DRAFT INTERNATIONAL STANDARD
ISO/IEC DIS 5087-1

ISO/IEC JTC **1** Secretariat: **ANSI**

Voting begins on: **2022-03-17** Voting terminates on: **2022-06-09**

Information technology — City data model —

Part 1: **Foundation level concepts**

ICS: 35.240.99; 13.020.20

Reference number
ISO/IEC DIS 5087-1:2022(E)

Contents

Page

Foreword ..... **v**

Introduction ..... **vi**

1 Scope **1**

2 Normative References **1**

3 Terms and Definitions **1**

4 Symbols and Abbreviated Terms **3**

5 Unique Identifiers **4**

6 Foundational Ontologies **5**

6.1 General 5

6.2 Generic Properties 5

6.2.1 General 5

6.2.2 Key Properties 5

6.2.3 Formalization 5

6.3 Location Pattern 5

6.3.1 General 5

6.3.2 Key Classes & Properties 6

6.3.3 Formalization 7

6.4 Time Pattern 7

6.4.1 General 7

6.4.2 Key Classes & Properties 8

6.4.3 Formalization 9

6.5 Change Pattern 9

6.5.1 General 9

6.5.2 Key Classes & Properties 9

6.5.3 Formalization 15

6.6 Agent Pattern 15

6.6.1 General 15

6.6.2 Key Classes & Properties 15

6.6.3 Formalization 15

6.7 Organization Structure Pattern 16

6.7.1 General 16

6.7.2 Key Classes & Properties 16

6.7.3 Formalization 17

6.8 Activity Pattern 18

6.8.1 General 18

6.8.2 Key Classes & Properties 18

6.8.3 Formalization 21

6.9 Recurring Event Pattern 22

6.9.1 General 22

6.9.2 Key Classes & Properties 23

6.9.3 Formalization 26

6.10 Resource Pattern 26

6.10.1 General 26

6.10.2 Key Classes & Properties 27

6.10.3 Formalization 28

6.11 Mereology Pattern 29

6.11.1 General 29

6.11.2 Key Classes & Properties 29

6.11.3 Formalization 31

6.12 City Units Pattern 31

6.12.1 General 31

6.12.2 Key Classes & Properties 31

6.12.3 Formalization 32
6.13 Agreement Pattern 33
6.13.1 General 33
6.13.2 Key Classes & Properties 33
6.13.3 Formalization 35
6.14 Provenance Pattern 36
6.14.1 General 36
6.14.2 Key Classes & Properties 36
6.14.3 Formalization 37

Annex A (informative) **Implementation Alternatives for Additional Change Semantics** **39**

Annex B (informative) **Example Use Cases** **41**

Annex C (informative) **Relationship to existing standards** **42**

Annex D (informative) **Extended Recurring Event Example** **48**

Bibliography **49**

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. 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. 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/IEC JTC 1.

ISO/IEC ##### is based on work developed in the Enterprise Integration Laboratory of the University of Toronto.

Introduction

The audience for this standard 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 is 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, the data must 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 traditionally have operated as silos. If during the process of becoming smarter, transportation, social services, utilities, etc. were to develop their own data models, then we would have smarter silos. To create truly smart cities, data must be shared across these silos which 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. It represents who occupies a home, age, occupations, where they work, abilities, etc. Though each city service may 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 standard.

Figure 1 identifies the three levels of the standard. The lowest level, defined in Part 1 of this standard provides the classes, properties, and logical computational definitions for representing the concepts that are foundational to representing any data. The middle level, defined in part 2 of this standard, provides 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. Part 3 of this standard defines the Transportation concepts. In the future, additional parts will be added to the standard covering services such as Education, Water, Sanitation, Energy, etc.

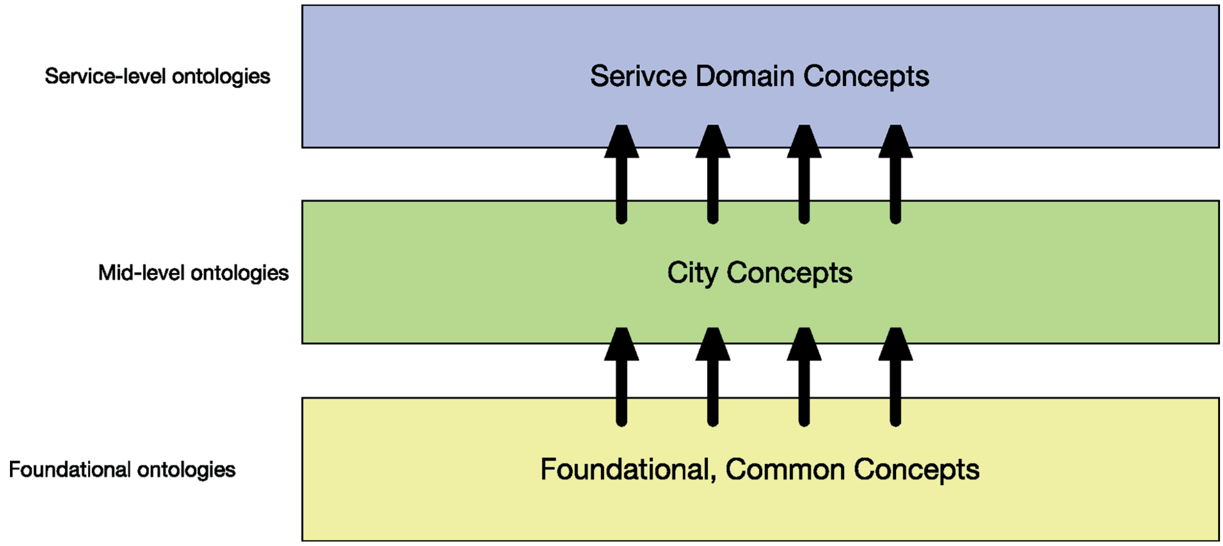

Figure 1 — Stratification of City Data Model.

Figure 2 depicts example concepts for the three levels. Level 1, as defined in part 1 of this standard includes concepts of Location, Time, Unit of Measure, Change, etc. Level 2, as defined in part 2 of this standard includes concept of Land Use, Building, Household, etc. Level 3, as defined in part 3 of this standard defines transportation concepts such as Vehicle, Trips, Transportation Network, etc.

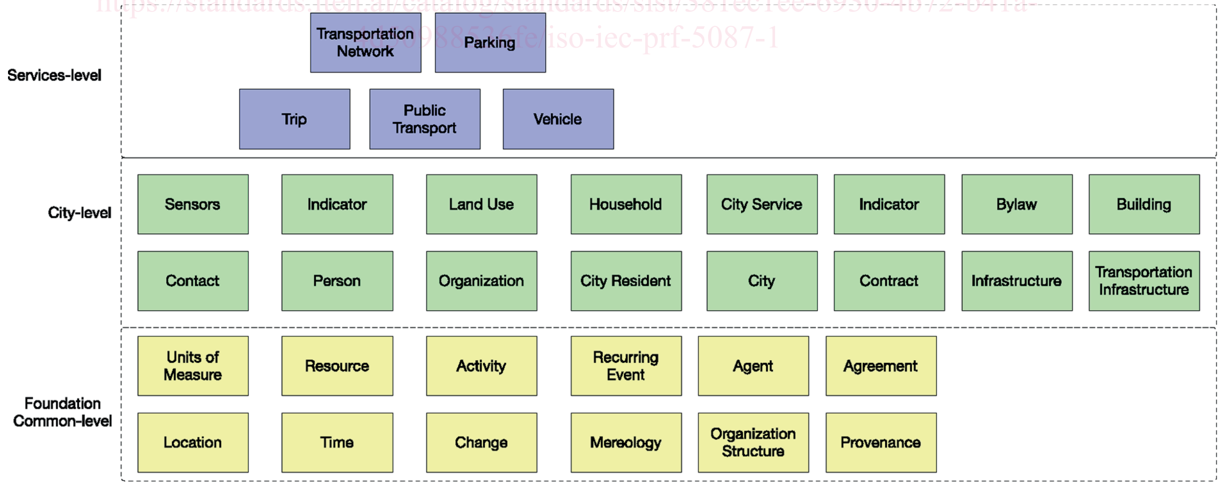

Figure 2 — Example Concepts for each Level.

It is important to distinguish between this standard series and the related, but distinct effort of ISO/IEC 30145-2. ISO/IEC 30145-2 "specifies a generic knowledge management framework for a smart city focusing on smart city knowledge creating, capturing, sharing, using and managing. It also gives the key practices which are needed to be implemented to ensure 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. 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. Part 2 of the standard addresses some of the Core Concept Model and cuts across the domain knowledge models. Subsequent parts of the standard (not yet defined) may define knowledge models for the services of citizen livelihood, urban management, and smart transportation illustrated in the diagram.

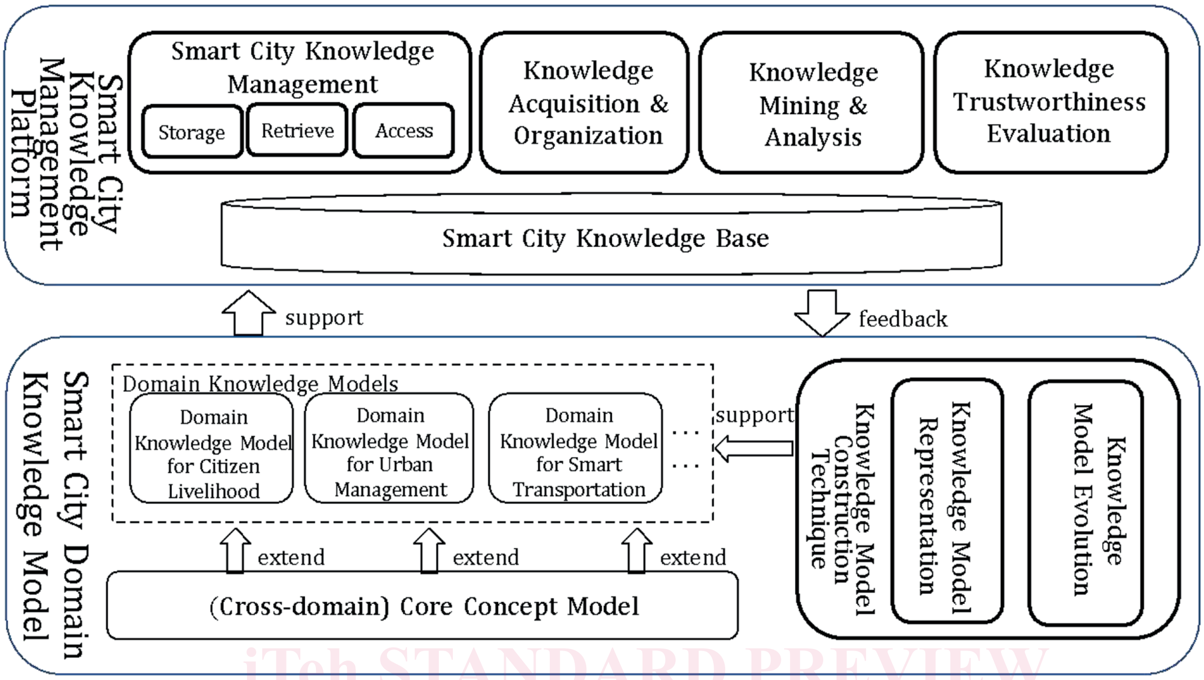

Figure 3 — The framework of smart city knowledge management

Information technology — City data model —

Part 1: **Foundation level concepts**

1 Scope

This is part 1 of the multi-part standard that specifies a common data model for cities. Part 1 is a standard for 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 documents (including any amendments) applies.

ISO/IEC 21972:2020, *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

Time 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/vocab-org/

3 Terms and Definitions

For the purposes of this document, the following apply.

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

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

— IEC Electropedia: available at http://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]

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

[SOURCE: https://www.w3.org/TR/owl2- manchester-syntax/]

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

[SOURCE: ISO/IEC 21972:2020, 3.5]

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]

[SOURCE: ISO 19150-4:2019, 3.1.19]

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/]

[SOURCE: ISO/IEC 21972:2020, 3.7]

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.]

[SOURCE: ISO 23386:2020, 3.16]

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.

[SOURCE: https://www.w3.org/standards/semanticweb/]

[SOURCE: ISO/IEC 21972:2020, 3.8]

3.10
unit_of_measure
actual units in which some quantity is measured

[SOURCE: ISO 11179-3:2003, 3.3.1334 modified.]

4 Symbols and Abbreviated Terms

- **DL**: Description Logic
- **OWL**: Ontology Web Language
- **RDF**: Resource Description Framework
- **RDFS**: Resource Description Framework Schema

The following namespace prefixes are used in this document:

- activity: http://ontology.eil.utoronto.ca/5087/1/Activity/
- agent: http://ontology.eil.utoronto.ca/5087/1/Agent/
- agreement: http://ontology.eil.utoronto.ca/5087/1/Agreement/
- change: http://ontology.eil.utoronto.ca/5087/1/Change/
- genprop: http://ontology.eil.utoronto.ca/5087/1/GenericProperties/
- geo: http://www.opengis.net/ont/geosparql#
- i72: http://ontology.eil.utoronto.ca/5087/2/iso21972/
- loc: http://ontology.eil.utoronto.ca/5087/1/SpatialLoc/
- org:http://www.w3c.org/ns/org#
- org_s: http://ontology.eil.utoronto.ca/5087/1/OrganizationStructure/
- owl: http://www.w3.org/2002/07/owl#
- partwhole: http://ontology.eil.utoronto.ca/5087/1/Mereology/
- prov: http://www.w3.org/ns/prov-o#
- 5087prov: http://ontology.eil.utoronto.ca/5087/1/Prov/
- rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
- rdfs: http://www.w3.org/2000/01/rdf-schema#
- time: http://www.w3.org/2006/time#
- time5087: http://ontology.eil.utoronto.ca/5087/1/Time
- xsd: http://www.w3.org/2001/XMLSchema#

The formalization of the classes in this document is specified using the following table format, which is a simplification of 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 is to 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, Agent is a subclass of the intersection of (Person or Organization) and org:memberOf only Organization.

Class	Property	Value Restriction
Agent	rdfs:subClassOf	Person or org:Organization
	org:memberOf	only Organization

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.

We use camelCase 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 must satisfy the class’s definition. The instance’s properties and values must 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 the table below hasPrivilege is a sub-property of the agentInvolvedIn property. Characteristics are specified using the Manchester syntax.

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.

5 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 a URI. When using 5087-1 in an application, a class is identified by the URI for the pattern it is a member of followed by the class name. In the Agent example in section 5, the Agent class’s unique identifier would be:

http://ontology.eil.utoronto.ca/5087/1/Agent/Agent

Breaking the URI down:

- http://ontology.eil.utoronto.ca refers to the server where the OWL file that defines the Agent pattern resides. This may change if ISO/IEC provides a server for storing ontologies.
- **5087** identifies the standard
- **1** identifies the part of the standard
- The first **Agent** identifies the Agent Pattern
- The second **Agent** identifies the Agent class within the Agent Pattern

The URI can be shortened using the prefix's defined in section 5:

agent:Agent

where agent: is the prefix for the Agent Pattern. Properties are identified in the same manner. The URI's of individuals created by an application of ISO/IEC 5087-1 would have URI's unique to the application.

6 Foundational Ontologies

6.1 General

Beyond the domain-specific subjects that are clearly identified in our 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 sections.

6.2 Generic Properties

6.2.1 General

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

6.2.2 Key Properties

The following generic properties have been identified:

- **hasName**: identifies the name of some object
- **hasDescription**: specifies a description of some object
- **hasIdentifier**: specifies an identifier for some object

6.2.3 Formalization

The full implementation of the encoding in OWL is available at http://ontology.eil.utoronto.ca/5087/1/GenericProperties.owl

6.3 Location Pattern

6.3.1 General

The ontology for representing location information shall conform to the vocabulary specified in OGC 11-052r4 [4]. To capture generic spatial features requires concepts of location, but also concepts of geometry in order to describe shapes that are more complex than a single point in space. In addition, there is a need to be able to describe the spatial relationship between various features (e.g. containment, overlap). The GeoSPARQL Ontology is used in the Location Pattern to achieve this. It is included in its entirety with the prefix 'geo'.

6.3.2 Key Classes & Properties

In the following, we replicate and specialize a subset of the GeoSPARQL ontology:

- **Feature**: represents some area in space. Features are distinguished by their geometric shape as well as their geographic location. Features may also be related to other Features via mereotopological relations such as containment and contact. Core properties are:
  - o **hasGeometry**: specifies the shape that defines the location of the Feature in order to capture quantitative spatial information.
- **Geometry**: represents a shape. Various types (subclasses) of Geometry are defined in the GeoSPARQL ontology including the classes: Point, Polygon, and Curve. Core properties are:
  - o **asWKT**: specifies the well-known text (WKT) encoding of a given geometry. The default reference system for the coordinate values is assumed to be WGS84. GeoSPARQL supports the identification of alternate reference systems, captured as IRIs and concatenated with the coordinates.

In addition, the pattern specifies the following generic properties to support the reference of locations by other classes:

- **hasLocation**: captures the relationship between objects and the spatial locations they occupy
- **associatedLocation**: introduced to capture the association of some object with a particular location. For example, a train station may occupy a fairly large spatial location but be associated with a particular point.

The Mereology Pattern (described in 6.11) is also imported in order to specify the relationship between the GeoSPARQL properties defined for Spatial Objects and the generic part-whole relations that may also apply to other types of entities.

The diagram in Figure 4 illustrates the use of the Location Pattern to represent both the Feature (as defined in GeoSPARQL) associated with a vehicle's location, as well as the specific geometry that can be used to describe the Feature. Note that the Geometry object may be a point or other more complex shape and may be associated with more than one data property based on different encoding systems (such as well-known text or WKT, shown here).

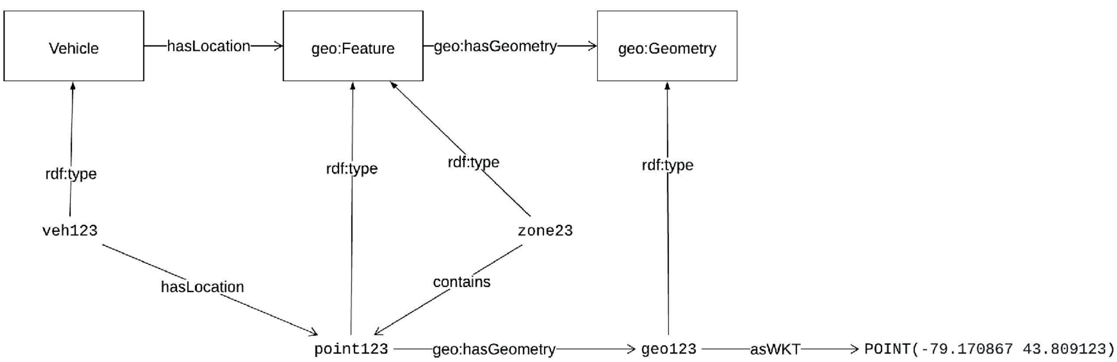

Figure 4 — Example use of the Location Pattern.

6.3.3 Formalization

Table 1 — Key classes in the Location Pattern.

Class	Property	Value Restriction
Feature	rdfs:subClassOf	geo:Feature
	hasGeometry	exactly 1 geo:Geometry
Geometry	rdfs:subClassOf	geo:Geometry
	asWKT	exactly 1 geo:wktLiteral

The following table extends a subset of properties in GeoSPARQL so that they are integrated with the Mereology pattern.

Table 2 — Additional properties introduced in the Location Pattern.

Property	Characteristic	Value (if applicable)
hasLocation	Range	Feature
associatedLocation	Range	Feature
partwhole:containedIn	inverseProperty	geo:sfContains
geo:sfContains	rdfs:subPropertyOf	partwhole:contains
geo:'tangential proper part'	rdfs:subPropertyOf	partwhole:properPartOf
geo:'non-tangential proper part'	rdfs:subPropertyOf	partwhole:properPartOf
hasGeometry	rdfs:subPropertyOf	geo:hasGeometry
	Functional	
asWKT	rdfs:subPropertyOf	geo:asWKT
	Functional	

The full implementation of the pattern encoding in OWL is available at http://ontology.eil.utoronto.ca/5087/1/SpatialLoc.owl. A copy of GeoSPARQL is imported (via http://ontology.eil.utoronto.ca/5087/1/geosparql.owl) in this OWL encoding.

6.4 Time Pattern

6.4.1 General

In order to define an ontology pattern for time, the objects of interest must be identified. What are the *things* that will be described? In general, three approaches to a representation of time may be identified: point-based, interval-based, and mixed. In a point-based representation, the objects of interest are timepoints. The passing of time is described as an ordering over time points, and periods of time may 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 a 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 Recommendation "Time Ontology in OWL" [5] originally presented in work by [6]. It is included in its entirety with the prefix 'time'.

