iCity Ontology Initial Release

Katsumi, Megan and Mark Fox

iCity: Urban Informatics for Sustainable Metropolitan Growth

A project funded by the Ministry of Research and Innovation of Ontario through the ORF-RE07 Program and by partners Cellint Traffic Solutions, Esri Canada, IBM Canada, Waterfront Toronto, the City of Toronto and the Region of Waterloo.

Report #17-01-01-01

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

UTTRI

iCity Ontology Initial Release

Prepared by: Megan Katsumi and Mark Fox

Prepared on: January 31, 2017

Contents

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.

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, Horrocks, Motik, Parsia, Patel-Schneider, & Sattler, 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 6.2 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 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)[1], created by mapping data from the iCity applications to the agreed-upon terminology defined in the iCity ontology. The architecture of the iCity project, relative to the ontology, is illustrated in **Error! Reference source not found.**.

[1] 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.

Beyond this, 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.

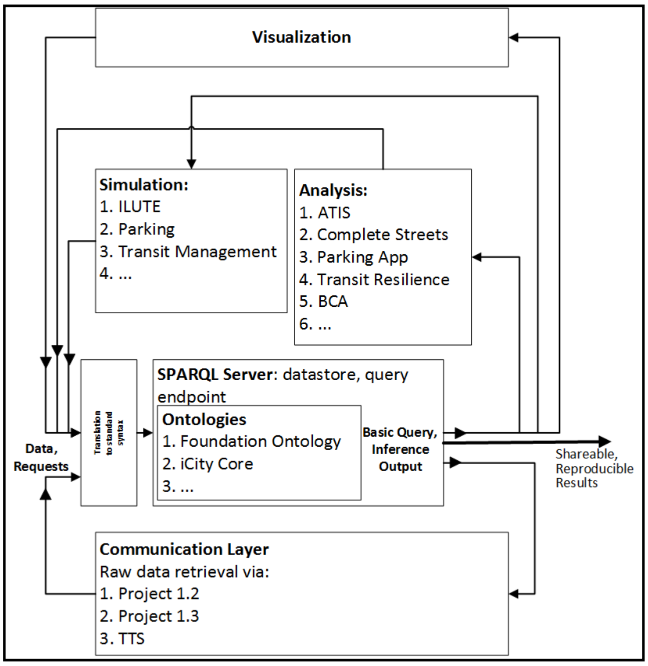


Figure 1: iCity Architecture

xThe sections that follow introduce the core ontology required to capture the iCity projects, in particular, to define the urban system.

5 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.

5.1 *Foundational Ontology (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.

5.1.1 Spatial Location Ontology (iCity-SpatialLoc.owl)

Namespace: spatial_loc

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. To achieve this, we combine an ontology of longitude and latitude, with simple spatial and geometry ontologies.

- Point: A Point in space.
  A Point has **a longitude**.
  A Point has **a latitude**.
  A Point has **an altitude**.
  Note that the inclusion of some other reference system is possible but not currently implemented.
- SpatialFeature[2]: A Spatial Feature is some Thing that exists in space. Many of the classes in the iCity ontology may be Spatial Features, such as Buildings, Vehicles, and Persons; even the more abstract notion of some location may also be defined as a SpatialFeature.
  A Spatial Feature **has some geometry** to describe its shape.
  A Spatial Feature may be qualitatively related to other Spatial features, specifically via the RCC-8 relations (e.g. contained in, part of,...)
- Geometry: Geometry describes the shape of some spatial thing.
  Geometry may **have a location** in space, defined by the points that it's comprised of.
  There are different types (subclasses) of Geometry: (2d) Polygon, (may be 2d or 3d) Multipolygon, Line, LinearRing, and Point.
  A Polygon is defined according to its **boundaries** (**exterior / interior**); these boundaries are specified as LinearRings.
  Point is defined according to WGS-84, so that the points that comprise the Geometries may be defined according to a coordinate system.

Object	Property	Value
wgs:Point	hasLatitude	exactly 1 om:Quantity
	hasLongitude	exactly 1 om:Quantity
	hasAltitude	exactly 1 om:Quantity
SpatialFeature	equivalentClass	spatial:Feature
	spatial:connectsWith	only SpatialFeature
	spatial:disconnectedFrom	only SpatialFeature
	spatial:externallyConnectedWith	only SpatialFeature
	spatial:equals	only SpatialFeature
	spatial:non-tangentialProperPartOf	only SpatialFeature
	spatial:overlaps	only SpatialFeature
	spatial:isPartOf	only SpatialFeature
	spatial:partiallyOverlaps	only SpatialFeature
	spatial:isPropertPartOf	only SpatialFeature
	spatial:tangentialPropertPartOf	only SpatialFeature
	geom:geometry	only geom:Geometry
geom:Geometry	geom:bbox	only geom:BoundingBox
geom:GeometryCollection	subclassOf	geom:Geometry
geom:LineString	subclassOf	geom:Geometry
	posList	only wgs:Point
geom:LinearRing	subclassOf	geom:LineString

[2] Renamed from spatial:Feature for clarity.

geom:MultiLineString	subclassOf	geom:GeometryCollection
	lineStringMember	only geom:LineString
geom:MultiPoint	subclassOf	geom:GeometryCollection
	pointMember	only wgs:Point
geom:MultiPolygon	subclassOf	geom:GeometryCollection
	polygonMember	only geom:Polygon
geom:Polygon	subclassOf	geom:Geometry
	boundary	only geom:LinearRing
	exterior	only geom:LinearRing
	interior	only geom:LinearRing

Property	Characteristic	Value (if applicable)
hasLocation	Ranges	SpatialFeature
hasAltitude	Ranges	om:Quantity
hasLongitude	Ranges	om:Quantity
hasLatitude	Ranges	om:Quantity

Reused Ontologies:

- wgs:WGS-84 Ontology[3]
  Note that this ontology was extended due to some discrepancies between the documentation and the axiomatization. In particular, the latitude and longitude are not specified as object or data properties but as annotation properties (which we cannot include in the axioms). The iCity Spatial ontology extends wgs in order to capture these concepts with the use of the Units of Measure ontology.
- geom: GeoVocab Geometry Ontology[4]
- spatial: GeoVocab Spatial Ontology[5]
- om: Units of Measure Ontology[6]

5.1.2 Time Ontology (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

[3] https://www.w3.org/2003/01/geo/wgs84_pos
[4] http://geovocab.org/geometry
[5] http://geovocab.org/spatial
[6] http://www.wurvoc.org/vocabularies/om-1.8/

	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

Reused Ontologies:

- time: OWL-Time Ontology[7] originally presented by (Hobbs & Pan, 2004)

5.1.3 Change Ontology (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[8]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.

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

[8] 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".

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
	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

5.1.4 Activity Ontology (iCity-Activity.owl)

Namespace: activity

- Activity: An Activity describes a type of Activity Occurrence.
  An Activity may or may not **occur as an ActivityOccurrence**.
  An Activity may be further defined by **(decomposed into) Subactivities**.
  An Activity may have **precondition and/or effect** State Types.
  An Activity may be **enabled by or cause** some State Types. An enabling of causing state is a generalization of a precondition/effect; an Activity is enabled by or causes some State Type if it has a subactivity with a precondition or effect (respectively) of that State Type.
  In other words, the state may not be required directly before, or cause directly after the activity, but by some more specialized sub-activity.
- ActivityOccurrence: An ActivityOccurrence is an **occurrence of** some Activity that **occurs at** some point in time and space. Note that the Activity and ActivityOccurrence relationship is *not the same* (though intuitively similar) to the relationship between a TimeVaryingConcept and its Manifestations; a manifestation is *part* of a time varying concept at a particular point in time, whereas an activity occurrence is an occurrence of the *entire* activity.
  An ActivityOccurrence takes place **during some interval**, and so **has** some **duration**.
  An ActivityOccurrence may have some Manifestations that participate in it.
- State Type: a state type 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 type may be complex and refer to some combination of classes of manifestations.
  A note on complex state types:
  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 State Types to capture aggregation, adopting the following approach used in the description of state trees in TOVE by (Fox, Chionglo, & Fadel, 1993).

A StateType 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.

NonTerminalStateType(x) v TerminalStateType(x) = StateType(x)

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

TerminalStateType disjointWith NonTerminalStateType

- A terminal state type 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 type may be conjunctive or disjunctive. Naturally, a conjunctive state type is defined by the conjunction of its child state types, whereas a disjunctive state type is defined by the disjunction of its child states.

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

A state type cannot be both conjunctive and disjunctive.

ConjunctiveStateType disjointWith DisjunctiveStateType

Conjunctive and disjunctive state types, which *do* have substates, are achieved at some time if their decomposition of state types 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 type. A more general relation that *is* transitive is the *substate* relation.

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

Object	Property	Value
Activity	hasOccurrence	only ActivityOccurrence
	hasSubactivity	only Activity
	hasPrecondition	only StateType
	enabledBy	only StateType
	hasEffect	only StateType
	causes	only StateType
ActivityOccurrence	occurrenceOf	only Activity
	occursAt	only time:TemporalEntity
	spatial_loc:hasLocation	only spatial_loc:SpatialFeature
	hasParticipant	only change:Manifestation
StateType	preconditionOf	only Activity
	enables	only Activity
	effectOf	only Activity
	causedBy	only Activity
	achievedAt	only time:TemporalEntity
TerminalStateType	subClassOf	StateType
	disjointWith	NonTerminalStateType
	stateTypeOf	only change:Manifestation
	hasDecomp	exactly 0 StateType
NonTerminalStateType	subClassOf	StateType
	disjointWtih	TerminalStateType
	hasDecomp	only StateType and min 2 StateType
	hasSubstate	only StateType
ConjunctiveStateType	subClassOf	NonTerminalStateType
	disjointWith	DisjunctiveStateType
DisjunctiveStateType	subClassOf	NonTerminalStateType
	disjointWith	ConjunctiveStateType

Property	Characteristic	Value (if applicable)
hasOccurrence	inverseOf	occurrenceOf
	Inverse Functional	-
hasSubactivity	Transitive	-
hasPrecondition	Domains	Activity
	Ranges	StateType
enabledBy	subPropertyOf	hasPrecondition
hasEffect	Domains	Activity
	Ranges	StateType
causes	subPropertyOf	hasEffect
occurrenceOf	inverseOf	hasOccurrence
	Functional	-
occursAt	Domains	ActivityOccurrence
	Ranges	time:TemporalEntity
hasParticipant	Domains	ActivityOccurrence
	Ranges	change:Manifestation
participatesIn	inverseOf	hasParticipant
preconditionOf	inverseOf	hasPrecondition
enables	inverseOf	enabledBy
effectOf	inverseOf	hasEffect
causedBy	inverseOf	Causes
achievedAt	Domains	StateType
	Ranges	time:TemporalEntity
stateTypeOf	Domains	TerminalStateType
	Ranges	Manifestation
hasDecomp	Domains	StateType
	Ranges	StateType
hasSubstate[9]	Domains	StateType
	Ranges	StateType
	subPropertyOf	hasDecomp

Reused Ontologies:

- iCity-Change
- iCity-SpatialLocation

5.1.5 Resource Ontology (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.

[9] Note that while we would like to specify the transitivity of the hasSubstate relation, we are limited by OWL due to the cardinality restriction on the hasDecomp relation (making it a non-simple property). For the present, we have removed the transitivity property in order to maintain the cardinality restriction.

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 *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:Quantity
	capacityInUse	only om:Quantity
	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:Quantity
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

Reused Ontologies:

- iCity-Activity

5.1.6 Mereology Ontology (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[10]	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

[10] 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.

Reused Ontologies:

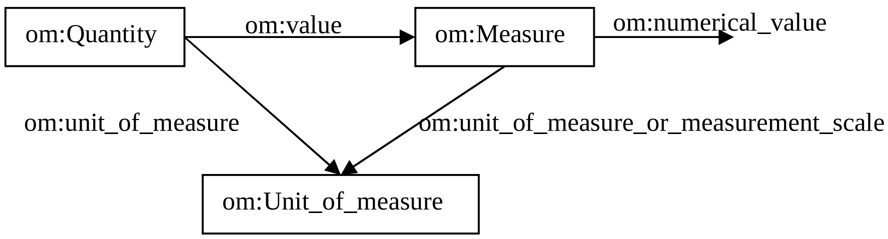
- None directly, but reused concepts as defined by (Bittner & Donnelly, 2005), however theirs is not an officially published ontology.

5.1.7 Ontology of Units of Measure (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, and may be associated with a Unit_of_measure or Measurement_scale
- A measured value (om:Measure) is associated with a unit of measure or measurement scale
- There are many types (subclasses) of units of measure, such as length, mass, speed, and currency.



Object	Property	Value
om:Quantity	om:value	some om:Measure
om:Measure	om:unit_of_measure_or_measurement_scale	some (om:Measurement_scale or Unit_of_measure)
om:Quantity	om:unit_of_measure	om:Unit_of_measure
om:Length_unit	subClassOf	om:Unit_of_measure
om:Mass_unit	subClassOf	om:Unit_of_measure
om:Amount_of_substance_unit	subClassOf	om:Unit_of_measure
om:Area_unit	subClassOf	om:Unit_of_measure
om:Volume_unit	subClassOf	om:Unit_of_measure
om:Acceleration_unit	subClassOf	om:Unit_of_measure
om:Speed_unit	subClassOf	om:Unit_of_measure
om:Currency	subClassOf	om:Unit_of_measure
...	subClassOf	om:Unit_of_measure
om:System_of_units	om:base_unit	some om:Unit_of_measure
	om:derived_unit	some om:Unit_of_measure

Reused Ontologies:

- om: Ontology of Units of Measure[11]

5.1.8 Monetary Value Ontology (iCity-MonetaryValue.owl)

Namespace: monetary

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

- 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**.
- Rate: A Rate is a Monetary Value **fee** that applies **for some Duration**.

Object	Property	Value
MonetaryValue	hasDollarValue	exactly 1 xsd:decimal
	hasRelativeYear	exactly 1 xsd:gYear
	om:unit_of_measure	exactly 1 om:Currency
Rate	subclassOf	MonetaryValue
	appliesFor	only time:DurationDescription

Property	Characteristic	Value (if applicable)
hasDollarValue	Ranges	xsd:Decimal
hasRelativeYear	Ranges	xsd:gYear
hasValue	Ranges	MonetaryValue

Reused Ontologies:
- om: Ontology of Units of Measure
- time: owl-time

5.2 *Person Ontology (iCity-Person.owl)*

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[12] (A vocabulary as opposed to an ontology)
- iCity-Change
- iCity-MonetaryValue
- owl-time

5.3 *Household Ontology (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 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.
- 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.
  We do not make any commitment regarding the identity of the Persons, and in fact a Person may belong to more than one 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

[12] http://schema.org/

	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[13]
- icity-foundation: iCity-Foundation Ontology

5.4 *Organization Ontology (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).

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

[13] 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[14], 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)

5.5 *Building Ontology (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
	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

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

	monetary:hasValue	only monetary:MonetaryValue
	hasRent	only monetary:MonetaryValue
	hasUnitSize	only om:Quantity
	hasRooms	only xsd:int
	hasFacility	only Facility

Reused Ontologies:

- iCity-Foundation

5.6 *Vehicle Ontology (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:Quantity
	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:Quantity
	spatial_loc:hasLocation	only spatial_loc:SpatialFeature
schema:QualitativeValue	subClassOf	om:Quantity

Ontologies Reused:

- Schema.org (vocabulary)
- iCity-Foundation

5.7 *Transportation System Ontology (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.

Currently, there is no need to capture an aggregate Arc; in other words, we do not require a subArc relation. It is possible this may change as the requirements evolve.

- Network: A collection of Nodes and Arcs that enables transportation. A Network may have some cost associated to its access.
- Arc: A directed link in the Network that enables transportation via a particular Mode(s) from one Node to another.
  An Arc begins and ends at a source and sink Node.
  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.
- Node**:** A point in the Network at which Arcs are connected.
  A Node may contain different types of controls: Network Transfer, Signal Control, and Flow Control.
- 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 have access to 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 Arcs 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 Arc(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.

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
	mereology:componentOf	only Network
Node	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some NodePD and change:manifestationOf only NodePD
	change:existsAt	exactly 1 TemporalEntity
	connectedTo	min 1 Arc
	hasControl	only (NetworkTransfer or SignalControl or FlowControl)
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
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
	hasSpeedLimit	exactly 1 SpeedLimit
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
SpeedLimit	hasMaxSpeed	only om:Quantity
	applicableDuring	time:TimePeriod or time:CalendarPeriod
otn:Road	hasRoadId	only RoadId
	aggregationOf	only RoadSegment
RoadSegmentPD	subclassOf	change:TimeVaryingConcept
	equivalentClass	change:hasManifestation some RoadSegment and change:hasManifestation only RoadSegment
	hasRoadSegmentId	only RoadSegmentId
	change:existsAt	exactly 1 time:Interval
RoadSegment	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 spatial_loc:SpatialFeature

Ontologies Reused:

- otn: Ontology of Transportation Networks[15] as presented by (Lorenz, Ohlbach, & Yang, 2005).
- iCity-Foundation

5.7.1 Travel Costs (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

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

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:Quantity
	travelCostOf	only transportation:Arc
AccessFee	subClassOf	TravelCost
	travelCostOf	only transportation:Network

Property	Characteristic	Value (if applicable)
travelCostOf	inverseOf	hasTravelCost

Ontologies Reused:

- iCity-Transportation Network

5.8 *Parking Ontology (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**.
  There are different types (**subclasses**) of Parking Area, such as Street Parking Area, Lot Parking Area, Garage Parking Area, Illegal Parking Area, Loading/Unloading Zone Parking Area, Accessibility Parking Area
- 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**.
  Rate: A Rate has a **dollar value** and an **associated duration**.
  A Rate has a **ParkingPaymentMethod** (e.g. mobile, license plate entry, cashier, meter).

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
ParkingArea	subclassOf	change:Manifestation
	equivalentClass	change:manifestationOf some ParkingAreaPD and change:manifestationOf only ParkingAreaPD
	change:existsAt	exactly 1 time:TemporalEntity
	mereology:hasProperPart	only ParkingArea
	hasParkingPolicy	only ParkingPolicy
	hasChargingStations	exactly 1 xsd:integer
ParkingPolicy	hasParkingRate	only ParkingRate
	maxDuration	only time:DurationDescription
	allowableDuring	only time:TimePeriod or time:CalendarPeriod
ParkingRate	hasMonetaryCost	only monetary:MonetaryValue
	forDuration	only time:DurationDescription
	hasPayment	only ParkingPaymentMethod

Ontologies Reused:

- iCity-Foundational

5.9 *Public Transit Ontology (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.
- 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.
  A Stop Point **has a Location**.
  A Person may enter or exit the transit vehicle at a Stop Point.
- 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:Quantity)
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
RouteLink	beginsAtStop	exactly 1 StopPoint
	endsAtStop	exactly 1 StopPoint
StopPoint	spatial_loc:hasLocation	exactly 1 spatial_loc:SpatialFeature
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

Ontologies Reused:

- iCity-Foundation

5.10 *Land Use Ontology (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.
- Land Use: Land Use provides a means of classifying the use of some land in the urban system.
  There are different types (**subclasses**) of Land Use: Activity, Function, Structure, Site, and Ownership Classifications. Each classification is further defined by a taxonomy of specialized classifications as defined in the LBCS.
- 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.

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
ActivityClassification	subclassOf	LandUseClassification
	equivalentClass	lbcs:Activity
FunctionClassification	subclassOf	LandUseClassification
	equivalentClass	lbcs:Function
StructureClassification	subclassOf	LandUseClassification
	equivalentClass	lbcs:Structure
SiteClassification	subclassOf	LandUseClassification
	equivalentClass	lbcs:Site
OwnershipClassification	subclassOf	LandUseClassification
	equivalentClass	lbcs:Ownership

Reused Ontologies:

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

5.11 *Trip Ontology (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.
  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
	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

[16] Not available online

	occursOn	min 1 transportation:TransportationComplex
TripSegment	mereology:containedIn	exactly 1 Trip
	during	exactly 1 time:Interval
	startLoc	only spatial_loc:SpatialFeature
	endLoc	only spatial_loc:SpatialFeature
	viaMode	min 1 vehicle:Mode
	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

5.11.1 Trip Costs (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 are also possible indirect costs such as vehicle wear and tear, gas, etc. In addition, 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 monetary:MonetaryValue
	tripCostOf	only (trip:Tour or trip:Trip or trip:TripSegment)
DurationCost	subclassOf	TripCost
	hasDurationCost	only time:DurationDescription
DistanceCost	subclassOf	TripCost
	hasDistanceCost	only om:Quantity
EnvironmentalCost	subclassOf	TripCost
	hasEnvironmentalCost	only CarbonEmissions
VehicleCost	subclassOf	TripCost

Reused Ontologies:

- iCity-Trip

5.12 *Urban System Ontology (iCity-UrbanSystem.owl)*

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**.
- 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)
parking:ParkingArea	hasOwner	only person:Person or org:Organization
	occupiedBy	min 0 vehicle:Vehicle
parking:ParkingPolicy	appliesTo	only person:Person or orgLOrganization
	vehicleRestriction	only vehicle:VehicleType
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

6 Future Work

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.

6.1 *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 [17], the format used by Google for travel information.

6.2 *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 of the iCity Ontology is illustrated in Figure 2.

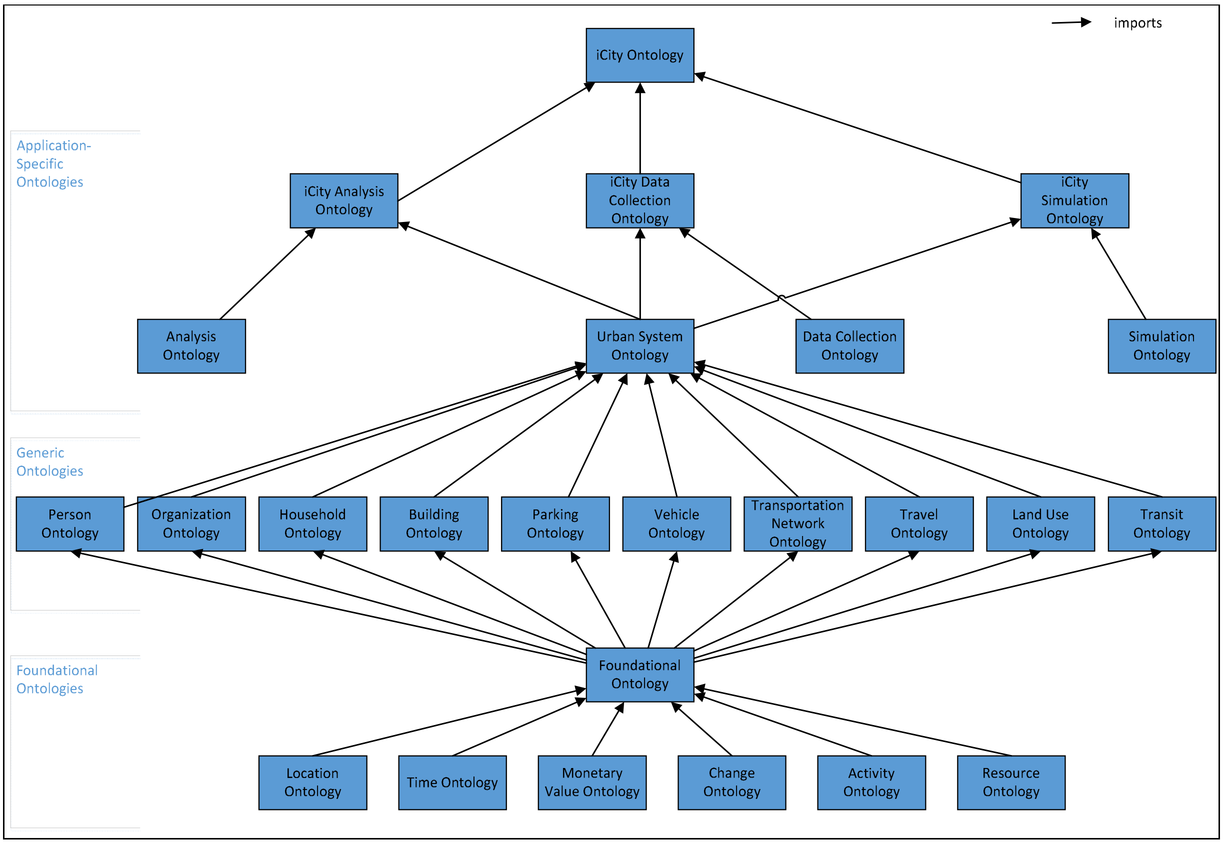


Figure 2: iCity Ontology Structure

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.

---

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

6.2.1 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.

6.2.2 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.

6.2.3 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.

6.2.4 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.

7 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.

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).