TOVE Digital City Manual  
Resource Pattern Reasoning

Mark S. Fox, [msf@eil.utoronto.ca](mailto:msf@eil.utoronto.ca)

21 March 2021

# Introduction

This report documents the Tove Digital City (TDC) Python OWLReady2 (Lamy, 2017) functions that support the use of the ISO/IEC 5087-1 Resource Pattern defined in OWL and can be found at: http://ontology.eil.utoronto.ca/5087/Resource.owl.

In the remainder of this report, we use the following ontology prefix’s:

|  |  |
| --- | --- |
| **Prefix** | **IRI** |
| 5087-1 | http://ontology.eil.utoronto.ca/5087/5087-1/ |

# Resource Representation

As per the ISO/IEC 5087-1 draft document, the Resource 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 [18] 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*". In this sense, Resources are a class of Manifestations, as they are subject to change over time. An object may be classified as a different type of resource, dependent on its participation in an activity. The Resource ontology reuses the Activity ontology.

A Resource class may be used by or consumed by some Activity; the specification of the Resource class defines the quantity of a particular resource that will be used or consumed by a particular activity. If some resource is used by an activity, then when the activity occurs, there must be some resource of that type that is (partially) not available.If a resource type is consumed by an activity, then the resource and the entity it is a manifestation of (partially) cease to exist by the end of the occurrence. Key properties of a Resource are:

* hasLocation: The location of the instance of the type of resource.
* hasCapacity: A Quantity that specifies how much of the resource exists, e.g., if it is liquid, or how much it can hold, if it is an oven.
* hasAvailableCapacity: A Quantity that specifies how much of the resource is available for use at this time (i.e., manifestation)
* capacityInUse: The portion of capacity in use, by some activities, at time of the manifestation.
* hasAllocation: Specifies the State and time interval that the Resource has been allocated to.
* participatesIn: Identifies the Activity the Resource is being used/consumed, etc.

An Allocation class specifies the planned allocation of a resource to an activity via a state. An Allocation is a Manifestation as the allocation may change over time. It has the following properties:

* forResource: specifies the resource that is being allocated
* forState: specifies the state that the resource is allocated to
* hasTime: specifies the DateTimeInterval that the resource is allocated to the state

|  |  |  |
| --- | --- | --- |
| **Object** | **Property** | **Value Restriction** |
| Asset | subClassOf | Manifestation |
| hasLocation | exactly 1 Feature |
| Resource | subClassOf | Asset |
| hasCapacity (functional) | exactly 1 i72:Quantity |
| capacityInUse (functional) | exactly 1 i72:Quantity |
| hasAvailableCapacity (functional) | exactly 1 i72:Quantity |
| participatesIn | min 1 Activity |
| hasAllocation | only Allocation |
|  | ~~hasTrace~~ | ~~only ResourceTrace~~ |
| DivisibleResource | subClassOf | Resource |
| disjointWith | NonDivisibleResource |
| NonDivisibleResource | subClassOf | Resource |
| disjointWith | DivisibleResource |
| Allocation | rdfs:subclassOf | Manifestation |
| hasTime | exactly 1 time:DateTimeInterval |
| forResource | exactly 1 Resource |
| forState | exactly 1 State |
| ~~ResourceTrace~~ | ~~disjointWith~~ | ~~Manifestation~~ |
|  |  |  |

# Resource Functions (http://ontology.eil.utoronto.ca/dt/code/Resource.owl)

|  |  |
| --- | --- |
| **resourceCapacityT(resType, dtd)** | |
| *Computes the capacity availability at time dtd of all resources of type resType.* | |
| **resType** | The class of Resource for which all manifestations are searched. |
| **dtd** | a DateTimeDescription specifying the instant that the manifestations exist |
| **Returns** | an integer representing the total capacity across all manifestations valid at time dtd |
|  | |
| **resourceCapacityLT(resType, locWKT, dtd)** | |
| *Computes the capacity availability at time t and location loc, where loc can specify any area.* | |
| **resType** | The class of Resource for which all manifestations are searched. |
| **locWKT** | The location in WKT format. Can specify a Polygon or Point. |
| **dtd** | a DateTimeDescription specifying the instant that the manifestations exist. |
| **Returns** | an integer representing the total capacity across all manifestations valid at time dtd, and contained within the spatial area specified by locWKT. |
|  | |
| **resourceAllocate(res, state, quant, ti, ns-None)** | |
| *Creates instance of Allocation that specifies the allocation of quantity ‘quant’ of resource ‘res’ to state ‘state’ at time point or interval ‘ti’. Appends allocation to the has Allocation property of the resource and state, and updates capacity properties.* | |
| **res** | The instance of a Resource that is being allocated. |
| **state** | The instance of a State to which the resource is allocated. |
| **quant** | An instance of i72:Quantity specifying the amount allocated. |
| **ti** | The time point or interval that the resource is allocated for. |
| **ns** | The namespace in which the Allocation instance is created. |
| **Returns** | The instance of Allocation. |
| Comment: this function does not create a new manifestation of the state nor resource when appending the Allocation instance. | |

# References

Lamy JB. 2017. Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. Artificial Intelligence In Medicine 2017;80:11-28