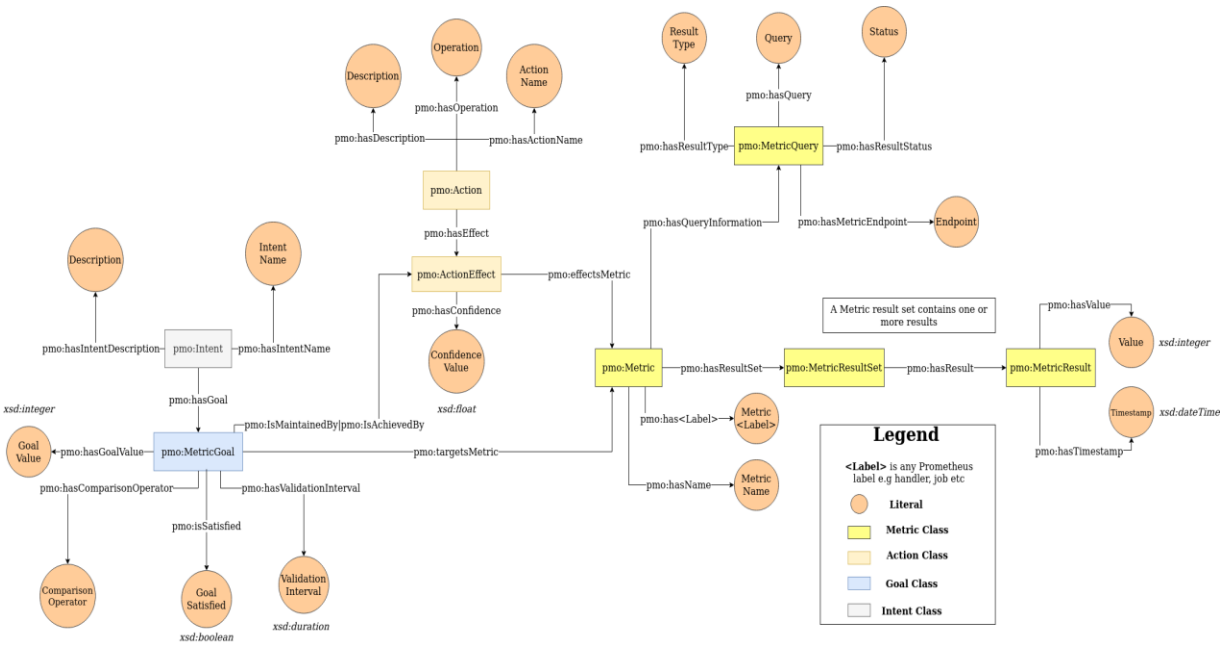
IBCLO Demonstration Walkthrough

The following document demonstrates how intent can be represented and validated using the Intent Based Control Loop Ontology (IBCLO). The processes involved in the demonstration are outlined below.



Class interaction diagram of IBCLO

Ontology specification: https://alex-randles.github.io/IBCLO/

Defining Metrics

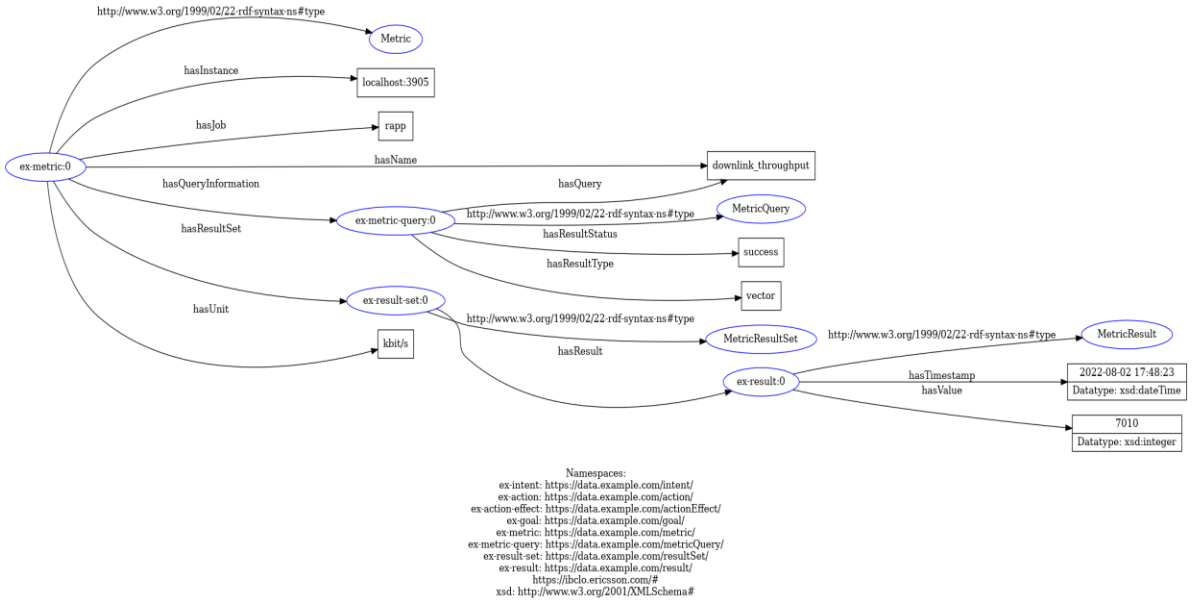
The target metric within the demonstration relates to downlink throughput. The RDF representation of the metric is shown below.

```
ex-metric:0 a :Metric ;
    :hasInstance "localhost:3905" ;
    :hasJob "rapp" ;
    :hasName "downlink_throughput" ;
    :hasQueryInformation ex-metric-query:0 ;
    :hasResultSet ex-result-set:0 ;
    :hasUnit "kbit/s" .

ex-metric-query:0 a :MetricQuery ;
    :hasQuery "downlink_throughput" ;
    :hasResultStatus "success" ;
    :hasResultType "vector" .

ex-result-set:0 a :MetricResultSet ;
    :hasResult ex-result:0 .

ex-result:0 a :MetricResult ;
    :hasTimestamp "2022-08-02 17:48:23"^^xsd:dateTime ;
    :hasValue 7010 .
```

RDF Representation of the the Metric

The metric (`:Metric`) has attributes related to it such as the instance (`:hasInstance`) and job (`:hasJob`). Furthermore, the query information (`:hasQueryInformation`) and values (`:hasResultSet`) are included.

The query information (`:MetricQuery`) includes the query and result type returned. The values of the metric are grouped (`:MetricResultSet`) and each value has a timestamp (`:hasTimestamp`) and value (`:hasValue`).



Visual Representation of the Metric

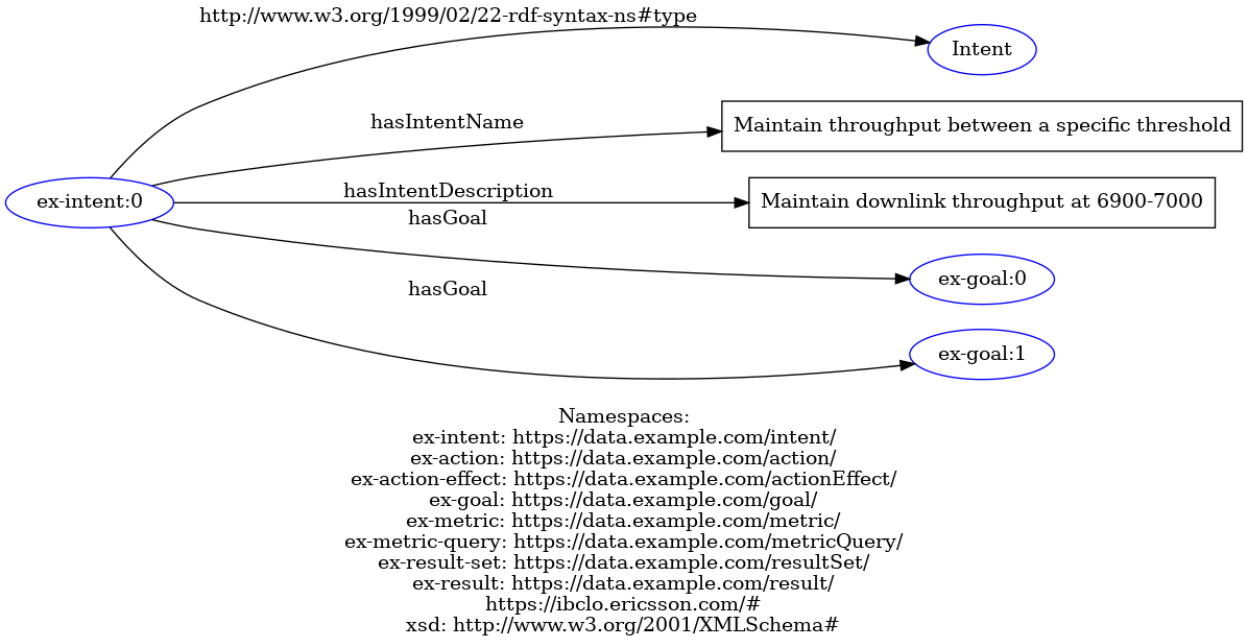
Defining Intent

The sample intent relates to ensuring that the metric which relates to downlink throughput is maintained between 6900-7000. The RDF representation of the intent is shown below.

```
ex-intent:0 a :Intent;
       :hasIntentName "Maintain throughput between a
specific threshold";
       :hasIntentDescription “Maintain downlink

          throughput at 6900-7000”;
:hasGoal ex-goal:0, ex-goal:1 .
```

RDF Representation of the Intent

The intent (`ex:intent:0`) relates to maintaining throughput (`:hasIntentName`) at a value of 7000 (`:hasIntentDescription`) and includes 2 goals `(ex-goal:0, ex-goal:1)`.



Visual Representation of the Intent

Defining Goals

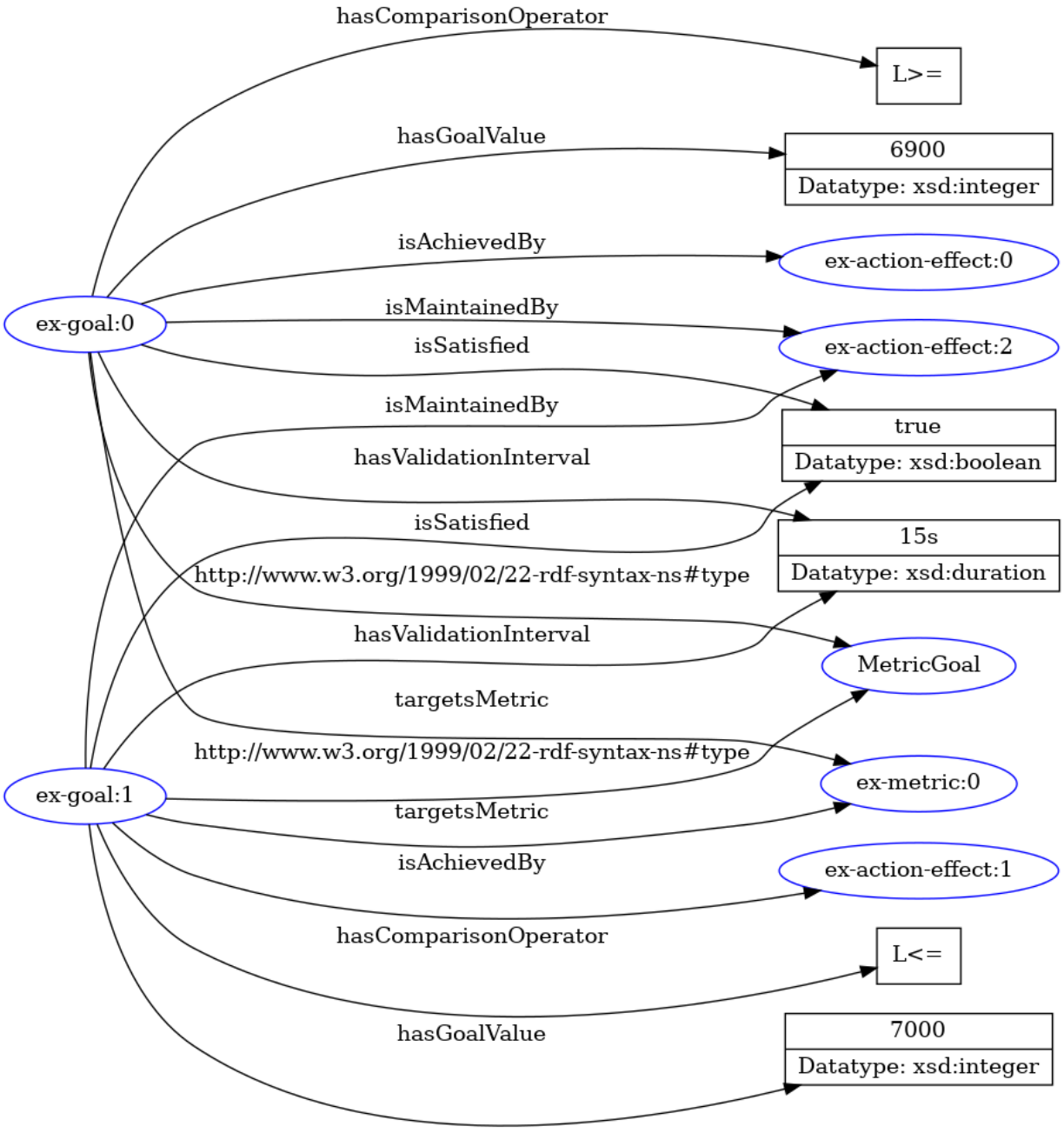
The two goals are designed to ensure that the value is within a specific threshold. The RDF representation is shown below.

```
ex-goal:0 a :MetricGoal ;
    :hasComparisonOperator ">=" ;
    :hasGoalValue 6900;
    :hasValidationInterval "15s"^^xsd:duration ;
    :isSatisfied "true"^^xsd:boolean;
    :isMaintainedBy ex-action-effect:2;
    :isAchievedBy ex-action-effect:0;
    :targetsMetric ex-metric:0 .
```

```
ex-goal:1 a :MetricGoal ;
    :hasComparisonOperator "<=" ;
    :hasGoalValue 7000;
    :hasValidationInterval "15s"^^xsd:duration ;
    :isSatisfied "true"^^xsd:boolean;
    :isAchievedBy ex-action-effect:1;
    :isMaintainedBy ex-action-effect:2;
    :targetsMetric ex-metric:0 .
```

RDF Representation of the Goals

The goals (`:MetricGoal`) compare (`:hasComparisonOperator`) the current value with the goal value (`:hasGoalValue`). The validation is run every 15 seconds (`:hasValidationInterval`) and changed is satisfied (`:isSatisified`). The maintenance action needs to be executed to ensure the goal value is maintained.



Namespaces:
ex-intent: https://data.example.com/intent/
ex-action: https://data.example.com/action/
ex-action-effect: https://data.example.com/actionEffect/
ex-goal: https://data.example.com/goal/
ex-metric: https://data.example.com/metric/
ex-metric-query: https://data.example.com/metricQuery/
ex-result-set: https://data.example.com/resultSet/
ex-result: https://data.example.com/result/
https://ibclo.ericsson.com/#
xsd: http://www.w3.org/2001/XMLSchema#

Visual Representation of the Goals

Defining Actions

Sample actions are shown in the table below. These actions are designed to affect the target metric.

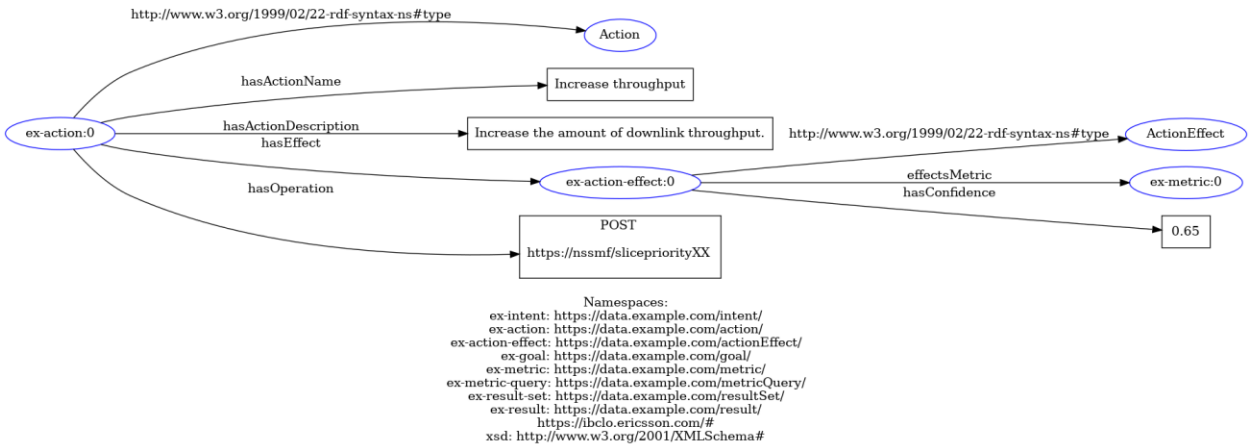
Increase throughput	`ex-action:0 a :Action ;`
	`:hasActionName "Increase throughput" ;`
	`:hasActionDescription "Increase the amount of downlink throughput." ;`
	`:hasEffect ex-action-effect:0 ;`
	`:hasOperation """POST https://nssmf/sliceprioritXX """ .`
	
	`ex-action-effect:0 a :ActionEffect ;`
	`:effectsMetric ex-metric:0 ;`
	`:hasConfidence "0.65" .`

Decrease throughput	`ex-action:1 a :Action ;`
	`:hasActionName "Decrease throughput" ;`
	`:hasActionDescription "Decrease the amount of downlink throughput." ;`
	`:hasEffect ex-action-effect:1 ;`
	`:hasOperation """POST https://nssmf/sliceprioritXX """ .`
	
	`ex-action-effect:1 a :ActionEffect ;`
	`:effectsMetric ex-metric:0 ;`
	`:hasConfidence "0.59" .`

NO-OP	`ex-action:2 a :Action ;`
	`:hasActionName "No operation" ;`
	`:hasActionDescription "Do not execute an operation." ;`
	`:hasEffect ex-action-effect:2 ;`
	`:hasOperation """POST https://nssmf/sliceprioritXX """ .`
	
	`ex-action-effect:2 a :ActionEffect ;`
	`:effectsMetric ex-metric:0 ;`
	`:hasConfidence "0.85" .`

Increase power	`ex-action:3 a :Action ;`
	`:hasActionName "Increase power" ;`
	`:hasActionDescription "Increase power provided for throughput." ;`
	`:hasEffect ex-action-effect:3 ;`
	`:hasOperation """POST https://nssmf/sliceprioritXX """.`
	
	`ex-action-effect:3 a :ActionEffect ;`
	`:effectsMetric ex-metric:0 ;`
	`:hasConfidence "0.29" .`

RDF Representation of the Actions



Visual Representation of the Increase throughput action

The value of the property which states if the goal has been satisfied (`:isSatisified`) can be updated using the SPARQL update query shown below.

```
DELETE { ?goal :isSatisfied ?goalSatisifed }
INSERT { ?goal :isSatisfied "true"^^xsd:boolean }
WHERE  {
     ?goal :isSatisfied ?goalSatisifed
}
```

Query to update goal satisfied status

Validating Intent

The intent can be validated using the SPARQL query shown below.

```
SELECT ?goal ?goalSatisifed
WHERE {
  ex-intent:0 a :Intent;
              :hasGoal ?goal .
  ?goal :isSatisfied ?goalSatisifed .
}
```

Query used to validate intent

The result of the query will include the goal IRI and their corresponding satisfaction status.

goal	goalSatisifed
<http://data.example.com/metricGoal/0>	"true"^^xsd:boolean

Sample result for the query

Maintaining Intent

The intent needs to be maintained once it has been validated. The query shown below can be used to retrieve the actions which maintain each of the goals within the intent.

```
SELECT ?goal ?mainteanceAction
WHERE {
  ex-intent:0 a :Intent;
              :hasGoal ?goal .
  ?goal :isMaintainedBy ?mainteanceAction . }
```

Query used to retrieve maintenance actions

The result includes each of the goals within an intent and their corresponding maintenance action.

goal	mainteanceAction
<https://data.example.com/goal/1>	<https://data.example.com/actionEffect/2>
<https://data.example.com/goal/0>	<https://data.example.com/actionEffect/2>

Sample result for the query

The latest metric values can be retrieved by executing an API request on the Prometheus RDF Generator. An example API request is shown below.

```
alex@alex-Latitude-7300:~/Prometheus-RDF-Generator$ curl "http://127.0.0.1:5000/retrieve-metric-graph?metric-selected=HTTP%20Code%20Responses&label-selected-1=code&label-value-1=200&server-ip-address=127.0.0.1:9090"
<http://example.org/HTTPCodeResponses-Code200> {
    <http://data.example.com/metricResult/1>
            a       <https://prometheus-metric-ontology.ericsson.com/#MetricResult> ;
            <https://prometheus-metric-ontology.ericsson.com/#hasTimestamp>
                    "2022-07-04 11:02:04"^^<http://www.w3.org/2001/XMLSchema#dateTime> ;
            <https://prometheus-metric-ontology.ericsson.com/#hasValue>
                    "341" .

    <http://data.example.com/metricResultSet/1>
            a       <https://prometheus-metric-ontology.ericsson.com/#MetricResultSet> ;
            <https://prometheus-metric-ontology.ericsson.com/#hasResult>
                    <http://data.example.com/metricResult/1> .

    <http://data.example.com/metricQuery/0>
            a       <https://prometheus-metric-ontology.ericsson.com/#MetricQuery> ;
            <https://prometheus-metric-ontology.ericsson.com/#hasQuery>
                    "prometheus_http_requests_total" ;
            <https://prometheus-metric-ontology.ericsson.com/#hasResultType>
                    "vector" ;
            <https://prometheus-metric-ontology.ericsson.com/#hasStatus>
                    "success" .

    <http://data.example.com/metric/0>
            a       <https://prometheus-metric-ontology.ericsson.com/#Metric> ;
            <https://prometheus-metric-ontology.ericsson.com/#hasCode>
                    "200" ;
            <https://prometheus-metric-ontology.ericsson.com/#hasHandler>
                    "/api/v1/query" ;
```

Sample API request to retrieve latest metric graph

The values can be retrieved and compared using the following query template.

```
SELECT ?goalReached ?value
WHERE {
  <Metric> a :Metric;
       :hasResultSet ?resultSet .
        ?resultSet :hasResult ?result .
        ?result :hasValue ?value .
        BIND (?value <Comparison_operator> <Goal_value> AS
?goalReached)
}
```

Query used to compare goal values

A sample query result is shown below.

goalReached	value
"false"^^xsd:boolean	"7010"^^xsd:integer

Sample goal validation