Taken from: <https://www.ogc.org/standards/sfa>

# Open Geospatial Consortium: Simple Feature Access

## Specification

### 6 Feature models

#### 6.1 Introduction to feature taxonomy

A feature is a representation of a real-world entity/object. This idea can trace its definition back to the 1960’s in the definition of Simula (a simulation language), the first object-oriented programming language. In Simula, "digital objects" represented "real-world objects"4. Since the 1980’s, spatial or geographic information systems have paralleled this same idea, at about the same time as Simula and OOPL were first being used. Much of the early standardization of GIS followed the same object oriented paradigm and defined feature classes as object classes (often in UML or it predecessors, such as OMT, object modeling technique).

The problem with digital objects representing real world objects is that, in most schematic modeling, all objects of the same class have essentially the same representation. The "real-world" is not as neat and clean. Any two real-world objects are distinct, often at a fundamental level and, for that reason, each may be a unique instance of their own "class", e.g. must be described in different manners with different properties, not just different values for the same property. While feature schemata are the core of many systems and their associated standards, there is another approach that supports schema but allows more flexibility as needed. This approach steps back from structural formalism and uses semantic taxonomy and ontology, to define feature types and prescribes descriptive properties of features as related parts of a taxonomy. In contrast with a schema, where a class is defined by its properties, a taxonomy-based system allows any feature to be associated with any property where the feature and property definitions are consistent with one another according to a given taxonomy.

A geographic feature taxonomy, ontology or schema will define a namespace that contains definitions of the keys that will be used to identify the semantics of key-value pairs associated to a feature:

• a set of feature types (a type name and a definition)

o including a "is a type of" relation where a definition may reference a more general type,

 e.g. "weir is a type of dam such that"…5

• a set of feature properties:

o a set of feature attributes (an attribute name, a definition and a data structure or object to  
represent its value), and

o a set of feature relations that associate features to one another (a name, a definition, a list   
of rolenames and pointer or reference type to identify the feature in each role).

A feature instance contains:

− a unique local identity, (possibly part of a universal identity scheme such as URN)

− one or more feature type names,

− a list of properties with attribute names and associated values,

− a list of feature relationship names each with a list of roles and the target feature identities   
for those roles.

A geographic feature schema is an ontology that further defines which properties and relations that can be (optional) or that must be (mandatory) associated to instances of which feature types. All properties not specifically mentioned in this list cannot be part of the structure of a feature of that type. Table 1 below shows how a set of features could be represented and gives some potential examples of a taxonomy for feature type definition that are used in the example.

Classical schema formalism arises from application requirements. For example, a radar-based application would concentrate on the attributes of a feature that affect its radar reflectivity, but a visual-based application would concentrate on the visual attributes of a feature such as color. Radar is not good at detecting color (wrong part of the spectrum), but human eyes are. A radar system would not care what color the Golden Gate Bridge is painted, but a system for visual identification would know "international orange" is a good visual recognition clue; in fact, the "color is used in the aerospace industry to set things apart from their surroundings".

#### Table 1 – Example feature entities in a key-value format

Object-1:

{

Feature: "URL 1" ‼ [some URL]

FeatureType: "Dam"

FeatureType: "Bridge"

Name: "Wheeler Dam"

On: "URL 3" ‼ Tennessee River

CenterPoint 34°48'14.99" N, -87°22'32.99" W

Geometry: Location ‼ [LineString expression]

}

Object-2:

{

Feature: "URL 2" ‼ [some URL]

FeatureType: "Lock"

On: "URL 3" ‼ Tennessee River

In: "URL 1" ‼ Wheeler Dam

}

Object-3:

{

Feature: "URL 3" ‼ [some URL]

FeatureType: River

Name: Tennessee River

}

4 The GIS definition used "real-world phenomena" and Simula used "real world objects". This is an extension needed for GIS since many "features" displayed in data sets are "conceptual" like political boundaries, buffer zones, voting districts and others whose existing in the real world has real world implications, but is not always a real-world physical but a conceptual "object".

5 The USSD (US Society of Dams, https://www.ussdams.org/) recognizes 12 basic types of dams; a weir is an "overflow dam".