Semantic Web practitioners lack clear and precise methods for how to apply these approaches to capture change at a domain level, whether reusing a temporal ontology or developing an ontology from scratch. The Change pattern serves as a clear guide to support a consistent approach to representing change over time.

6.6.2 Key classes and properties

The key classes and properties are formalized in Table 6 and Table 7, respectively. An approach to representing changing properties, or "fluents", that leverages the 4-dimensionalist perspective was proposed in Reference [11]. A similar approach is adopted in this document, based on the design pattern presented by Reference [13], requiring the representation of objects that are subject to change as subclasses of the Manifestation class. Manifestations may be interpreted as "snapshots" of an object at some point in time. This enables the representation of changing values of properties of an object, without losing information about its past values. The properties of the class can then be identified as properties that are (and are not) subject to change, in order to distinguish between the static and dynamic aspects of a particular entity.

The Manifestation class is defined using the following properties:

— **existsAt**: identifies the TemporalEntity that reflects the time Instant or Interval during which the snapshot of the object holds (is valid).

— **hasNextManifestation**: identifies the immediate successor Manifestation (i.e. the subsequent snapshot of the object).

— **hasPreviousManifestation**: identifies the immediate prior Manifestation (i.e. the prior snapshot of the object).

— **hasFirstManifestation**: identifies the First Manifestation that is related to a particular Manifestation (i.e., for some snapshot of an object, it identifies the first such snapshot of the object).

The class FirstManifestation denotes the first recorded manifestation of an individual. No prior manifestations (in time) exist for the individual. It is a subclassOf Manifestation and contains the following additional properties:

— **precedesManifestation**: identifies all subsequent Manifestations that follow a FirstManifestation. In other words, the FirstManifestation is related to each and every subsequent manifestation via the precedesManifestation.

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 5087-1_2023