iCity Ontology Version 1.2 Report

Design Documentation

Megan Katsumi
Post-Doctoral Fellow
Enterprise Integration Lab
Mechanical and Industrial Engineering
University of Toronto

Mark Fox
Distinguished Professor of Urban Systems Engineering
Professor of Industrial Engineering and Computer Science
Director of the Centre for Social Services Engineering
University of Toronto

Last Updated: July 16, 2018

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING
Transportation Research Institute

UTTRI

1 Purpose

The purpose of this document is to present the first release of the iCity ontology. This document provides a concrete outline of the concepts defined in the ontology; its purpose is to provide a point of reference to facilitate discussion and feedback on of the ontology, and to facilitate its eventual implementation across the iCity projects. This initial release focuses on the identification of the classes and properties that will form the ontology. These classes and properties provide a clear indication of the breadth of the scope of the iCity ontology, and are thus a critical first step in the development of the final artifact. Based on this release, we may begin the process of implementation by capturing relevant information in the language defined by the ontology, and thus providing a shared, commonly understood model of the knowledge being used and generated by the various iCity projects.

Note that for each ontology we also maintain html documentation, automatically generated using Widoco[1].

2 Scope

The scope of this document is limited to the identification of the vocabulary within the ontology. More specifically, it is restricted to the identification of vocabulary in the Urban System Ontology; at this stage, it does not capture the application-specific concepts of the individual iCity projects. We provide an initial specification of the classes and relationships (properties) to support formalization in OWL 2 (Grau, et al., 2008). Future versions will expand on the depth of these definitions, providing more detailed semantics in a complementary logical language. This document aims only at conveying the vocabulary currently defined in the iCity ontology, implementation of the ontology shall be addressed in a separate iCity report.

3 Outline

The report will begin with an introduction to the role of the ontology within the iCity project. The core concepts pertaining to the characteristics and behaviour of the urban system will then be presented in Section 5. Section 0 identifies directions for future iterations of the ontology; in particular, Section 8.2 outlines top-level concepts required for data collection, simulation, and analysis applications. At the current stage of development, we have not identified any requirements specific to the Visualization application (Theme 3 in the iCity project). It is our understanding that the Theme 3 work will interpret the iCity ontology in order to generate the required visualizations. Should this change in the future, it is likely that an extension to capture the visualization applications will also be required.

4 Role of the Ontology

Given that all of the projects within iCity are situated in the urban domain, and therefore it is not surprising to find many common concepts between them. It stands to reason that some integration between the different applications should be possible. For example, if data is collected about the population, it should be usable by ILUTE and other simulations, but also by the projects developing analysis tools, such as the smart parking application. Unfortunately, there is also ambiguity in how different concepts are used, and in some cases the same concept may be

[1] https://github.com/dgarijo/Widoco

defined differently in different applications. This provides a challenge not only for integration of the iCity applications, but for shareability of results: if the knowledge generated by iCity is not defined sufficiently, it will be difficult for any other researchers to understand and leverage it.
The iCity ontology provides a common set of terms with which data can be stored and accessed. The ontology will resolve any ambiguities and disagreements between terms by defining a common set of concepts that completely captures the domain, with agreed-upon definitions. In the case that two applications attribute a different meaning to the same term, the result will be two distinct terms with distinct, precisely defined meanings. In this way we can recognize these differences and clearly identify the relationships between different concepts. The ontology will be used to organize and describe data within the iCity project. It may also be used as means of publishing or sharing data with the research community.
The resulting artifact, often referred to as the *knowledge base* will take the form of a triple-store(s)[2], created by mapping data from the iCity applications to the agreed-upon terminology defined in the iCity ontology. The architecture for the ontology's implementation in the context of the iCity project, is illustrated in Figure 1.
The precise and formal nature of the ontology will support the use of services such as inference and data validation. Based on the definitions, we may be able to infer new information that was not originally part of the knowledge base. Data validation is supported as a result of the consistency-checking mechanism. We also hope that identification of relationships may serve to uncover synergies between the projects, by illustrating how data from one project may serve to inform the work of another.

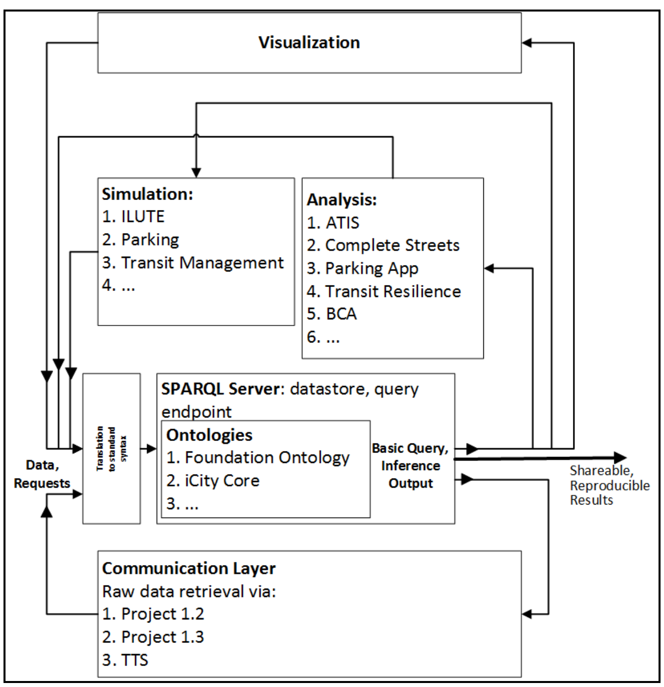


Figure 1: iCity Knowledge Base Architecture

---

[2] Note that the data may not in fact be stored in the triple-store, but maintained in the application's own database, with mappings from the ontology to the database performed on-the-fly, as required.

The sections that follow introduce the core ontology required to capture the iCity projects, in particular, to define the urban system. Beyond this, the iCity ontology may be implemented to support specific applications. Examples of this are discussed in further detail in Section 7.

5 Summary of Changes from Previous Version

The "iCity-" prefix was removed from all ontology filenames and IRIs in order to improve readability and convey generality. All other changes are summarized by ontology below.

Activity ontology:

- revised representation such that an activity is a class of occurrences (activities and occurrences are not separate entities); removal of ActivityOccurrence, definition of State instead of StateType.

Transportation Network ontology:

- added Link and LinkPD classes to serve as "containers" for multiple arcs (e.g. vehicle lanes, bicycle lanes, walkways); introduced some additional properties and changed the mode of an Arc from an invariant to variant property.
- Associated Nodes with location information.
- Links and Arcs represent access on some part of the physical infrastructure, which has an associated location. Although Nodes do not represent access in the same way, they are still associated with some physical location. A property was added to specify the *associated* location of a Node. It should be possible to infer which Transportation Complexes (e.g. road segments) meet or contain the node based on the links it is connected to. Future extensions may consider capturing the relationship between the nodes location and the location of the Transportation Complexes accessed by its related links.

Added: iContact.owl Ontology

The notion of addresses is required to represent information about buildings, parking lots, the start and end of trips, and so on. Contact information for individuals and organizations may also be required and captured for some applications.

Added: Calendar/Hours of Operation Ontology:

Beyond the representation of individual timepoints and time intervals, there is often a requirement to reference concepts from a calendar. In particular, the specification of hours of operation (e.g. fora business or transportation network policy) relies on the representation of these concepts, such as the days of the week or times of day. The Calendar Ontology is introduced to define these concepts.

Imported SSN/SOSA Ontology:

Sensor observations are an integral part of ITS operations and research. To capture these sensors and the data they generate, we import the SSN/SOSA ontology [ref].

Units of Measure Ontology:

- Extended and merged with monetary value ontology.

- Updated with new release OM 2.0
- Extend with specializations of quantities, measures, etc, as required by use cases.

Spatial Location ontology:

- Replaced original representation with geoSPARQL terminology, primarily due to it being better supported (geoSPARQL works for simple linked data but also offers the potential for specialized query abilities) and more current.
  Features have geometry (geo:hasGeometry); geometries can be defined as simple features (points, polygons,…); these geometries can be *serialized* as WKT or GML, special purpose datatypes that allow for, e.g. a series of coordinates. Both serializations support the specification of a reference system, therefore (for now) we do not need to extend OM with NAD83 and WGS84.
  The representation of the reference system is not ideal as in the current implementation of geoSPARQL it is appended within the same IRI as the coordinate data, (it is also not clear what code is to be used for NAD83). Future revisions should investigate a possible extension to this representation that will capture the reference system in a more convenient way (while still leveraging the capabilities of geoSPARQL).
  Ideally, in future work we would like to look consider the spatial-location theory in more detail as it geoSPARQL provides a vocabulary and a tool but lacks a complete declarative semantics
- This change impacts many of the other ontologies within iCity (those with spatial information). Each was modified to address this.
- Extended SpatialLoc to include a hasLocation property. This allows us to separate objects from their spatial embodiment as required. Whether an object is related to geosparql:Feature or is a subclass of geosparql:Feature is a foundational ontological decision. In either case, we can describe location of these objects more precisely via the hasGeometry property.

Foundation Ontology: removed

- In the initial design Foundation is imported by all of the domain ontologies, this requires a revision to all of the Urban System ontologies. We observe that this design is not ideal as there may be cases where foundational concepts (e.g. activities or resources) are not used in a particular domain ontology. In such cases, updates to Foundation.owl will result in unnecessary updates to unaffected domain ontologies. This was originally done for convenience, however we observe that it may be more clear and effective to only consider the foundational grouping conceptually, and individually import whichever ontologies are required.
- In version 1.2 of the ontology, we replace all imports of the Foundation ontology only with the ontologies that are used. In some cases, this may be equivalent to importing the Foundation ontology, however in other cases this will be a sub-theory of the Foundation ontology.

Land Use Ontology:

- Defined new subclasses of Parcel based on sample data: TrafficZone, PlanningDistrict, Municipality. It's unclear what the logical distinction between these classes will be, but they are distinct types of parcels used by the domain experts.
- Introduced additional land use classifications, aligned with lbcs classifications where possible, based on CLUMP and AAFC systems.
- Added TrafficZone subclass of Parcel
- Added population properties for Parcels

PublicTransit Ontology:

- Added specializations of the Activity class: TransitTrip and TransitIncident
- Included some additional properties and added to the definitions of some existing classes

- Imported Transportation System ontology
- Imported Activity ontology

Time Ontology:

- Revised to reuse the new version of OWL-Time (updated via W3C in fall of 2017)

Parking Ontology:

- Based on requirements identified by CUHK use cases, the parking ontology has been extended to capture additional concepts to provide a more detailed picture of existing car parks.

6 Urban System Characteristics and Behaviour

In the urban system, we recognize the following key concepts that must be defined:

- Person
- Organization
- Household
- Building
- Parking
- Vehicle
- Transportation Networks
- Transit
- Land Use
- Travel

The semantics of each of these concepts will be defined by a generic ontology. These generic ontologies will then be used in the iCity ontology to define the urban system and its behaviour; its population, land use, transportation infrastructure, and the travel that occurs within it. This representation may then be extended to capture the individual iCity applications so that they may be integrated with one another and sufficiently well-defined so as to be shareable and reproducible with the research community. A Foundational Ontology will be also required in order to define the core concepts that apply across the transportation domain. This is introduced first, followed by the presentation of each generic ontology in more detail. Where warranted, we provide a brief description of the domain and role of the ontology prior to describing its classes and their properties.

To do: include discussion on practices adopted for the reuse of existing ontologies

- Save and re-publish with new IRI to ensure availability and consistency (there may be issues if some ontologies update and do not use Version IRI metdata)
- Entities created for organization: XThing, XObjectProperty, XDataProperty…

6.1 Foundational Ontologies

~~*https://w3id.org/icity/Foundation.owl*~~

In addition to the concepts that are specific to an urban system, there exist foundational concepts that are required to fully define the domain. In particular, the foundational ontology captures the concepts of time, space, change, activities, and resources; each concept is defined its own sub-ontology.

6.1.1 Spatial Location Ontology

https://w3id.org/icity/SpatialLoc.owl

Namespace: geo

To capture generic spatial features, we require 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, we need to be able to describe the spatial relationship between various features (e.g.

containment, overlaps). To achieve this, we leverage geoSPARQL[3]. geoSPARQL specifies the required vocabulary and provides specialized functions for querying spatial data that are supported by some implementations. The Spatial Location Ontology also includes the property hasLocation to capture the relationship between non-spatial objects and their associated spatial locations.

- geo:SpatialObject
- geo:Feature
- geo:Geometry
- sf:Point
- sf:Poly
- …
- geo:wktLiteral
- geo:gmlLiteral
-

Object	Property	Value
geo:Feature	subClassOf	geo:SpatialObject
geo:Geometry	subClassOf	geo:SpatialObject

The ontology specifies properties for the topological relationship between spatial objects, among these are:

Property	Characteristic	Value (if applicable)
geo:sfEquals	Domain and Range	geo:SpatialObject
geo:sfDisjoint	Domain and Range	geo:SpatialObject
geo:sfIntersects	Domain and Range	geo:SpatialObject
geo:sfTouches	Domain and Range	geo:SpatialObject
geo:sfWithin	Domain and Range	geo:SpatialObject
geo:sfContains	Domain and Range	geo:SpatialObject
geo:sfOverlaps	Domain and Range	geo:SpatialObject
geo:sfCrosses	Domain and Range	geo:SpatialObject
icity:hasLocation	Range	geo:Feature

RCC8 and Egenhofer relations are also defined.

[3] http://www.opengeospatial.org/standards/geosparql

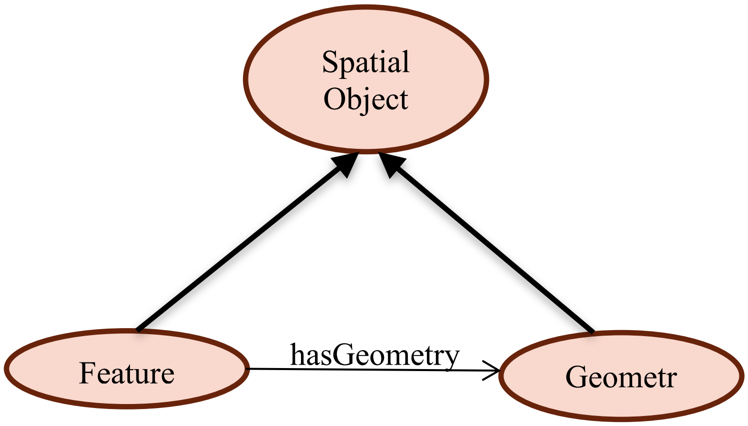


Figure 2: The key classes in geoSPARQL, from (Kolas & Battle, 2012).

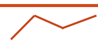
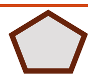
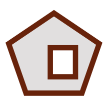
TYPE	SHAPE	Geometry Class	SYNTAX
POINT		`sf:Point`	`POINT(longitude latitude)`
LINESTRING		`sf:LineString`	`LINESTRING(long1 lat1, long2 lat2, …)`
POLYGON		`sf:Polygon`	`POLYGON((long1 lat1, long2 lat2, … , long1 lat1))`
POLYGON (WITH HOLE)		`sf:Polygon`	`POLYGON((long1 lat1, long2 lat2, … , long1 lat1), (longA latA, longB latB, …, longA latA))`

Figure 3: An example of some basic shape encodings in WKT, from (Kolas & Battle, 2012).

To capture geometries as RDF literals, they are given the `geo:WktLiteral` datatype, for example: `"POINT(-77.03524 38.889468)"^^geo-sf:wktLiteral`.

```
ex:Monument1 a ex:Monument;
   rdfs:label "Washington Monument";
   geo:hasGeometry ex:Point1 .
ex:Point1 a geo:Point;
  geo:asWKT "POINT(-77.03524 38.889468)"^^geo-sf:WktLiteral.
Ex:Park1 a ex:Park;
     rdfs:label "Example Park";
     geo:hasGeometry ex:Polygon1 .
ex:Polygon1 a geo:Polygon;
  geo:asWKT "POLYGON((-77.05 38.87, -77.02 38.87, -77.02
  38.9, -77.05 38.9, 77.05 38.87))"^^geo-sf:WktLiteral.
```

Figure 4: An example encoding of geospatial information for the Washington Monument, from (Kolas & Battle, 2012).

To do: include diagram of the above example.

Regarding the specification of the reference system used, the default is assumed to be WGS84. In theory, GeoSPARQL supports the identification of alternate reference systems (however, these are captured as IRIs and concatenated with the coordinates). Currently, a translation between these systems is not implemented in geoSPARQL, however it appears as though other systems (e.g. Virtuoso) have implemented their own translations.

Reused Ontologies:

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

6.1.2 Time Ontology

https://w3id.org/icity/Time.owl

Namespace: time

- Temporal Entity: A Temporal Entity may refer to an instant or an interval in time.
  A Temporal Entity may be described as being **before** or **after** some other Temporal Entity(s).
  A Temporal Entity has a **beginning** and **ending** time Instant.
  A Temporal Entity has a **duration**.
- Instant
  An Instant may be **inside** some Interval.
- Interval
  An Interval may be described as **before**, **meets, overlaps, starts, during, finishes, equals** some other Interval(s).

Object	Property	Value
time:TemporalEntity	EquivalentClass	time:Instant and time:Interval
	time:before	only time:TemporalEntity
	time:after	only time:TemporalEntity
	time:hasBeginning	only time:Instant
	time:hasEnding	only time:Instant
	time:hasDuration	only time:Duration
time:Instant	subClassOf	time:TemporalEntity
	time:inside	only time:Interval
	time:inTimePosition	max 1 time:TimePosition
	time:inXSDDateTime	max 1 xsd:DateTime
time:Interval	subClassOf	time:TemporalEntity
	time:before	only time:Interval
	time:meets	only time:Interval
	time:overlaps	only time:Interval
	time:starts	only time:Interval
	time:finishes	only time:Interval
	time:during	only time:Interval
	time:equals	only time:Interval
time:DateTimeDescription	time:day	max 1 rdfs:Literal
	time:dayOfWeek	max 1 owl:Thing
	time:dayOfYear	max 1 rdfs:Literal
	time:hour	max 1 rdfs:Literal
	time:minute	max 1 rdfs:Literal
	time:month	max 1 rdfs:Literal
	time:second	max 1 rdfs:Literal
TimePeriod	subClassOf	time:DateTimeDescription
CalendarPeriod	subClassOf	time:DateTimeDescription

Update: Note, version 1.1 updated to reflect revision to owl-time released as a W3C recommendation in October 2017.

Reused Ontologies:

- time: OWL-Time Ontology[4] originally presented by (Hobbs & Pan, 2004)

6.1.3 Change Ontology

https://w3id.org/icity/Change.owl

Namespace: iCity-Change

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 may 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. An approach to representing changing properties, or *fluents*, that leverages the 4-dimensionalist perspective was proposed by (Welty, Fikes, & Makarios, 2006). We adopt a similar approach, requiring the division of classes that are subject to change into two parts: invariant and variant parts of the concept; we refer to these as TimeVaryingConcept and Manifestation classes, respectively. By distinguishing between these class types and recognizing the properties that are (and aren't) subject to change, the ontology supports the capture of both the static and dynamic aspects of a particular entity.

- TimeVaryingConcept: A class that is subject to change is defined as a type of TimeVaryingConcept (e.g. Vehicle may be a subclass of TimeVaryingConcept). The TimeVaryingConcept itself is invariant and defined by properties that do not change over time. As per (Krieger, 2008), we view TimeVaryingConcepts as perdurants (things that occur over time, i.e. processes).
  A TimeVaryingConcept has **Manifestations** that demonstrate their changing (variant) properties over time. Different types (**subclasses**) of TimeVaryingConcept may be defined based on the Manifestations that are part of them. For example, VehiclePD[5]s have manifestations that are Vehicles.
  A TimeVaryingConcept **exists at** some Interval.
  The class of TimeVaryingConcepts is equivalent to the class of things that have some Manifestations - and *only* Manifestations - in the hasManifestation relation.
- Manifestation: A Manifestation of some TimeVaryingConcept at a particular point/interval in time.
  A Manifestation **exists at** some Instant (or possibly Interval).
  The class of Manifestations is equivalent to the class of things that are manifestations of some TimeVaryingConcept - and *only* time varying concepts - in the manifestationOf relation.

Object	Property	Value
TimeVaryingConcept	disjointWith	time: TemporalEntity and Manifestation
	existsAt	exactly 1 time:Interval
	hasManifestation	only Manifestation
	equivalentClass	hasManifestation some Manifestation and hasManifestation only Manifestation
Manifestation	disjointWith	TimeVaryingConcept and time: TemporalEntity
	equivalentClass	manifestationOf some TimeVaryingConcept and manifestationOf only TimeVaryingConcept

[4] https://www.w3.org/TR/owl-time/

[5] Note: in order to avoid confusion that may result from the use of the "-Process" suffix (e.g. VehicleProcess,OrganizationProcess), we opt instead to use the suffix "PD", i.e. short for "Perdurant".

	manifestationOf	only TimeVaryingConcept
	existsAt	exactly 1 time:TemporalEntity

Property	Characteristic	Value (if applicable)
hasManifestation	inverseOf	manifestationOf
	Inverse Functional	-
manifestationOf	Functional	-
existsAt	Ranges	time:TemporalEntity

Reused Ontologies:

- iCity-Time

6.1.4 Activity Ontology

https://w3id.org/icity/Activity.owl

Namespace: activity

- Activity: An Activity describes something that occurs in the domain.
  An Activity may be further defined by **(decomposed into) Subactivities**.
  An Activity may have **precondition and/or effect** States.
  An Activity may be **enabled by or cause** some States. An enabling of causing state is a generalization of a precondition/effect; an Activity is enabled by or causes some State if it has a subactivity with a precondition or effect (respectively) of that State.
  In other words, the state may not be required directly before, or cause directly after the activity, but by some more specialized sub-activity.
  An Activity **occurs at** some point in time and space.
  An Activity takes place **during some interval**, and so **has** some **duration**.
  An Activity may have some Manifestations that participate in it.
- ActivityOccurrence: An ActivityOccurrence is an **occurrence of** some Activity that **occurs at** some point
- State: a state refers to a class of manifestations. It may be an immediate precondition or effect of some Activity, or more generally it may enable or be caused by some Activity (in which case, it might be a direct precondition or effect of some subactivity of the activity).
  A state may be complex and refer to some combination of classes of manifestations.
  A note on complex states:
  Say that a shopping activity, Activity-Shop, requires both the state of a vehicle having at least 30L of gas in the tank (let's call this state VehicleW30LGas), but also some state type wherein the mall is open, (we'll call this state OpenMall). Each state type would first be defined separately. This precondition could bet stated as:

  precondition(VehicleW30LGas,Activity-Shop) AND precondition(OpenMall,Activity-Shop)

  were the preconditions required disjunctively, we could state:

  precondition(VehicleW30LGas,Activity-Shop) OR precondition(OpenMall,Activity-Shop)

  However, in large and complex domains, there will be cases in which the above approach is undesirable. In particular, due to the complexity of the description that results as the state type becomes more detailed. In many cases it will be more natural and convenient to be able to refer to a single, aggregate state. We therefore extend the representation of States to capture aggregation, adopting the following approach used in the description of state trees in TOVE by (Fox, Chionglo, & Fadel, 1993).
  A State may be either non-terminal or terminal. A terminal state has no child states, and therefore refers directly to a class of manifestations, whereas a non-terminal state has child states, which may define some classes of manifestations, or further define some other complex state types.

  NonTerminalState(x) v TerminalState(x) = State(x)

A state type cannot be both non-terminal and terminal.

TerminalState disjointWith NonTerminalState

- A terminal state has no substates (cannot be decomposed). It corresponds to a particular class of manifestations. A terminal state is achieved at some time if and only if there exists a manifestation within its defined classification, that exists at that time.
- A non-terminal state may be conjunctive or disjunctive. Naturally, a conjunctive state is defined by the conjunction of its child state, whereas a disjunctive state is defined by the disjunction of its child states.

ConjunctiveStateType(x) v DisjunctiveStateType(x) = NonTerminalStateType(x)

A state cannot be both conjunctive and disjunctive.

ConjunctiveStateType disjointWith DisjunctiveStateType

Conjunctive and disjunctive states, which *do* have substates, are achieved at some time if their decomposition of state is achieved.

Note that in this representation, *decomp_of* is not a transitive relation, it only refers to the direct children of a non-terminal state. A more general relation that *is* transitive is the *substate* relation.

decomp_of(x,y) -> substate(x,y)

Object	Property	Value
Activity	hasSubactivity	only Activity
	hasPrecondition	only State
	enabledBy	only State
	hasEffect	only State
	causes	only State
	occursAt	some time:Interval
	beginOf	some time:Instant
	endOf	some time:Instant
	spatial_loc:hasLocation	only spatial_loc:SpatialFeature
	hasParticipant	only change:Manifestation
State	preconditionOf	only Activity
	enables	only Activity
	effectOf	only Activity
	causedBy	only Activity
	achievedAt	only time:TemporalEntity
TerminalState	subClassOf	State
	disjointWith	NonTerminalState
	subClassOf	change:Manifestation and (preconditionOf some Activity or effectOf some Activity)
	hasDecomp	exactly 0 StateType
NonTerminalState	subClassOf	State
	disjointWtih	TerminalState
	hasDecomp	only State and min 2 State
	hasSubstate	only State
ConjunctiveState	subClassOf	NonTerminalState
	disjointWith	DisjunctiveState
DisjunctiveState	subClassOf	NonTerminalState

	disjointWith	ConjunctiveState

Property	Characteristic	Value (if applicable)
hasSubactivity	Transitive	-
hasPrecondition	Domains	Activity
	Ranges	State
~~enabledBy~~	~~subPropertyOf~~	~~hasPrecondition~~
hasPrecondition	subPropertyOf	enabledBy
hasEffect	Domains	Activity
	Ranges	State
~~causes~~	~~subPropertyOf~~	~~effectOf~~
hasEffect	subPropertyOf	causedBy
occursAt	Domains	Activity
	Ranges	time:TemporalEntity
hasParticipant	Domains	Activity
	Ranges	change:Manifestation
participatesIn	inverseOf	hasParticipant
preconditionOf	inverseOf	hasPrecondition
enables	inverseOf	enabledBy
effectOf	inverseOf	hasEffect
causedBy	inverseOf	causes
achievedAt	Domains	State
	Ranges	time:TemporalEntity
	superPropertyOf	inverse(preconditionOf) o beginOf
	superPropertyOf	inverse(effectOf) o endOf
	superPropertyOf	inverse (hasDecomp) o conjunctiveAchievedAt o during
hasDecomp	Domains	State
	Ranges	State
	subPropertyOf	hasSubstate
hasSubstate	Domains	State
	Ranges	State
	~~subPropertyOf~~	~~hasDecomp~~
	Transititve	
beginOf	Domains	Activity
	Ranges	time:Instant
	superPropertyOf	occursAt o time:hasBeginning
endOf	Domains	Activity
	Ranges	time:Instant
	superPropertyOf	occursAt o time:hasEnd
terminalAchievedAt	subPropertyOf	achievedAt
	subPropertyOf[6]	existsAt
	Domains	TerminalState

[6] equivalent property?

disjunctiveAchievedAt	subPropertyOf	achievedAt
	superPropertyOf	hasDecomp o achievedAt
	Domains	DisjunctiveState
conjunctiveAchievedAt	subPropertyOf	achievedAt
	Domains	ConjunctiveState

Reused Ontologies:

- iCity-Change
- iCity-SpatialLocation

6.1.5 Resource Ontology

https://w3id.org/icity/Resource.owl

Namespace: resource

This ontology provides a generic representation of resources that contain core properties generic across all transportation uses. We take the view presented in the TOVE model (Fadel, Fox, & Gruninger, 1994) that "*...being a resource is not an innate property of an object but a property that is derived from the role the object plays with respect to an activity*". The definition of a resource is dependent on its participation in an activity occurrence, so the Resource ontology is in fact an extension of the Activity ontology. In this sense, Resources are a class of manifestations, so that rather than have a specialized Resource-perdurant (PD) class, a Resource is a manifestation of some *other* perdurant class in the ontology. For example, an instance of a Vehicle that is a manifestation of some VehiclePD may also be an instance of a resource, whereas some other instance of a Vehicle that is some later manifestation of the same VehiclePD may not be a Resource, or it may be a *different* Resource.

- A Resource is a generic representation of some Thing that can be "used" in an Activity.
  A Resource may have some Location, amount or availability, according to the definition of the Manifestation or TimeVaryingEntity.
  A Resource must be **classified as** some Resource Type.
  A Resource may **participate in** some Activity Occurrence.
  A *specific* Resource may be **used in** or **consumed in** some activity *occurrence*. As with the precondition and effect properties defined in the Activity Ontology, these relationships are specific to a particular activity occurrence; more general properties may be defined (analogous to enables and causes) should this be required.
  A Resource may have some **associated location**. This property captures an approximate location that may or may not differ from its actual location. For example, a landmark may be associated with a particular point (single set of coordinates), whereas the landmark itself in fact has a location that extends beyond this single point. It may even be the case that the associated location is distinct from (ie. not contained in) the actual location.
  A Resource may have some **owner**.
- A Resource may *either* be a Divisible Resource or a Non-Divisible Resource. On the surface this may seem counterintuitive -- consider a vehicle being used as a non-divisible resource for transportation, and then later as a divisible resource for scrap metal. However, while these examples might refer to the same car over the span of its lifetime, each one in fact refers to a different manifestation of the car, and hence a different resource. The resources differ in their divisibility because each one is defined with respect to a different activity occurrence (e.g. travel, versus metal recycling).
  A divisible resource may be used by or consumed by more than one activity occurrence, whereas a non-divisible resource may only be used by one activity occurrence (i.e. the object may only be used by one activity at a time).
- A Resource Type describes a class of Resources, (intuitively similar to the State Type class).
  A Resource Type may be **usedBy** or **consumedBy** some Activity; the specification of the Resource Type defines the quantity of a particular resource that will be used or consumed by a particular activity

occurrence.
If some resource type is used by an activity, then for all occurrences of the activity, there is a resource of that type that is (partially) not available. Further, the resource and the entity it is a manifestation of (partially) cease to exist by the end of the occurrence.
usedBy and consumedBy are subproperties of preconditionOf.

Object	Property	Value
Resource	subClassOf	change:Manifestation
	change:existsAt	exactly 1 TemporalEntity
	spatial_loc:hasLocation	only spatial_loc:SpatialFeature
	hasCapacity	only om:CapacitySize
	capacityInUse	only om:CapacitySize
	hasResourceType	only ResourceType
	activity:participatesIn	min 1 activity:ActivityOccurrence
	usedInOccurrence	only activity:ActivityOccurrence
	consumedInOccurrence	only activity:ActivityOccurrence
DivisibleResource	subClassOf	Resource
	disjointWith	NonDivisibleResource
	hasAvailableCapacity	only om:CapacitySize
NonDivisibleResource	subClassOf	Resource
	disjointWith	DivisibleResource
	usedInOccurrence	exactly 1 activity:ActivityOccurrence
ResourceType	usedBy	only activity:Activity
	consumedBy	only activity:Activity

Property	Characteristic	Value (if applicable)
hasResourceType	-	-
usedInOccurrence	Functional	-
consumedInOccurrence	Functional	-
usedBy	subPropertyOf	preconditionOf
consumedBy	subPropertyOf	preconditionOf
hasOwner	domain	Resource
hasAssociatedLocation	domain	Resource

Reused Ontologies:

- iCity-Activity

6.1.6 Mereology Ontology

https://w3id.org/icity/Mereology.owl

Namespace: mereology

While sometimes conflated, there are distinctly different types of "parthood". Mereology focuses on identifying and defining these differences. In particular, we define the following different types of parthood: proper-part-of, component-of, and contained-in. The distinction between these types of parthood may be best explained with the use of examples. An item may be *contained in* my car, but that does not make it a *component of* my car. For example, we may wish to describe passengers or cargo being *contained in* a vehicle, but this relation must be distinguished from the parts and components that make up a vehicle. The distinction between component-of and proper-part-of is slightly more subtle, however there is a difference in semantics. While we may define components of a vehicle, different zone systems (wards, postal codes) are not components, but proper parts of larger areas. Two areas that have the same area as a proper-part do not necessarily share a proper-part relation (i.e. they may simply overlap), whereas two car parts that share the same part as a component must somehow be related through the component-of relation.

- Something may be a Proper Part of some other thing.
  An object cannot be a proper part of itself. Thus, any object must have more than one proper part.
  Proper Parthood is transitive.
  Proper parthood is dense and so there exist no immediate proper parts; in other words, given some object, whatever proper part, *x*, we choose, there exists some slightly larger proper part of the object that also has *x* as a proper part.
- Something may be a Component of some other thing
  More specifically, something may be a *immediate* component of something; in other words, if *x* is an immediate component of *y*, then there does not exist any other object that is a component of *y* and has *x* as a component.
  Component-of is transitive. Immediate component-of is not transitive.
  Immediate component-of is a subproperty of component-of.
- Something may be contained-in some other thing; more specifically it may be *immediately* contained in something.
  Containment is transitive. Immediate containment is not transitive.
  Immediate containment is a subproperty of containment.

Object	Property	Value
Thing	subClassOf	hasProperPart exactly 0 Thing or hasProperPart min 2 Thing

Property	Characteristic	Value (if applicable)
partOf		
hasPart	inverseOf	partOf
properPartOf[7]	subPropertyOf	partOf
hasProperPart	inverseOf	properPartOf
	subPropertyOf	hasPart
componentOf	subPropertyOf	partOf
hasComponent	inverseOf	componentOf
	subPropertyOf	hasPart
immediateComponentOf	subPropertyOf	componentOf
containedIn	subPropertyOf	partOf
contains	inverseOf	containedIn
	subPropertyOf	hasPart
immediatelyContainedIn	subPropertyOf	containedIn

Reused Ontologies:

- None directly, but reused concepts as defined by (Bittner & Donnelly, 2005), however theirs is not an officially published ontology.

[7] Note that while we would like to specify the transitivity of the properPartOf relation, we are limited by OWL due to the fact that we wish to define cardinality restrictions on this relation (making it a non-simple property). For the present, we have removed the transitivity property in order to maintain the cardinality restriction. Likewise with the containedIn and componentOf relations.

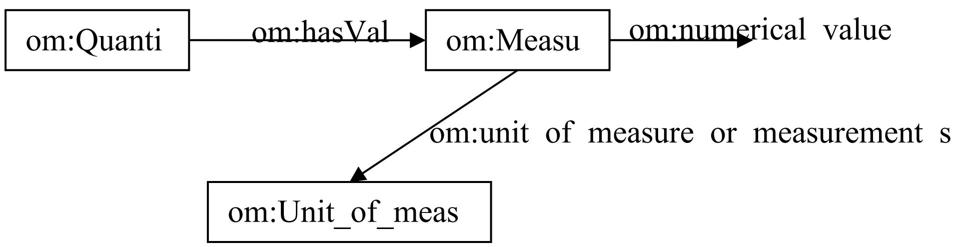
6.1.7 Ontology of Units of Measure

https://w3id.org/icity/OM.owl

Namespace: om

The Ontology of Units of Measure provides a structured vocabulary to describe, among other things, the different values (measures) that we associate to given quantities. This allows us to provide greater detail regarding specific measurements that are defined in the ontology. Rather than simply have a simple data property to describe the length of some road segment as "10 m", with the units of measure ontology we are able to describe the nature of the quantity (i.e. length), its value as a Measure (10 m), and also describe the unit that the measure's numerical value is given in (e.g. metres).

- A quantity has some measured value
- A measured value (om:Measure) is associated with a unit of measure
- There are many types (subclasses) of units of measure, such as length, mass, speed, and currency.



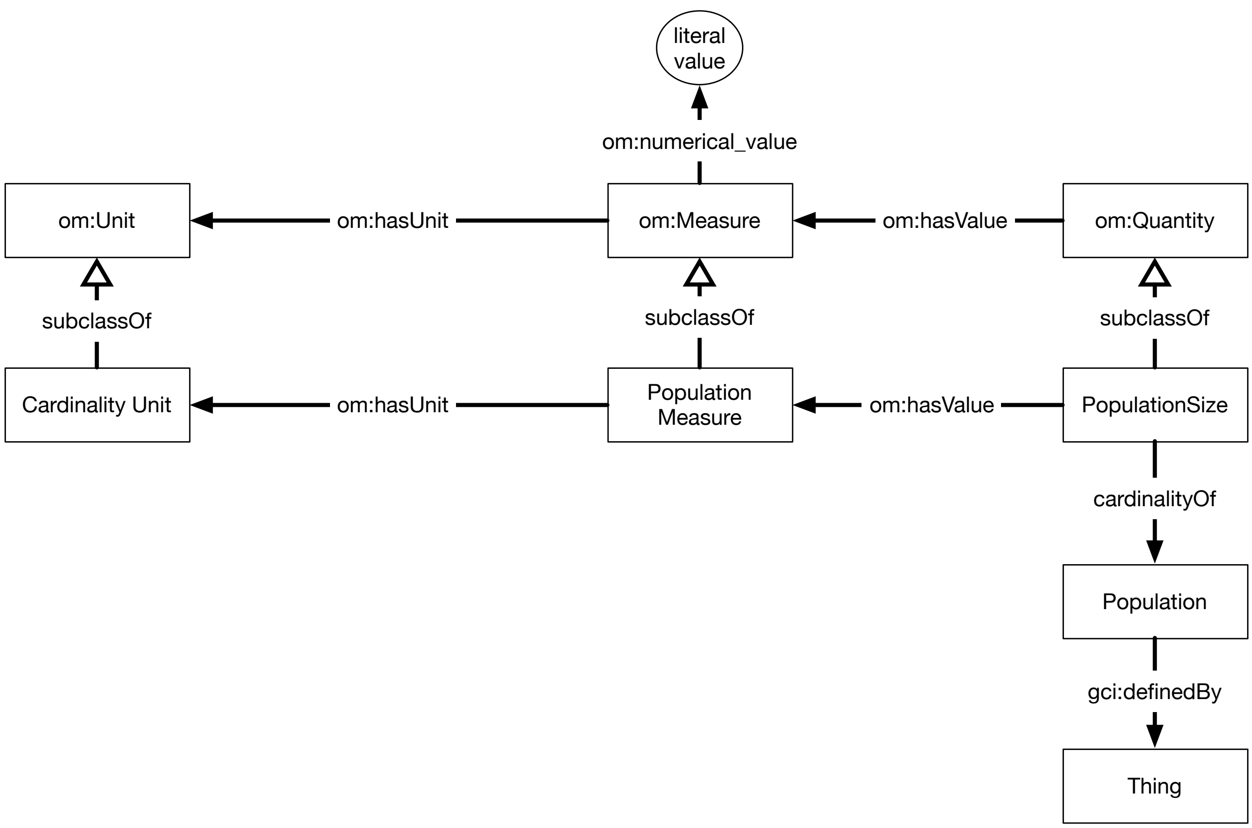


Figure 5: Representation of populations as reused from the GCI Ontology.

In order to represent populations, we reuse the following classes from the GCI-Foundation ontology: ~~govstat:Population,~~ gci:PopulationSize, gci:PopulationSizeMeasure, and gci:CardinalityUnit. Refer to the working paper on the GCI Ontology for more details on this approach. The meaning of population is general here, while it may define a population of residents within some zone, it may also be used to describe the population of vehicles occupying some stretch of the road network.

The quantity of interest (population size being measured/described) is defined as gci:Population_Size, a subclass of Quantity. Population_Size has some unit of measure (a cardinality unit), and has_value some Population_Measure (with an associated numeric value). The elements associated with a population quantity are captured through the defined_by property that relates a Population to some class of objects. For example, consider the measurement of the number of cars on some road segment, we could specify: Population_Size and cardinalityOf only (Population and definedBy only (Vehicle)). The defining population might be even more precisely captured for a given Road Segment, X, as depicted in Figure 6: definedBy only (Vehicle and onSegment value X). These specializations are defined, as required, within the relevant module; for example, a vehicle population would be defined in a module that contains the required concepts of vehicles and road segments. The units of measure ontology captures only to core concepts of Population Size, Population Measure, Cardinality Unit, and Population, as depicted in Figure 5.

Capacity and its associated quantity and measure are defined similar to population.

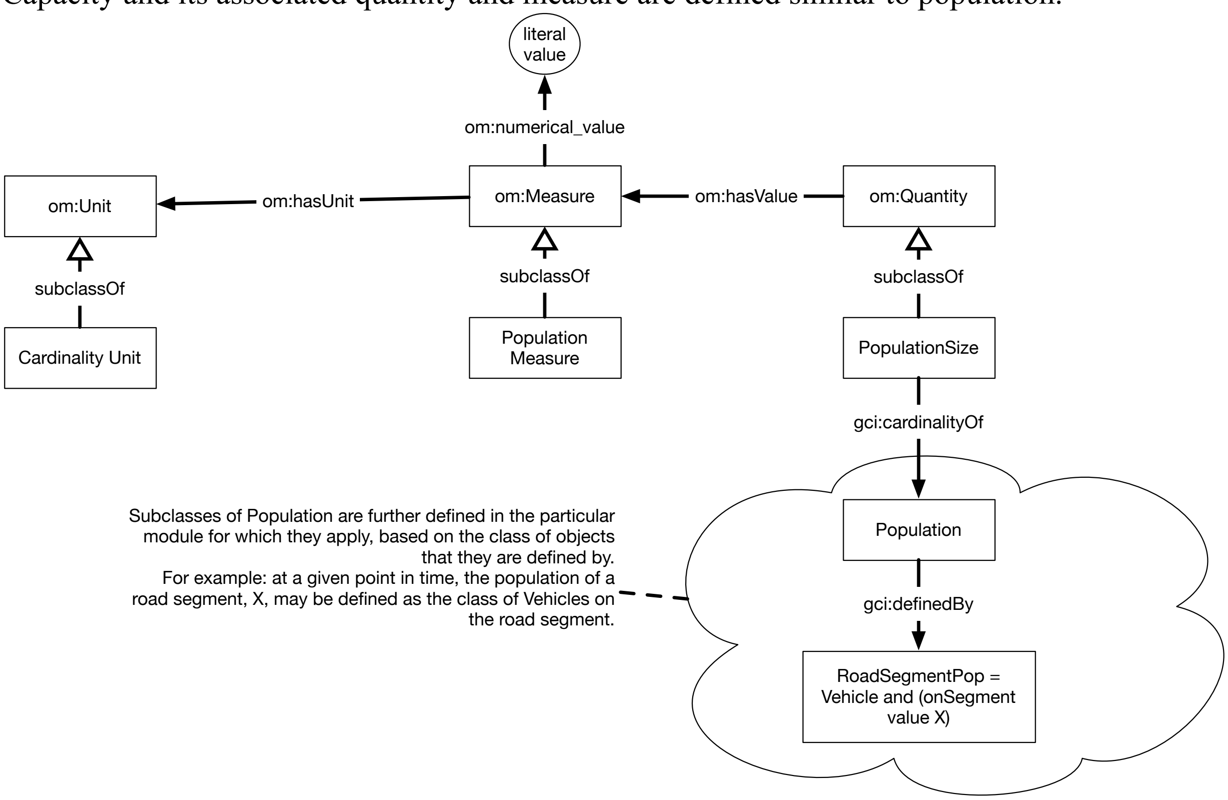


Figure 6: Specialization of populations.

Future work:

Q: is it more accurate to describe the position coordinates as quantities that are measured in degrees *that are relative to a geodetic datum* (e.g. NAD83)? In this approach, we are considering "degrees relative to a particular datum" as a kind of measure. In any case, it is important that we are able to distinguish between different position systems as latitudinal and longitudinal values cannot be assumed to use the same sort of system. In particular, WGS84 and NAD83, which were originally nearly equal are now considerably different (depending on the area) due to changes that have occurred to the earth since 1984. Note that http://data.ign.fr/def/ignf/20150505.en.htm may be a relevant ontology.

Extensions:

- Updated to OM2.0
- OM:MonetaryValue a om:Unit_of_measure
- Monetary Value: Monetary Values may be attributed to things such as the purchase of a dwelling, or the salary of some Job.
  A Monetary Value has a **dollar value** relative to a particular **date** (year).
  A Monetary Value has some associated **currency**.
- Population (via GCI): The population class is defined as a subclass of om:Quantity. It represents the quantities of counted sets. Various subclasses of Population will be defined as required, e.g. Resident_Population, Vehicle_Population, etc. A population has a measure (Population_measure) in some cardinality unit
- Population_measure
- Population_unit_of_measure
- Cardinality_unit
- Capacity: subclass of Quantity and hasUnit CardinalityUnit
- CapacityRate, e.g.
  - OM:lanecap a om:Unit_of_measure (vehicles/hour)

General protocol: quantities, units, and/or measures that are defined with domain-specific concepts (e.g. vehicles, lanes) are defined by reusing and extending the units of measure ontology in the relevant modules. E.g.,

- Lane capacity (vehicles/hr):
  define unit, define cardinality on capacity of vehicles
- Link capacity(vehicles/hr/lane): subclassOf Quantity;
  define unit: hasValue only (hasUnit only UnitDivision and hasNumerator only Cardinality_unit_per_hour; hasValue only (hasUnit only UnitDivision and hasDenominator only Cardinality_unit
  define numerator cardinality on capacity of vehicles
  define denominator cardinality on lanes

Object	Property	Value
om-2:Quantity	om-2:hasValue	only om-2:Measure
om-2:Measure	om-2:hasUnit	only om-2:Unit
om-2:Length_unit	subClassOf	om-2:Unit
om-2:Mass_unit	subClassOf	om-2:Unit
om-2:Area_unit	subClassOf	om-2:Unit
om-2:Acceleration_unit	subClassOf	om-2:Unit
om-2:Volume_unit	subClassOf	om-2:Unit
om-2:Speed_unit	subClassOf	om-2:Unit
om-2:Amount_of_money_unit	subClassOf	om-2:Unit
Geo_Position_unit	subClassOf	om-2:Unit
gci:Cardinality_unit	subClassOf	om-2:Unit

om-2:UnitDivision	subClassOf	om-2:Unit
Cardinality_unit_per_time	subClassOf	om-2:UnitDivision
	om-2:hasNumerator	only gci:Cardinality_unit
	om-2:hasDenominator	only om-2:TimeUnit
...	subClassOf	om-2:Unit_of_measure
MonetaryValue	subClassOf	om-2:Measure
	hasRelativeYear	exactly 1 xsd:gYear
	om-2:hasUnit	only om-2:Amount_of_money_unit
gci:Population_measure	subClassOf	om-2:Measure
	subClassOf	CardinalityMeasure
CardinalityMeasure	subClassOf	om-2:Measure
	hasUnit	only gci:Cardinality_unit
ValueOfMoney	subClassOf	om-2:Quantity
	subClassOf	om-2:AmountOfMoney
	om-2:hasValue	only MonetaryValue
om-2:Length	subClassOf	om-2:Quantity
	om-2:hasValue	only (om-2:Measure and om-2:hasUnit only om-2:Length_unit)
gci:PopulationSize	subClassOf	om-2:Quantity
	om-2:hasValue	only gci:Population_measure
	gci:cardinalityOf	exactly 1 gci:Population
CapacitySize	subClassOf	om-2:Quantity
	om-2:hasValue	only gci:Cardinality_measure
	gci:cardinalityOf	exactly 1 Capacity
CapacityRate	subclassOf	om-2:Quantity
	om-2:hasValue	only (om-2:hasUnit only CardinalityUnitPerTime)
	gci:cardinality_of	exactly 1 Capacity
om-2:Mass	subClassOf	om-2:Quantity
	om-2:hasValue	only (om-2:hasUnit only om-2:Mass_unit)
om-2:Area	subClassOf	om-2:Quantity
	om-2:hasValue	only (om-2:hasUnit only om-2:Area_unit)
om-2:Volume	subClassOf	om-2:Quantity
	om-2:hasValue	only (om-2:hasUnit only om-2:Volume_unit)
om-2:Acceleration	subClassOf	om-2:Quantity
	om-2:hasValue	only (om-2:hasUnit only om-2:Acceleration_unit)
om-2:Speed	subClassOf	om-2:Quantity

	om-2:hasValue	only (om-2:hasUnit only om-2:Speed_unit)

Property	Characteristic	Value (if applicable)
om-2:hasBaseUnit	domain	om-2:System_of_units
om-2:hasBaseUnit	range	om-2:Unit
om-2:hasDenominator	domain	om-2:UnitDivision
om-2:hasDenominator	range	om-2:Unit
om-2:hasNumerator	domain	om-2:UnitDivision
om-2:hasNumerator	domain	om-2:Unit

Reused Ontologies:

- ~~om: Ontology of Units of Measure~~[8]
- om-2: Ontology of Units of Measure 2.0[9]
- gci: Global City Indicators Foundation Ontology[10]

6.1.8 Monetary Value Ontology

~~*https://w3id.org/icity/MonetaryValue.owl*~~

~~**Namespace: monetary**~~

- Removed. Monetary Value now integrated with Units of Measure Ontology

6.2 SSN Ontology

https://w3id.org/icity/SSN.owl

Namespace: ssn

The SSN (Semantic Sensor Network) ontology is reused directly from http://www.w3.org/ns/ssn/. SOSA is the foundation for the SSN (Semantic Sensor Network) and provides the scope required for this application, it does not specify any definitions for the terms, therefore we have opted to reuse the SSN Ontology. The SSN Ontology is a W3C recommendation that has been widely adopted to represent sensors and their observations. The relevant terms are highlighted here.

- A Sensor is a device that makes some observation and may be triggered by some stimulus.
- An Observation has some feature of interest – the thing whose property is being detected by the sensor. An observation observes some ObservableProperty. A phenomenon time (i.e. the time at which the property was demonstrated) and result time may be associated with an observation.

Object	Property	Value
Sensor	sosa:madeObservation	only sosa:Observation
	sosa:observes	only sosa:ObservableProperty
	ssn:detects	only ssn:Stimulus

[8] om:http://www.wurvoc.org/vocabularies/om-1.6/

[9] om: http://www.ontology-of-units-of-measure.org/resource/om-2/

[10] http://ontology.eil.utoronto.ca/GCI/Foundation/GCI-Foundation-v2.owl#

Observation	sosa:madeBySensor	exactly 1 sosa:Sensor
	sosa:hasFeatureOfInterest	exactly 1 owl:Thing and only sosa:FeatureOfInterest
	sosa:hasResult	exactly 1 owl:Thing and only sosa:Result
	sosa:observedProperty	exactly 1 owl:Thing and only sosa:ObservableProperty
	sosa:phenomenonTime	exactly 1 owl:Thing
	sosa:resultTime	exactly 1 rdfs:Literal
	ssn:wasOriginatedBy	exactly 1 owl:Thing and only ssn:Stimulus
ObservableProperty	inverse ('is proxy for')	only ssn:Stimulus
	inverse ('observed property')	only sosa:Observation
	sosa:'is observed by'	only sosa:Sensor
FeatureOfInterest	ssn:'has property'	min 1 owl:Thing and ssn:Property
Result	sosa:'is result of'	min 1 owl:Thing

Reused Ontologies:

- SSN
- SOSA

6.3 *iContact Ontology*

https://w3id.org/icity/iContact

Namespace: icontact

Contact information is relevant for a range of concepts in the transportation domain. For example, a building may have some associated address, similarly a person or an organization may have some contact address (or phone number, email, etc). Note that a person's contact address may differ from their place of residence. The iContact ontology provides the core concepts necessary to define this type of information. It also uses concepts from the spatial location ontology in order to associate an address with a location.

To do: import and extend as required for Person, Building, Organization(?) and Parking ontologies

Object	Property	Value
iContact:Address	hasStreetNumber	exactly 1 xsd:nonNegativeInteger
	hasStreet	only xsd:string
	hasCity	exactly 1 schema:city
	…	
	spatialloc:hasLocation	exactly 1 geo:Feature

Reused Ontologies:

- iContact: http://ontology.eil.utoronto.ca/icontact.owl
- iCity Spatial Location: *https://w3id.org/icity/SpatialLoc/*

6.4 *Person Ontology*

https://w3id.org/icity/Person

Namespace: person

- Person: A Person may have a **unique identifier**.
  A Person has a **date of birth**, and may have a **date of death**.
  A Person has a **mother** and **father**, and may have a **spouse** and/or **child**(ren). Note that we define the parent relation as the legal relation as opposed to biological. This property may be specialized and restricted, for example hasBiologicalMother: exactly 1 Person.
  A Person may **have** some **Job** and associated **Income**.
  A Person has an **address** of residence and may have other contact information such as **E-mail**, **phone number**, etcetera.

Object	Property	Value
PersonPD	subclassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some Person and change:hasManifestation only Person
	change:existsAt	exactly 1 time:Interval
	hasPersonID	only PersonId
	schema:birthDate	exactly 1 time:Instant
	hasSex	exactly 1 Sex
Person	equivalentClass	change:manifestationOf some PersonPD and change:manifestationOf only PersonPD
	subclassOf	change:Manifestation
	change:existsAt	exactly 1 time:TemporalEntity
	schema:deathDate	max 1 time:Instant
	schema:parent	only Person
	schema:spouse	only Person
	schema:children	only Person
	hasIncome	only MonetaryValue
	schema:address	some schema:PostalAddress
	hasSkill	only Skill
	hasQualification	only Qualification

Reused Ontologies:

- schema.org[11] (A vocabulary as opposed to an ontology)
- iCity-Change
- iCity-MonetaryValue
- owl-time

6.5 Household Ontology

https://w3id.org/icity/Household.owl

Namespace: household

In order to define a Household, we require the following classes and properties:

- Family: We may define different types of Family (e.g. Immediate, Extended). Here, we simply make the commitment that it is a group of people who are connected via the has-spouse or has-child properties. From these, we can derive grandparents, aunts, uncles, etcetera.
  One question to consider is to what degree the general/extended Family concept makes sense or is useful. After a few generations the concept of a family will become quite large and confusing, with Persons

[11] http://schema.org/

belonging to many different Families. At a certain point it may be more useful to consider a relatedTo property between Persons, or only defining restricted subclasses of Family.
to do: add hasMember only Person

- Household: A Household **occupies** a particular Dwelling, according to some **tenure** type. It is defined by this location, so that if the members move (even collectively), the new residence constitutes a new Household.
  Note that a Household, and likely many other classes may have different definitions in different contexts/applications. To address this we may be required to introduce specializations of the class (e.g. ILUTE_Household, TTS_Household) in future extensions.
- Dwelling Unit: A Dwelling Unit is **occupied** by a Household.
  A Dwelling Unit has a **market value**.
  A Dwelling Unit has some Location.

Object	Property	Value
FamilyPD	subclassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some Family and change:hasManifestation only Family
	change:existsAt	exactly 1 time:Interval
Family	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some FamilyPD and change:manifestationOf only FamilyPD
	change:existsAt	exactly 1 time:TemporalEntity
HouseholdPD	subclassOf	change:timeVaryingConcept
	equivalentClass	change:hasManifestation some Household and change:hasManifestation only Household
	change:existsAt	exactly 1 time:Interval
	occupies	exactly 1 DwellingUnit
gci:Household	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some HouseholdPD and change:manifestationOf only HouseholdPD
	change:existsAt	exactly 1 time:TemporalEntity
DwellingUnitPD	subclassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some DwellingUnit and change:hasManifestation only DwellingUnit
	change:existsAt	exactly 1 time:Interval
	schema:address	only schema:PostalAddress
	spatial_loc:hasLocation	only spatial_loc:SpatialFeature
DwellingUnit	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some DwellingUnitPD and change:manifestationOf only DwellingUnitPD
	change:existsAt	exactly 1 time:TemporalEntity
	occupiedBy	exactly 1 Household
	hasValue	only monetary:MonetaryValue

	tenureType	only Tenure

Property	Characteristic	Value (if applicable)
occupiedBy	inverseOf	occupies

Reused Ontologies:

- schema.org
- gci: GCI-Shelter Ontology[12]
- icity-foundation: iCity-Foundation Ontology

6.6 Organization Ontology

https://w3id.org/icity/Organization.owl

Namespace: org

- Organization: A company or other sort of group of individuals in the urban system with some goal(s).
  An Organization may **own** Property, including different types of Buildings.
  An Organization may have an address.
  An Organization has at least 2 members.
  An Organization has some Goal(s); this represents some state or complex states, and allows for the representation of various groups' responsibilities.
  An Organization may be divided into Divisions.
- Organization Agent: Members of an organization.
  Organization Agents have goals, authority, and may be members of some team.
  An Organization Agent plays a Role within the Organization.
- Role: A Role has a single (possibly complex) Goal.
  A Role has some authority, requires some skill, and may also have some associated processes.
- Firm: A Firm is a type of organization.
  A Firm has an address and an industry type, and some Employees.
  A Firm may have a Business Establishment(s).
- Business Establishment: A Business establishment is a physical location where a Firm conducts business.
  A Business Establishment has a Location and may have an address.
- Employee: An Firm **has** some **Employees**, whom it **employs for** some Occupation.
  An Employee is a type of Organization Agent.
  An Employee may be employed at a particular Business Establishment.
  An Employee may be responsible for one or more Roles within the Organization.
  An Employee is **employed by** some Organization, unless the Person is self-employed.
  An Employee **has a Wage/Salary** and **may work at** some **Location** (this may be the location of the Firm, an alternate Location, or a Location that is subject to change).
  to do: Employee subclass of Person

Object	Property	Value
OrganizationPD	subclassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some Organization and change:hasManifestation only Organization
	change:existsAt	exactly 1 time:Interval
tove:Organization	subclassOf	change:Manifestation

[12] http://ontology.eil.utoronto.ca/GCI/Shelters/GCI-Shelters.html

	equivalentClass	change:manifestationOf some OrganizationPD and change:manifestationOf only OrganizationPD
	change:existsAt	exactly 1 time:TemporalEntity
	schema:address	only schema:PostalAddress
	tove:has_goal	only tove:Goal
	tove:consists_of	only tove:Division
tove:Role	tove:has_goal	only tove:Goal
	tove:has_process	only (tove:Process or activity:Activity)
	tove:has_authority	only tove:Authority
	tove:requires_skill	only tove:Skill
	tove:has_resource	only resource:ResourceType
tove:Goal	subClassOf	StateType
FirmPD	subclassOf	tove:Organization
	hasFirmId	only FirmId
	equivalentClass	change:hasManifestation some Firm and change:hasManifestation only Firm
	change:existsAt	exactly 1 time:Interval
Firm	subclassOf	tove:Organization
	equivalentClass	change:manifestationOf some FirmPD and change:manifestationOf only FirmPD
	change:existsAt	exactly 1 time:TemporalEntity
	schema:address	exactly 1 schema:PostalAddress
	hasIndustryType	only IndustryType
	hasEstablishment	only BusinessEstablishment
BusinessEstablishmentPD	subclassOf	change:TimeVaryingConcept
	change:existsAt	exactly 1 time:Interval
	hasBusinessId	only BusinessId
	equivalentClass	change:hasManifestation some BusinessEstablishment and change:hasManifestation only BusinessEstablishment
BusinessEstablishment	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some BusinessEstablishmentPD and change:manifestationOf only BusinessEstablishmentPD
	change:existsAt	exactly 1 time:TemporalEntity
	spatial_loc:hasLocation	exactly 1 spatial_loc:SpatialFeature
	schema:address	only schema:PostalAddress
tove:OrganizationAgent	tove:member_of	only tove:Division
	tove:plays	only tove:Role
	tove:has_goal	only tove:Goal

	tove:has_authority	only tove:Authority
Employee	subclassOf	tove:OrganizationAgent
	employedAs	some Occupation
	hasPay	some Wage or Salary
	worksAt	some spatial_loc:SpatialFeature
Wage	hourlyPay	exactly 1 monetary:MonetaryValue
	overtimePay	only monetary:MonetaryValue
Salary	hasAnnualPay	exactly 1 monetary:MonetaryValue
tove:Activity	equivalentClass	activity:Activity
tove:Resource	equivalentClass	resource:Resource

Reused Ontologies:

- tove: The TOVE Organization ontology[13], as originally presented by (Fox, Barbuceanu, Gruninger, & Lin, 1998) with modifications to account for the difference in our representation of states, where a Goal is a subclass of StateType, and where Activities are enabled/caused by state types.
  This modification also results in the removal of the StateEmpowerment class. Note that it is possible to introduce a similar concept if required, however this would likely take the form of a property that relates an organization agent to some state-types (where the states they are empowered to take an object to, and the object itself, are described by the state type).
- icity-foundation: iCity-Foundation Ontology
- schema.org (vocabulary)

6.7 Building Ontology

https://w3id.org/icity/Building.owl

Namespace: building

- Building: A Building is a structure with some location in the urban system. The location of the Building in space may change due to construction, but the Parcel/Lot of land it is located on cannot.
  There are different types (**subclasses**) of buildings, such as House, Apartment Building, Office Building, and so on.
  A Building has a market **value**.
  A Building **has** some **Location**.
  A Building contains one or many units.
- BuildingUnit: A BuildingUnit has a size (square footage, number of rooms)
  A BuildingUnit may contain some Facilities, e.g. kitchen, bath.(Note that contain is distinct from the notion of including amenities, which may be part of the Tenure)
  A BuildingUnit has an address.
  A BuildingUnit has a value, and may have some rental fee.

Object	Property	Value
BuildingPD	subClassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some Building and change:hasManifestation only Building
	change:existsAt	exactly 1 Interval
Building	equivalentClass	change:manifestationOf some BuildingPD and change:manifestationOf only BuildingPD
	subClassOf	change:Manifestation
	change:existsAt	exactly 1 TemporalEntity

[13] http://ontology.eil.utoronto.ca/tove/organization.html

	spatial_loc:hasLocation	exactly 1 spatial_loc:SpatialFeature
	monetary:hasValue	only monetary:MonetaryValue
	mereology:contains	only BuildingUnit
House	subclassOf	Building
ApartmentBuilding	subclassOf	Building
OfficeBuilding	subclassOf	Building
IndustrialBuilding	subclassOf	Building
BuildingUnitPD	subclassOf	change:TimeVaryingConcept
	change:existsAt	exactly 1 Interval
	equivalentClass	change:hasManifestation some BuildingUnit and change:hasManifestation only BuildingUnit
	mereology:containedIn	exactly 1 Building
	schema:address	exactly 1 schema:PostalAddress
BuildingUnit	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some BuildingUnitPD and change:manifestationOf only BuildingUnitPD
	change:existsAt	exactly 1 TemporalEntity
	monetary:hasValue	only monetary:MonetaryValue
	hasRent	only monetary:MonetaryValue
	hasUnitSize	only om:area
	hasRooms	only xsd:int
	hasFacility	only Facility

Reused Ontologies:

- ~~iCity-Foundation~~
- Change
- Units of measure
- Mereology
- Spatial location

6.8 Vehicle Ontology

https://w3id.org/icity/Vehicle.owl

Namespace: icity-vehicle

- Vehicle: A Vehicle provides a means of transportation within the urban system.
  A Vehicle is **associated with some Mode** of transportation.
  A Vehicle has a Vintage.
  A Vehicle has a Manufacturer (make).
  There are different types (**subclasses**) of vehicles: Motorcycle, Sedan, Truck, Bus, Commercial Cargo Vehicle, Train, Bicycle...
  A Vehicle **has a capacity** of passengers
  A Vehicle **has a capacity** of cargo
  A Vehicle **has a Speed** at some point in time
  A Vehicle **has a location** at some point in time.

Object	Property	Value

VehiclePD	equivalentClass	change:hasManifestation some Vehicle and change:hasManifestation only Vehicle
	subclassOf	change:TimeVaryingConcept
	change:existsAt	exactly 1 time:Interval
	hasMode	only Mode
	schema:productionDate	only time:DateTimeDescription
	schema:brand	only schema:Brand
	schema:vehicleSeatingCapacity	exactly 1 xsd:int
	schema:cargoVolume	only om:volume
	hasCargoCapacityLoad	only om:Quantity
	schema:driveWheelConfiguration	schema:DriveWheelConfigurationValue
	schema:fuelConsumption	schema:QuantitativeValue
	schema:fuelEfficiency	schema:QuantitativeValue
	schema:fuelType	schema:QualitativeValue
	schema:mileageFromOdometer	schema:QuantitativeValue
	schema:numberOfDoors	only xsd:int
	schema:numberOfAxels	only xsd:int
Vehicle	equivalentClass	change:manifestationOf some VehiclePD and change:manifestationOf only VehiclePD
	subclassOf	change:Manifestation
	change:existsAt	exactly 1 time:TemporalEntity
	schema:purchaseDate	only time:DateTimeDescription
	hasSpeed	only om:speed
	spatial_loc:hasLocation	only spatial_loc:SpatialFeature
schema:QualitativeValue	subClassOf	om:quantity

Ontologies Reused:

- Schema.org (vocabulary)
- iCity-Foundation

6.9 Transportation System Ontology

https://w3id.org/icity/TransportationSystem.owl

Namespace:transport

While most existing work attempts to describe the network based on its physical constructs, we model the network flow and the physical infrastructure separately. The motivation for this is that the constraints on transportation flow are something that is *applied to* the physical infrastructure. These constraints are distinct from the physical characteristics and so should be defined separately. Although some constraints may be related, such as flow constraints imposed by the size of the lane that an arc accesses, this is a specific relationship that should be captured rather than conflating the concepts. For example, there is nothing to stop a vehicle from going the wrong way on a road, except for the flow of traffic that is imposed on the system (and these constraints may change with time). This results in the identification of two key concepts: the Transportation Network (a directed graph), and the Transportation Infrastructure (a physical feature where transportation occurs).

We relate the Network and the Infrastructure by relating an Arc to a Transportation Complex (or other Road Segment) with the "accesses" property. In this way, we may define an Arc accessing various Transportation Complexes at different Levels of Detail (LOD).
In this representation Nodes do not access the Transportation Infrastructure nor are they part of it in any way. Both Nodes and Arcs may have implicit locations based on the infrastructure they access, however unlike the infrastructure classes, Nodes and Arcs are *not* Spatial Things. A Node may have a control (e.g. a signal) with a physical presence somewhere else (traffic lights apply to one side of the intersection, but are actually located on the other side of the intersection); by separating the physical infrastructure and the network flow we are able to accurately represent this.

The OTN (Ontology of Transportation Networks[14]) ontology, as presented by (Lorenz, Ohlbach, & Yang, 2005) ,also defines terms such as nodes, arcs, and road/rail elements. Although its scope is similar, we have elected not to reuse it in the design of this ontology.

- Lack of modular structure
- Lack of maintenance/activity
-

- Network: A collection of Nodes and Arcs that enables transportation. A Network may have some cost associated to its access.
- Link: A directed connection in the Network that enables transportation via some Mode(s) from one Node to another.
  A link contains one or more Arcs that represent individual flows of traffic (e.g. traffic lanes, bicycle lanes).
  A link begins and ends at a source and sink Node.
  A link has some (straight-line) length description, in km.
  A link is associated with, or considered to be *in*, a municipality and a planning district.
  A link has one or more Mode(s) of access.
- Arc: A directed connection in the Network that enables transportation via a particular Mode(s) from one Node to another.
  An Arc begins and ends at the source and sink of the Link it is contained in.
  An Arc has access to some Spatial Thing (such as a road), which may change over time.
  An Arc may impose access restrictions (for example, based on the size of vehicle), which are subject to change.
  An Arc may have some cost associated to its travel.
  There is a relationship between the modes of access of a link and those of the arcs it contains that should be captured in a more detailed representation.
  An Arc may have some posted and/or free flow speed. It may also be described with a volume delay function (VDF).
- Node**:** A point in the Network at which Arcs are connected. A node as a unique identifier; for example, as defined in the EMME NCS11.
  A Node may contain different types of controls: Network Transfer, Signal Control, and Flow Control.
  A Node may be associated with specific location information (e.g. coordinates). Note that this may be

[14] http://www.pms.ifi.lmu.de/rewerse-wga1/otn/OTN.owl

subject to change. The physical location of a node (generally larger than a single point) may be inferred based on the locations of the transportation complexes which it connects.
A Node accesses some TransportationComplex, such as an Intersection. In the future, it may be useful to define other specific types of TransportationComplexes that are accessed by nodes, (e.g. bus stops).

- Network Transfer: Enables transfer between networks at a given Node.
- Signal Control: Controls the flow of transportation between some of the incoming and outgoing arcs that the Node connects. Signal Controls have specialized attributes such as the number of phases, phase length, signal timing, type of signal. Note that the phases and/or the phase length may vary as a function of time of day or other triggers (e.g. ground sensors, traffic sensors).
- Flow Control: Controls the flow of traffic at a given Node.
  A Flow Control may be operative/inoperative at different times. For example, "no left turns from 4-6pm".
  A Flow Control may be a generalization of Signal Control.
- Mode: A mode of transportation is a **means of** performing travel within the urban system.
  There are various types (instances) of Mode: Foot, Bike, PersonalVehicle, PublicTransit, Cab, CommercialVehicle, Plane, Boat, Train.

The physical Infrastructure of the transportation system is defined, as required, at different levels of detail (LOD). Specific types of Transportation Complex (a term we adopt from the CityGML schema) may be defined according to the Arcs that access them. We define the following types of Transportation Complex.

- Road
- Rail
- Waterway
- Airway
- Bike Trail
- Footpath
- Parking

Each Transportation Complex may be further defined as follows:

- **Road**: An aggregation of Road Segments with the same name.
- **RoadSegmentPD**: accessed only by Links that are not accessible by water or air modes.
  Different RoadSegments Perdurants will be accessed by Arcs that are accessible by various other Modes, not necessarily *everything* else. A Road Segment Perdurant is comprised of Road Segments that exist over time.
- **RoadSegment**: A RoadSegment has variant attributes.
  A RoadSegment has an owner, access restrictions, and is accessed by some Arc(s) -- all of which may change over time.
  A RoadSegment has some location, which is co-located with (contains the locations of) the Arcs and Nodes it contains.
- **Rail**: An aggregation of Rail Segments with the same name.
- **RailSegmentPD**: Accessed only by Arcs that are accessible by rail modes.
  A RailSegment Perdurant has an invariant location, which is co-located with (contains the locations of) the Arcs and Nodes it contains. A Rail Segment Perdurant is comprised of Rail Segments that exist over time.
- **RailSegment**: A RailSegment has an owner, access restrictions, and is accessed by some Link(s).
- Note that the location of a RoadSegment is variable (e.g. road widening or other activities do not change the identity of the road element), whereas a RailSegment's is not.
- **IntersectionPD:** Accessed only by NodePDs. An Intersection Perdurant captures the physical entity of an intersection, which is co-located with various other transportation complexes (e.g. roads, paths) that pass through it. An Intersection Perdurant is comprised of Intersections that exist over time.
- **Intersection:** An Intersection exists at some time. It has some location. It may have some owner and is accessed by some Node. In the future, it may be useful to extend this class and relate it to physical infrastructure such as signs, signals, etc.

Classes may be defined for footpaths, bicycle lanes/trails, and so on. Should it be useful, this representation could be extended to define individual traffic lanes, (e.g. the transportation complex that is accessed by a single arc).

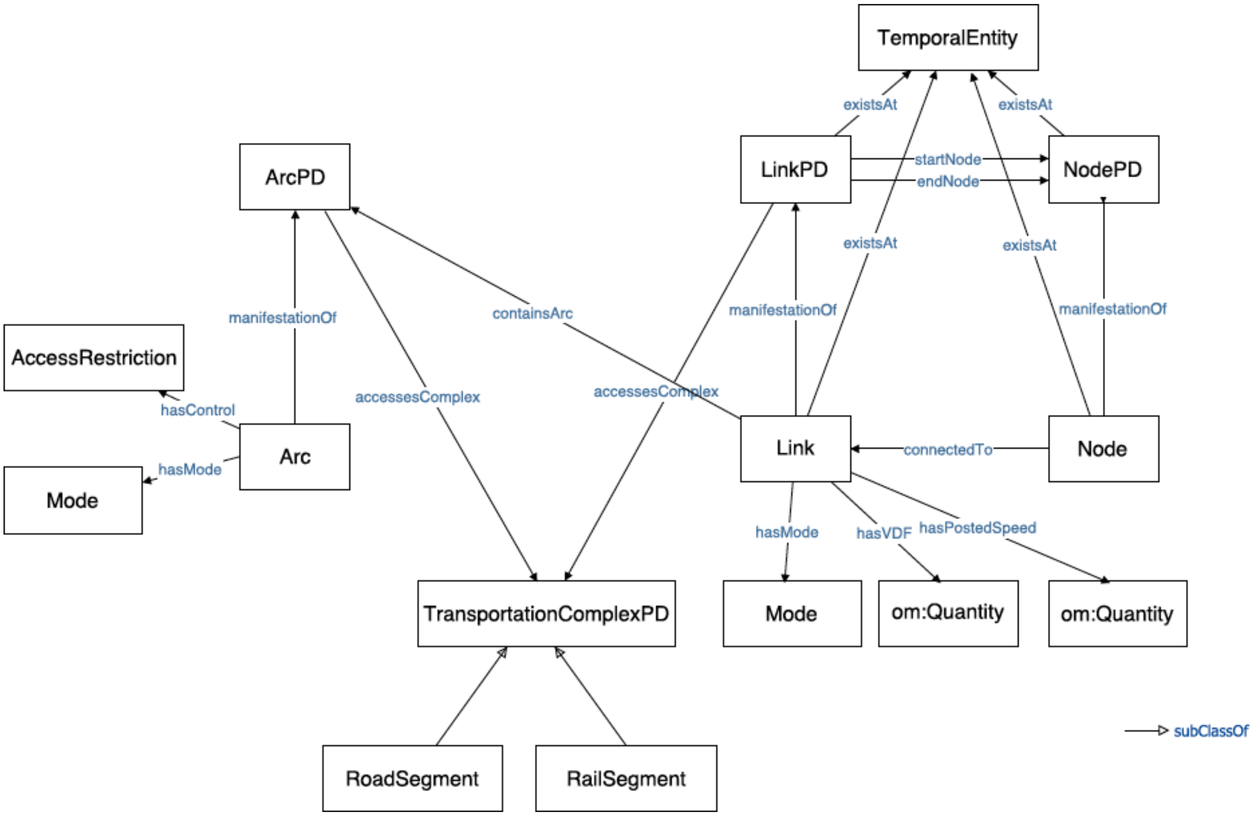


Figure 7: Structure of the Transportation Network (some attributes omitted).

Object	Property	Value
NetworkPD	subclassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some Network and change:hasManifestation only Network
	change:existsAt	exactly 1 time:Interval
Network	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some NetworkPD and change:manifestationOf only NetworkPD
	change:existsAt	exactly 1 time:TemporalEntity
	mereology:hasComponent	only Arc or Node
NodePD	subclassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some Node and change:hasManifestation only Node
	change:existsAt	exactly 1 time:Interval
	hasNodeID	max 1 NodeId
Node	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some NodePD and change:manifestationOf only NodePD

	change:existsAt	exactly 1 TemporalEntity
	mereology:componentOf	only Network
	connectedTo	min 1 Arc
	hasControl	only (NetworkTransfer or SignalControl or FlowControl)
	associatedLocation	only spatial_loc:Feature
LinkPD	subclassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some Link and change:hasManifestation only Link
	change:existsAt	exactly 1 time:Interval
	mereology:componentOf	only Network (variant or invariant?)
	startNode	exactly 1 NodePD
	endNode	exactly 1 NodePD
	accessesComplex	only TransportationComplexPD
Link	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some LinkPD and change:manifestationOf only LinkPD
	change:existsAt	exactly 1 time:TemporalEntity
	containsArc	min 1 ArcPD
	associatedLinkLength	exactly 1 om:length
	hasMode	min 1 Mode
	hasNumLanes	exactly 1 xsd:integer
	hasVDF	max 1 om: Quantity
	hasLinkCapacity	max 1 (om:Quantity and om:'has value' only (om:'has unit' only (om:'has numerator' only om:CardinalityUnitPerTime) and (om:'has denominator' only (om:'Cardinality Unit' and inverse(om:'has unit') only (inverse(om:'has value') only (gci:cardinality_of only (gci:defined_by only Arc)))))))
	hasFreeFlowSpeed	max 1 om:speed
	hasPostedSpeed	max 1 om:speed
	hasToll	only MonetaryValue
	inMunicipality	exactly 1 Municipality
	inPlanningDistrict	exactly 1 PlanningDistrict
ArcPD	subclassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some Arc and change:hasManifestation only Arc
	startNode	exactly 1 NodePD
	endNode	exactly 1 NodePD
	change:existsAt	exactly 1 time:Interval

	accessesComplex	only TransportationComplexPD
	containedInLink	exactly 1 LinkPD
Arc	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some ArcPD and change:manifestationOf only ArcPD
	change:existsAt	exactly 1 time:TemporalEntity
	accessesComplex	only TransportationComplex
	mereology:componentOf	only Network
	hasControl	only AccessRestriction
	hasMode	min 1 Mode
	hasLaneCapacity	exactly 1 om:CapacityRate
	hasVDF	max 1 om:quantity
	hasFreeFlowSpeed	max 1 om:speed
	hasPostedSpeed	max 1 om:speed
	hasToll	only MonetaryValue
	inMunicipality	exactly 1 Municipality (?) tbd – where should municipalities be defined
	inPlanningDistrict	exactly 1 PlanningDistrict
NetworkTransfer	controlFor	only Node
	connectsNetworks	min 2 Network
FlowControl	controlFor	only Node
	hasInflow	min 1 Arc
	hasOutflow	min 1 Arc
SignalControlPD	subClassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some SignalControl and change:hasManifestation only SignalControl
	change:existsAt	exactly 1 time:Interval
	controlFor	only Node
	hasInflow	min 1 Arc
	hasOutflow	min 1 Arc
SignalControl	subClassOf	change:Manifestation
	equivalentClass	change:manifestationOf some SignalControlPD and change:manifestationOf only SignalControlPD
	change:existsAt	exactly 1 time:TemporalEntity
	hasPhase	only SignalPhase
SignalPhase	signalLength	only time:DurationDescription
TransportationComplexPD	subClassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some TransportationComplex and

		change:hasManifestation only TransportationComplex
TransportationComplex	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some TransportationComplexPD and change:manifestationOf only TransportationComplexPD
	spatial_loc:hasLocation	only spatial_loc:Feature
otn:Road	hasRoadId	only RoadId
	aggregationOf	only RoadSegment
RoadSegmentPD	subclassOf	TransportationComplexPD
	equivalentClass	change:hasManifestation some RoadSegment and change:hasManifestation only RoadSegment
	hasRoadSegmentId	only RoadSegmentId
	change:existsAt	exactly 1 time:Interval
RoadSegment	equivalentClass	otn:RoadElement
	subClassOf	TransportationComplex
	equivalentClass	change:manifestationOf some RoadSegmentPD and change:manifestationOf only RoadSegmentPD
	change:existsAt	exactly 1 time:TemporalEntity
	spatial_loc:hasLocation	only spatial_loc:SpatialFeature
Mode	equivalentClass[15]	{C,E,F,H,I,J,B,G,L,M,P,Q,R,S,A,K,T,U,V,W,Y}
Municipality		
PlanningDistrict		
LoopDetector	subClassOf	ssn:Sensor
	sosa:observes	only RoadOccupancy
	sosa:madeObservation	Only OccupancyObservation
RoadOccupancy	subClassOf	sosa:ObservableProperty
	ssn:isPropertyOf* optional?	RoadSegment
OccupancyObservation	subClassOf	sosa:Observation
	sosa:hasSimpleResult	exactly 1 xsd:Boolean
	sosa:hasFeatureOfInterest	exactly 1 RoadSegment

1. Note that the classes of observable properties are primarily introduced for consistency with the SSN representation as a means of capturing the semantics of a class of Sensors (in this case, Loop Detectors).

[15] More options may be added as required. This list comes from the options specified in the EMME NCS11.

Any instance of, e.g. RoadOccupancy simply corresponds to a RoadSegment occupied by some thing, or occupied by nothing:
RoadOccupancy(x) ⇔ isPropertyOf(x,y) & RoadSegment(y) & [exists (t) occupiedBy(y,t) | -exists(t) occupiedBy(y,t)]
As a consequence of the 4D representation, an instance of the observable property RoadOccupancy refers to a property of a road segment at some time, t.

2. Additional semantics of sensors and observations: temporal restrictions, connection to object properties
3. A RoadSegment's vehicle count can be calculated based on the KB, but can we formalize the relationship between the count and the KB? i.e. number of observations in a given interval?

IntersectionPD	subclassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some Intersection and change:hasManifestation only Intersection
	inverse(accessesComplex)	only NodePD
	change:existsAt	exactly 1 time:Interval
Intersection	equivalentClass	otn:RoadElement
	subclassOf	change:Manifestation
	subClassOf	TransportationComplex
	equivalentClass	change:manifestationOf some RoadSegmentPD and change:manifestationOf only RoadSegmentPD
	change:existsAt	exactly 1 time:TemporalEntity
	spatial_loc:hasLocation	only geosparql:Feature
	inverse(accessesComplex)	only Node

Ontologies Reused:

- Change
- SpatialLoc
- SSN: Semantic Sensor Network ontology to capture sensors and their observations. These observations are processed or used directly as attributes of the network.

Notes:

- We observe that the properties *inMunicipality* and *inPlanningDistrict* may apply to other areas of the domain (e.g. land use, building ontologies), in which case they will be better defined at a lower (more foundational) level within the ontology. However, as they are currently only required for the Transportation System sub-ontology, it is currently not clear where and how this should be done. For now, we define these properties within the Transportation Network System ontology and leave the final organization for a future iteration if and when requirements for their widespread use are identified.

6.9.1 Travel Costs

https://w3id.org/icity/TravelCost.owl

Namespace: icity-travelcost

An extension of the transportation network (and other generic ontologies) is required in order to represent the different costs associated with accessing and travelling on the networks. These may take the form of direct costs such as tolls and fares, or possible indirect costs such as vehicle wear and tear, gas, etc. In addition, there may be non-monetary costs associated with travel such as pollution and travel time. Costs are associated with Network access, but also with individual

Arcs. They may also be dependent on situational factors such as time of day, or age of traveler. Travel Costs define the costs associated with accessing the transportation system; a travel cost is a property of an arc or its network. We define a separate extension of Trip Costs to capture other, indirect costs that may vary between individual trips; a trip cost is a property of some instance of travelling.

- Travel Cost: There are different types of Travel Costs which are derived from different factors, and may be defined in different ways. Travel Costs apply to Arcs and / or Networks.
- Distance Fee is a type of Travel Cost
  Distance Fee has an associated Cost
  It applies for a certain distance (between nodes, or per km)
  It applies to some Arc
  It may have an associated time-of-day applicability
  It may be associated to specific modes of transport
- Access Fee is a type of Travel Cost
  Access Fee has an associated Cost
  It may have an associated time-of-day applicability
  It may be associated to specific modes
  It applies to some Network

Object	Property	Value
TravelCost	travelCostOf	only (transportation:Arc or transportation:Network)
	applicableFor	only time:TimePeriod or time:CalendarPeriod
	applicableTo	only vehicle:Mode
	hasMonetaryCost	only monetary:MonetaryValue
transportation:Arc	hasTravelCost	only TravelCost
transportation:Network	hasTravelCost	only TravelCost
DistanceFee	subclassOf	TravelCost
	forDistance	only om:length
	travelCostOf	only transportation:Arc
AccessFee	subClassOf	TravelCost
	travelCostOf	only transportation:Network

Property	Characteristic	Value (if applicable)
travelCostOf	inverseOf	hasTravelCost

Ontologies Reused:

- iCity-Transportation Network

6.10 Parking Ontology

https://w3id.org/icity/Parking.owl

Namespace: parking

- Parking Area: Parking Area refers to some area that enables parking of Vehicles.
  A Parking Area may contain **sub-Parking Areas**, the area of which may change.
  A Parking Area has some Parking **Policy**
  A Parking Area may provide **car changing** stations.
  A Parking Area has some **Location**.
  A Parking Area has some vacancy (or occupancy) at some point in time.
  A Parking Area may be associated with some Building or Organization,…
  A Parking Area may have some hours of operation.

A Parking Area may have some limit on the dimensions of allowed vehicles (height/width/length)
associated location information (e.g. nearby crossroads, landmark, etc)
Different types (subclasses) of Parking Area may be defined as required, such as Street Parking Area, Lot Parking Area, Garage Parking Area, Illegal Parking Area, Loading/Unloading Zone Parking Area,…

- Parking Space: A Parking Space is a Parking Area with the capacity for a single vehicle. (hasCapacity 1, hasVacancy 0 or 1). Specializations of parking space may be defined based on accommodated vehicle type (e.g. small vehicles, commercial vehicles, electric vehicles,…).
  has Reservations?
  has Schedules?
  A Parking Space may or may not be occupied by some vehicle at a particular point in time. If a space is occupied, its availability may be determined (or approximated) based on the scheduled/purchased time by its current occupant.
- Accessible Parking Space: A type of parking space reserved for users with disabilities
- EV Space: A type of parking space that provides access to some EV Charger(s)
- Parking Policy: A Parking Policy dictates under what terms some Parking Area is accessible for parking.
  A Parking Policy may have a **Rate**.
  A Parking Policy may have a **max duration**.
  A Parking Policy may have **allowable periods** *(these periods must be during the hours of operation of the parking area)*.
  A Parking Policy may apply only to a particular class of users.
- Rate: A Rate has a **monetary value** and an **associated duration**.
  A Rate has a **ParkingPaymentMethod** (e.g. mobile, license plate entry, cashier, meter).
- EV charger: A charger for electric vehicles is an amenity which may be provided by some parking spaces.
  An EV charger has some model and is capable of charging certain classes of vehicles.
  An EV charger may be available or unavailable at a given time. This availability may be predetermined based on the scheduled duration of a vehicle's occupancy, and the time left to charge the vehicle.

Object	Property	Value
ParkingAreaPD	subclassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some ParkingArea and change:hasManifestation only ParkingArea
	change:existsAt	exactly 1 time:Interval
	spatial_loc:hasLocation	exactly 1 spatial_loc:SpatialFeature
	spatial_loc:hasAssociatedLocation	only spatial_loc:SpatialFeature
	maxAdmittableHeight	exactly 1 om:length
	maxAdmittableWidth	exactly 1 om:length
	maxAdmittableLength	exactly 1 om:length
	has Address	only icontact:Address
ParkingArea	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some ParkingAreaPD and change:manifestationOf only ParkingAreaPD
	change:existsAt	exactly 1 time:TemporalEntity
	mereology:hasProperPart	only ParkingArea

	hasVehicleCapacity	only (CapacitySize and gci:cardinality_of only (gci:defined_by only Vehicle))
	hasParkingPolicy	only ParkingPolicy
	hasChargingStations	exactly 1 xsd:integer
	hasOwner	some Person or Organization
	occupiedBy	only Vehicle
	isOpen	exactly 1 xsd:boolean
	hasParkingService	only ParkingService
ParkingSpace	subclassOf	ParkingArea
	hasVehicleCapacity	some (om:hasValue some (om:has_numerical_value value 1))
AccessibleSpace	subclassOf	ParkingSpace
	hasParkingPolicy	only AccessibilityParkingPolicy (to define)
EVSpace	subclassOf	ParkingSpace
	hasParkingPolicy	only EVParkingPolicty (to define)
ParkingService	*may be defined in greater detail in the future	
Valet	subclassOf	ParkingService
Carwash	subclassOf	ParkingService
ParkingPolicy	hasParkingRate	only ParkingRate
	maxDuration	only time:DurationDescription
	allowableDuring	only time:TimePeriod or time:CalendarPeriod
	appliesTo	only person:Person
	appliesFor	only vehicle:VehicleType
	hasGracePeriod	max 1 time:DurationDescription
	excludesPublicHoliday	exactly 1 xsd:boolean
ParkingRate	hasMonetaryCost	only monetary:MonetaryValue
	forDuration	only time:DurationDescription
	hasPayment	only ParkingPaymentMethod
	appliesTo	only person:Person

Ontologies Reused:

- iCity-Foundational
- iCity-Person
- iCity-Vehicle
- icontact.owl

6.11 Public Transit Ontology

[*https://w3id.org/icity/PublicTransit.owl*](https://w3id.org/icity/PublicTransit.owl)

Namespace: transit

- TransitSystem: A TransitSystem is a **collection of Routes**.
  A TransitSystem may be **accessed by** some Fare or Transit Pass.
- Route: A Route consists of a series of Route Links and may contain larger Route Sections.
  A Route has some directionality (captured by the route links).

- Route Section: A Route Section is part of some Route and **consists of** Route Links.
  A Route Section begins and ends at a Stop Point.
- Route Link: A Route Link **begins and ends** at a Stop Point.
  A Route Link **operates on** an Arc.
- Stop Point: A Stop Point marks the **start or end of** a Route Link (e.g. a subway stop or bus stop).
  A Stop Point is a subclass of a Node, as defined in the Transportation System ontology.
  Like a Node, a Stop Point **has an associated Location**.
  A Person may enter or exit the transit vehicle at a Stop Point.
  (to do: Station subclass of StopPoint)
- Transit Incidents, broadly, are events of interest that occur on a particular transit trip. Typically, they are problematic, unplanned issues resulting in some delay.
  A TransitIncident is a subclass of Activity.
  It is associated with some station or stop point.
  An incident may be described (and so classified) by a predefined code: hasCode only xsd:String
  An incident may cause some delay to the planned transit trip. Future extensions will define the data property with respect to the activity's caused states; the definition should relate the incident activity and the affected transit trip activity(s).
  An incident also has information regarding its "caused gap".
- TransitTrip is a subclass of Trip.
  Transit Trips have specific restrictions and specialized properties. A Transit Trip occurs on some predefined route. A Transit Trip may also describe some smaller part of a Route, i.e. a Route Link. The start and destination of a Transit Trip must be a Stop Point, and all Transit Trips must be performed with a Transit Vehicle.
  TBD: a Block may be a subclass of TransitTrip
- TransitVehicle is a subclass of Vehicle.
  A TransitVehicle has a transit vehicle id. This refers to the identifier assigned by the transit authority, as opposed to a serial number.
  Transit Vehicles are owned and operated by some transit authority. There are specialized types of transit vehicles (e.g. different types of streetcars), and a restricted set of modes. Transit Vehicles typically only operate on pre-defined routes, however there are exceptions (e.g. detours, travel for maintenance, etc).
- AccessMethod: An Access Method is the means of **access to** a Line
  An AccessMethod **has a Monetary Value**.
  An AccessMethod may be **valid for** a specific distance or time.
- RouteTimetable: A Timetable represents schedule information for a particular Route, or Route Link.
  A RouteTimetable has an **expected travel time (Duration)** for the Route, or Route Link.
- A StopTimetable has an **expected arrival time (Time Instant)** for some Stop Point.

Object	Property	Value
TransitSystemPD	subclassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some TransitSystem and change:hasManifestation only TransitSystem
	change:existsAt	exactly 1 time:Interval
TransitSystem	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some TransitSystemPD and change:manifestationOf only TransitSystemPD
	change:existsAt	exactly 1 time:TemporalEntity
	hasRoutes	only Route
	accessBy	only AccessMethod

AccessMethod	hasMonetaryCost	only monetary:MonetaryValue
	validFor	only (time:DurationDescription or om:length)
Fare	subclassOf	AccessMethod
TransitPass	subclassOf	AccessMethod
RoutePD	hasRouteId	exactly 1 RouteId
	subClassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some Route and change:hasManifestation only Route
	change:existsAt	only time:Interval
Route	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some RoutePD and change:manifestationOf only RoutePD
	change:existsAt	only time:TemporalEntity
	hasSection	only RouteSection
	mereology:contains	only RouteLink
RouteSection	mereology:contains	only RouteLink
	beginsAtStop	exactly 1 StopPoint
	endsAtStop	exactly 1 StopPoint
RouteLink	beginsAtStop	exactly 1 StopPoint
	endsAtStop	exactly 1 StopPoint
StopPoint	subclassOf	transport:Node
	spatial_loc:hasLocation	exactly 1 spatial_loc:SpatialFeature
	transit:hasStopCode	exactly 1 xsd:string
	foaf:name	min 1 xsd: string
	transit:wheelchairBoarding	exactly 1 {0,1,2}
AccessibleStopPoint	equivalentClass	StopPoint and transit:wheelchairAccessible value 1
RouteTimetable	timetableFor	exactly 1 Route or RouteLink
	arrivalTime	exactly 1 time:DateTimeDescription
	travelTime	exactly 1 time:DurationDescription
StopTimetable	timetableFor	exactly 1 StopPoint
	arrivalTime	min 1 time:DateTimeDescription
TransitIncident	subclassOf	activity:Activity
	associatedWithLoc	only StopPoint
	hasIncidentCode	min 1 xsd:string
	causedDelay	only xsd:duration
TransitTrip	subclassOf	trip:Trip
	trip:occursOn	only Route or RouteSection or RouteLink
	trip:viaVehicle	exactly 1 TransitVehicle
TransitVehicle	subclassOf	vehicle:Vehicle
	hasTransitVehicleId	exactly 1 xsd:string

Ontologies Reused:

- iCity-Foundation
- iCity-TransportationSystem
- iCity-Activity
- iCity-Trip

6.12 ***Land Use Ontology***

https://w3id.org/icity/LandUse.owl

Namespace: landuse

- Parcel: A Parcel is a way of defining some area in an urban system.
  A Parcel **has a Location**.
  A Parcel may be classified as **having some type of Land Use**.
  There may be other types (**subclasses**) of Parcel, defined in more precise or different ways, such as a Zone.
  A Parcel may have some associated Area. This is currently a variant property and we have yet to determine whether this is equivalent to the area of the Geometry of the Parcel's location (e.g. there may be various values with different accuracy from different sources).
  A Parcel may have some population that is subject to change over time.
  A Parcel may have a number of employed residents that is subject to change over time.
  Future work will look at extending the SpatialLoc Ontology (specifically the Geometry class) with a representation of various measurements such as surface area, volume, length, and so on.
  Future work will also extend population representation to captured defined_as, residence, and other restrictions.
- LandUseClassification: Land Use Classifications provide a means of describing the land cover/use in a standard way. Various classification systems are used to identify types of land use. Currently, we include LBCS, CLUMP, and AAFC.
- The LBCS recognizes different dimensions of Land Use: Activity, Function, Structure, Site, and Ownership Classifications. Each dimension is further defined by a taxonomy of specialized classifications. For each dimension, we introduce an equivalent class name for disambiguation, e.g. to distinguish between the Activity dimension of land use (we refer to this as ActivityClassification) and the notion of an Activity in icity.
  - Activity Classification: An Activity Classification identifies the **activity use** of some Land Parcel.
    - Residential Activities
    - Shopping Activities
    - Industrial Activities
    - ...
  - Function Classification: A Function Classification identifies the **economic function of** some Land Parcel,
  - Structure Classification: A Structure Classification identifies the **type of structure(s) on** some Land Parcel.
  - Site Classification: A Site Classification identifies the **state of the site development on** some Land Parcel (e.g. is it developed or not?)
  - Ownership Classification: An Ownership Classification identifies any **constraints on the use of the land and its ownership for** some Land Parcel.
- CLUMPClassification: Canada Land Use Monitoring Program Classification is a type (subclass) of Land Use classification. CLUMP identifies 15 different types of land use, each with an associated code used in datasets. We have made the design decision that the code need not be unique to a particular land use classification, as a classification from one system may correspond to multiple classifications in CLUMP. CLUMP introduces the following land use classifications:
  - B - Urban built-up area
  - E - Mines, quarries, sand and gravel pits
  - O - Outdoor recreation
  - H - Horticulture
  - G - Orchards and vineyards

- A - Cropland
  - P - Improved pasture and forage crops
  - K - Unimproved pasture and range land
  - T - Productive woodland
  - U - Non-productive woodland
  - M - Swamp, marsh or bog
  - S - Unproductive land - sand
  - L - Unproductive land - rock
  - 8 - Unmapped areas (technically not a CLUMP classification but it is used in the land use data)
  - Z - Water areas (technically not a CLUMP classification but it is used in the land use data)
- AAFCClassification: Agriculture and Agri-Foods Canada Classification is a type (subclass of) land use classification. The codes are based on the IPCC (International Panel on Climate Change) protocol. We have made the design decision that the code need not be unique to a particular land use classification, as a classification from one system may correspond to multiple classifications in AAFC. AAFC uses the following land use classifications:
  - Unclassified
  - Settlement
  - Roads
  - Water
  - Forest
  - Forest Wetland
  - Trees
  - Treed Wetland
  - Cropland
  - Grassland Managed
  - Grassland Unmanaged
  - Wetland
  - Wetland Shrub
  - Wetland Herb
  - Other land
- TrafficZone: traffic zone is a kind of (subclass of) Parcel. It may be identified with a predefined set of identifiers, corresponding to its centroid node ID.

Object	Property	Value
ParcelPD	subclassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some Parcel and change:hasManifestation only Parcel
	change:existsAt	exactly 1 time:Interval
	spatial_loc:hasLocation	exactly 1 spatial_loc;SpatialFeature
lbcs:Parcel	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some ParcelPD and change:manifestationOf only ParcelPD
	change:existsAt	exactly 1 time:TemporalEntity
	hasLandUse	min 1 LandUseClassification
	associatedArea	only om:area
	hasPopulation	only Population
ResidentPopulation	subclassOf	govstat:Population

EmployedPopulation	subclassOf	ResidentPopulation
LBCSClassification	subclassOf	LandUseClassification
ActivityClassification	subclassOf	LBCSClassification
	equivalentClass	lbcs:Activity
FunctionClassification	subclassOf	LBCSClassification
	equivalentClass	lbcs:Function
StructureClassification	subclassOf	LBCSClassification
	equivalentClass	lbcs:Structure
SiteClassification	subclassOf	LBCSClassification
	equivalentClass	lbcs:Site
OwnershipClassification	subclassOf	LBCSClassification
	equivalentClass	lbcs:Ownership
CLUMPClassification	subclassOf	LandUseClassification
	equivalentTo	hasCLUMPCode min 1 xsd:string
AAFCClassification	subclassOf	LandUseClassification
	equivalentTo	hasAAFCCode min 1 xsd:string
Unclassified	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "11"
Settlement	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "21"
Roads	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "25"
Water	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "31"
Forest	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "41"
ForestWetland	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "42"
Trees	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "45"
TreedWetland	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "46"
AAFCCropland	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "51"
GrasslandManaged	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "61"
GrasslandUnmanaged	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "62"
Wetland	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "71"
WetlandShrub	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "73"
WetlandHerb	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "74"

OtherLand	subclassOf	AAFCClassification
	equivalentTo	hasAAFCCode value "91"
UrbanBuiltUp	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "B"
MinesQuarriesSandGravelPits	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "E"
CLUMPCropland	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "A"
CLUMPWater	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "Z"
Horticulture	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "H"
ImprovedPasture	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "P"
NonProductiveWoodland	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "U"
OrchardsVineyards	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "G"
OutdoorRecreation	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "O"
ProductiveWoodland	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "T"
SwampMarshBog	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "M"
UnimprovedPasture	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "K"
Unmapped	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "8"
UnproductiveRock	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "L"
UnproductiveSand	subclassOf	CLUMPClassification
	equivalentTo	hasCLUMPCode value "S"

Reused Ontologies:

- lbcs: Land Based Classification Standards (LBCS) Ontology[16] presented by (Montenegro, Gomes, Urbano, & Duarte, 2011).
- iCity-Foundation

6.13 Trip Ontology

https://w3id.org/icity/Trip.owl

Namespace: trip

- Trip: A Trip describes the movement of a Person(s) from one location to another **via some** Mode(s).
  A Trip **starts at** some **Location** and **ends at** some **Location**.
  A Trip **occurs during** some Interval.

[16] Not available online

A Trip **occurs in** some Network(s).
A Trip **occurs via** some Arc(s).
A Trip **occurs on** some Transportation Complex. (e.g. a road or a rail)
A Trip contains some Trip Segments.
A Trip may incur some cost (monetary or otherwise).

- A Trip Segment describes part of a trip. It may be used, for example, to identify different parts of a the Trip by Mode.
  The restrictions on the Mode and possibly Vehicle used will become more complicated as we begin to incorporate restrictions based on a Persons access to a vehicle (age, household).
  A Trip Segment is **part of** some Trip.
  A Trip Segment **occurs during** some Interval.
  A Trip Segment **occurs in** some Network(s).
  A Trip Segment occurs via some Arc(s).
  A Trip occurs on some Transportation Complex.
  A Trip Segment may incur some cost (monetary or otherwise).
- Tour: A sequence of Trips made by one Person.
  A Tour **starts and ends at** the same Location.

Object	Property	Value
Trip	mereology:containedIn	exactly 1 Tour
	subclassOf	activity:Activity
	startLoc	only spatial_loc:SpatialFeature
	endLoc	only spatial_loc:SpatialFeature
	~~during~~	~~exactly 1 time:Interval~~
	accessesNetwork	min 1 transportation:Network
	accessesArc	min 1 transportation:Arc
	occursOn	min 1 transportation:TransportationComplex
	hasMode	min 1 vehicle:Mode?
TripSegment	subclassOf	Trip
	mereology:containedIn	exactly 1 Trip
	during	exactly 1 time:Interval
	startLoc	only spatial_loc:SpatialFeature
	endLoc	only spatial_loc:SpatialFeature
	viaMode	~~min~~ exactly 1 vehicle:Mode
	viaVehicle	only vehicle:Vehicle
	accessesNetwork	min 1 transportation:Network
	accessesArc	min 1 transportation:Arc
	occursOn	min 1 transportation:TransportationComplex
Tour	mereology:contains	min 1 Trip
	startLoc	only SpatialThing
	endLoc	only SpatialThing
	during	only time:Interval
	accessesNetwork	min 1 transportation:Network
	accessesArc	min 1 transportation:Arc
	occursOn	min 1 transportation:TransportationComplex

Reused Ontologies:

- iCity-TransportationSystem

- iCity-Vehicle

6.13.1 Trip Costs

https://w3id.org/icity/TripCost.owl

Namespace: tripcost

Different costs are associated with the performance of Trips. These may take the form of direct costs such as those presented in the Travel Cost Ontology, but there may be non-monetary costs associated with travel over different arcs such as pollution and travel time. Trip Costs capture these indirect costs that may vary between individual trips; a trip cost is a property of some instance of travelling.

- A Duration Cost is a Trip Cost.
  A duration cost has an associated cost in terms of duration; e.g. the length of time to perform the trip or trip segment
  A duration cost may have an associated monetary cost (valuation); e.g. the monetary cost applied to the length of time taken to perform the trip or trip cost.
- A Distance is a Trip Cost
  A distance has an associated cost in terms of the distance travelled.
  It may also have an associated monetary cost (valuation)
- An Environmental Cost is a Trip Cost
- A Vehicle Cost is a Trip Cost

Object	Property	Value
TripCost	hasMonetaryCost	only om:MonetaryValue
	tripCostOf	only (trip:Tour or trip:Trip or trip:TripSegment)
DurationCost	subclassOf	TripCost
	hasDurationCost	only time:DurationDescription
DistanceCost	subclassOf	TripCost
	hasDistanceCost	only om:length or om:MonetaryValue
EnvironmentalCost	subclassOf	TripCost
	hasEnvironmentalCost	only CarbonEmissions
VehicleCost	subclassOf	TripCost

Reused Ontologies:

- iCity-Trip

6.14 Urban System Ontology

https://w3id.org/icity/UrbanSystem.owl

Namespace: urbansys

Earlier in this report, we recognized that the urban system covers many different concepts, thus motivating the design of the preceding, so-called generic ontologies. However, it must be recognized that in isolation, these concepts do not effectively capture the urban system. The urban system not only includes these concepts, but relationships between them. For example, the relationship between its population and trips taken and vehicles used. The Urban System Ontology extends all of the previously defined ontologies in order to capture the relationships between them, in the context of the urban system.

- A Person may be a **member of** a Family and/or a Household.
  A Person may work for another Person, or some Organization.

A Person may **have access to** some **Vehicle**.
A Person may **have access to** some **Bicycle**.
A Person may **have** some **TransitPass**.
A Person has a **Schedule** for a given point (period) in time.

- A Schedule is a plan for some Activity to occur at/over some point in time.
- A Family **has members** who are Persons, and who are related via the has-spouse or has-child properties.
- A Household has one or more Persons as **members**. We do not make any commitment regarding the identity of the Persons, and in fact a Person may belong to more than one Household.
- A Dwelling Unit is **located in some Building** (e.g. House, Apartment,...)
- An Organization must have at least 2 Person(s) as members(s).
- A Firm or a Business Establishment may have a Person as an employee
- An Employee is a type of Person(s).
- Occupation: An Occupation **is performed by** some Person.
  An Occupation has a type (e.g. sales, skilled trades)
- A Building may be located on some Parcel of land (this is an invariant property of any building).
  A Building **has an owner**, which may be a Persons or some Organization.
  A Building **has occupants**, which may or may not be the same Persons or Firm who own it.
  A Building may **provide** some **Parking**.
- A Building Unit may be **occupied by** some Persons or Organization.
  A Building Unit may be **provide some Parking**.
- A Vehicle may be **occupied by** at least one Person, and some cargo.
  A Vehicle is **owned by** some Person(s) or Firm.
- Occupant: An occupant is a Person who is occupying a Vehicle during transit.
  An Occupant may be a Driver or a Passenger
- Cargo: A Cargo is some Thing that is not a Person and is occupying a Vehicle during transit.
- An *entire* Arc is accessible by a single set of Mode(s).
- A Road Segment is accessed by some Arc(s) with modes that are not water, air, or rail.
- A Parking Area has some **owner**.
  A Parking Area may be **occupied by** some Vehicle (however, it might also be occupied by some debris or activities such as construction).
- A Parking Policy may **apply to** a specific group of Persons or Organizations.
  A Parking Policy may have a **vehicle type restriction.**
- A TransitSystem may be **owned by** some Organization.
- A Route is **executed by** various Vehicles at different points in time.
- A Vehicle Block is a schedule assigned to some Vehicle for a given time period.
- A Trip is **made by** a Person to **facilitate participation in some Activity**.

Object	Property	Value
person:Person	memberOf	min 1 household:Family
	memberOf	min 0 household:Household
	schema:worksFor	some (person:Person or org:Organization)
	hasAccess	some (vehicle:Vehicle or Bicycle)
	hasPass	some transit:Pass
	hasSchedule	some Schedule
Schedule	hasActivity	only activity:Activity
	scheduledFor	exactly 1 time:Interval
household:Family	hasMember	only person:Person
household:Household	hasMember	min 1 (household:Family or person:Person)
household:DwellingUnitPD	locatedIn	some building:Building

org:Organization	org:hasOrgMember	min 2 person:Person
org:Firm	hasEmployee	only person:Person
org:BusinessEstablishment	hasEmployee	only person:Person
org:Employee	equivalentClass	person:Person and employedBy some (tove:Organization or person:Person)
Occupation	performedBy	some person:Person
	hasOccupationType	only OccupationType
building:BuildingPD	locatedOn	only landuse:Parcel
building:Building	hasOwner	min 1 (person:Person or org:Organization)
	hasOccupant	some person:Person or org:Organization or org:BusinessEstablishment
	hasParking	only parking:ParkingArea
vehicle:Vehicle	occupiedBy	only (Occupant or Cargo)
	hasOwner	only (person:Person or org:Organization)
Occupant	equivalentClass	person:Person and occupies some vehicle:Vehicle
Cargo	equivalentClass	not(person:Person) and occupies some vehicle:Vehicle
transport:ArcPD	hasMode	only vehicle:Mode
transport:RoadSegment	accessedBy	only transport:Arc and manifestationOf only (hasMode value water)
transit:TransitSystem	hasOwner	only org:Organization
transit:Route	executedBy	only vehicle:Vehicle
VehicleBlock	assignedTo	exactly 1 vehicle:Vehicle
	assignedFor	exactly 1 time:Interval
	hasSchedule	exactly 1 Schedule
trip:Trip	subClassOf	activity:ActivityOccurrence
	performedBy	some person:Person
	associatedWith	only activity:Activity

7 Applications

Beyond providing formalizing a vocabulary for an integrated, iCity knowledge base, the ontology may be employed more directly support applications for urban informatics. Three such examples are currently being explored: (1) ontology support for the IT-SoS framework, (2) ontology support for survey design and results storage, and (3) an ontology-based path finding tool. In this section we focus on the IT-SoS application and describe the progress made thus far. More detail will be added as each application is explored further.

7.1 iCity Ontology for the IT-SoS Framework

The IT-SoS framework proposed by [ref Elshenawy] enables a solution for the design of transportation applications capable of dynamically executing services based on a given context.

Using the framework, services can easily be added and integrated as part of applications as appropriate.
[more detail/diagram]
Ontologies play a key role in this framework. Here, we focus on the interface between the iCity ontology and the rest of the system. This interface must be well-understood and clearly defined in order to implement this framework in the context of the iCity project. In the following sections, we outline the functional requirements and describe the design of an interface to satisfy these requirements.

7.1.1 Ontology Interface: Functional Requirements

More detail on the complete framework can be found in [ref]. The role of the ontology and its interface with the rest of the system is specified with the following requirements (revised from the requirements identified in [ref thesis]):

- Ontology representation of services (WFS)
  - To support discovery
  - To support composition
- Ontology representation of applications
  - To enable (context-based) composition

The above requirements are formalized and elaborated with the following use cases:

1. A user formalizes the relevant semantics of a new WFS service for the ArcGIS server, to be part of the IT-SoS framework.
2. Given a process definition and some contextual information, the system identifies the information requirements required to perform it.
3. Given an information requirement, identify which services are capable of providing the required information.

Use Case 1	A user formalizes the relevant semantics of a new WFS service for the ArcGIS server.	
Goal in Context	Users create new WFS services for the ArcGIS server, to be incorporated into the IT-SoS framework.	
Preconditions	WFS service designed correctly.	
Success End	WFS service created, formalized with metadata.	
Failure End	WFS service not created and/or metadata not captured	
Primary & Secondary Actors	Primary: user	
Trigger	New WFS service created.	
Description	**Step**	**Action**
	1.	User creates new WFS service.

	2.	User defines additional required metadata for service.
	3.	User submits the service to IT-SoS.
	4.	Service description formalized in the language of the ontology.
Extensions	**Step**	**Branching Actions**
Variations	**Step**	**Branching Actions**
	2a.	No sources are capable of supplying all of the required data 2a1. System or user notified query is unknown...
Related Information		
Priority	High	
Performance	TBD	
Frequency	High	
Open Issues	What metadata is required?	

Use Case 2	Given a process definition and some contextual information, the system identifies the information requirements required to perform it.	
Goal in Context	In order to perform automatic and dynamic composition of applications, the system must be able to determine what information is required in a particular context.	
Preconditions	Application process correctly formalized. Contextual information available	
Success End	All and only the correct information requirements are identified.	
Failure End	Information requirements not (fully or exclusively) identified.	
Primary & Secondary Actors	Primary: application composer	
Trigger	Application execution.	
Description	**Step**	**Action**
	1.	Application process selected.
	2.	Contextual information input.

	3.	Information requirements to support application process generated.
Extensions	**Step**	**Branching Actions**
Variations	**Step**	**Branching Actions**
Related Information		
Priority	High	
Performance	TBD	
Frequency	High	
Open Issues		

Use Case 3	Given an information requirement, the system identifies which services are capable of providing the required information.	
Goal in Context	In order to perform automatic and dynamic composition of applications, the system must be able to determine which services are capable of satisfying which information requirements.	
Preconditions	WFS services correctly formalized. Application process correctly formalized.	
Success End	All and only the required services are identified.	
Failure End	Required services not (fully or exclusively) identified.	
Primary & Secondary Actors	Primary: application composer	
Trigger	Application execution.	
Description	**Step**	**Action**
	1.	User provides a set of information requirements.
	2.	WFS services capable of satisfying the information requirements are identified.
Extensions	**Step**	**Branching Actions**
Variations	**Step**	**Branching Actions**

Related Information		
Priority	High	
Performance	TBD	
Frequency	High	
Open Issues		

7.1.2 System Design

Illustrated in Figure 8, the creation of ITS applications using the IT-SoS framework is supported within the iCity project with three key components: the WFS creator, which supports the definition of individual services; the Application creator, which supports the definition of applications as processes; and the Application engine, which supports the automated composition and execution of services.

The WFS Creator (illustrated in Figure 9) supports the creation of a WFS service from some existing data source. The WFS is created and published to the ArcGIS server. Additional metadata is also collected during the creation process in order to achieve a complete description of the service (e.g. the equipment it uses, the information it provides). This description is mapped into a formal representation using terminology defined in the iCity ontology, and stored in a triplestore which may be accessed as required by other components.

The Application Creator component supports the definition of ITS applications – in particular the context-specific decomposition of the underlying process (including accounting for possible dependencies between sub-processes). As with the creation of WFS services, each application definition shall be mapped into an ontology-based representation in the IT-SoS triplestore.

The iCity ITS Application Engine (illustrated in Figure 10) is responsible for building and executing ITS applications on-the-fly, according to the definitions specified by the ITS Application Creator.
In the context of the iCity project, each application shall be accessible (buildable and executable) via the iCity dashboard. Based on the selected application and additional contextual information (supplied by the dashboard), the engine queries the triplestore for the appropriate composition in order to determine how the application shall be built. The resulting composition then serves as input for the execution engine, which combines the WGS services from the ArcGIS server in accordance with the prescribed composition in order to execute the application. Since all of the WFS services are represented using the iCity ontology, information between services may be combined easily, using the ontology as the interlingua.

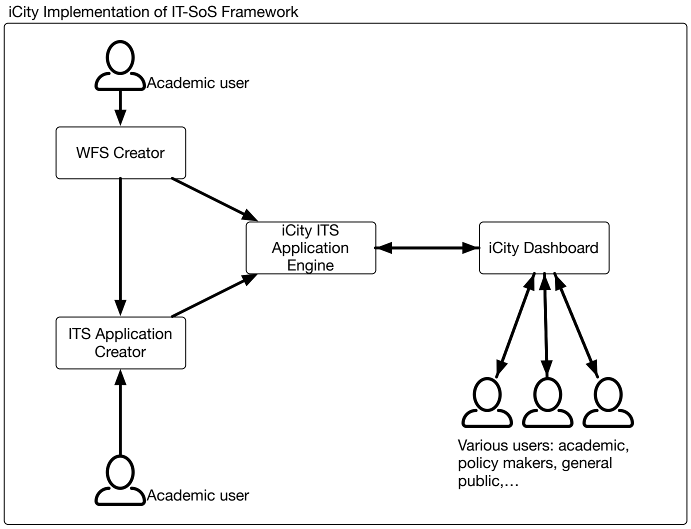


Figure 8: Ontology-focused view of the iCity implementation of the IT-SoS framework.

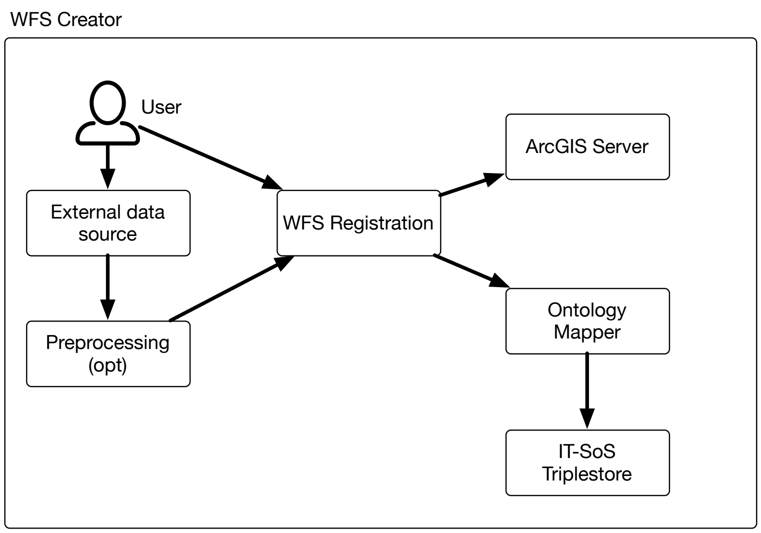


Figure 9: Illustration of the key components within the WFS Creator.

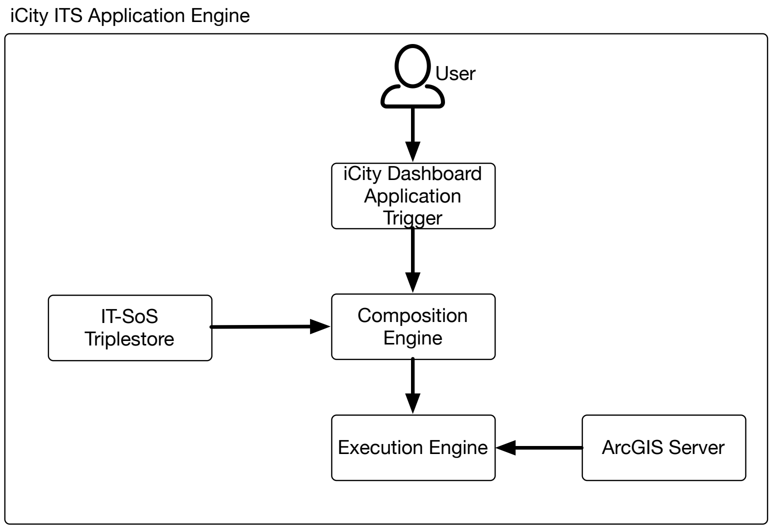


Figure 10: Illustration of the key components within the ITS Application Engine.

7.1.3 Ontology Design: Required Extensions

The iCity ontology must be extended in order to capture WFS and their possible compositions. This will be approached by formalizing some sample application definitions and compositions in order to guide the metadata that will be required to support the system's functionality.

7.1.4 Next Steps

The next steps toward implementation of this application will be to clearly define the required, ontology-based representation of applications and services within the system. This will be incorporated into a workflow that will become part of the system's processes to add applications and services. In this way, the ontology interface to the IT-SoS system will be baked-in to the creation process, requiring no additional overhead on the part of the user. The system shall be implemented to take advantage of the ontology-based representation in order to support the automated, intelligent composition of applications as envisioned by the IT-SoS framework.

8 Future Work

The iCity Ontology, presented in the previous section, has been classified with the Hermit reasoner in Protegé 5.1[17] and shown to be consistent. An initial, informal evaluation has been performed through a review of its contents with iCity project members serving as domain experts. Future iterations shall be informed by and evaluated against a more precisely defined series of competency questions to be elicited from the iCity project team.

Future iterations of the iCity Ontology will develop a deeper semantics for the concepts identified here, in addition to an expansion of scope. This will be dictated largely by use cases identified by the various project groups, which will not only determine additional requirements for representation, but potential applications for additional functionality that may be supported by the ontology.

[17] http://protege.stanford.edu/

8.1 Extensions to the Urban System Ontology

In developing a richer semantics for the iCity concepts, we will also look to identify more detailed connections between them. This will serve to facilitate shareability between the various projects and domains within iCity. Consider for example, the identification of relationship between common property types, such as hasId, memberOf. While there is likely a shared semantics between these relations in, for example the Person/Family and the Organization ontology, in this initial release, we opt to maintain a distinction between these relations (through specialized names, e.g. personId). Future work should, if required, investigate and make explicit exactly what the relationship is.

In a similar vein, future work will also look to integration of the iCity ontology with other existing vocabularies, which may provide opportunities to improve its shareability. For example, in the design of the iCity ontology we identified some vocabularies that were not directly reusable, (specified as XML schemas, for example), however based on their applications, it might be advantageous to incorporate the representations in some way. For example, GTFS [18], the format used by Google for travel information.

8.2 Extensions for iCity Applications

The first release of the iCity ontology is designed to capture the urban system. However, we anticipate additional concepts will be required for each iCity project to capture the nature of the data within a given application. Varying definitions of concepts within the urban system should be captured as part of the appropriate ontology (for example, multiple definitions of a Household should be represented by different definitions of Household in the Household ontology), on the other hand the iCity projects also introduce other concepts that are beyond the domain of the urban system, and more related to the applications themselves. For example, a simulation may produce output that captures information about an urban system, but we must also represent that this information is the result of a particular model being applied to some data to explain how it was generated and why it is of interest. We divide the iCity projects into 4 categories based on the nature of the applications: Data Collection, Simulation, Analysis, and Visualization. In the following subsections, we consider the classes and properties for each extension. The resulting structure for this future state of the iCity Ontology is illustrated inFigure 11.

[18] https://developers.google.com/transit/gtfs/

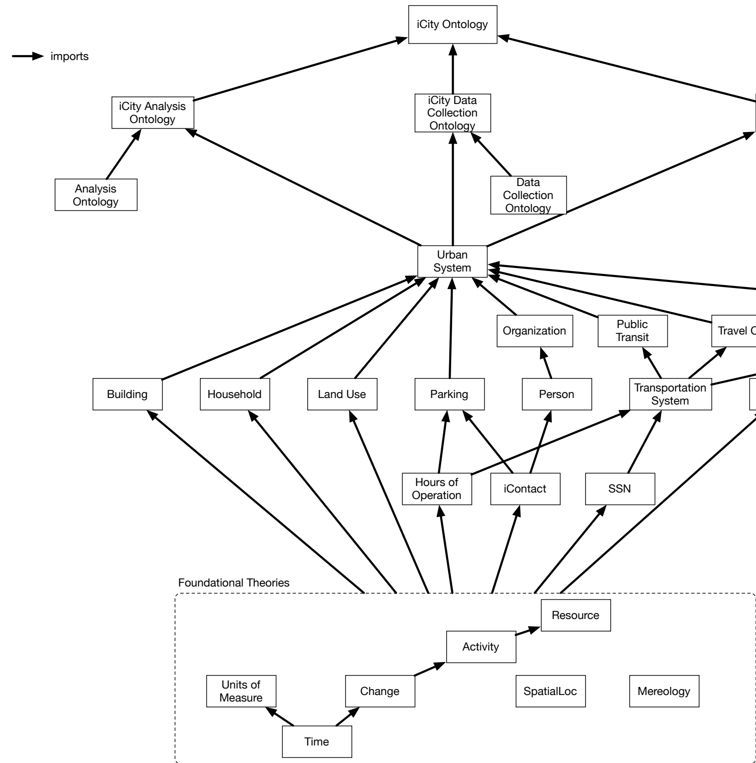


Figure 11: iCity Ontology Structure

In identifying these concepts, a key question is: "What question(s) is the project/application trying to answer?"
Note that it is unclear whether or to what degree there may be some overlap between the requirements for Analysis and Simulation in that they both require some aspect of experiment management. This report concludes with some preliminary notes on the requirements for each category of application in the following sections.

8.2.1 Data Collection

Related projects: 1.2, 1.3, 2.1, 2.2, 2.3

To completely capture collected data requires representation of its origin: what was the means of collection? When was it collected? How may the data be accessed? It requires the representation of concepts *about* the data collection itself. The following additional concepts may be required for the data collection extension:

- Data Entity: A Data Entity refers to some instance that is defined within the urban system, according to some source.
  A Data Collection is a type of (**subclass of**) Data Entity.
  A Data Collection **contains** one or many Data Entities.
  A Data Entity **is generated by** some Collection Activity.
  A Data Entity may be found **at some Location**.
- Data Entity: A Data Entity is any instance **contained in** some Dataset.
- Collection Activity: A Collection Activity indicates the origin of the data; i.e. how was it collected?

A Collection Activity **starts** and **ends** at some Time
There are different types (**subclasses**) of Collection Activity: Survey Activity, Sensor Activity, Data Fusion Activity, Simulation Activity, etcetera.
A Collection Activity may be found **at some Location** (e.g. location of the sensor or survey, could be physical or virtual).

- Data Fusion: A Data Entity may be the result of the Fusion of two or more Data Collections.
  Data Fusion **is informed by** at least 2 Collection Activities.
- Data Collection Agent: The agent responsible for some Collection Activity.
  A Collection Activity may be **associated with** some Data Collection Agent.
  A Data Entity may be **attributed to** some Data Collection Agent.

8.2.2 Simulation of Urban Systems

Related projects: 2.2, 2.3, 2.4

Capturing the simulation activities that occur within the iCity project, at this stage, appears to be very much an effort of experiment management. We need to be able to represent the simulation runs that are performed -- but also, more specifically the model(s) that was used, as well as the results that were obtained. The following additional concepts may be required for the Simulation extension:

- Simulation: A Simulation is an execution of some Model System.
  A Simulation **executes** some Model System.
  A Simulation has some **input** and **output** Dataset(s)
  A Simulation has an **initial** State, **sequence** of States, and **final** State.
  A Simulation has a **run date** and **duration**.
- State: A State is **comprised of** some instantiation of (part of) the urban system, at some specified point in time.
- Model System: A Model System is some configuration of model(s) that has been designed for simulation.
  A Model System **contains** some Model(s)
  A Model System may contain rules for how the Model(s) interact. (sequentially, in parallel, etcetera).
- Model: A Model is a means of advancing some current state within a Simulation.
  A Model **applies to** some classes in the domain.
  There are different types (**subclasses**) of Models, identified based on their perspective: State-oriented Model, Event-oriented Model, Activity-oriented Model, PD-oriented Model.
  A Model **has** some **Parameter**(s).
  A Model may **execute in parallel with** some other Model(s).
  A Model may **execute directly after** some other Model(s).
- State-oriented Model. There are different types (**subclasses**) of State-oriented Models that can be defined, according to the application.
  A State-oriented Model has some State Space
  A State-oriented Model has some Event Set
  A State-oriented Model has some Time Set
  A State-oriented Model has some Transition Function to transition between states.

A State-oriented Model has some Clock Function to advance "time".
A State-oriented Model has some Initial State.

8.2.3 Analysis of Urban Systems

Related projects: Project 1.2, 2.1, 2.2, 2.3, 2.4

Similar to the previous section, capturing the various analysis applications may be seen as a sort of experiment management. We must capture the concepts of analysis input, output, as well as the analysis itself: in other words, how is the output determined from the input? The following concepts may be required for the Analysis extension:

- Analysis: A set of rules or criteria applied to some Analysis Input to obtain some Analysis Output.
  An Analysis may take only certain class(es) of instances as Input.
  An Analysis will output only certain class(es) of instances as Output.
- Analysis Input: An Analysis Input is input for some Analysis.
- Analysis Output: An Analysis Output is output from some Analysis.

8.2.4 Visualization of Urban Systems

Related projects: All of Theme 3

The concepts defined in the iCity ontology (and the data they define) shall be interpreted for visual renderings; to-date no additional requirements have been identified.

9 Extra-logical Design Practices

Here, we summarize and explain the design practices that were adopted in the creation of the ontologies. These practices do not pertain to the semantic definitions, but rather are adopted to address pragmatic concerns regarding the organization and maintenance of the ontologies.

- Organizational terms
- For reuse (full import) of existing, external ontologies, e.g. owl-time. In order to create the required groupings under organizational subclasses, it is easiest to merge the imported ontology into the iCity container (e.g. icity/Time/). This allows for the addition of organizational subclass assertions (e.g. TemporalEntity subclassOf TimeOntologyThing) and also ensures that the appropriate version is captured/reused as a snapshot. This prevents any issues should versioning IRIs not be used by the ontology's author.

Acknowledgements

This project is supported by the Ontario Ministry of Research and Innovation through the ORF-RE program.

Bibliography

Bittner, T., & Donnelly, M. (2005). Computational ontologies of parthood, componenthood, and containment. *Proceedings of the 19th international joint conference on Artificial intelligence (IJCAI).* Morgan Kaufmann Publishers Inc.

Fadel, F. G., Fox, M. S., & Gruninger, M. (1994). A Generic Enterprise Resource Ontology. *Third Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises* (pp. 117-128). IEEE.

Fox, M. S., Barbuceanu, M., Gruninger, M., & Lin, J. (1998). An Organization Ontology for Enterprise Modelling. In K. C. M. Prietula, *Simulating Organizations: Computational Models of Institutions and Groups* (pp. 131-152). Menlo Park CA: AAAI/MIT Press.

Fox, M. S., Chionglo, J. F., & Fadel, F. G. (1993). A Common Sense Model of the Enterprise. *Proceedings of the 2nd Industrial Engineering Research Conference*, (pp. 425-429). Norcross GA.

Grau, B. C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., & Sattler, U. (2008). OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 309-322.

Hobbs, J. R., & Pan, F. (2004). An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing (TALIP)*, 66-85.

Krieger, H.-U. (2008). Where temporal description logics fail: Representing temporally-changing relationships. *Annual Conference on Artificial Intelligence* (pp. 249-257). Springer Berlin Heidelberg.

Lorenz, B., Ohlbach, H. J., & Yang, L. (2005). *Ontology of transportation networks.*

Montenegro, N., Gomes, J., Urbano, P., & Duarte, J. (2011). An OWL2 Land Use Ontology: LBCS. *International Conference on Computational Science and its Applications* (pp. 185-198). Springer Berlin Heidelberg.

Welty, C., Fikes, R., & Makarios, S. (2006). A reusable ontology for fluents in OWL. *Formal Ontology in Information Systems (FOIS)*, (pp. 226-236).