

I To be included from

<https://docs.google.com/document/d/1HhHCjzkVwIQoEJANtLliFyhelFxuvT6MORDlgOg6xUo/edit#heading=h.m1e7pdr04dtz>

- *Tables 1 and 3 (which are in the document) and 2 and 4 (which are in spreadsheets linked to in that document): Laurian can write this*
- *The Scope (first part): Katrina can write this, after you are done with the main body*
- *Diagrammatic descriptions that you made - the prov and uml diagrams*

I think follow the approach of documentation such as <https://www.w3.org/TR/prov-dm/> or maybe better <https://www.w3.org/TR/vocab-dcat-2/>, as much as is relevant for what is an implementation rather than a “model” (tho I am not sure why PLB suggested that, you will have to explain it to me later). So not using table, but rather to have each “class” in a section, with description and an example for the use of that class (maybe here one in eml, one in ttl), and then a list of its properties and for each one of those, also a subsection with description and example (eml and ttl again, where both can be done). Where those properties are themselves classes, then refer to another section where those classes/properties are similarly described. Hmm, you will have to think about it - but at the end of the day, this is to be reference documentation, not explanation documentation, so the idea is that people can find a class/property easily, understand it, and know to what else it is linked, but they do not need to understand from this document how to implement in a provenance document.

I am unsure how you do this with our physicalActivities being a class in which other classes are instances of....be creative. Go for whatever is best first, and we can talk about it if you are unsure afterwards that this is the best approach.

Reference implementation of provenance for biological data: technical documentation

Documenting provenance information

Provenance information is often classified into 3 major groups: information about entities, activities and agents. For the provenance information of biological data, we are following this approach.

For biological data, the entities/things are things like permits and protocols, devices and platforms, samples and software (figure 1). The activities are split up into two types of activities: physical activities and digital activities (figure 2). And agents consist of people and organizations (figure 3).

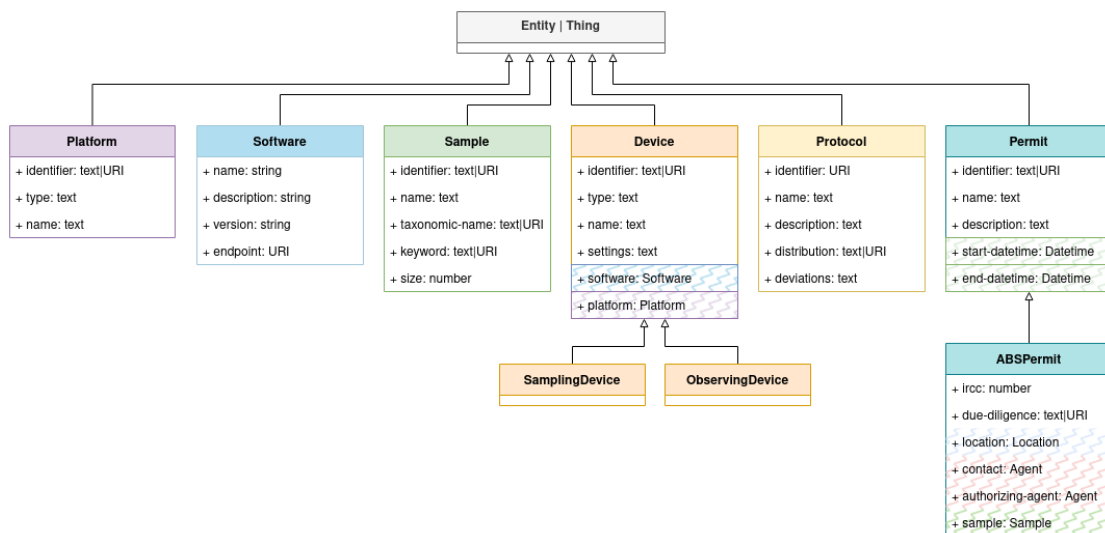


Figure 1.1: Class diagram of Entity classes.

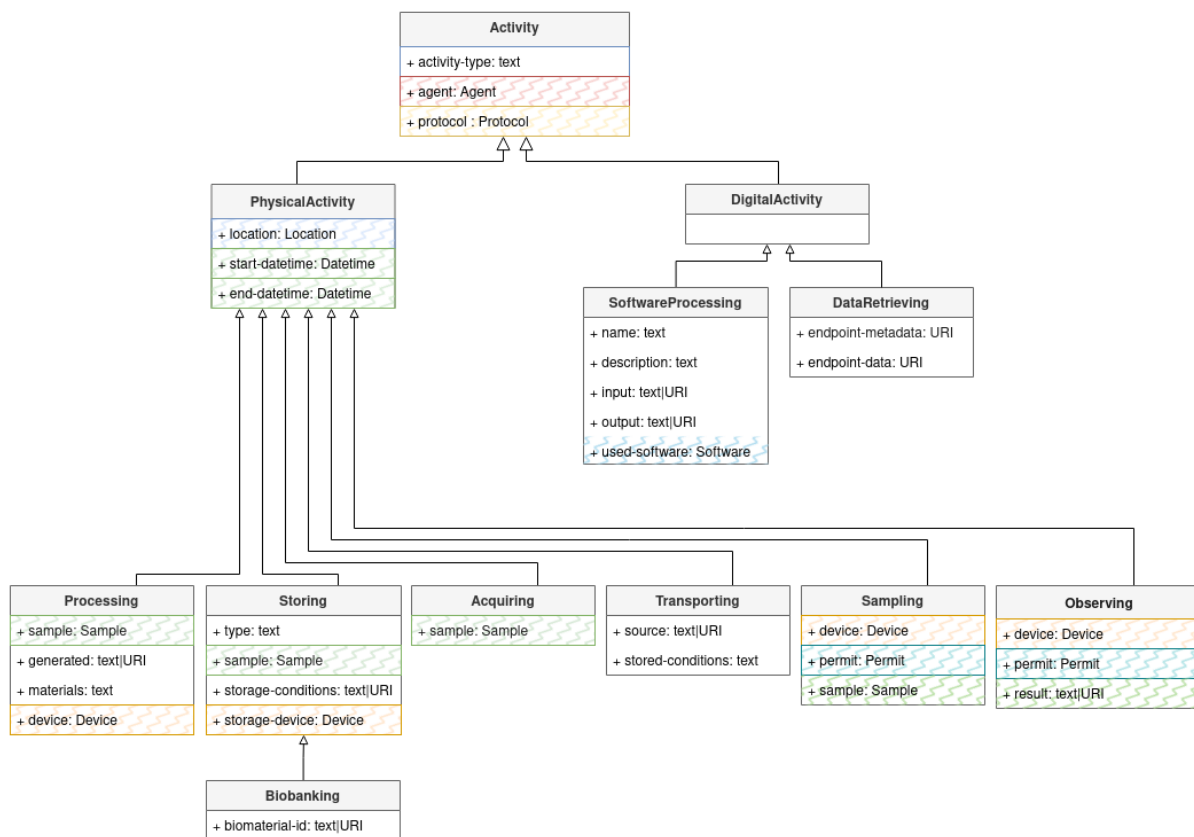


Figure 2.1 - Class diagram of Activity classes

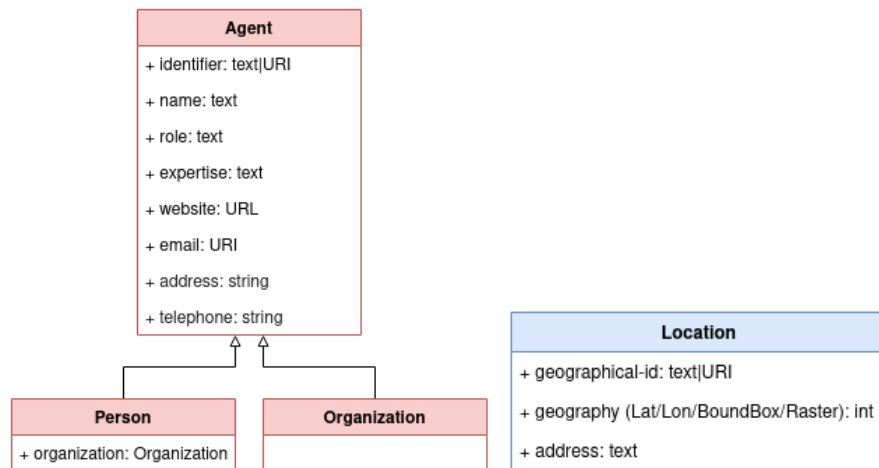


Figure 3.1 - Class diagram of Agent and Location classes.

Documenting provenance information of biological data in RDF

For our reference implementation of provenance information for biological data in RDF we based ourselves on well known standards, such as the W3C standard [prov-dm](#), [Dublin Core](#) and [schema.org](#). Rather than creating new types or a new ontology, we defined each of the agnostic information groups (figure 1.1, 1.2 and 3.1) as a combination of existing classes in RDF (figure 1.2, 2.2 and 3.2). This approach increases findability and interoperability since third parties are able to identify and query the property paths of the resulting provenance graph. It also infers certain requirements on the shape of the provenance graph, which are exposed/specified in the SHaCL shapes provided in appendix C.

The prov-dm serves as the backbone for our reference implementation, with the classes selected to describe provenance information being the core structures of prov-dm: prov:Entity, prov:Activity and prov:Agent (figure 4).

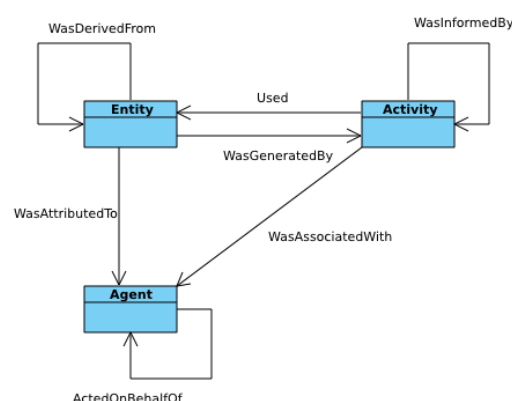


Figure 4: Prov core structures (source: <https://www.w3.org/TR/prov-dm/>)

Instances of these classes (tables 1 - 3) are described using properties from [schema.org](#) and [Dublin Core](#), two well known ontologies that provide general property terms (tables 1.A-G, 2.A-J and 3.A-C). To describe specific biological aspects of provenance, we used properties of the [Semantic Sensor Network](#).

In order to be semantically sound with regards to the used properties, some concepts (o.a. Activity) are described by multiple classes. As can be seen in the class diagrams below (figure 1.2, 2.2 and 3.2).

The given properties are our suggestion as to what should be documented in relation to provenance of biological data. The reference implementation has been constructed in such a way that it can be extended with other properties in case your use-case isn't fully covered. We recommend trying to use existing terms as much as possible.

A template, in turtle syntax, of our reference implementation in RDF is available in appendix A and an example can be found in appendix B.

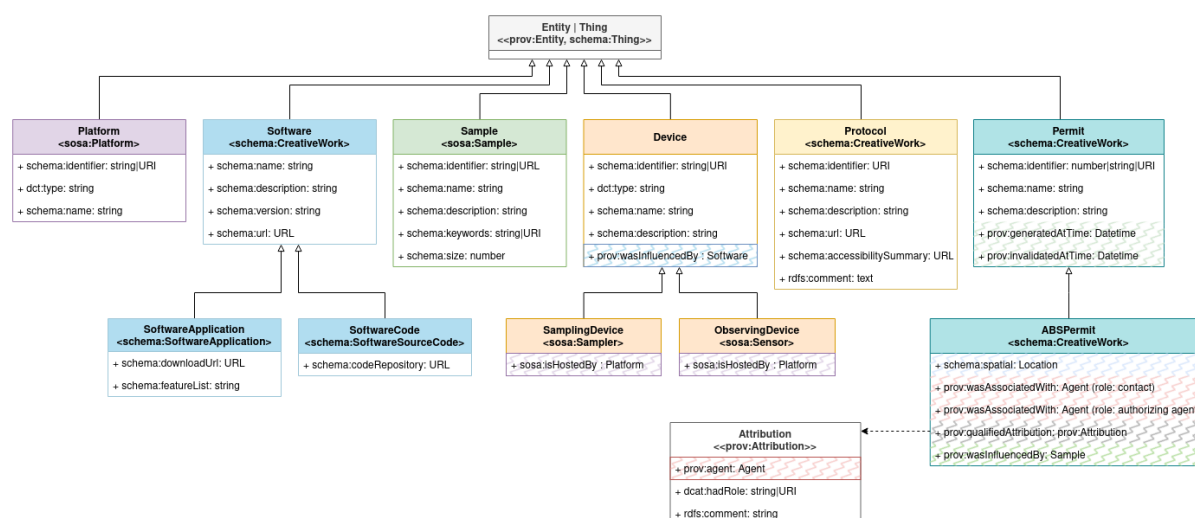


Figure 1.2 - Class diagram of the RDF implementation of Entity classes and their properties.

Table 1 - Definitions of Entity classes, with their RDF implementation

| Class | Description | RDF implementation |
|------------------|--|---|
| Entity | An abstract class for representing entities, which is a physical, digital, conceptual, or other kind of thing with some fixed aspects; entities may be real or imaginary. | prov:Entity , schema:Thing |
| Protocol | A class for representing protocols, which are entities that describe how an activity was done, including the design, methodology, materials, statistical considerations, etc. AKA Standard Operating Procedure | prov:Entity , schema:Thing |
| Device | A class for representing devices, which are entities that are used to carry out an activity: an instrument, a sensor, a container, etc. It includes also human-based devices (i.e. manual observations) | prov:Entity , schema:Thing |
| Sampling-Device | A class for representing devices that are used in relation to sampling activities. | prov:Entity , schema:Thing , sosa:Sampler |
| Observing-Device | A class for representing devices that are used in relation to observing activities. | prov:Entity , schema:Thing , sosa:Sensor |

| | | |
|-----------|--|---|
| Platform | A class for representing platforms, which are entities on/from which a device is deployed (e.g. a ship, a buoy, a CTD carousel, an ROV, etc.) | prov:Entity , schema:Thing , sosa:Platform |
| Sample | A class for representing samples, which are entities that constitute an extract of physical material | prov:Entity , schema:Thing , sosa:Sample |
| Permit | A class for representing permits, which are entities that describe the authorisation to carry out an activity, often needing to be approved by a legal entity. | prov:Entity , schema:Thing , schema:CreativeWork |
| ABSPermit | A class for representing Access and Benefit Sharing permits, which are a kind of permit specifically for the Nagoya Protocol on Access and Benefit Sharing | prov:Entity , schema:Thing , schema:CreativeWork |
| Software | A class for representing software, which are entities that constitute software applications or computer programming source code, but can also be computational workflows, notebooks and also simple processes such as basic arithmetical calculations. | prov:Entity , schema:Thing , schema:SoftwareApplication schema:SoftwareSourceCode |

Table 1.A. - Properties associated to a Software entity, with their RDF implementation

| property | Description | RDF implementation |
|-------------|--|---|
| name | The name of the software application or the source code. | schema:name |
| description | A description of the software application or the source code. | schema:description |
| version | A version of the software application or the source code. | schema:version |
| endpoint | A link to gain access to the software. Depending on the kind of Software (i.e. schema:SoftwareApplication or schema:SoftwareSourceCode or another class) this can be described with different properties. (sidenote: given the use-case, feel free to describe with other existing properties) | schema:url |
| | | schema:codeRepository |
| | | schema:downloadUrl , schema:featureList |

Table 1.B. - Properties associated to a Permit entity, with their RDF implementation

| property | Description | RDF implementation |
|----------------|---|--|
| identifier | The identifier of the permit. | schema:identifier |
| name | The name of the permit. | schema:name |
| description | A description of the permit. | schema:description |
| start-datetime | The start date of the temporal range depicting the applicability of the permit. | prov:generatedAtTime |
| end-datetime | The end date of the temporal range depicting the applicability of the permit. | prov:invalidatedAtTime |

Table 1.C. - Properties associated to an ABSPermit entity, with their RDF implementation

| property | Description | RDF implementation |
|-------------------|---|--|
| ircc | The 'Internationally-Recognised Certificate of Compliance', which is the code on the certificate granted by the ABS clearing house. | schema:identifier |
| due-diligence | A description of due-diligence, which should be provided in the case an IRCC cannot be given. | schema:description |
| location | The spatial coverage describing the spatial applicability of the permit. | schema:spatial |
| contact | The agent that acts as the contactpoint for this permit. | prov:wasAssociatedWith |
| authorizing-agent | The agent that has authorized the permit. | prov:wasAssociatedWith |
| sample | A sample that is associated with this permit. | prov:wasInfluencedBy |

Table 1.D. - Properties associated to a Sample entity, with their RDF implementation

| property | Description | RDF implementation |
|----------------|--|------------------------------------|
| identifier | An identifier of the sample. | schema:identifier |
| name | The name of the sample. | schema:name |
| taxonomic-name | Any taxonomic names that are associated with the sample. | schema:description |
| keyword | Any keywords that describe the sample (type, content, etc.). | schema:keywords |
| size | The size of the sample (e.g. volume, weight, ...). | schema:size |

Table 1.E. - Properties associated to a Device entity, which can be either a SamplingDevice or an ObservingDevice, with their RDF implementation

| property | Description | RDF implementation |
|------------|--|--------------------------------------|
| identifier | An identifier of the device, being either a code or URL. | schema:identifier |
| type | The type of device. | dct:type |
| name | The name of the device. | schema:name |
| settings | The settings on the device that were relevant/of importance during the activity. | schema:description |
| software | A reference to the software that is used by the device. | prov:wasInfluencedBy |
| platform | The platform on which the device is mounted or hosted on. | sosa:isHostedBy |

Table 1.F. - Properties associated to a Platform entity, with their RDF implementation

| property | Description | RDF implementation |
|------------|---|-----------------------------------|
| identifier | An identifier of the platform, being a URL or code. | schema:identifier |
| type | The type of platform. | dct:type |
| name | The name of the platform. | schema:name |

Table 1.G. - Properties associated to a Protocol entity, with their RDF implementation

| property | Description | RDF implementation |
|--------------|---|---|
| identifier | An identifier of the protocol. | schema:identifier |
| name | The name of the protocol. | schema:name |
| description | A description of the protocol. | schema:description |
| distribution | A description of where the protocol can be found and/or accessed. | schema:url |
| | | schema:accessibilitySummary |
| deviations | A description of any deviations that occurred from the protocol. | rdfs:comment |

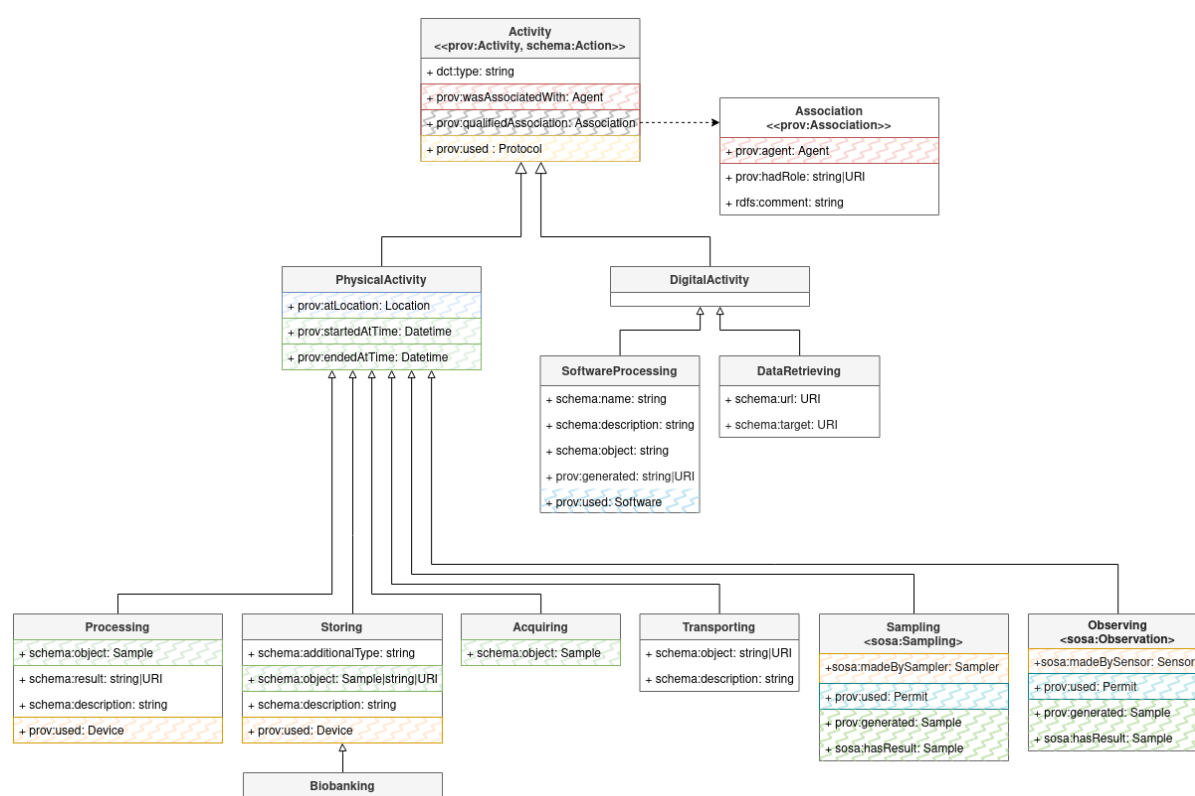


Figure 2.2. - Class diagram of the RDF implementation of Activity classes and their properties.

Table 2 - Definitions of Activity classes, with their RDF implementation

| Class | Description | RDF implementation |
|-------------------|---|--|
| Activity | An abstract class for representing activities, which is something that occurs over a period of time, is performed by an agent and acts upon or with objects/entities. | prov:Activity , schema:Action |
| Physical Activity | An abstract class for representing any activity that deals with physical material. | prov:Activity , schema:Action |

| | | |
|---------------------|---|--|
| Acquiring | A class for representing physical activities where physical material is acquired (from the field, from the lab, ordered from a biobank, taken out of local storage, etc) | prov:Activity , schema:Action ; dct:type "Acquiring" |
| Sampling | A class for representing acquiring activities where the material is collected from the field. | prov:Activity , schema:Action ; dct:type "Sampling" |
| Storing | A class for representing physical activities where physical material is retained in a specific location (e.g. storage facility, freezer, etc.), for some length of time. | prov:Activity , schema:Action ; dct:type "Storing" |
| Biobanking | A class representing storing activities where the physical material is stored long-term in a long-term storage facility (such as a biobank or culture collection facility) specifically to make it available for others to use. | prov:Activity , schema:Action ; dct:type "Biobanking" |
| Processing | A class representing physical activities where the work done on the physical material transforms that material, e.g. adding preservatives, filtering, centrifuging, extracting, etc. | prov:Activity , schema:Action ; dct:type "Processing" |
| Transporting | A class representing physical activities where the physical material is moved, posted or shipped. | prov:Activity , schema:Action ; dct:type "Transporting" |
| Observing | A class representing physical activities where observations are made of physical material (such as organisms, media, states, etc). | prov:Activity , schema:Action ; dct:type "Observing" |
| Digital Activity | <i>An abstract class for representing any activity involving digital objects (mainly being datasets).</i> | prov:Activity , schema:Action |
| Software Processing | A class representing digital activities where the digital objects are processed such as in quality control of data or the workflow used by instruments; 'processing' can range from simple calculations to complete software workflows. | prov:Activity , schema:Action ; dct:type "Software Processing" |
| Data Retrieving | A class representing digital activities where the digital objects are obtained from another digital source, e.g. an online catalog, colleague's spreadsheet | prov:Activity , schema:Action ; dct:type "Data Retrieving" |

Table 2.A. - Properties associated to an Activity, with their RDF implementation

| property | Description | RDF implementation |
|---------------|---|--|
| activity-type | The type of activity carried out (recommended activity types are names of the classes mentioned above). | dct:type |
| agent | The agent that is associated with the activity. | prov:wasAssociatedWith |
| protocol | A protocol or standard operating procedure (S.O.P.) that is associated with the activity. | prov:used |

Table 2.B. - Properties associated to a SoftwareProcessing activity, with their RDF implementation

| property | Description | RDF implementation |
|----------|-------------|--------------------|
|----------|-------------|--------------------|

| | | |
|---------------|---|------------------------------------|
| name | The name associated with the software processing activity. | schema:name |
| description | A description of the software processing activity. | schema:description |
| input | A reference or description of the input used in the software processing activity. | schema:object |
| output | A reference to or description of the output used in the software processing activity. | prov:generated |
| used-software | A reference to the software used during the software processing activity. | prov:used |

Table 2.C. - Properties associated to a DataRetrieving activity, with their RDF implementation

| property | Description | RDF implementation |
|-------------------|---|-------------------------------|
| endpoint-metadata | A link to the description of the metadata relating to the data retrieving activity. | schema:url |
| endpoint-data | The endpoint / link to retrieve or get the data. | schema:target |

Table 2.D. - Properties associated to a Physical Activity, with their RDF implementation

| property | Description | RDF implementation |
|----------------|--|------------------------------------|
| location | The location where the physical activity took place. | prov:location |
| start-datetime | The date and time of the start of the physical activity. | prov:startedAtTime |
| end-datetime | The date and time of the end of the physical activity*. | prov:endedAtTime |

(*Note: if the physical activity constitutes a point in time, then start-datetime and end-datetime can be the same.)

Table 2.E. - Properties associated to a Processing activity, with their RDF implementation

| property | Description | RDF implementation |
|-----------|--|--|
| sample | The sample that is processed. | schema:object |
| generated | The resulting entity that is generated from the sample during the processing activity. | schema:result |
| materials | A description of the materials that were used in the processing activity (e.g. chemicals, etc.). | schema:description |
| device | The device that was used in the processing activity. | prov:used schema:instrument |

Table 2.F. - Properties associated to a Storing activity (and more specifically also a Biobanking activity), with their RDF implementation

| property | Description | RDF implementation |
|----------|--|---------------------------------------|
| type | The type of storing activity (e.g. long term, short term, biobanking, etc.). | schema:additionalType |
| sample | The identifier or name of the sample (or | schema:object |

| | | |
|--------------------|---|--|
| (~ biomaterial-id) | biosample, in case of a biobanking activity) that was stored in the storing activity. | |
| storage-conditions | A description of the storage conditions of the storing activity. | schema:description |
| storage-device | The device that was used for storing the sample. | prov:used schema:instrument |

Table 2.G. - Properties associated to an Acquiring activity, with their RDF implementation

| property | Description | RDF implementation |
|----------|--|-------------------------------|
| sample | The identifier or name of the sample that is acquired in the acquiring activity. | schema:object |

Table 2.H. - Properties associated to a Transporting activity, with their RDF implementation

| property | Description | RDF implementation |
|-------------------|---|------------------------------------|
| source | | schema:object |
| stored-conditions | A description of the storage conditions during the transporting activity. | schema:description |

Table 2.I. - Properties associated to a Sampling activity, with their RDF implementation

| property | Description | RDF implementation |
|----------|--|--|
| device | The device used in the sampling activity. (note: the object in this triple should be an instance of the SamplingDevice class) | sosa:madeBySampler |
| permit | A permit that was used in relation to the sampling activity. | prov:used |
| result | The sample that resulted from the sampling activity. | prov:generated sosa:hasResult |

Table 2.J. - Properties associated to an Observing activity, with their RDF implementation

| property | Description | RDF implementation |
|----------|---|--|
| device | The device used to carry out the observing activity (note: the object in this triple should be an instance of the ObservingDevice class) | sosa:madeBySensor |
| permit | A permit that was used in relation to the sampling activity. | prov:used |
| result | The result of the observing activity. | prov:generated sosa:hasResult |

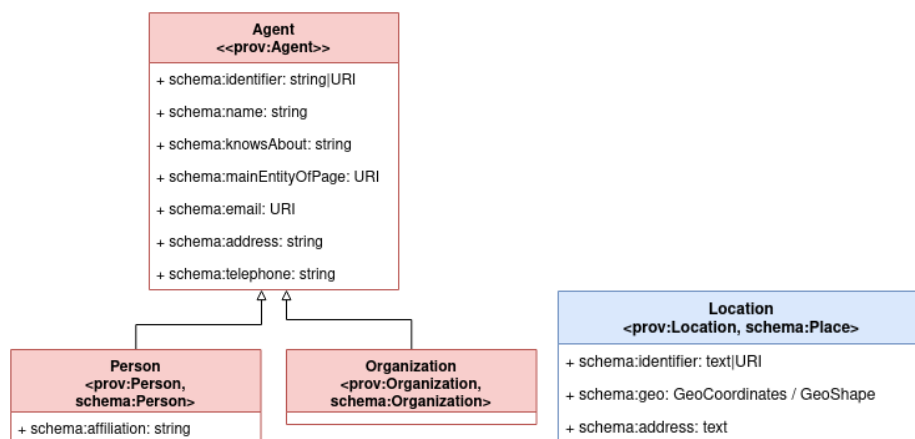


Figure 3.2 - Class diagram of the RDF implementation of Agent and Location classes and their properties.

Table 3 - Agent and Location classes, with their RDF implementation

| Class | Description | RDF implementation |
|--------------|--|--|
| Agent | An abstract class representing the agent responsible for an activity. | prov:Agent |
| Person | A class to represent agents which are people. | prov:Person , schema:Person |
| Organization | A class to represent agents which are social or legal organizations such as a company, society, etc. | prov:Organization , schema:Organization |
| Location | The place at which an activity occurs. Note: field activities should be defined by geographical coordinates, lab activities by address. | prov:Location , schema:Place |

Table 3.A. - Properties associated to an Agent, with their RDF implementation

| property | Description | RDF implementation |
|------------|---|---|
| identifier | An identifier of the agent. | schema:identifier |
| name | The name of the agent. | schema:name |
| role | The role of the agent in relation to an activity. (note: object of the triple should have following structure - see also appendix A: prov: qualifiedAssociation [a prov:Association, prov:agent :Agent, prov:hadRole "specific role"] ;) | prov:qualifiedAssociation |
| | The role of the agent in relation to an entity. (note: object of the triple should have following structure - see also appendix A: prov: qualifiedAssociation [a prov:Attribution, prov:agent :Agent, | prov:qualifiedAttribution |

| | | |
|-----------|---|---|
| | dcat:hadRole "specific role"] ;) | |
| expertise | The domain(s) of expertise of the agent. | schema:knowsAbout |
| website | A website of the agent, being either an organizational website or a personal website. | schema:mainEntityOfPage |
| email | The email address of the agent. | schema:email |
| address | The address of the agent. | schema:address |
| telephone | A telephone number of the agent. | schema:telephone |

Table 3.B. - Properties related to a Person agent, with their RDF implementation

| property | Description | RDF implementation |
|--------------|--|------------------------------------|
| organization | The organization that the person is affiliated with. | schema:affiliation |

Table 3.C. - Properties associated to a Location, with their RDF implementation

| property | Description | RDF implementation |
|-----------------|---|-----------------------------------|
| geographical-id | The identifier of the location. | schema:identifier |
| geography | The geography of the location, described either through the geographic coordinates of the location or the geographic shape of the location. | schema:geo |
| address | The address of the location. | schema:address |

Documenting provenance information of biological data in EML

For the EML implementation, we tried to apply the [EML 2.2.0. schema](#) to the agnostically defined classes (figure 1 - 3) as best as possible. However, for a number of classes, EML elements are not present as needed in the current schema. These were highlighted in red in the diagrams below (figure 5.1 - 7.1).

One approach *to include this (missing) provenance information* is to add these fields in the following EML schema. We propose, for example, to create a Permit and Sample class (figure 5.1 - 7.1). However, this is not within our control and takes time to implement - hence, it doesn't allow us to provide provenance information for biological data at this moment.

Another approach is to include the information relevant to EML fields currently missing in EML2.2.0. in other fields by expanding their definitions. We followed this approach to include information from numerous properties in the agnostic model, as can be seen by the description element in various classes (EML templates 1 - 21). But this didn't provide enough of a solution. Besides that, this approach results in more ambiguous semantics, hence also decreasing interoperability.

For reasons mentioned above, we alternatively propose to provide provenance information in RDF and reference to it in the eml-xml document using the *<annotation>* element, as follows:

```

<dataset id="....">
  <title></title>
  ....
  <annotation>
    <propertyURI label="provenance">http://purl.org/dc/terms/provenance</propertyURI>
    <valueURI label=""> {URL of RDF file with provenance information } </valueURI>
  </annotation>
</dataset>

```

Or:

```

<dataset id="....">
  <title></title>
  ....
  <annotation>
    <propertyURI label="conforms to">http://purl.org/dc/terms/conformsTo</propertyURI>
    <valueURI label=""> {URL of RDF file with provenance information } </valueURI>
  </annotation>
</dataset>

```

To give an indication on how to use the EML implementation, EML templates are given below.

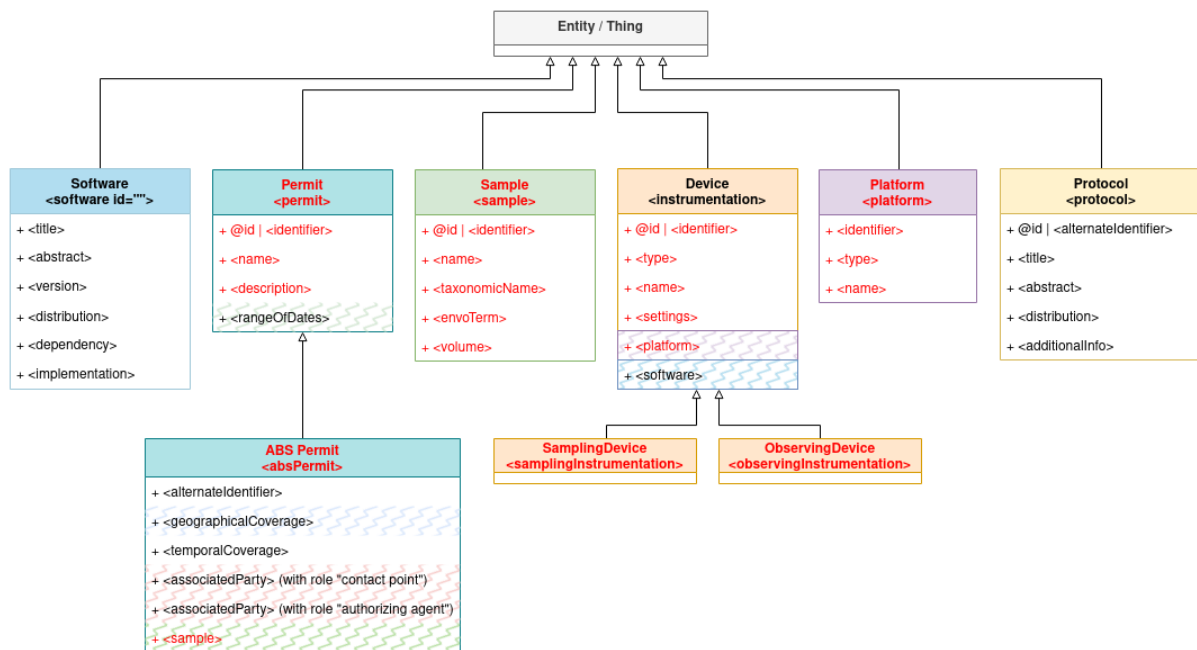


Figure 5.1 - Diagram of the EML elements of Entity classes and their child-elements.

```

<software>
  <title>{name of the software used in the activity}</title>
  <abstract>{description of the software used in the activity}</abstract>
  <version>{version number of the software used in the activity}</version>
  <distribution>
    <online>
      <url>{endpoint to gain access to the software used in the activity }</url>
    </online>

```

```

<distribution>
<!-- Depending on the Software, it can be described with different properties -->
<dependency>{dependency info of the software}</dependency>
<implementation>{implementation info of the software}</implementation>
</software>

```

EML template 1 - Software entity class with its properties.

```

<permit id="...">
  <identifier>{an identifier/code of the permit}</identifier>
  <name>{Name of the permit}</name>
  <description>{A short description of the scope of the permit}</description>
  <temporalCoverage>
    <rangeOfDates>
      <beginDate>{start date of the applicability of the permit}</beginDate>
      <endDate>{end date of the applicability of the permit}</endDate>
    </rangeOfDates>
  </temporalCoverage>
</permit>

```

EML template 2 - Permit entity class with its properties

```

<absPermit id="...">
  <identifier>{an identifier/code of the permit}</identifier>
  <alternateIdentifier>{the IRCC code}</alternateIdentifier>
  <name>{Name of the permit}</name>
  <description>
    {A short description of the scope of the permit
    + due-diligence in case of missing IRCC code}
  </description>
  <temporalCoverage>
    <rangeOfDates>
      <beginDate>{start date of the applicability of the permit}</beginDate>
      <endDate>{end date of the applicability of the permit}</endDate>
    </rangeOfDates>
  </temporalCoverage>
  <geographicCoverage>
    {spatial coverage describing the spatial applicability of the permit}
  </geographicCoverage>

  <associatedParty>
    {the agent that serves as the contact-point of the permit}
    <role>contact</role>
  </associatedParty>

  <associatedParty>
    {the agent that serves as the authorizing agent of the permit}
    <role>authorizing agent</role>
  </associatedParty>

  <sample>
    {a sample that is associated with this permit}
  </sample>

```

```
</sample>
</absPermit>
```

EML template 3 - ABS Permit entity class with its properties.

```
<sample id="...">
  <identifier>{the identifier of the sample}</identifier>
  <name>{the name of the sample}</name>
  <taxonomicName>{taxonomic name associated with the sample}</taxonomicName>
  <envoTerm>{envo term associated with the sample}</envoTerm>
  <volume>{the size (volume, weight, ...) of the sample}</volume>
</sample>
```

EML template 4 - Sample entity class with its properties.

```
<instrumentation id="...">
  <identifier>{the identifier/code of the device}</identifier>
  <type>{the type of device}</type>
  <name>{the name of the device}</name>
  <settings>{the settings of the device}</settings>
  <software>{reference to the software that is used by the device}</software>
  <platform>{reference to the platform on which the device is mounted/attached}</platform>
</instrumentation>
```

EML template 5 - Device entity class with its properties.

```
<samplingInstrumentation id="...">
  <identifier>{the identifier/code of the device}</identifier>
  <type>{the type of device}</type>
  <name>{the name of the device}</name>
  <settings>{the settings of the device}</settings>
  <software>{reference to the software that is used by the device}</software>
  <platform>{reference to the platform on which the device is mounted/attached}</platform>
</samplingInstrumentation>
```

EML template 6 - Sampling Device entity class with its properties.

```
<observingInstrumentation id="...">
  <identifier>{the identifier/code of the device}</identifier>
  <type>{the type of device}</type>
  <name>{the name of the device}</name>
  <settings>{the settings of the device}</settings>
  <software>{reference to the software that is used by the device}</software>
  <platform>{reference to the platform on which the device is mounted/attached}</platform>
</observingInstrumentation>
```

EML template 7 - Observing Device entity class with its properties.

```
<platform id="...">
  <identifier>{the identifier of the platform, being a URL or code}</identifier>
  <type>{the type of platform}</type>
  <name>{the name of the platform}</name>
</platform>
```

EML template 8 - Platform entity class with its properties.


```
<protocol id="...">
  <identifier>{identifier of the protocol}</identifier>
  <!-- though alternateIdentifier element already available in current EML schema
  could serve this purpose-->
  <alternateIdentifier>{identifier of the protocol}</alternateIdentifier>
  <title>{name of the protocol}</title>
  <abstract>{description of the protocol}</abstract>
  <distribution>{where the protocol can be found and/or accessed}</distribution>
  <additionalInfo>
    {description of any deviations that occurred from the protocol}
  </additionalInfo>
</protocol>
```

EML template 9 - Protocol entity class with its properties.

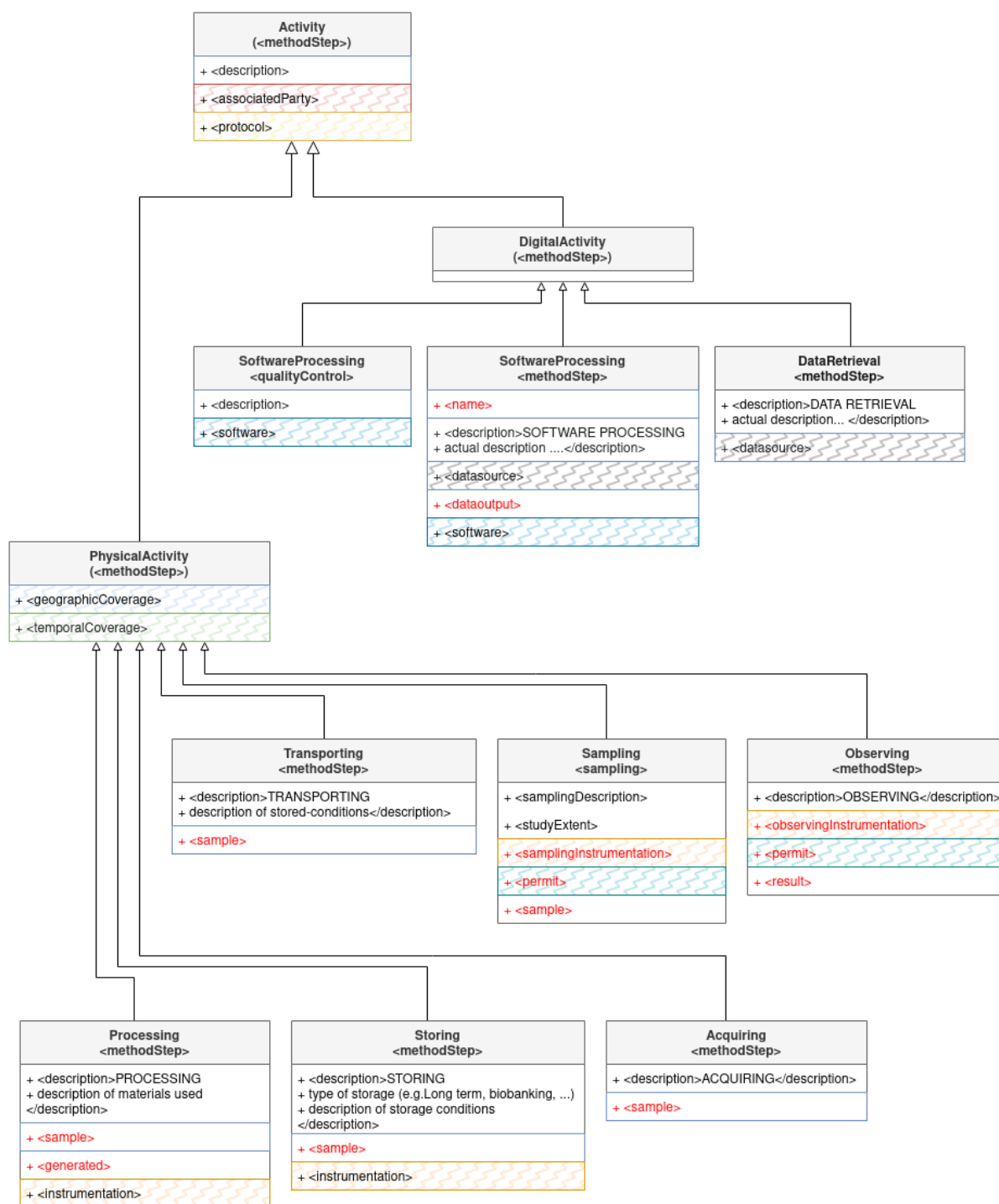


Figure 6.1 - Class diagram of the EML implementation of Activity classes and their properties.

```

<methods>
  <qualityControl>
    <description>
      {A description of the quality control activity - which is a type of software
       processing activity }
    </description>
  
```

```

        <software>
            {Reference to the software used in the quality control activity}
        </software>
    </qualityControl>
</methods>

```

EML Template 10 - Software processing activity class, with its properties; in case of Quality control.

```

<methods>
    <methodStep>
        <name>{name of the software processing activity}</name>
        <description>
            SOFTWARE PROCESSING
            {...
            + A description of the software processing activity}
        </description>

        <associatedParty>
            {the agent that is associated with the activity}
        </associatedParty>

        <protocol>
            {A protocol or SOP that is associated with the activity}
        </protocol>

        <dataSource>
            {A reference to the input used in the software processing activity.}
        </dataSource>

        <dataOutput>
            {A reference to the output generated in the software processing activity.}
        </dataOutput>

        <software>
            {A reference to the software used in the software processing activity}
        </software>
    </methodStep>
</methods>

```

EML Template 11 - Software processing activity class, with its properties.

```

<methods>
    <methodStep>
        <description>
            DATA RETRIEVING
            {...
            + A description of the data retrieving activity}
        </description>

        <associatedParty>

```

```

        {the agent that is associated with the activity}
    </associatedParty>

    <protocol>
        {A protocol or SOP that is associated with the activity}
    </protocol>

    <dataSource>
        {A reference to the input used in the data retrieving activity.
        (with child elements that can hold information relating to
        1. endpoint metadata ~ link to metadata relating to this activity
        2. endpoint-data ~ entry-point to retrieve/get to the data}
    </dataSource>
</methodStep>
</methods>

```

EML Template 12 - Data retrieving activity class, with its properties.

```

<methods>
  <methodStep>
    <description>
      PROCESSING
      {...
      + description of the materials (e.g. chemicals, etc.) that were used in the
      processing activity}
    </description>

    <geographicCoverage>
      {Reference to the location or geographicCoverage of the physical activity}
    </geographicCoverage>
    <temporalCoverage>
      {Reference to the temporal range of the physical activity
      note: can be either <rangeOfDates> or <singleDateTime>}
    </temporalCoverage>

    <associatedParty>
      {the agent that is associated with the activity}
    </associatedParty>

    <protocol>
      {A protocol or SOP that is associated with the activity}
    </protocol>

    <sample>
      {Reference to the sample that served as the source in the storing activity}
    </sample>

    <generated>
      {Reference to the output that is generated by the processing activity}
    </generated>

```

```

        <instrumentation>
            {Reference to the device(s) that was/were used in the processing activity}
        </instrumentation>
    </methodStep>
</methods>

```

EML Template 13 - Processing activity class, with its properties.

```

<methods>
    <methodStep>
        <description>
            STORING
            {...
            + the type of storage (e.g. biobanking, long term, etc.
            + a description of storing conditions}
        </description>

        <geographicCoverage>
            {Reference to the location or geographicCoverage of the physical activity}
        </geographicCoverage>

        <temporalCoverage>
            {Reference to the temporal range of the physical activity
            note: can be either <rangeOfDates> or <singleDateTime>}
        </temporalCoverage>

        <associatedParty>
            {the agent that is associated with the activity}
        </associatedParty>

        <protocol>
            {A protocol or SOP that is associated with the activity}
        </protocol>

        <sample>
            {Reference to the sample that served as the source in the storing activity
            (with the child-element <alternateIdentifier> to hold the identifier of the
            sample/biomaterial-id)}
        </sample>

        <instrumentation>
            {Reference to the device that was used in the storing activity}
        </instrumentation>
    </methodStep>
</methods>

```

EML Template 14 - Storing activity class, with its properties.

```

<methods>
    <methodStep>
        <description>

```

```

        ACQUIRING
        {...}
    </description>

    <geographicCoverage>
        {Reference to the location or geographicCoverage of the physical activity}
    </geographicCoverage>
    <temporalCoverage>
        {Reference to the temporal range of the physical activity
        note: can be either <rangeOfDates> or <singleDateTime>}
    </temporalCoverage>

    <associatedParty>
        {the agent that is associated with the activity}
    </associatedParty>

    <protocol>
        {A protocol or SOP that is associated with the activity}
    </protocol>

    <sample>
        {Reference to the sample that served as the source in the acquiring
        activity}
    </sample>
</methodStep>
</methods>

```

EML Template 15 - Acquiring activity class, with its properties.

```

<methods>
    <methodStep>
        <description>
            TRANSPORTING
            {...
            + a description of the stored-conditions during the transporting activity}
        </description>

        <geographicCoverage>
            {Reference to the location or geographicCoverage of the physical activity}
        </geographicCoverage>
        <temporalCoverage>
            {Reference to the temporal range of the physical activity
            note: can be either <rangeOfDates> or <singleDateTime>}
        </temporalCoverage>

        <associatedParty>
            {the agent that is associated with the activity}
        </associatedParty>

        <protocol>
            {A protocol or SOP that is associated with the activity}
        </protocol>
    </methodStep>
</methods>

```

```

        </protocol>

        <sample>
            {Reference to the sample that served as the source in the transporting
            activity}
        </sample>
    </methodStep>
</methods>

```

EML Template 16 - Transporting activity class, with its properties.

```

<methods>
    <sampling>
        <samplingDescription>
            {...}
        </samplingDescription>
        <studyExtent></studyExtent>

        <geographicCoverage>
            {Reference to the location or geographicCoverage of the physical activity}
        </geographicCoverage>

        <temporalCoverage>
            {Reference to the temporal range of the physical activity
            note: can be either <rangeOfDates> or <singleDateTime>}
        </temporalCoverage>

        <associatedParty>
            {the agent that is associated with the activity}
        </associatedParty>

        <permit>
            {Reference to the permit that is associated with the sampling activity}
        </permit>

        <sample>
            {Reference to the sample that is generated in the sampling activity}
        </sample>

        <samplingInstrumentation>
            {Reference to the device that is used in the sampling activity}
        </samplingInstrumentation>
    </sampling>
</methods>

```

EML Template 17 - Sampling activity class, with its properties.

```

<methods>
    <methodStep>
        <description>
            OBSERVING

```



```

    {...}
  </description>

  <geographicCoverage>
    {Reference to the location or geographicCoverage of the physical activity}
  </geographicCoverage>

  <temporalCoverage>
    {Reference to the temporal range of the physical activity
     note: can be either <rangeOfDates> or <singleDateTime>}
  </temporalCoverage>

  <associatedParty>
    {the agent that is associated with the activity}
  </associatedParty>

  <protocol>
    {A protocol or SOP that is associated with the activity}
  </protocol>

  <permit>
    {Reference to the permit that is associated with the observing activity}
  </permit>

  <result>
    {Reference to the result that is generated in the observing activity}
  </result>

  <observingInstrumentation>
    {Reference to the device used in the observing activity}
  </observingInstrumentation>
</methodStep>
</methods>

```

EML Template 18 - Observing activity class, with its properties.

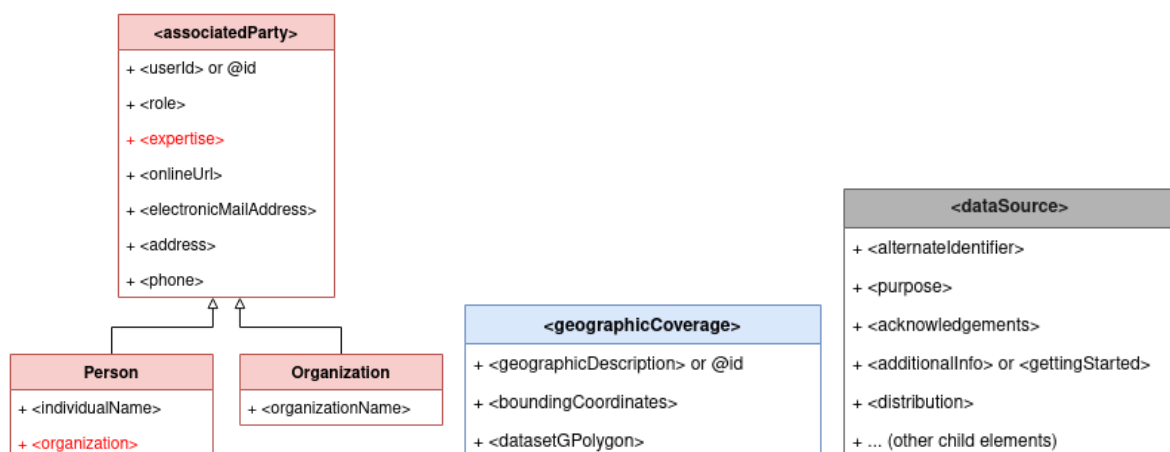


Figure 7.1 - Class diagram of the EML implementation of Agent and Location classes and their properties.

```
<associatedParty id="...">
  <userId>{Identifier(s) of the organization and/or people agent(s)}</userId>
  <individualName>{The name of the agent}</individualName>
  <organization>{The organization to which the agent is associated with respect to the
  activity / entity}</organization>
  <role>{The role of the agent with respect to the activity / entity}</role>
  <expertise>{The domains of expertise of the agent}</expertise>
  <onlineUrl>{A link to information about the agent, usually a (personal) website}</onlineUrl>
  <electronicMailAddress>{The email address of the agent}</electronicMailAddress>
  <address>{The address of the agent}</address>
  <phone>{The contact telephone number of the agent}</phone>
</associatedParty>
```

EML Template 19 - Person agent class with its properties.

```
<associatedParty id="...">
  <userId>{Identifier(s) of the organization and/or people agent(s)}</userId>
  <organizationName>{The name of the agent}</organizationName>
  <role>{The role of the agent with respect to the activity / entity}</role>
  <expertise>{The domains of expertise of the agent}</expertise>
  <onlineUrl>{A link to information about the agent, usually a (personal) website}</onlineUrl>
  <electronicMailAddress>{The email address of the agent}</electronicMailAddress>
  <address>{The address of the agent}</address>
  <phone>{The contact telephone number of the agent}</phone>
</associatedParty>
```

EML Template 20 - Organization agent class with its properties.

```
<geographicCoverage id="...">
  <geographicDescription>
    {A short description of the geographic area}
  </geographicDescription>
  <boundingCoordinates>
    {Reference to the bounding coordinates that cover the geographic area}
  </boundingCoordinates>
  <datasetGPolygon>
    {Reference to the doughnut shaped spatial ring that covers the geographic area}
  </datasetGPolygon>
</geographicCoverage>
```

EML Template 21 - Location class with its properties.

Appendix A: Template of RDF implementation, in turtle syntax:

```
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix schema: <http://schema.org/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix ssn: <http://www.w3.org/ns/ssn#> .
@prefix sosa: <http://www.w3.org/ns/ssn#> .
@prefix dwc: <http://rs.tdwg.org/dwc/terms/#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

ACTIVITIES

PHYSICAL ACTIVITIES

#TRANSPORTING#

```
<http://example.org/transport/transportExample>
  a prov:Activity, schema:Action ;
  dct:type "Transporting"^^xsd:string ;
```

```
  schema:object {the identifier of the sample/object that is transported} ;
  schema:description {A short description of transport storage conditions} ;
```

```
  prov:atLocation [
    a prov:Location, schema:Place ;
    schema:identifier {ID};
    schema:address {address} ;
  ] ;
  prov:startedAtTime {datetime} ;
  prov:endedAtTime {datetime} ;
```

```
  prov:wasAssociatedWith <http://example.org/person/1> ;
  prov:qualifiedAssociation [
    a prov:Association ;
    prov:agent <http://example.org/person/1> ;
    prov:hadRole {The role of the agent during the transport} ;
    rdfs:comment {Any other comments} ;
  ] ;
```

```
  prov:used <http://example.org/Protocol/1> ;
```

#PROCESSING#

```
<http://example.org/processing/processingExample>
  a prov:Activity, schema:Action ;
  dct:type "Processing"^^xsd:string ;
```

```
  schema:object {ID of the thing on which the processing activity is performed} ;
  schema:result {ID of the result output of this activity} ;
```

```

schema:description {material & chemical objects that helped the agent perform the action}
;
schema:instrument <http://example.org/device/2> ;
prov:used <http://example.org/device/2> ; #valid equivalent for schema:instrument

prov:atLocation <http://example.org/location-type2> ;
prov:startedAtTime {datetime} ;
prov:endedAtTime {datetime} ;

prov:wasAssociatedWith <http://example.org/person/2> ;
prov:qualifiedAssociation [
  a prov:Association ;
  prov:agent <http://example.org/person/2> ;
  prov:hadRole {The role of the agent during the transport} ;
  rdfs:comment {Any other comments} ;
];

prov:used <http://example.org/protocol/2> ;
.

```

#STORING#

```

<http://example.org/storage/storageExample>
  a prov:Activity, schema:Action ;
  dct:type "Storing"^^xsd:string ;

  schema:additionalType {the type of storage} ;

  schema:object {ID of the sample/object that is stored} ;
  schema:description {description of the storage conditions} ;
  schema:instrument <http://example.org/StorageDevice/1> ;
  prov:used <http://example.org/StorageDevice/1> ; #valid equivalent for
schema:instrument

  prov:atLocation <http://example.org/location-type1> ;
  prov:startedAtTime {datetime} ;
  prov:endedAtTime {datetime} ;

  prov:wasAssociatedWith <http://example.org/person/2> ;
  prov:qualifiedAssociation [
    a prov:qualifiedAssociation ;
    prov:agent <http://example.org/person/2> ;
    prov:hadRole {The role of the agent during the storage} ;
    rdfs:comment {Any other comments} ;
  ];
#Case of Bio-banking --> Agent associated with this activity is the bio-banking facility
prov:wasAssociatedWith <http://example.org/BiobankInstitute> ;
prov:qualifiedAssociation [

```

```
    a prov:qualifiedAssociation ;
    prov:agent <http://example.org/BiobankInstitute> ;
    prov:hadRole "Biobanking facility"^^xsd:string ;
    rdfs:comment {Any other comments} ;
];
```

```
prov:used <http://example.org/protocol/3> ;
```

#ACQUIRING#

```
<http://example.org/acquisition/acquisitionExample>
```

```
    a prov:Activity, schema:Action ;
    dct:type "Acquiring"^^xsd:string ;
```

```
    schema:object {ID of the thing that is acquired in this activity} ;
```

```
    prov:atLocation <http://example.org/location-type1> ;
    prov:startedAtTime {datetime} ;
    prov:endedAtTime {datetime} ;
```

```
    prov:wasAssociatedWith <http://example.org/person/4> ;
    prov:qualifiedAssociation [
        a prov:Association ;
        prov:agent <http://example.org/person/4> ;
        prov:hadRole {The role of the agent during the acquisition} ;
        rdfs:comment {Any other comments} ;
    ] ;
```

```
    prov:used <http://example.org/protocol/4> ;
```

#SAMPLING#

```
<http://example.org/Sampling/1>
```

```
    a prov:Activity, schema:Action, sosa:Sampling ;
    dct:type "Sampling"^^xsd:string ;
```

```
    sosa:madeBySampler <http://example.org/device/sampler_a> ;
```

```
    prov:used <http://example.org/permit/1> ;
```

```
    prov:generated <http://example.org/sample1> ;
```

```
    prov:atLocation <http://example.org/location-type1>, <http://example.org/location-type2> ;
    prov:startedAtTime {datetime} ;
    prov:endedAtTime {datetime} ;
```

```
    prov:wasAssociatedWith <http://example.org/person/3> ;
    prov:qualifiedAssociation [
```

```
    a prov:Association ;
    prov:agent <http://example.org/person/3> ;
    prov:hadRole {The role of the agent during the sampling} ;
    rdfs:comment {Any other comments} ;
];
```

```
prov:used <http://example.org/samplingprotocol/1> ;
```

#OBSERVING#

```
<http://example.org/Observing/1>
```

```
    a prov:Activity, schema:Action, sosa:Observation ;
    dct:type "Observation"^^xsd:string ;
```

```
sosa:madeBySensor <http://example.org/device/sensor_01> ;
```

```
prov:used <http://example.org/permit/2> ;
```

```
## to have look at!! ##
```

```
prov:generated <http://example.org/observation1> ;
sosa:hasResult { a sosa:Result... } ;
```

```
    prov:atLocation <http://example.org/location-type1>, <http://example.org/location-type2>,
<http://example.org/location-type3> ;
    prov:startedAtTime {datetime} ;
    prov:endedAtTime {datetime} ;
```

```
    prov:wasAssociatedWith <http://example.org/person/3> ;
    prov:qualifiedAssociation [
        a prov:Association ;
        prov:agent <http://example.org/person/3> ;
        prov:hadRole {The role of the agent during the observation} ;
        rdfs:comment {Any other comments} ;
    ] ;
```

```
prov:used <http://example.org/observationprotocol/1> ;
```

DIGITAL ACTIVITIES

#SOFTWARE PROCESSING#

```
<http://example.org/softwareprocessingExample1>
```

```
    a prov:Activity, schema:Action ;
    dct:type "Software Processing"^^xsd:string ;
```

```
    schema:name {name of the software processing activity} ;
    schema:description {description of the software processing activity} ;
    schema:object {ID of the thing/data that is processed in this activity} ;
```

to have a look at! ~ what is the exact meaning of this...

prov:generated {} ;

prov:used <http://example.org/software/1> ;

prov:wasAssociatedWith <http://example.org/person/3> ;

prov:qualifiedAssociation [

 a prov:Association ;

 prov:agent <http://example.org/person/3> ;

 prov:hadRole {The role of the agent during the software processing} ;

 rdfs:comment {Any other comments} ;

];

prov:used <http://example.org/protocol/softwareprotocol> ;

#DATA RETRIEVING#

<http://example.org/DataRetrieval/1>

 a prov:Activity, schema:Action ;

 dct:type "Data retrieving"^^xsd:string ;

 schema:description {description of the data retrieval} ;

 schema:identifier {id/name/code/... of the data that is retrieved} ;

 schema:target {the target EntryPoint, or url, for this Action, being a data retrieval activity} ;

prov:wasAssociatedWith <http://example.org/person/3> ;

prov:qualifiedAssociation [

 a prov:Association ;

 prov:agent <http://example.org/person/3> ;

 prov:hadRole {The role of the agent during the data retrieval} ;

 rdfs:comment {Any other comments} ;

];

prov:used <http://example.org/protocol/softwareprotocol> ;

#####LOCATION#####

<http://example.org/location-type1>

 a prov:Location, schema:PostalAddress;

 schema:addressCountry {the country name of the location} ;

 schema:postalCode {the postal code of the location} ;

 schema:streetAddress {the street address of the location} ;

<http://example.org/location-type2>

 a prov:Location, schema:Place ;

schema:name {ISO country name} ;
schema:latitude {the latitude of the location} ;
schema:longitude {the longitude of the location} ;
owl:sameAs {MRGID url} ;

.
<http://example.org/location-type3>
a prov:Location, dct:Location ;

locn:geometry {geometry could be encoded as WKT (geosparql:asWKT)} ;
dcat:bbox {bbox description of the location} ;
dcat:centroid {centroid description of the location} ;
.

#####AGENT#####

<http://example.org/person/1>
a prov:Person, schema:Person ;

schema:identifier {orcid} ; #Katrina will dig through homework to check how orcid is
referenced

schema:givenName {firstname} ;
schema:surName {surname} ;
schema:email {mail} ;
schema:knowsAbout {domains of expertise} ;
prov:hadRole [
a prov:Role ;
dct:type {role during activity} ;
] ;
.

<http://example.org/person/2>
a prov:Person, schema:Person ;

schema:identifier {orcid} ;
schema:givenName {firstname} ;
schema:surName {surname} ;
schema:email {mail} ;
schema:knowsAbout {domains of expertise} ;
prov:hadRole [
a prov:Role ;
dct:type {role during Biobanking activity} ;
] ;
schema:affiliation <http://example.org/BiobankInstitute> ;
.

<http://example.org/person/3>
a prov:Person, schema:Person ;

```

schema:identifier {orcid} ;
schema:giveName {firstname} ;
schema:surName {surname} ;
schema:email {mail} ;
schema:knowsAbout {domains of expertise} ;
prov:hadRole [
    a prov:Role ;
    dct:type {role during activity} ;
] ;
schema:affiliation {organization that person works for} ;
.

```

```

<http://example.org/person/4>
    a prov:Person, schema:Person ;

```

```

schema:identifier {orcid} ;
schema:giveName {firstname} ;
schema:surName {surname} ;
schema:email {mail} ;
prov:hadRole [
    a prov:Role ;
    dct:type {role during activity} ;
] ;
.

```

```

<http://example.org/BiobankInstitute>
    a prov:Agent, schema:Organization ;

```

```

schema:identifier {ROR ID, ...} ;
schema:name {institute name} ;
schema:email {mail} ;
schema:telephone {institute telephone number} ;
schema:address {institute address} ;
.

```

ENTITIES

#PROTOCOL#

```

<http://example.org/samplingprotocol/1>
    a schema:Thing, prov:Entity, schema:CreativeWork ;

```

```

schema:identifier {identifier of the protocol url/code/...} ;
schema:name {name given to the protocol} ;
schema:description {the description of the protocol} ;
schema:url {url of the location of the protocol} ;
schema:accessibilitySummary {description of how the protocol can be accessed} ;
rdfs:comment {description of any deviations from the protocol that occurred} ;
.

```

```

<http://example.org/Protocol/1>
  a schema:Thing, prov:Entity, schema:CreativeWork ;

  schema:identifier {identifier of the protocol url/code/...} ;
  schema:name {protocol 1} ;
  schema:description {this is the description of the protocol used during the transport activity
... } ;
  schema:url {url of the location of the protocol} ;
  schema:accessibilitySummary {description of how the protocol can be accessed} ;
  rdfs:comment {description of any deviations from the protocol that occurred} ;
.

```

#DEVICE#

```

<http://example.org/Instrument/1>
  a schema:Thing, prov:Entity ;

  dct:type {type of device} ;
  schema:identifier {ID of the device/instrument} ;
  schema:name {name of the device} ;
  schema:description {the description/listing of the configuration settings of the device} ;

  #in case the device has relevant software settings / uses relevant software
  prov:wasInfluencedBy <http://example.org/software/1> ;
.

```

```

<http://example.org/StorageDevice/1>
  a schema:Thing, prov:Entity ;

  dct:type {type of device} ;
  schema:identifier {ID of the device} ;
  schema:name {name of the device} ;
  schema:description {the description/listing of the configuration settings of the device} ;

  #in case the (storage) device has relevant software settings / uses relevant software
  prov:wasInfluencedBy <http://example.org/software/1> ;
.

```

###note: use of either SamplingDevice (sosa:Sampler) or ObservingDevice (sosa:Sensor) is recommended

#SamplingDevice#

```

<http://example.org/device/sampler_a>
  a schema:Thing, prov:Entity, sosa:Sampler ;

  dct:type {the type of sampler} ;
  schema:identifier {ID of the sampler} ;
  schema:name {name of the sampler} ;

```

schema:description {the description/listing of the configuration settings} ;
prov:wasInfluencedBy <http://example.org/software/1> ;

#Case when device is hosted on a platform:
sosa:isHostedBy <http://example.org/Platform/A> ;

#ObservingDevice#

<http://example.org/device/sensor_01>

a schema:Thing, prov:Entity, sosa:Sensor ;

dct:type {the type of sensor} ;
schema:identifier {ID of the sensor} ;
schema:name {name of the sensor} ;
schema:description {the description/listing of the configuration settings} ;
prov:wasInfluencedBy <http://example.org/software/1> ;

#Case when device is hosted on a platform:
sosa:isHostedBy <http://example.org/Platform/A> ;

#PLATFORM#

<http://example.org/Platform/A>

a schema:Thing, prov:Entity, sosa:Platform ;

schema:identifier {the universal/standard identifier or code of the platform} ;
schema:name {A locally used name of the platform} ;
dct:type {name for the type of platform} ;

#SAMPLE#

<http://example.org/sample1>

a schema:Thing, prov:Entity, sosa:Sample;

schema:identifier {ID of the sample} ;
schema:name {name of the sample} ;
schema:description {a description of the sample} ;
schema:keywords {keywords that are applicable to this sample (e.g. ENVOterms, ...)} ;
schema:size {size of the sample} ;

prov:wasAssociatedWith {agent with which this sample is associated} ;

#SOFTWARE#

#Software application

<http://example.org/software/1>

a prov:Entity, schema:CreativeWork, schema:SoftwareApplication ;

schema:name {name of the software application} ;
schema:description {description of the software application} ;
schema:version {version of the software application} ;
schema:downloadUrl {downloadUrl of the software application} ;
schema:featureList {Features or modules provided by the software application} ;

#Software source code

<http://example.org/software/1>

a prov:Entity, schema:CreativeWork, schema:SoftwareSourceCode ;

schema:name {name of the software source code} ;
schema:description {description of the software source code} ;
schema:version {version of the software source code} ;
schema:codeRepository {url to the repository where the code and related code is located};

#in case schema:SoftwareApplication or schema:SoftwareSourceCode do not match/apply to the used software

<http://example.org/software/1>

a prov:Entity, schema:CreativeWork ;

schema:name {name of the software} ;
schema:description {description of the software} ;
schema:version {version of the software} ;
schema:url {url of the software - to reference page / download page / ...} ;

#PERMIT#

<http://example.org/Permit/1>

a schema:Thing, prov:Entity, schema:CreativeWork ;

schema:identifier {ID of the permit; can be the IRCC or proof of due diligence} ;
schema:name {name of the permit} ;
schema:description {description of the permit} ;

prov:generatedAtTime {datetime when permit was created and also became valid} ;
prov:invalidatedAtTime {datetime when permit became invalid} ;

#case of an ABS permit

<http://example.org/Permit/ABS>

a schema:Thing, prov:Entity, schema:CreativeWork ;

schema:identifier {ID of the permit; can be the IRCC or proof of due diligence} ;
schema:name {name of the permit} ;
schema:description {description of the permit} ;

prov:generatedAtTime {datetime when permit was created and valid} ;
prov:invalidatedAtTime {datetime when permit became invalid} ;

schema:spatial [
 a prov:Location, schema:Place ;
 schema:identifier {url identifier of the location} ;
 schema:geo {when applicable, the geo coordinates or geo shape of the location} ;
 schema:address {when applicable, the address of the location} ;
];

#Agent: contact point

prov:wasAssociatedWith [
 a schema:Person ;
 schema:identifier {orcid of the person} ;
 schema:givenName {first name of the person} ;
 schema:surName {surname of the person} ;
 schema:email {mail address of the person} ;
 schema:knowsAbout {domains of expertise of the person} ;
];

prov:qualifiedAttribution [
 a prov:Attribution ;
 prov:agent {id of the agent - orcid, ...} ;
 dcat:hadRole "contact" ;
 rdfs:comment {any other comments} ;
];

#Agent: authorizing agent

prov:wasAssociatedWith [#or use a local identifier here...

 a schema:Organization ;
 schema:identifier {orcid} ;
 schema:name {institute name} ;
 schema:email {mail} ;
 schema:knowsAbout {domains of expertise} ;
];

prov:qualifiedAttribution [
 a prov:Attribution ;
 prov:agent {id of the agent - orcid, ...} ;
 dcat:hadRole "Authorizing agent" ;
 rdfs:comment {any other comments} ;
];

#to indicate entities associated with this permit (e.g. collected samples)

prov:wasInfluencedBy <http://example.org/sample1>, <http://example.org/sample2>,
<http://example.org/sample3> ;

.

Appendix B: *Example from MOC...*

Appendix C: SHACL Shapes of provenance information groups

C.1 Entity shapes

```
_:ProvPlatformShape
  a sh:NodeShape ;
  sh:targetClass prov:Entity, schema:Thing, sosa:Platform ;
  sh:property [
    sh:path schema:identifier ;
    sh:maxCount 1 ;
    sh:or (
      [
        sh:datatype xsd:string ;
      ]
      [
        sh:datatype xsd:anyURI ;
      ]
    )
  ];
  sh:property [
    sh:path schema:name ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
  ];
  sh:property [
    sh:path dct:type ;
    sh:datatype xsd:string ;
  ];

  #sh:closed true ;
  #sh:ignoredProperties ( rdf:type )
.

_:ProvSoftwareShape
  a sh:NodeShape ;
  sh:targetClass prov:Entity, schema:Thing, schema:CreativeWork ;
  sh:property [
    sh:path schema:name ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
```



```

];
  sh:property [
    sh:path schema:description ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
  ];
  sh:property [
    sh:path dct:version ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
  ];
  sh:property [
    sh:path schema:url ;
    sh:datatype xsd:anyURI ;
  ];
];
.

```

```

_:ProvSoftwareApplicationShape
  a sh:NodeShape ;
  sh:targetClass prov:Entity, schema:Thing, schema:SoftwareApplication ;
  sh:property [
    sh:path schema:name ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
  ];
  sh:property [
    sh:path schema:description ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
  ];
  sh:property [
    sh:path dct:version ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
  ];
  sh:property [
    sh:path schema:downloadUrl ;
    sh:datatype xsd:anyURI ;
  ];
  sh:property [
    sh:path schema:featureList ;
    sh:datatype xsd:string ;
  ];
];
.

```

```

_:ProvSoftwareCodeShape
  a sh:NodeShape ;
  sh:targetClass prov:Entity, schema:Thing, schema:SoftwareSourceCode ;

```

```

sh:property [
    sh:path schema:name ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];

sh:property [
    sh:path schema:description ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];

sh:property [
    sh:path dct:version ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];

sh:property [
    sh:path schema:codeRepository ;
    sh:datatype xsd:anyURI ;
];
.

_:ProvSampleShape
    a sh:NodeShape ;
sh:targetClass prov:Entity, schema:Thing, sosa:Sample ;
sh:property [
    sh:path schema:identifier ;
    sh:maxCount 1 ;
    sh:or (
        [
            sh:datatype xsd:string ;
        ]
        [
            sh:datatype xsd:anyURI ;
        ]
    )
];

sh:property [
    sh:path schema:name ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];

sh:property [
    sh:path schema:description ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];

sh:property [
    sh:path schema:keywords ;

```

```

        sh:or (
        [
            sh:datatype xsd:string ;
        ]
        [
            sh:datatype xsd:anyURI ;
        ]
        )
];
    sh:property [
        sh:path schema:size ;
        sh:datatype xsd:double ;
    ];
.

_:ProvDeviceShape
    a sh:NodeShape ;
sh:targetClass prov:Entity, schema:Thing ;
sh:property [
    sh:path schema:identifier ;
    sh:maxCount 1 ;
    sh:or (
        [
            sh:datatype xsd:string ;
        ]
        [
            sh:datatype xsd:anyURI ;
        ]
    )
];
    sh:property [
        sh:path dct:type ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path schema:name ;
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path schema:description ;
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path prov:wasInfluencedBy ;
        sh:or (
            [

```

```

        sh:class prov:Entity, schema:Thing, schema:CreativeWork ;
    ]
    [
        sh:class prov:Entity, schema:Thing, schema:SoftwareApplication ;
    ]
    [
        sh:class prov:Entity, schema:Thing, schema:SoftwareSourceCode ;
    ]
    )
];
.

```

_:ProvSamplingDeviceShape

```

    a sh:NodeShape ;
    sh:targetClass prov:Entity, schema:Thing, sosa:Sampler ;
    sh:property [
        sh:path schema:identifier ;
        sh:maxCount 1 ;
        sh:or (
            [
                sh:datatype xsd:string ;
            ]
            [
                sh:datatype xsd:anyURI ;
            ]
        )
    ];
    sh:property [
        sh:path dct:type ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path schema:name ;
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path schema:description ;
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path prov:wasInfluencedBy ;
        sh:or (
            [
                sh:class prov:Entity, schema:Thing, schema:CreativeWork ;
            ]
        )
    ];

```

```

    [
        sh:class prov:Entity, schema:Thing, schema:SoftwareApplication ;
    ]
    [
        sh:class prov:Entity, schema:Thing, schema:SoftwareSourceCode ;
    ]
)
];
sh:property [
    sh:path sosa:isHostedBy ;
    sh:class prov:Entity, schema:Thing, sosa:Platform ;
];
.

```

```

_:ProvObservingDeviceShape
    a sh:NodeShape ;
sh:targetClass prov:Entity, schema:Thing, sosa:Sensor ;
sh:property [
    sh:path schema:identifier ;
    sh:maxCount 1 ;
    sh:or (
        [
            sh:datatype xsd:string ;
        ]
        [
            sh:datatype xsd:anyURI ;
        ]
    )
];
sh:property [
    sh:path dct:type ;
    sh:datatype xsd:string ;
];
sh:property [
    sh:path schema:name ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];
sh:property [
    sh:path schema:description ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];
sh:property [
    sh:path prov:wasInfluencedBy ;
    sh:or (
        [
            sh:class prov:Entity, schema:Thing, schema:CreativeWork ;

```

```

    ]
    [
        sh:class prov:Entity, schema:Thing, schema:SoftwareApplication ;
    ]
    [
        sh:class prov:Entity, schema:Thing, schema:SoftwareSourceCode ;
    ]
    )
];
    sh:property [
        sh:path sosa:isHostedBy ;
        sh:class prov:Entity, schema:Thing, sosa:Platform ;
    ];
.

_:ProvProtocolShape
    a sh:NodeShape ;
    sh:targetClass prov:Entity, schema:Thing, schema:CreativeWork ;
    sh:property [
        sh:path schema:identifier ;
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path schema:name ;
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path schema:description ;
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path schema:url ;
        sh:maxCount 1 ;
        sh:datatype xsd:anyURI ;
    ];
    sh:property [
        sh:path schema:accessibilitySummary ;
        sh:maxCount 1 ;
        sh:datatype xsd:anyURI ;
    ];
    sh:property [
        sh:path rdfs:comment ;
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
    ];
];

```

```

.

_:ProvPermitShape
  a sh:NodeShape ;
  sh:targetClass prov:Entity, schema:Thing, schema:CreativeWork ;
  sh:property [
    sh:path schema:identifier ;
    sh:maxCount 1 ;
    sh:or (
      [
        sh:datatype xsd:integer ;
      ]
      [
        sh:datatype xsd:string ;
      ]
      [
        sh:datatype xsd:anyURI ;
      ]
    )
  ];
  sh:property [
    sh:path schema:name ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
  ];
  sh:property [
    sh:path schema:description ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
  ];
  sh:property [
    sh:path prov:generatedAtTime ;
    sh:maxCount 1 ;
    sh:datatype xsd:dateTime ;
  ];
  sh:property [
    sh:path prov:invalidatedAtTime ;
    sh:maxCount 1 ;
    sh:datatype xsd:dateTime ;
  ];
];
.

```

```

_:ProvABSPermitShape
  a sh:NodeShape ;
  sh:targetClass prov:Entity, schema:Thing, schema:CreativeWork ;
  sh:property [
    sh:path schema:identifier ;
    sh:maxCount 1 ;
  ];
.

```

```

    sh:or (
    [
        sh:datatype xsd:integer ;
    ]
    [
        sh:datatype xsd:string ;
    ]
    [
        sh:datatype xsd:anyURI ;
    ]
    )
];

sh:property [
    sh:path schema:name ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];

sh:property [
    sh:path schema:description ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];

sh:property [
    sh:path prov:generatedAtTime ;
    sh:maxCount 1 ;
    sh:datatype xsd:dateTime ;
];

sh:property [
    sh:path prov:invalidatedAtTime ;
    sh:maxCount 1 ;
    sh:datatype xsd:dateTime ;
];

sh:property [
    sh:path schema:spatial ;
    sh:class prov:Location, schema:Place ;
];

sh:property [
    sh:path prov:wasAssociatedWith ;
    sh:class prov:Agent, prov:Person, schema:Person ;
    ## need to be able to define restrictions on the Agent's role
];

sh:property [
    sh:path prov:qualifiedAttribution ;
    sh:class prov:Attribution ;
    ## need to be able to define restrictions on the Agent's role
];

sh:property [
    sh:path prov:wasInfluencedBy ;

```



```

    sh:class prov:Entity, schema:Thing, sosa:Sample ;
];

```

C.2 Activity shapes

Activities

_:ProvSoftwareProcessingShape

```

    a sh:NodeShape ;
    sh:targetClass prov:Activity, schema:Action ;
    sh:property [
        sh:path dct:type ;
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path prov:wasAssociatedWith ;
        sh:or (
            [
                sh:class prov:Agent, prov:Person, schema:Person ;
            ]
            [
                prov:Agent, prov:Organization, schema:Organization ;
            ]
        )
    ];
    sh:property [
        sh:path prov:qualifiedAssociation ;
        sh:class prov:Association ;
        ## how to add restrictions on the prov:Association class?
    ];
    sh:property [
        sh:path prov:used ;
        sh:class prov:Entity, schema:Thing, schema:CreativeWork ;
        ## check if other Entity cannot be defined by these 3 classes...
    ];
    sh:property [
        sh:path schema:name ;
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path schema:description ;
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path schema:object ;

```

```

    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];
    sh:property [
      sh:path prov:generated ;
      sh:maxCount 1 ;
      sh:or (
        [
          sh:datatype xsd:dateTime ;
        ]
        [
          sh:datatype xsd:anyURI ;
        ]
      )
];
    sh:property [
      sh:path prov:used ;
      sh:or (
        [
          sh:class prov:Entity, schema:Thing, schema:CreativeWork ;
        ]
        [
          sh:class prov:Entity, schema:Thing, schema:SoftwareApplication ;
        ]
        [
          sh:class prov:Entity, schema:Thing, schema:SoftwareSourceCode ;
        ]
      )
];
.

```

_:ProvDataRetrievingShape

```

  a sh:NodeShape ;
  sh:targetClass prov:Activity, schema:Action ;
  sh:property [
    sh:path dct:type ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
  ];
  sh:property [
    sh:path prov:wasAssociatedWith ;
    sh:or (
      [
        sh:class prov:Agent, prov:Person, schema:Person ;
      ]
      [
        prov:Agent, prov:Organization, schema:Organization ;
      ]
    )
  ]

```

```

    )
];
  sh:property [
    sh:path prov:qualifiedAssociation ;
    sh:class prov:Association ;
];
  sh:property [
    sh:path prov:used ;
    sh:class prov:Entity, schema:Thing, schema:CreativeWork ;
];
  sh:property [
    sh:path schema:url ;
    sh:datatype xsd:anyURI ;
];
  sh:property [
    sh:path schema:target ;
    sh:datatype xsd:anyURI ;
];
.

```

```

_:ProvProcessingActivityShape
  a sh:NodeShape ;
  sh:targetClass prov:Activity, schema:Action ;
  sh:property [
    sh:path dct:type ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];
  sh:property [
    sh:path prov:wasAssociatedWith ;
    sh:or (
      [
        sh:class prov:Agent, prov:Person, schema:Person ;
      ]
      [
        prov:Agent, prov:Organization, schema:Organization ;
      ]
    )
];
  sh:property [
    sh:path prov:qualifiedAssociation ;
    sh:class prov:Association ;
];
  sh:property [
    sh:path prov:used ;
    sh:class prov:Entity, schema:Thing, schema:CreativeWork ;
];

```

```

    sh:property [
      sh:path prov:atLocation ;
      sh:class prov:Location, schema:Place ;
];
    sh:property [
      sh:path prov:startedAtTime ;
      sh:maxCount 1 ;
      sh:datatype xsd:dateTime ;
];
    sh:property [
      sh:path prov:endedAtTime ;
      sh:maxCount 1 ;
      sh:datatype xsd:dateTime ;
];
    sh:property [
      sh:path schema:object ;
      sh:class prov:Entity, schema:Thing, sosa:Sample ;
];
    sh:property [
      sh:path schema:result ;
      sh:or (
        [
          sh:datatype xsd:string ;
        ]
        [
          sh:datatype xsd:anyURI ;
        ]
      )
];
    sh:property [
      sh:path schema:description ;
      sh:maxCount 1 ;
      sh:datatype xsd:string ;
];
    sh:property [
      sh:path prov:used ;
      sh:or (
        [
          sh:class prov:Entity, schema:Thing ;
        ]
        [
          sh:class prov:Entity, schema:Thing, sosa:Sampler ;
        ]
        [
          sh:class prov:Entity, schema:Thing, sosa:Sensor ;
        ]
      )
];

```

];

_:ProvStoringActivityShape

 a sh:NodeShape ;

 sh:targetClass prov:Activity, schema:Action ;

 sh:property [

 sh:path dct:type ;

 sh:maxCount 1 ;

 sh:datatype xsd:string ;

];

 sh:property [

 sh:path prov:wasAssociatedWith ;

 sh:or (

 [

 sh:class prov:Agent, prov:Person, schema:Person ;

]

 [

 prov:Agent, prov:Organization, schema:Organization ;

]

)

];

 sh:property [

 sh:path prov:qualifiedAssociation ;

 sh:class prov:Association ;

];

 sh:property [

 sh:path prov:used ;

 sh:class prov:Entity, schema:Thing, schema:CreativeWork ;

];

 sh:property [

 sh:path prov:atLocation ;

 sh:class prov:Location, schema:Place ;

];

 sh:property [

 sh:path prov:startedAtTime ;

 sh:maxCount 1 ;

 sh:datatype xsd:dateTime ;

];

 sh:property [

 sh:path prov:endedAtTime ;

 sh:maxCount 1 ;

 sh:datatype xsd:dateTime ;

];

 sh:property [

 sh:path schema:additionalType ;

 sh:datatype xsd:string ;

```

];
  sh:property [
    sh:path schema:object ;
    sh:or (
      [
        sh:datatype xsd:string ;
      ]
      [
        sh:datatype xsd:anyURI ;
      ]
      [
        sh:class prov:Entity, schema:Thing, sosa:Sample ;
      ]
    )
];
  sh:property [
    sh:path schema:description ;
    sh:datatype xsd:string ;
];
  sh:property [
    sh:path prov:used ;
    sh:or (
      [
        sh:class prov:Entity, schema:Thing ;
      ]
      [
        sh:class prov:Entity, schema:Thing, sosa:Sampler ;
      ]
      [
        sh:class prov:Entity, schema:Thing, sosa:Sensor ;
      ]
    )
];
.

```

```

_:ProvAcquiringActivityShape
  a sh:NodeShape ;
  sh:targetClass prov:Activity, schema:Action ;
  sh:property [
    sh:path dct:type ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];
  sh:property [
    sh:path prov:wasAssociatedWith ;
    sh:or (

```

```

    [
        sh:class prov:Agent, prov:Person, schema:Person ;
    ]
    [
        prov:Agent, prov:Organization, schema:Organization ;
    ]
)
];
sh:property [
    sh:path prov:qualifiedAssociation ;
    sh:class prov:Association ;
];
sh:property [
    sh:path prov:used ;
    sh:class prov:Entity, schema:Thing, schema:CreativeWork ;
];
sh:property [
    sh:path prov:atLocation ;
    sh:class prov:Location, schema:Place ;
];
sh:property [
    sh:path prov:startedAtTime ;
    sh:maxCount 1 ;
    sh:datatype xsd:dateTime ;
];
sh:property [
    sh:path prov:endedAtTime ;
    sh:maxCount 1 ;
    sh:datatype xsd:dateTime ;
];
sh:property [
    sh:path schema:object ;
    sh:class prov:Entity, schema:Thing, sosa:Sample ;
];
.

```

```

_:ProvTransportingActivityShape
    a sh:NodeShape ;
    sh:targetClass prov:Activity, schema:Action ;
    sh:property [
        sh:path dct:type ;
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path prov:wasAssociatedWith ;
        sh:or (

```

```

    [
        sh:class prov:Agent, prov:Person, schema:Person ;
    ]
    [
        prov:Agent, prov:Organization, schema:Organization ;
    ]
)
];
sh:property [
    sh:path prov:qualifiedAssociation ;
    sh:class prov:Association ;
];
sh:property [
    sh:path prov:used ;
    sh:class prov:Entity, schema:Thing, schema:CreativeWork ;
];
sh:property [
    sh:path prov:atLocation ;
    sh:class prov:Location, schema:Place ;
];
sh:property [
    sh:path prov:startedAtTime ;
    sh:maxCount 1 ;
    sh:datatype xsd:dateTime ;
];
sh:property [
    sh:path prov:endedAtTime ;
    sh:maxCount 1 ;
    sh:datatype xsd:dateTime ;
];
sh:property [
    sh:path schema:object ;
    sh:or (
        [
            sh:datatype xsd:string ;
        ]
        [
            sh:datatype xsd:anyURI ;
        ]
    )
];
sh:property [
    sh:path schema:description ;
    sh:datatype xsd:string ;
];

```



```

_:ProvSamplingActivityShape
  a sh:NodeShape ;
  sh:targetClass prov:Activity, schema:Action, sosa:Sampling ;
  sh:property [
    sh:path dct:type ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
  ];
  sh:property [
    sh:path prov:wasAssociatedWith ;
    sh:or (
      [
        sh:class prov:Agent, prov:Person, schema:Person ;
      ]
      [
        prov:Agent, prov:Organization, schema:Organization ;
      ]
    )
  ];
  sh:property [
    sh:path prov:qualifiedAssociation ;
    sh:class prov:Association ;
  ];
  sh:property [
    sh:path prov:used ;
    sh:class prov:Entity, schema:Thing, schema:CreativeWork ;
  ];
  sh:property [
    sh:path prov:atLocation ;
    sh:class prov:Location, schema:Place ;
  ];
  sh:property [
    sh:path prov:startedAtTime ;
    sh:maxCount 1 ;
    sh:datatype xsd:dateTime ;
  ];
  sh:property [
    sh:path prov:endedAtTime ;
    sh:maxCount 1 ;
    sh:datatype xsd:dateTime ;
  ];
  sh:property [
    sh:path sosa:madeBySampler ;
    sh:class prov:Entity, schema:Thing, sosa:Sampler ;
  ];
  sh:property [
    sh:path prov:used ;
    sh:class prov:Entity, schema:Thing, schema:CreativeWork ;
  ];

```

```

];
    ## need to check if this is correct!!
    sh:or (
      [
        sh:path prov:generated ;
        sh:class prov:Entity, schema:Thing, sosa:Sample ;
      ]
      [
        sh:path sosa:hasResult ;
        sh:class prov:Entity, schema:Thing, sosa:Sample ;
      ]
    )
.

```

```

_:ProvObservingActivityShape
  a sh:NodeShape ;
  sh:targetClass prov:Activity, schema:Action, sosa:Observation ;
  sh:property [
    sh:path dct:type ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
  ];
  sh:property [
    sh:path prov:wasAssociatedWith ;
    sh:or (
      [
        sh:class prov:Agent, prov:Person, schema:Person ;
      ]
      [
        prov:Agent, prov:Organization, schema:Organization ;
      ]
    )
  ];
  sh:property [
    sh:path prov:qualifiedAssociation ;
    sh:class prov:Association ;
  ];
  sh:property [
    sh:path prov:used ;
    sh:class prov:Entity, schema:Thing, schema:CreativeWork ;
  ];
  sh:property [
    sh:path prov:atLocation ;
    sh:class prov:Location, schema:Place ;
  ];
  sh:property [

```

```

        sh:path prov:startedAtTime ;
        sh:maxCount 1 ;
        sh:datatype xsd:dateTime ;
];
    sh:property [
        sh:path prov:endedAtTime ;
        sh:maxCount 1 ;
        sh:datatype xsd:dateTime ;
];
    sh:property [
        sh:path sosa:madeBySampler ;
        sh:class prov:Entity, schema:Thing, sosa:Sensor ;
];
    sh:property [
        sh:path prov:used ;
        sh:class prov:Entity, schema:Thing, schema:CreativeWork ;
];
    ## need to check if this is correct!!
    sh:or (
        [
            sh:path prov:generated ;
            sh:class prov:Entity, schema:Thing, sosa:Sample ;
        ]
        [
            sh:path sosa:hasResult ;
            sh:class prov:Entity, schema:Thing, sosa:Sample ;
        ]
    )
.

```

C.3 Agent shapes

```

_:ProvPersonShape
  a sh:NodeShape ;
  sh:targetClass prov:Agent, prov:Person, schema:Person ;
  sh:property [
    sh:path schema:identifier ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
  ];
  sh:property [
    sh:path schema:name ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
  ];
  sh:property [

```

```

        sh:path schema:knowsAbout ;
        sh:datatype xsd:string ;
];
    sh:property [
        sh:path schema:mainEntityOfPage ;
        sh:datatype xsd:anyURI ;
    ];
    sh:property [
        sh:path schema:email ;
        sh:datatype xsd:anyURI ;
    ];
    sh:property [
        sh:path schema:address ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path schema:telephone ;
        sh:datatype xsd:string ;
    ];
    sh:property [
        sh:path schema:affiliation ;
        sh:maxCount 1 ;
        sh:datatype xsd:string ;
    ];
    sh:closed true ;
    sh:ignoredProperties ( rdf:type )
.

```

_:ProvOrganizationShape

```

a sh:NodeShape ;
sh:targetClass prov:Agent, prov:Organization, schema:Organization ;
sh:property [
    sh:path schema:identifier ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];
sh:property [
    sh:path schema:name ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
];
sh:property [
    sh:path schema:knowsAbout ;
    sh:datatype xsd:string ;
];
sh:property [
    sh:path schema:mainEntityOfPage ;
    sh:datatype xsd:anyURI ;
];

```

```

];
  sh:property [
    sh:path schema:email ;
    sh:datatype xsd:anyURI ;
  ];
  sh:property [
    sh:path schema:address ;
    sh:datatype xsd:string ;
  ];
  sh:property [
    sh:path schema:telephone ;
    sh:datatype xsd:string ;
  ];
  sh:closed true ;
  sh:ignoredProperties ( rdf:type )
.

```

C.4 Location Shape

```

_:ProvLocationShape
  a sh:NodeShape ;
  sh:targetClass prov:Location, schema:Place ;
  sh:property [
    sh:path schema:identifier ;
    sh:maxCount 1 ;
    sh:or (
      [
        sh:datatype xsd:string ;
      ]
      [
        sh:datatype xsd:anyURI ;
      ]
    )
  ];
  sh:property [
    sh:path schema:geo ;
    sh:or (
      [
        sh:class schema:GeoCoordinates
      ]
      [
        sh:class schema:GeoShape
      ]
    )
  ];

```

```
sh:property [  
  sh:path schema:address ;  
  sh:datatype xsd:string ;
```

```
];
```

```
.
```