Evolution of Environmental Information Models

Reusable Properties Bound to a Persistent URI

Katharina Schleidt(✉)

DataCove, Vienna, Austria
Kathi@datacove.eu

Abstract. Reusability of environmental data is essential for environmental research and control; standardized data models are being created by various organizations to facilitate this process. Due to the evolving nature of environmental science, these data models must be continuously extended for the support of new concepts, thus rapidly breaking the level of standardization achieved. The definition of reusable properties would allow for standardization of this extension process. In this paper, we first analyze the requirements to reusable properties, and explain the rational for the decision that reusable properties tightly bound to a URI would be the most apt solution; the following list of requirements was defined in order to compare the viability of the options proposed: URI Coupling, DataType Coupling, Semantics Coupling and Persistence. We then go on to explore possible avenues for implementation of reusable URI-Properties, whereby the following approaches where analysed for applicability: Data Types, Interfaces, MOF level adjustment of UML and a solution utilizing stereotypes for the definition and use of reusable URI-Properties. Of these approaches, all were deemed feasible except for the MOF level adjustment of UML; MOF level adjustment is not possible due to cardinality constraints within the MOF definition. Examples were created for the other 3 possibilities, including serialization options towards XML Schema. These examples were then compared with the requirements defined for URI-Properties; based on this analysis, the UML Stereotype based solution for the specification and use of reusable URI-Properties was deemed as most viable and is described in further detail.

1 Introduction

Access to environmental data is necessary for environmental research and control. Due to the complexity of environmental concerns, this task often requires the reuse of data originating from various sources. Triggered by this requirement, organizations from various environmental domains have started initiatives aimed at the provision of standardized access to environmental data; standardized data models for the exchange of various types of environmental data are now available, together with the required service definitions for discovery and access.

J. Hřebíček et al. (Eds.): ISESS 2017, IFIP AICT 507, pp. 338–348, 2017.
https://doi.org/10.1007/978-3-319-89935-0_28

These developments should guarantee easy access to standardized and harmonized environmental data. However, many standardized data models cover only common concepts pertaining to a wide range of usage areas; thematic extensions are required for the support of specific community's requirements. As the thematic extensions develop, domains with similar requirements create parallel extensions; while these may be semantically identical, this fact cannot be simply verified [1]. Thus, after much effort, we find ourselves dangerously close to the starting point.

After discussions with various stakeholder representatives, the conclusion was reached that the most efficient mechanism to allow for flexible extension in a complex environment as described would be the introduction of reusable properties, such as commonly used within semantic technologies. An overview of implementation options for reusable properties is provided, together with an analysis of their viability.

2 Background and State of the Art

Various initiatives have been launched in the last years aimed at providing easy access to relevant data stemming from various environmental sub-domains through the standardization of data models and service specifications. While data standards are provided for core concepts, there is always a need to extend these concepts in order to support new or alternative requirements.

2.1 Background

In this paper, we use the European INSPIRE Initiative [2] as an illustrative example, and thus focus on the first approach.

The INSPIRE Directive

The INSPIRE Directive (2007/2/EC), specifying an Infrastructure for Spatial Information in the European Community, entered into force on the 15th of May 2007 with the aim to assure easy availability of high quality spatial data as required for the definition and enforcement of European Community environmental policy. 34 spatial data themes are covered by INSPIRE; data models and service specifications have been created accordingly. While aiming to be technology agnostic through flexibility in the serialization technology, the data modelling process was solely based on the ISO/OGC Suite of Spatial Standards together with its inherent data modelling requirements.

ISO/OGC Suite of Spatial Standards

As environmental data almost invariably has a spatial component, the International Standards Organisation (ISO) & Open Geospatial Consortium (OGC) Suite of Spatial Standards is increasingly being used for the creation of thematic application schemas. This trend has in turn led to the creation of various standards beyond the classical spatial domain, including data and service standards covering the provision of measurement data, be they individual observations, time-series or multidimensional coverages.

2.2 State of the Art

The technological basis both for the creation of the underlying data models defining the structure of the data as well as the formats and technologies used for data provision varies across initiatives; the following approaches have been identified:

1. Definition of individual defined concepts that can be combined to data structures. Examples:
   (a) Clinical Data Interchange Standards Consortium (CDISC) foundational standards supporting clinical and non-clinical research processes;[1]
   (b) Darwin Core standard for biodiversity observations.[2]
2. Definition of data structures in Unified Modelling Language(UML), provision Extendible Markup Language (XML), sometimes JSON. Examples:
   (a) Most ISO/OGC standards and extensions, i.e. INSPIRE;
   (b) American National Information Exchange Model (NIEM).[3]
3. Definition of data structures directly in XML. Examples:
   (a) Geography Markup Language (GML; ISO 19136)
4. Definition of data structures using semantic technologies (Resource Description Framework (RDF) and Web Ontology Language (OWL))
   (a) Open Biomedical Ontology (OBO);[4]
   (b) The Extensible Observation Ontology (OBOE).[5]

3 Methodology

Based on an analysis of existing approaches as well as a workshop ISESS 2015, the requirements for harmonized data model extension were analyzed; URI-Properties, defined as reusable properties bound to a persistent Uniform Resource Identifier (URI) [3], were identified as a potential solution. A set of requirements that must be fulfilled by URI-Properties was defined:

1. A URI-Property must be uniquely identifiable through an URI
2. The datatype of a URI-Property must be tightly coupled with its definition
3. The semantics of a URI-Property must be tightly coupled with its definition
4. A URI-Property must be persistent. We shall define persistence in analog manner to the definition used for Global Unique Identifiers (GUIDs) referencing data: A URI-Property may not be redefined with different semantics while retaining the same URI; while the definition of a URI-Property may at some point no longer be available, the reuse of the URI is not allowed.

[1] https://www.cdisc.org/.

[2] http://rs.tdwg.org/dwc/.

[3] https://www.niem.gov/.

[4] https://www.bioontology.org/.

[5] https://github.com/NCEAS/oboe/.

The following sections describe the viability of the options identified for the provision of reusable properties within UML, as well as their conformance to the requirements defined above.

3.1 Data Types

Defining the semantics of data types via derivation hierarchies is state-of-the art. However, pushing the complexity of semantics into data type definition could cause difficulties, as a complex derivation hierarchy must be created and maintained; should this approach be pursued methods of coupling required data types with a formal ontology, i.e. formulated in OWL, should be explored [4]. In addition, while base semantics are defined, the usage of these concepts as data types allows for definition of class attributes using the same data type but with subtly different meanings. Such differentiation could be as simple as the provision of a preferred concept together with an alternative concept, with no additional information on the subtle difference between these two concepts.

Finally, as XML Schema doesn't currently support multiple inheritance, while the semantics stemming from the derivation hierarchy are available within the UML data model, no indication of this additional information is available within the XML Schema.

3.2 Interfaces

Interfaces are state of the art for provision of reusable attributes. However we encounter problems due to the fact that XML doesn't support multiple inheritance. While GML MIXIN overcomes this shortcoming by copying attributes and associations (copy down), this technique provides no information as to the source of these attributes and associations in the final XML Schema. Further, the utilization of interfaces for the representation of reusable properties would break a great deal of the visual clarity of UML; the properties provided by the interface are not visible in the class inheriting from the interface, nor for classes derived from this class. Thus, while the benefits of reusable properties would be valuable, the cost for both the creation as well as the interpretation of the model would be a great deal higher than with normal methodologies.

3.3 MOF Level Adjustment of UML

Initially, the approach of defining reusable URI-Properties at the Meta Object Facility (MOF) level seemed the most promising, as this would integrate the concept at the UML definition level. However, this proved not to be possible, as both attributes and associations have a minimal cardinality of 1 in the MOF definition. Thus a property cannot be defined without it being directly used.

3.4 Stereotypes

Stereotypes are well suited for the definition of reusable URI-Properties. Through the tight binding of the URI-Property to the URI, the semantics of the URI-Property can be provided through an external ontology referencing this URI. This URI is visible within the XML Schema defining the URI-Property via the appinfo element, allowing applications encountering this property to resolve the URI for more information on this attribute. In the final schema, the element name and data type are automatically supplied through the element reference. The schema encoding rules are in alignment with the requirements of the underlying GML and ISO standards, and should be easy to implement.

The only problems currently identified with to this solution pertain to its integration in UML development tools. At present the use of URI-Properties requires discipline from the data modelers, as the constraints on URI-Properties are not checked by the UML tools, and thus inconsistencies will only be flagged during the schema generation process. In addition, registries of reusable URI-Properties would need to be developed and ideally integrated within the UML tools.

A final advantage of the use of URI-Properties is the fact that the definition is agnostic of the final serialization form. While well suited to serialization in XML, the logic behind the URI-Properties is also in alignment with the requirements ensuing from semantic serialization technologies such as RDF.

3.5 Analysis Against Requirements

The following table shows the approaches analyzed against the individual requirements identified (Table 1).

Table 1. Analysis approaches against requirements

Requirement approach	URI coupling	DataType coupling	Semantics coupling	Persistence
Data types	✗	~	~	✗
Interfaces	✗	~	~	✗
MOF adjustment	✗	✗	✗	✗
Stereotypes	✓	✓	✓	✓

4 Stereotype Solution

Based on the insights presented above, UML Stereotypes were selected for the implementation of reusable URI-Properties is the use of UML Stereotypes. In the following section this is illustrated through the creation of the URIProp Stereotype.

4.1 UML Example

The URIProp stereotype, defined on both attributes and associations, adds the following tags to the attributes and associations it is applied to:

- URI: a unique URI for this property
- Name: the name of the attribute or association role
- Datatype: the datatype of the attribute or of the target of the association

In addition, the following three constraints are added to the URIProp stereotype:

- Property unique per class: A URI property can only occur once per class
- Name aligned: The attribute name must be the same as the Name tag of the attribute, which must in turn be the same as that stored for the specified URI Property under the referenced URI
- Datatype aligned: The attribute datatype must be the same as the Datatype tag of the attribute, which must in turn be the same as that stored for the specified URI Property under the referenced URI

For the definition of reusable URI-Properties, the stereotype must first be applied to the definition of the URI-Property, be it for an attribute or for an association role. In the example below, we define two URI-Properties:

- euStationName: this URI-Property provides an attribute named euStationName referencing the data type CharacterString. The following Tagged Values are added through the URIProp stereotype:
  - URI: http://www.props.eu/euStationName.
  - name: euStationName
  - dataType: CharacterString
- euStationNameAss: this URI-Property provides an association named euStationNameAss referencing the data type GeographicalName. The following Tagged Values are added through the URIProp stereotype:
  - URI: http://www.props.eu/euStationNameAss.
  - name: euStationNameAss
  - dataType: GeographicalName

The following diagram shows the UML Encoding of the URI-Properties:

As part of the definition process for URI-Properties, the Tagged Values from the URIProp stereotype must be provided (Fig. 1). This stereotype must then be added to the class attributes or associations that are utilizing an URI-Property as shown in the following diagrams (Fig. 2).

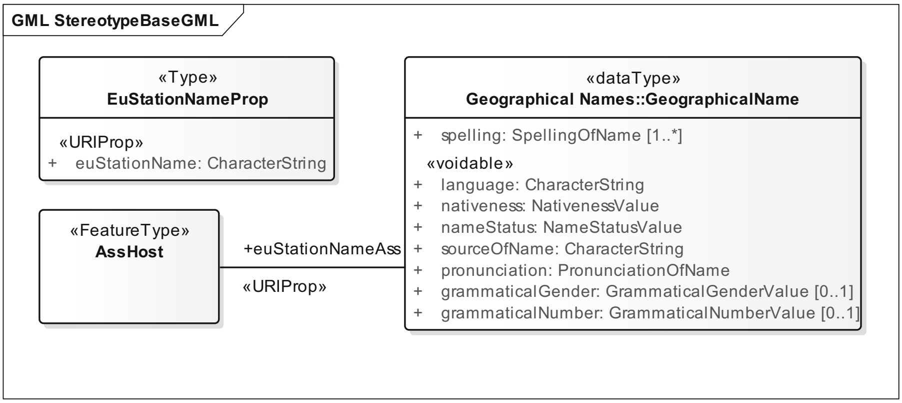
GML StereotypeBaseGML
«Type»
EuStationNameProp
«URIProp»
+ euStationName: CharacterString
«FeatureType»
AssHost
+euStationNameAss
«URIProp»
«dataType»
Geographical Names::GeographicalName
+ spelling: SpellingOfName [1..*]
«voidable»
+ language: CharacterString
+ nativeness: NativenessValue
+ nameStatus: NameStatusValue
+ sourceOfName: CharacterString
+ pronunciation: PronunciationOfName
+ grammaticalGender: GrammaticalGenderValue [0..1]
+ grammaticalNumber: GrammaticalNumberValue [0..1]

Fig. 1. Definition of URI-properties using stereotypes

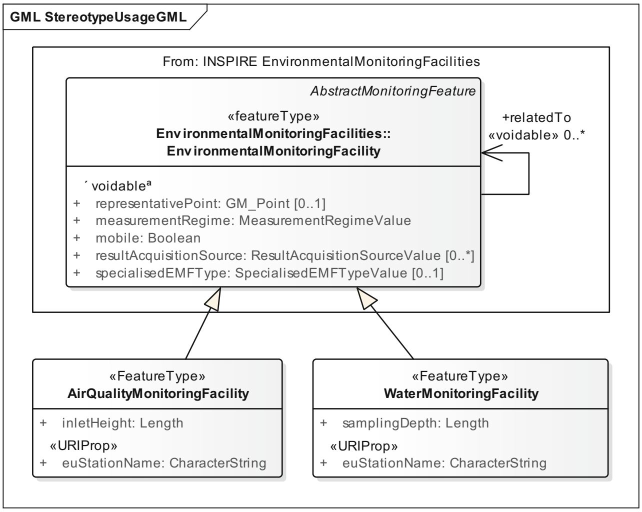
GML StereotypeUsageGML
From: INSPIRE EnvironmentalMonitoringFacilities
AbstractMonitoringFeature
«featureType»
EnvironmentalMonitoringFacilities::
EnvironmentalMonitoringFacility
´ voidableª
+ representativePoint: GM_Point [0..1]
+ measurementRegime: MeasurementRegimeValue
+ mobile: Boolean
+ resultAcquisitionSource: ResultAcquisitionSourceValue [0..*]
+ specialisedEMFType: SpecialisedEMFTypeValue [0..1]
+relatedTo
«voidable» 0..*
«FeatureType»
AirQualityMonitoringFacility
+ inletHeight: Length
«URIProp»
+ euStationName: CharacterString
«FeatureType»
WaterMonitoringFacility
+ samplingDepth: Length
«URIProp»
+ euStationName: CharacterString

Fig. 2. Usage of attribute URI-properties using stereotypes

The same tagged values as defined above for the definition of the URI-Properties must also be provided for each usage instance. The constraints defined for URI-Properties must be complied with, assuring alignment to the original URI-Property definition (Fig. 3).

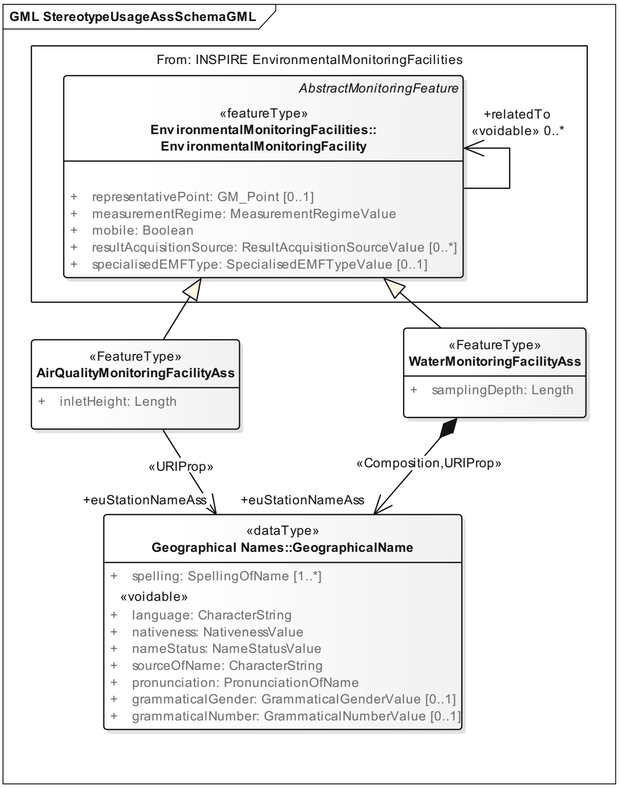
GML StereotypeUsageAssSchemaGML
From: INSPIRE EnvironmentalMonitoringFacilities
AbstractMonitoringFeature
«featureType»
EnvironmentalMonitoringFacilities::
EnvironmentalMonitoringFacility
+ representativePoint: GM_Point [0..1]
+ measurementRegime: MeasurementRegimeValue
+ mobile: Boolean
+ resultAcquisitionSource: ResultAcquisitionSourceValue [0..*]
+ specialisedEMFType: SpecialisedEMFTypeValue [0..1]
+relatedTo
«voidable» 0..*
«FeatureType»
AirQualityMonitoringFacilityAss
+ inletHeight: Length
«FeatureType»
WaterMonitoringFacilityAss
+ samplingDepth: Length
«URIProp»
«Composition,URIProp»
+euStationNameAss
+euStationNameAss
«dataType»
Geographical Names::GeographicalName
+ spelling: SpellingOfName [1..*]
«voidable»
+ language: CharacterString
+ nativeness: NativenessValue
+ nameStatus: NameStatusValue
+ sourceOfName: CharacterString
+ pronunciation: PronunciationOfName
+ grammaticalGender: GrammaticalGenderValue [0..1]
+ grammaticalNumber: GrammaticalNumberValue [0..1]

Fig. 3. Usage of association URI-properties using stereotypes

4.2 Serialization

While the schema encoding rules for data types and interfaces are specified in the GML and ISO standards, we must first define encoding rules for the use the URIProp Stereotype.

For the definition of URI-Properties, we will make use of the XML Schema option of defining an element by reference. The URI defining the URI property is provided within the appinfo section of the annotation element.

The element declarations for the URI-Properties pertaining to attributes are as follows:

```
<xs:element name="euStationName" type="xs:string">
  <xs:annotation>
    <xs:appinfo
source="http://www.props.eu/euStationName">URI-
Property</xs:appinfo>
  </xs:annotation>
</xs:element>
```

A similar pattern is utilized in the element declaration for URI-Properties pertaining to associations, taking into account the encoding requirements stemming from the GML and ISO standards:

```
<xs:element name="euStationNameAss" >
  <xs:annotation>
    <xs:appinfo
source="http://www.props.eu/euStationNameAss">URI-
Property</xs:appinfo>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="gml:AbstractMemberType">
        <xs:sequence minOccurs="0">
          <xs:element ref="gn:GeographicalName"/>
        </xs:sequence>
        <xs:attributeGroup
ref="gml:AssociationAttributeGroup"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Once the URI-Property has been defined, it can then be referenced from the XML Schemas reusing this property as follows:

```
<xs:element ref="st:euStationName"/>
```

The same pattern can also be used pertaining to associations:

```
<xs:element ref="st:euStationNameAss"/>
```

The following XML snippet shows the serialization of the AirQualityMonitoringFacility station name attribute using stereotypes:

```
<st:euStationName>AT_AQ_Station1</st:euStationName>
```

Namespaces:

- st: interface property schema

When the URI-Property is defined as an association, it is possible to provide the information either inline, or via xlink to an external instance.

4.3 Reflection

Stereotypes are well suited for the definition of reusable URI-Properties. Through the tight binding of the URI-Property to the URI, the semantics of the URI-Property can be provided through an external ontology referencing this URI. This URI is visible within the XML Schema defining the URI-Property via the appinfo element, allowing applications encountering this property to resolve the URI for more information on this attribute. In the final schema, the element name and data type are automatically supplied through the element reference. The schema encoding rules are in alignment with the requirements of the underlying GML and ISO standards, and should be easy to implement.

The only problems currently identified with to this solution pertain to its integration in UML development tools. At present the use of URI-Properties requires discipline from the data modelers, as the constraints on URI-Properties are not checked by the UML tools, and thus inconsistencies will only be flagged during the schema generation process. In addition, registries of reusable URI-Properties would need to be developed and ideally integrated within the UML tools.

A final advantage of the use of URI-Properties is the fact that the definition is agnostic of the final serialization form. While well suited to serialization in XML, the logic behind the URI-Properties is also in alignment with the requirements ensuing from semantic serialization technologies such as RDF.

5 Conclusion and Outlook

Based on the analysis of the implementation options, the current best candidate for the implementation of URI-Properties is the stereotype solution.

Further analyzing the potential of the stereotype solution, it becomes apparent that the addition of URI-Properties via stereotypes serves to bring traditional UML data modelling closer to emerging semantic technologies, where properties are traditionally first class citizens. If an alignment between URI-Properties within a UML model and predicates as utilized within RDF and OWL is provided, it becomes possible to easily traverse between UML based data models and semantic data models. This would be beneficial, as the spatial data community is progressively moving towards semantic technologies, while wishing to retain as much as possible of the existing data model standards. Thus, by properly utilizing URI-Properties, it is possible to reuse the UML based data models for data serialization both via semi-structured technologies such as XML as well as semantic technologies such as RDF and OWL, opening up the scope of potential end users for the data provided.

References

1. Tóth, K., Portele, C., Illert A., Lutz, M., de Lima, M.A.: A conceptual model for developing interoperability specifications in spatial data infra-structures, JRC reference report (2012). http://inspire.ec.europa.eu/documents/Data_Specifications/IES_Spatial_Data_Infrastructures_(online).pdf
2. Schleidt, K.: Evolution of Environmental Information Models. In: Denzer, R., Argent, R.M., Schimak, G., Hřebíček, J. (eds.) Environmental Software Systems. Infrastructures, Services and Applications, vol. 448, pp. 71–80. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15994-2_6
3. Schleidt K.: Evolution of Environmental Information Models. Workshop Outcomes (2015). http://datacove.eu/data/documents/EIM_WS_Outcomes.pdf
4. Janssen, S., Andersen, E., Athanasiadis, I.N., van Ittersum, M.K.: A database for integrated assessment of European agricultural systems. Environ. Sci. Policy **12**(5), 573–587 (2009)