1 ⟨Instrument⟩ Pattern

1.1 Description

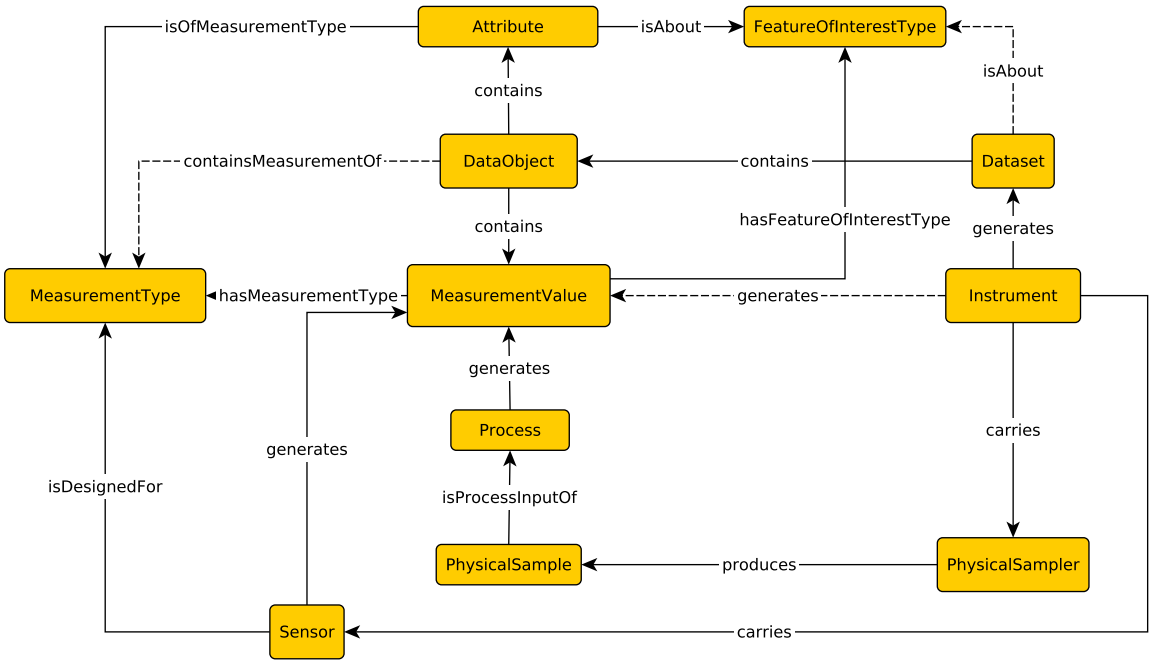

Figure 1.1: ⟨Instrument⟩ pattern

The ⟨Instrument⟩ pattern is designed to capture the relations among instruments, acquired measurement values, measurement types, datasets, and other concepts. In this pattern, an Instrument refers to a device used in scientific experiments to perform measurements and collect data. One Instrument can carry multiple Sensors and PhysicalSamplers, which can generate MeasurementValues. The generated MeasurementValues are contained in DataObjects which are further contained in a Dataset.

1.2 Competency Questions

The following competency questions have been proposed to stimulate the development of the ⟨Instrument⟩ pattern.

1. "Find me abundance of Euphausia pacifica sampled with MOCNESS."
2. "Give me calibration data for Durafet-type pH meters."
3. "Show me all cruises that deployed CTD."

1.3 Axiomatization

1.3.1 List of IRIs

Entity IRIs in the ⟨Instrument⟩ pattern are given as follows.

```
PREFIX glinstrument:  <http://schema.geolink.org/instrument#>
```

(1) Instrument (class): `glinstrument:Instrument`
(2) PhysicalSampler (class): `glinstrument:PhysicalSampler`
(3) Dataset (class): `glinstrument:Dataset`
(4) PhysicalSample (class): `glinstrument:PhysicalSample`
(5) Process (class): `glinstrument:Process`
(6) MeasurementValue (class): `glinstrument:MeasurementValue`
(7) MeasurementType (class): `glinstrument:MeasurementType`
(8) Sensor (class): `glinstrument:Sensor`
(9) DataObject (class): `glinstrument:DataObject`
(10) Attribute (class): `glinstrument:Attribute`
(11) FeatureOfInterestType (class): `glinstrument:FeatureOfInterestType`
(12) carries (object property): `glinstrument:carries`
(13) produces (object property): `glinstrument:produces`
(14) isProcessInputOf (object property): `glinstrument:isProcessInputOf`
(15) generates (object property): `glinstrument:generates`
(16) isDesignedFor (object property): `glinstrument:isDesignedFor`
(17) hasMeasurementType (object property): `glinstrument:hasMeasurementType`
(18) containMeasurementOf (object property): `glinstrument:containMeasurementOf`
(19) isOfMeasurementType (object property): `glinstrument:isOfMeasurementType`
(20) isAbout (object property): `glinstrument:isAbout`
(21) hasFeatureOfInterestType (object property): `glinstrument:hasFeatureOfInterestType`
(22) contains (object property): `glinstrument:contains`

1.3.2 Core Axioms

An Instrument can be considered as a carrier for Sensors or PhysicalSamplers.

$$\text{Instrument} \sqsubseteq \exists \text{carries.PhysicalSampler} \sqcup \exists \text{carries.Sensor} \tag{1.1}$$

An Instrument can also be associated with its generated MeasurementValues. This association can be through the PhysicalSampler which produces PhysicalSamples, or through the Sensor which generates MeasurementValues.

$$\text{carries} \circ \text{produces} \circ \text{isProcessInputOf} \circ \text{generates} \sqsubseteq \text{generates} \tag{1.2}$$

$$\text{carries} \circ \text{generates} \sqsubseteq \text{generates} \tag{1.3}$$

MeasurementType refers to the type of the MeasurementValue, such as O&M, OBOE, and so forth. One DataObject can contain a number of MeasurementValues which have specific MeasurementTypes. Meanwhile, One DataObject can also contain multiple Attributes (e.g., column names in a tabular form), and each Attribute is associated with a MeasurementType.

$$\text{contains} \circ \text{isOfMeasurementType} \sqsubseteq \text{containsMeasurementOf} \tag{1.4}$$

$$\text{contains} \circ \text{hasMeasurementType} \sqsubseteq \text{containsMeasurementOf} \tag{1.5}$$

One Attribute can be about a FeatureOfInterestType, such as pica pica bird, water, and so forth. If a Dataset contains DataObject which connects to FeatureOfInterestTypes through Attribute, then this Dataset is also about these FeatureOfInterestTypes.

$$\text{contains} \circ \text{contains} \circ \text{isAbout} \sqsubseteq \text{isAbout} \tag{1.6}$$

1.4 Example SPARQL Queries

1. "Find me abundance of Euphausia pacifica sampled with MOCNESS."

```
SELECT ?measurementValue WHERE
{
    :MOCNESS glinstrument:generates ?meansurementValue .
    ?measurementValue glinstrument:hasFeatureOfInterestType :EuphausiaPacifica .
    ?measurementValue glinstrument:hasMeasurementType :abundance .
}
```

2. "Give me calibration data for Durafet-type pH meters. "

(note from Yingjie: I assume here the competency question is try to find all data from the Durafet-type pH meters. Pascal: I don't think that's correct, but we probably haven't included calibration data in our pattern yet. We need to revsit this with someone who understands calibration data.)

```
SELECT ?measurementValue WHERE
{
    :Durafet-type_pH_meter glinstrument:generates ?meansurementValue .
}
```

Note from Yingjie: from the two examples above, it seems that we may need to add *SensorType* to the pattern to differentiate a type of sensors (e.g., PH meter) from a specific sensor (e.g., a particular PH meter). Pascal: To be revisited. I imagined such information to be captured by MeasurementType (i.e., PH values), but I'm not sure.

3. "Show me all cruises that deployed CTD."

This competency question requires a combination between the potential **Deployment** pattern, and the **Instrument** pattern.

1.5 Questions and Comments (for discussions only)

Please leave your questions and comments here for further discussion:

1. Two properties generates and produces seem very similar. Shall we merge them together?
2. Pascal: I'm not sure if this should really be called the "Instrument" pattern. My feeling is that we don't have this one clear yet. Are we probably rather talking about "measuring" as an event or probably even process?
3. Pascal: we need to rethink property names.
4. Pascal: time and place are somehow missing.

2 ⟨Station⟩ Pattern

2.1 Description

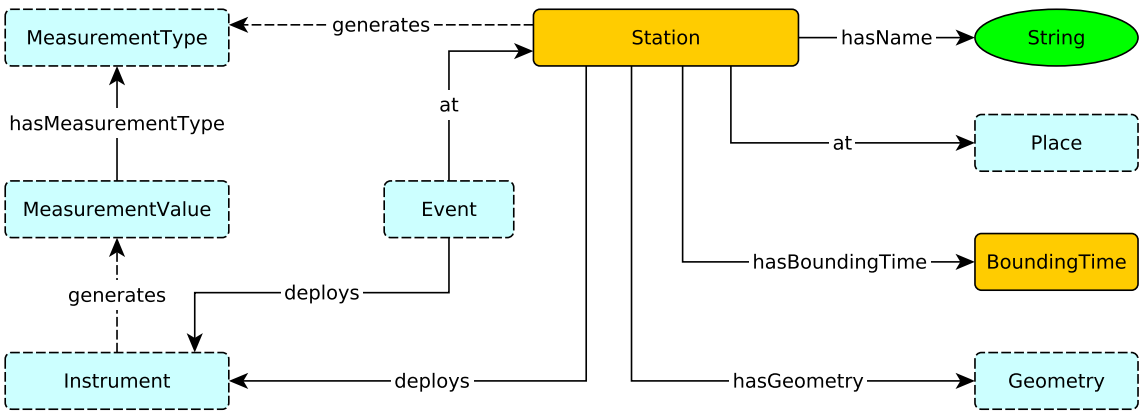

Figure 2.1: ⟨Station⟩ pattern

The ⟨Station⟩ pattern formalizes the corresponding concept of Station in oceanography. A Station may refer to a location where sampling events may take place (Persistent Station), or an event which associates with cruises (Ephemeral Station). Thus, a Station has its temporal component, which has been captured by the class of BoundingTime. Meanwhile, a Station also has its geospatial part, which has been captured by the Geometry. While the geometry (or the bounding shape) is important to convey the geospatial meaning of a Station, the boundary may not be crispy. Accordingly, a Station is also associated with the concept of Place. A Station can have a name for its identity, such as *Station Papa*. Sampling events can happen at a Station, and Instruments will be involved in sampling events. As a result, the ⟨Station⟩ pattern can also be linked to the ⟨Instrument⟩ pattern and models the relations between MeasurementType and a Station.

2.2 Competency Questions

The following competency questions have been proposed to stimulate the development of the ⟨Station⟩ pattern.

1. "Show me all measurements of salinity taken at Station Papa over the last 40 years."
2. "Show me all the stations in the gulf of Alaska."
3. "Show me all the NSF awards / AGU abstracts from these stations/this station."

2.3 Axiomatization

2.3.1 List of IRIs

Entity IRIs in the ⟨Station⟩ pattern are given as follows.

```
PREFIX glstation:  <http://schema.geolink.org/station#>
```

(1) Station (class): `glstation:Station`

(2) BoundingTime (class): `glstation:BoundingTime`
(3) at (object property): `glstation:at`
(4) hasBoundingTime (object property): `glstation:hasBoundingTime`
(5) deploys (object property): `glstation:employs`
(6) hasName (data property): `glstation:hasName`

2.3.2 Core Axioms

From the temporal perspective, an Station has a corresponding BoundingTime.

$$\text{Station} \sqsubseteq \exists \text{hasBoundingTime}.\text{BoundingTime} \tag{2.1}$$

From the geospatial perspective, an Station is associated with Place and Geometry.

$$\text{Station} \sqsubseteq \exists \text{at}.\text{Place} \sqcup \exists \text{hasGeometry}.\text{Geometry} \tag{2.2}$$

An Event (e.g., sampling event) can happen at a Station. An Event can employ an Instrument.

$$\exists \text{at}.\text{Station} \sqcap \exists \text{deploys}.\text{Instrument} \sqsubseteq \text{Event} \tag{2.3}$$

If the Instrument employed by the Station generates some MeasurementValue which belongs to certain MeasurementType, then this Station also generates this MeasurementType.

$$\text{deploys} \circ \text{generates} \circ \text{hasMeasurementType} \sqsubseteq \text{generates} \tag{2.4}$$

2.4 Example SPARQL Queries

1. "Show me all measurements of salinity taken at Station Papa over the last 40 years."

```
SELECT ?measurementValue WHERE
{
   ?station glinstrument:generates :salinity .
   ?station glstation:hasName ?name .
   ?station glstation:hasBoundingTime ?boundingTime .
   Filter(regex(?name, 'Station␣Papa', 'i') && ?boundingTime > xsd:dateTime(1974-01-01))
   ?station glstation:employs ?instrument .
   ?instrument glinstrument:generates ?measurementValue .
   ?measurementValue glinstrument:hasMeasurementType :salinity .
}
```

2. "Show me all the stations in the gulf of Alaska."

```
SELECT ?station WHERE
{
   ?station glstation:at dbpedia:Gulf_of_Alaska .
}
```

3. "Show me all the NSF awards / AGU abstracts from these stations/this station."

This competency question may need an integration of the ⟨FundingAward⟩ pattern with the ⟨Station⟩ pattern.

2.5 Questions and Comments (for discussions only)

Please leave your questions and comments here for further discussion:

1. From Yingjie: How should we represent the BoundingTime concept? As a time interval? This can directly influence how we are going to formalize the SPARQL query for competency question 1.

3 ⟨PhysicalSample⟩ Pattern

3.1 Description

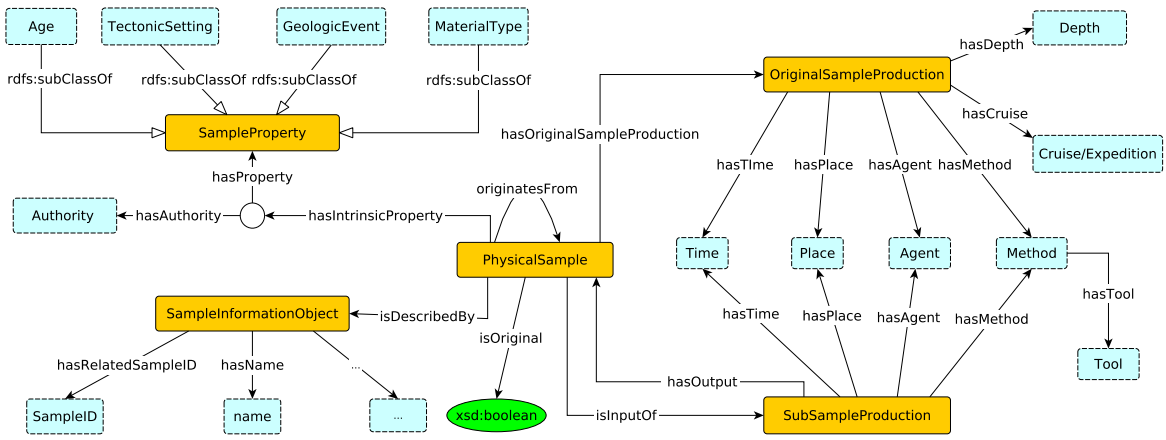

Figure 3.1: ⟨PhysicalSample⟩ pattern

The ⟨PhysicalSample⟩ pattern is designed to capture the process in which a physical sample is created. Specifically, this pattern focuses on samples in oceanography in which a sample was often derived from a *section* taken from a *core*. We also differentiate the samples which were directly taken from the natural environment from the samples (or sub-samples) which are derived from a parent sample. A data property, isOriginal, has been defined, which serves as a label to indicate whether this sample is original. Two classes, OriginalSampleProduction and SubSampleProduction, have been defined to capture the two different types of procedures to obtain physical samples. Elements in these two procedures, such as Time, Place, Agent, Method, and so forth, have been associated with the two procedure classes. In order to incorporate the situations in which the same sample may be given different analytical results (e.g., different ages) by multiple authorities, we also introduce a blank node to incorporate the source of the information. The class SampleProperty has been defined to differentiate properties which are intrinsic to the sample from the information which describes the sample, captured by the class SampleInformationObject.

3.2 Competency Questions

The following competency questions have been proposed to stimulate the development of the ⟨PhysicalSample⟩ pattern. These competency question can be put into several groups:

Group 1: Querying the intrinsic property of a sample

1. "Show me the type of a sample" (e.g. "Sediment", "Rock", "Mineral")
2. "Show me the age of the sample"

Group 2: Querying the extrinsic property of a sample

4. "Show me the name of a sample"
5. "Show me the registration date of a sample"

Group 3: Querying the original sample production

6. "Show me the original Location of the sample"
7. "Show me the cruise that the sample is originally collected"

Group 4: Querying the sub sample production

8. "Show me the collector of the sample"
9. "Show me the sampling method of this physical sample"

3.3 Axiomatization

3.3.1 List of IRIs

Entity IRIs in the ⟨PhysicalSample⟩ pattern are given as follows.

```
PREFIX glphysicalSample:  <http://schema.geolink.org/physicalSample#>
```

(1) PhysicalSample (class): `glphysicalSample:PhysicalSample`
(2) OriginalSampleProduction (class): `glphysicalSample:OriginalSampleProduction`
(3) SubSampleProduction (class): `glphysicalSample:SubSampleProduction`
(4) SampleProperty (class): `glphysicalSample:SampleProperty`
(5) SampleInformationObject (class): `glphysicalSample:SampleInformationObject`
(6) originatesFrom (object property): `glphysicalSample:originatesFrom`
(7) hasIntrinsicProperty (object property): `glphysicalSample:hasIntrinsicProperty`
(8) isInputOf (object property): `glphysicalSample:isInputOf` `glphysicalSample:isInputOf`
(9) hasOutput (object property): `glphysicalSample:hasOutput`
(10) hasOriginalSampleProduction (object property): `glphysicalSample:hasOriginalSampleProduction`
(11) isOriginal (data property): `glphysicalSample:isOriginal`

3.3.2 Core Axioms

Firstly, we define the property originatesFrom as transitive. Thus, if sample A originates from sample B which originates from sample C, then A also originates from sample C.

$$\mathsf{originatesFrom} \circ \mathsf{originatesFrom} \sqsubseteq \mathsf{originatesFrom} \tag{3.1}$$

A physical sample can be the input of a sub sample production. Meanwhile, a sample can also be the output of the sub sample production.

$$\mathsf{PhysicalSample} \sqsubseteq \exists \mathsf{isInputOf}.\mathsf{SubSampleProduction} \tag{3.2}$$

$$\mathsf{SubSampleProduction} \sqsubseteq \exists \mathsf{hasOutput}.\mathsf{PhysicalSample} \tag{3.3}$$

Based on the isOriginal property, we can also define a class of OriginalSample

$$\mathsf{PhysicalSample} \sqcap \exists \mathsf{isOriginal}.\top \sqsubseteq \mathsf{OriginalSample} \tag{3.4}$$

For the OriginalSampleProduction, it must have Depth and Place

$$\mathsf{OriginalSampleProduction} \sqsubseteq \exists \mathsf{hasDepth}.\mathsf{Depth} \sqcap \exists \mathsf{hasPlace}.\mathsf{Place} \tag{3.5}$$

3.4 Example SPARQL Queries

For competency questions in **group 1**, we can use a SPARQL query like below:
1. "Show me the type of a sample" (note: here we assume that we know the uri for the specific sample instance, as *physicalSample1*, and we will follow this style in the other SPARQL queries for competency questions.)

```
SELECT ?sampleType WHERE
{
  :physicalSample1 :hasIntrinsicProperty ?blank .
  ?blank :hasProperty ?sampleType .
  ?sampleType rdf:type :MaterialType .
}
```

Note: this query may return multiple material types according to different authorities.
2. "Show me the age of the sample"

```
SELECT ?sampleAge WHERE
{
  :physicalSample1 :hasIntrinsicProperty ?blank .
  ?blank :hasProperty ?sampleAge .
  ?sampleAge rdf:type :Age .
}
```

For competency question in **group 2**, we can use the following SPARQL queries:
3. "Show me the name of a sample"

```
SELECT ?sampleName WHERE
{
  :physicalSample1 :isDescribedBy ?sampleInformationObject .
  ?sampleInformationObject :hasName ?sampleName .
}
```

4. "Show me the registration date of a sample"

```
SELECT ?registrationDate WHERE
{
  :physicalSample1 :isDescribedBy ?sampleInformationObject .
  ?sampleInformationObject :hasRegistrationDate ?registrationDate .
}
```

For competency question in **group 3**, we can use the following SPARQL queries:
5. "Show me the original location of the sample"

```
SELECT ?originalLocation WHERE
{
  :physicalSample1 :originatesFrom* ?parentSample .
  ?parentSample :isOriginal true^^xsd:boolean .
  ?parentSample :hasOriginalSampleProduction ?originalSampleProduction .
  ?originalSampleProduction :hasPlace ?originalLocation .
}
```

Note: the SPARQL query above makes use of the transitivity of the originatesFrom property. originatesFrom$*$ means this property show up 0 or more times. Thus, if *physicalSample1* is already the original sample, this query will directly retrieve the location information from the OriginalSampleProduction.
6. "Show me the cruise that the sample is originally collected"

```
SELECT ?cruise WHERE
{
  :physicalSample1 :originatesFrom* ?parentSample .
  ?parentSample :isOriginal true^^xsd:boolean .
  ?parentSample :hasOriginalSampleProduction ?originalSampleProduction .
  ?originalSampleProduction :hasCruise ?cruise .
}
```

For competency question in **group 4**, we can use the following SPARQL queries:
7. "Show me the collector of the sample"

```
SELECT ?collector WHERE
{
  :physicalSample1 ^:hasOutput ?subSampleProduction .
  ?subSampleProduction :hasAgent ?collector .
}
```

8. "Show me the sampling method of the physical sample"

```
SELECT ?method WHERE
{
   :physicalSample1 ^:hasOutput ?subSampleProduction .
    ?subSampleProduction :hasMethod ?method .
}
```

3.5 Questions and Comments (for discussions only)

Please leave your questions and comments here for further discussion:

1. From Yingjie: Do we really have to differentiate *intrinsic* and *extrinsic* properties? I think they can easily get the data managers confused.