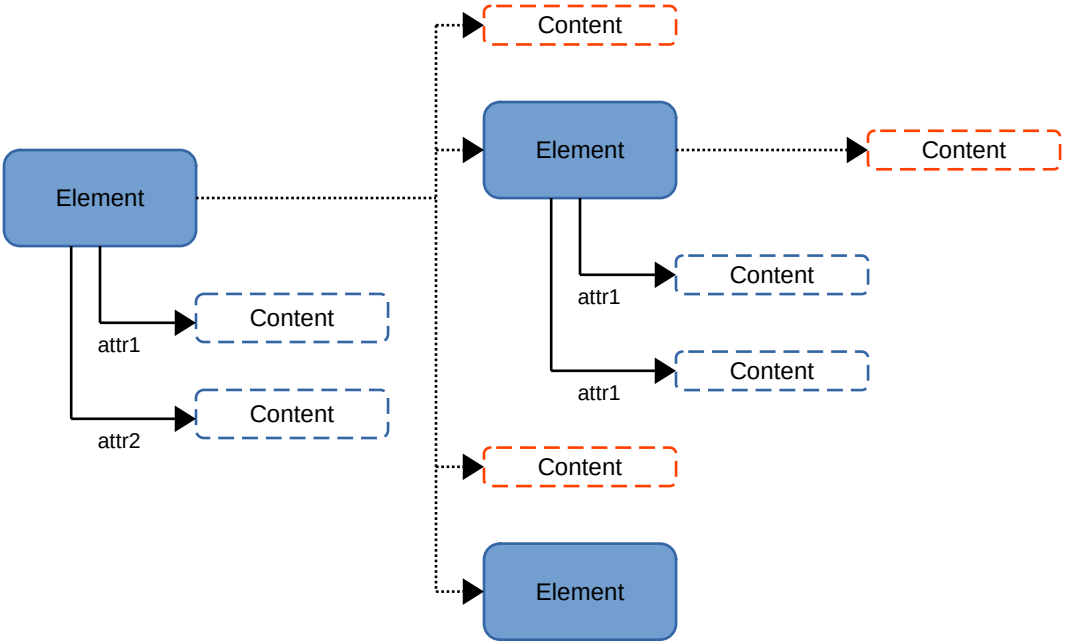
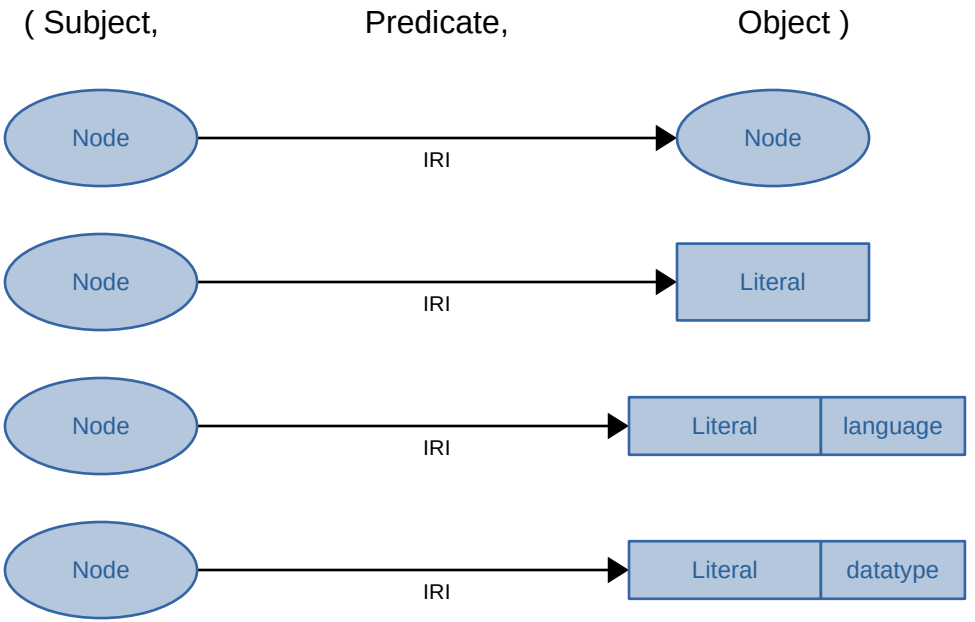


The basic XML document model consists of a tree of Elements where each Element may have attributes which have a value, and child elements which may be content or other elements. Parent/child relationships are implicit, other references between elements may use special ID and IDREF attributes.



The basic RDF graph model consists of a set of (subject, predicate, object) statements. The subject and predicates are IRIs, the object may be another node or a literal. Literals may have a language or datatype (XSD Datatype).

Subject and object IRIs may be globally unique (good for sharing) or “blank” (unique only within a graph). **IRIs do not have encoded content - things, not strings.**

`http://www.w3.org/2001/XMLSchema#float`

An IRI is an internationalized URI which often looks like a URL. It starts with a *scheme* and is followed by a bunch of content with syntax that is specified by the scheme. For RDF content that is intended to be shared via the Semantic Web, the scheme is often *http* or *https*.

Content negotiation is used to determine what serialization syntax is acceptable to the client and available from the server.

http://www.w3.org/2001/XMLSchema#float

<http://www.w3.org/2001/XMLSchema#float>

Depending on the serialization, the IRI is enclosed in angle brackets. In RDF/XML the IRI might be an attribute value, so it is simply enclosed in quotes.

http://www.w3.org/2001/XMLSchema#float

<http://www.w3.org/2001/XMLSchema#float>

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> .
```

xsd:float

On of the more common serialization formats called *Turtle* is designed to be easily exchanged between users in chat rooms and email so there is a mechanism to provide a prefix shortcut and drop the angle brackets. There are also syntax shortcuts for blank nodes…

```
ex:a ex:likes [ a ex:Car ] .
```

...repeating elements…

```
ex:a ex:hasChild ex:b, ex:c, ex:d .
```

...and lists:

```
ex:a ex:contacts ( ex:b ex:c ex:d ) .
```

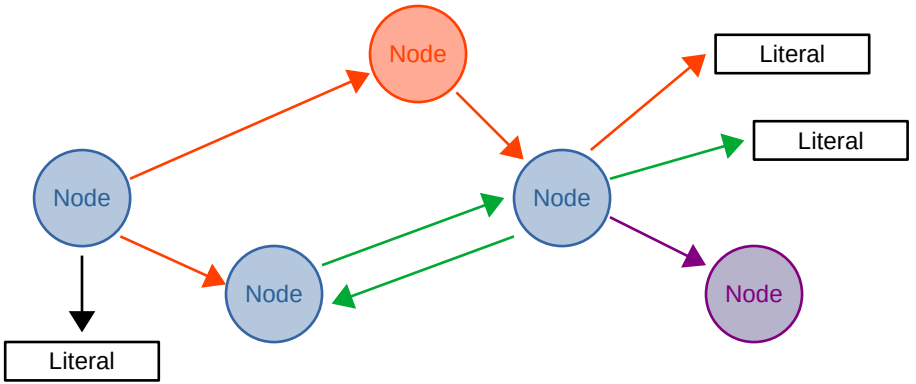
<http://data.ashrae.org/bacnet/2020#present-value>

```
PREFIX bacnet: <http://data.ashrae.org/bacnet/2020#> .
```

bacnet:present-value

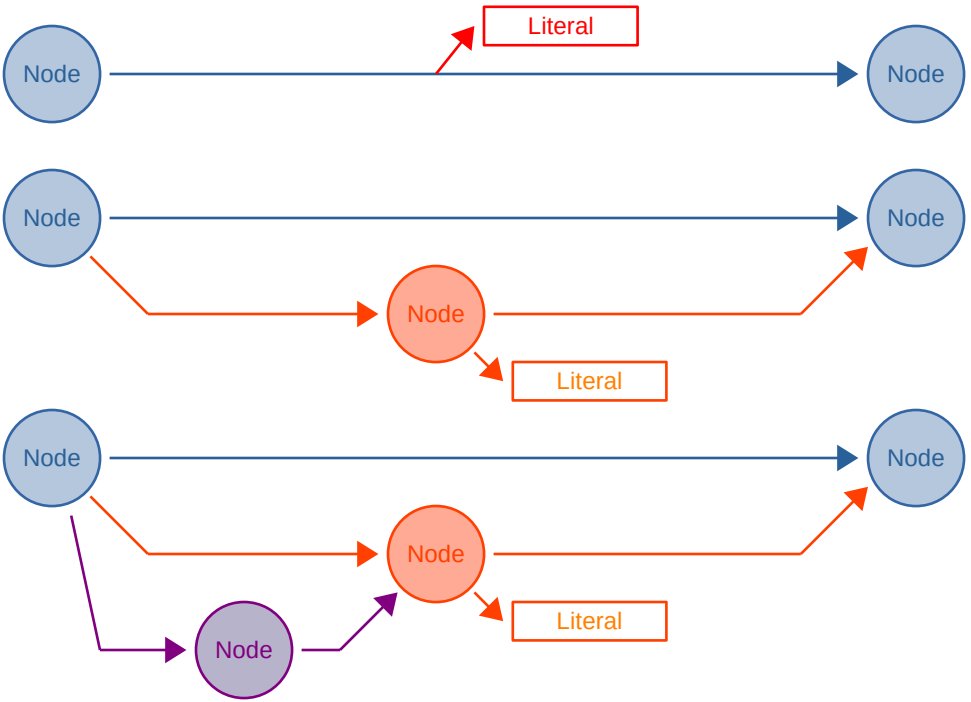
I'm assuming that this IRI is available to use.

- **UpperCamelCase** for class names and special "named individuals"
- **camelCase** for RDF predicate names that are implicit in other serializations
- **Pascal_Snake_Case** where it is used in the standard
- **spinal-case** to match ASN.1 names

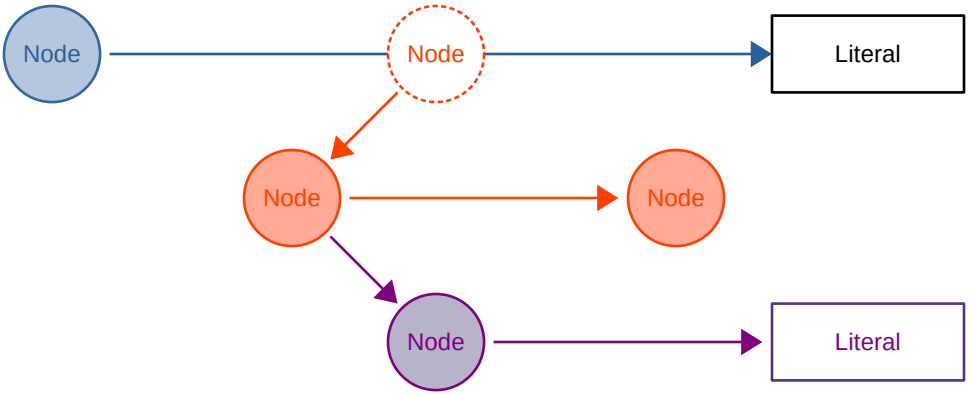


Nodes are “in” a graph in the context of a statement, and do not belong to any specific graph (with the exception of “blank nodes”). Where the IRI for a node is a shared identifier for a specific concept, it is typically “defined” in a specific graph (vocabulary, taxonomy, ontology) using other shared concept identifiers.

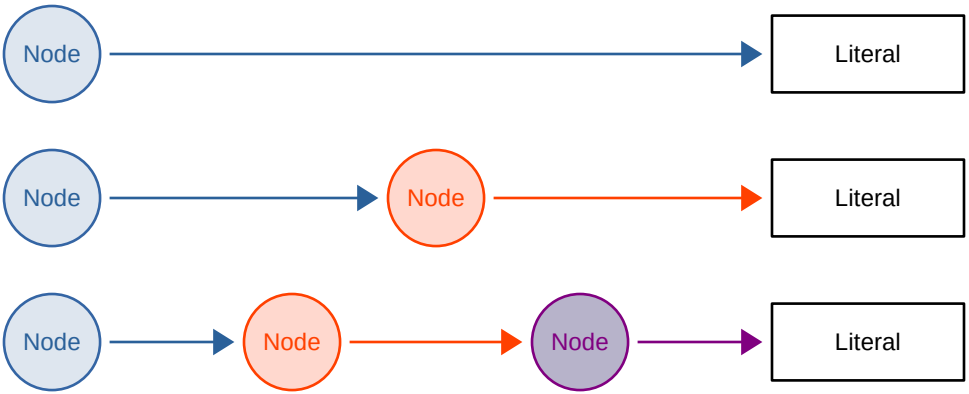
RDF contains core definitions, RDFS contains higher layer concepts like Class (set) subClassOf (membership relation between sets), and Properties (the names of things that are predicates).



RDF Predicates (properties or relationships) do not have additional attribute/value content (leading to the growth of *labeled property graphs*) so to refine a relationship an additional node is introduced. This successive refinement called **reification** can be repeated as necessary.



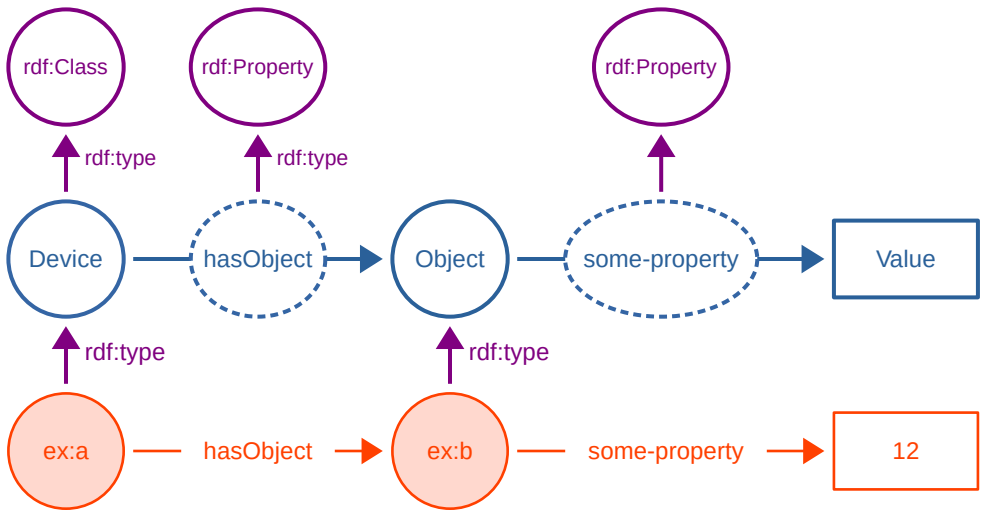
While predicate IRIs as used in a statement do not have a value, they are still IRIs and appear as a node in some other ontology. For example, the predicate ex:hasFather is used in a graph of family members, which is defined to be an rdf:Property with an rdfs:domain and rdfs:range of ex2:Person.



A refinement design pattern is also used for literal values. In many cases (like names and identifiers) the literal value is directly encoded in the graph, these are “L1” values.

Where there is additional provenence information about the value, or a reference for where to go to get a value, an intermediate node is introduced, “L2” values.

In some rare cases where versioning is used or the same value is represented in multiple ways, another node is introduced, an “L3” value.



The basic BACnet/RDF model is very simple, there is a rdf:Class called Device, an rdf:Property called hasObject, which references another rdf:Class called Object.

```
<ex:a> bacnet:hasObject <ex:b> .
<ex:b> bacnet:some-property 12 .
```

Primitive Data Type	Encoded Literal
Null	bacnet:Null
Boolean	`"true"^^xsd:boolean`
Unsigned Integer	`"64"^^xsd:nonNegativeInteger`
Signed Integer	`"64"^^xsd:integer`
Real	`"75.3"^^xsd:float`
Double	`"75.3"^^xsd:double`
Octet String	`"DEADBEEF"^^xsd:hexString`
Character String	`"Hello, world!"`

Primitive Data Type	Encoded Literal
Enumerated	`"degrees-fahrenheit"`
	`"64"`
	`64`
	bacnet:EngineeringUnits.degrees-fahrenheit

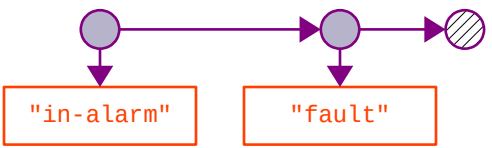
Primitive Data Type

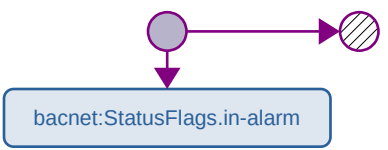
Encoded Literal

"in-alarm;fault"

"0;1"

Bit String

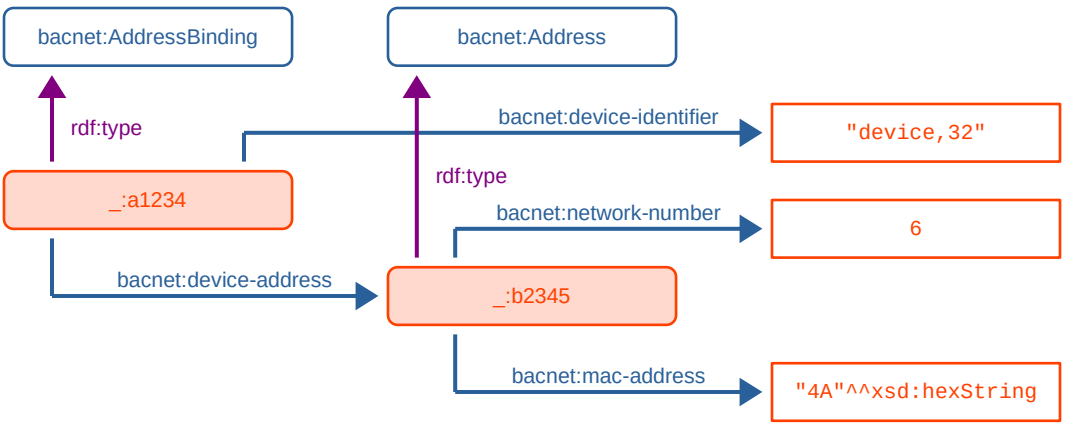




Primitive Data Type	Encoded Literal
Date	`"2021-10-21"^^xsd:date`
Date Pattern	`"2021-10-21 *"`
	`"2021-10-21 *"^^bacnet:date`
Time	`"11:30:35.23"^^xsd:time`
Time Pattern	`"11:30:*.*"^^bacnet:time`
Object Identifier	`"analog-value,12"`
	`"2,12"`

Sequence

Pattern

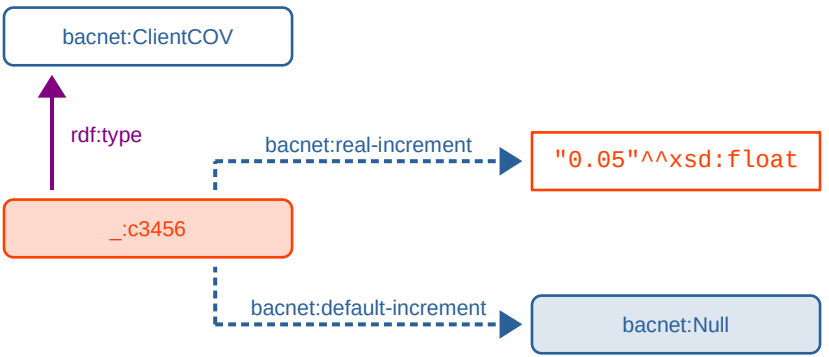


This is simpler than what you might expect:

```
<ex:a> bacnet:device-address-binding (
        [ bacnet:device-identifier “device,32” ;
          bacnet:device-address [
            bacnet:network-number 6 ;
            bacnet:mac-address “4A”^^xsd:hexString
          ]
        ]
    ) .
```

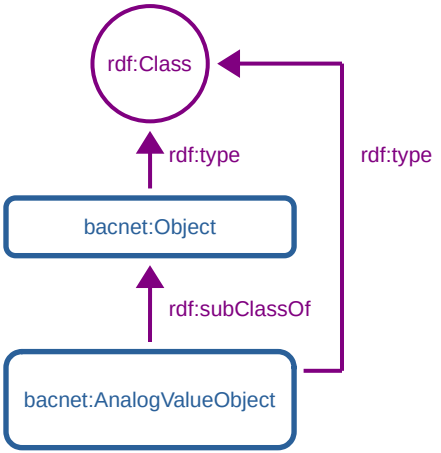
Choice

Pattern



What you might expect:

```
<ex:a> bacnet:client-cov-increment [
            bacnet:real-increment “0.05”^^xsd:float
    ] .
```

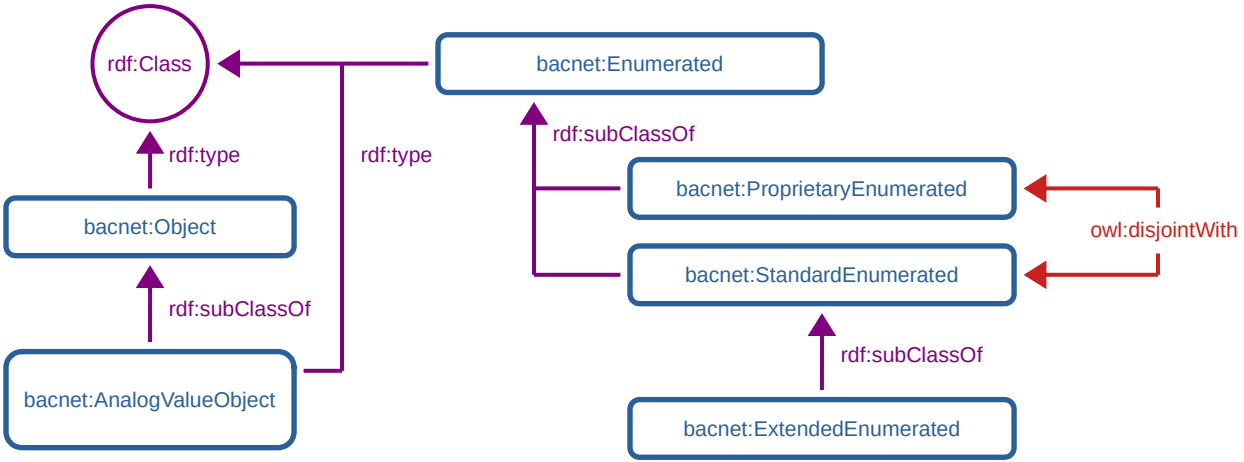


This is an explicit reference to the type:

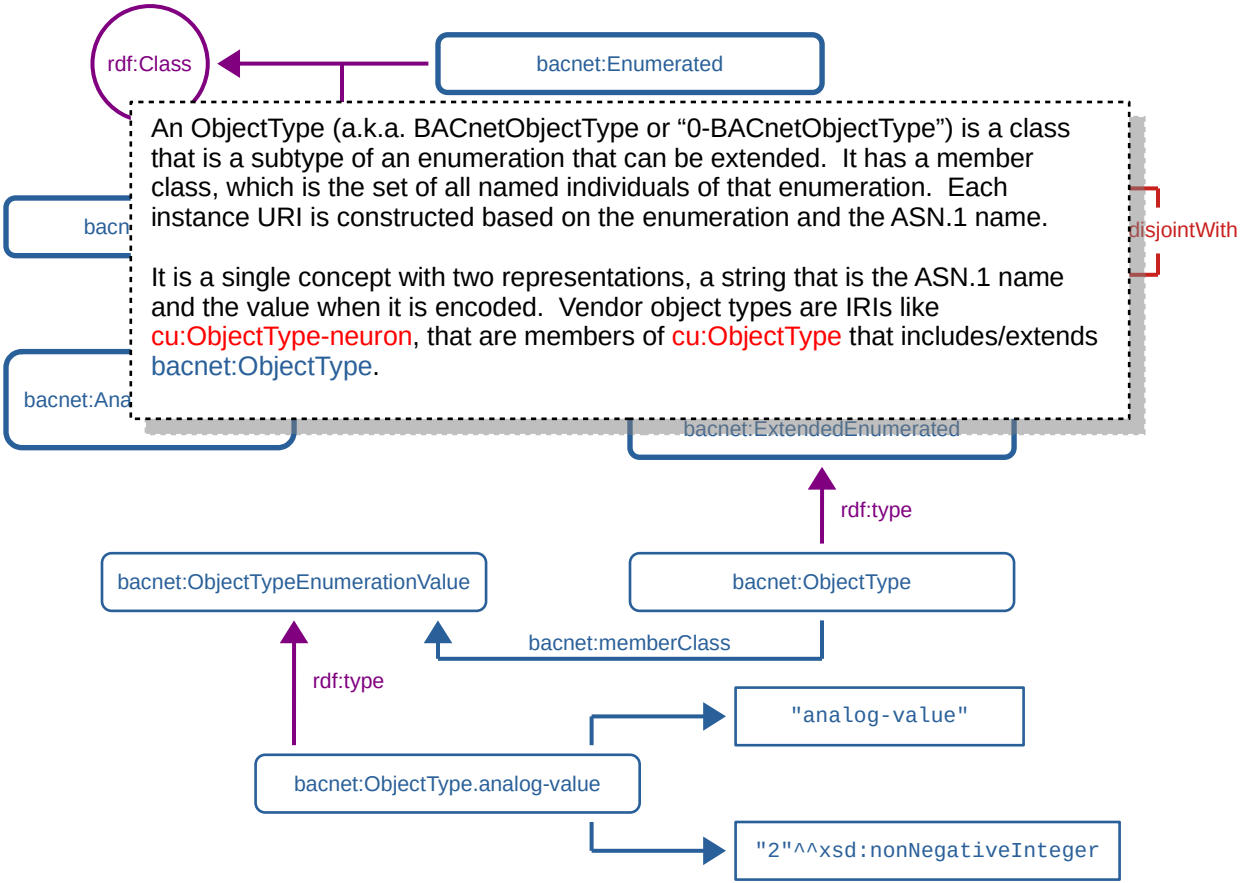
```
<ex:a> a bacnet:AnalogValueObject .
```

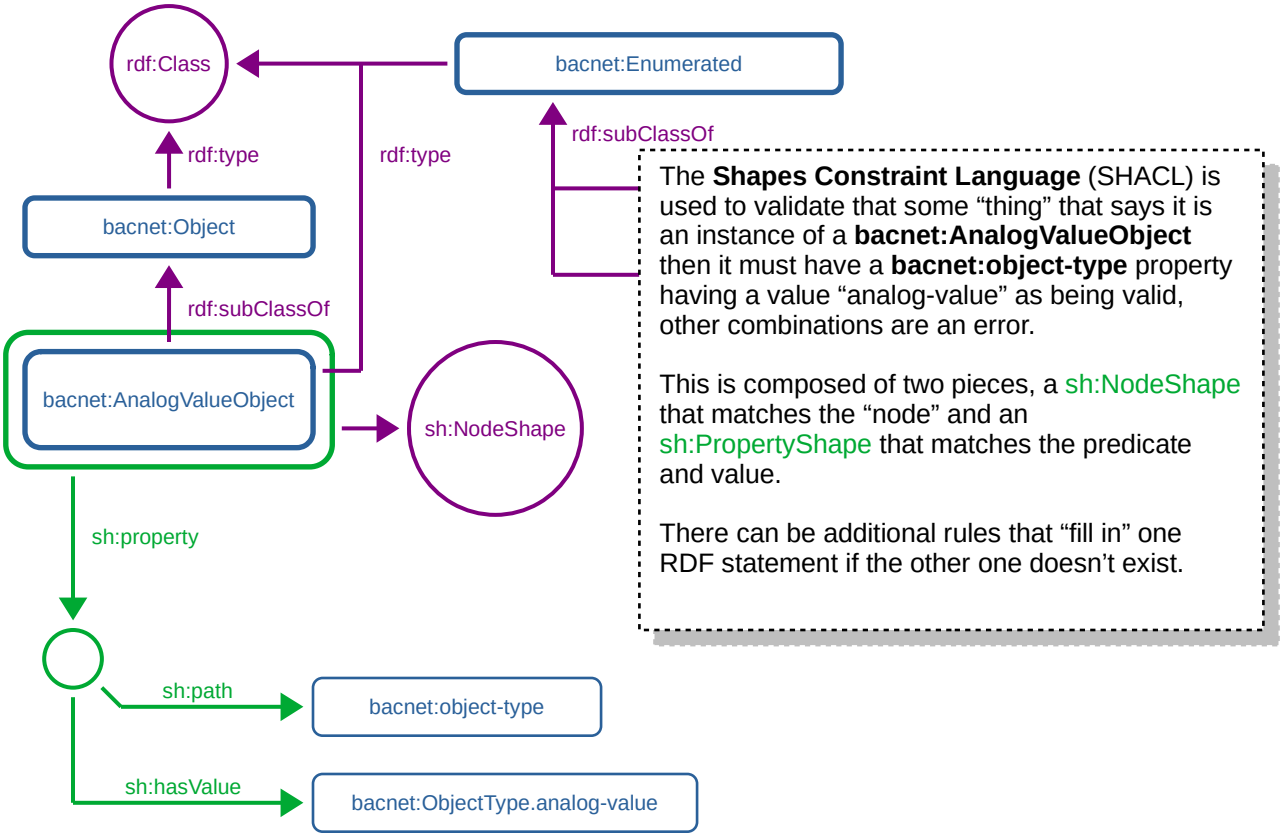
This is an implicit reference to the type with a property and enumerated value:

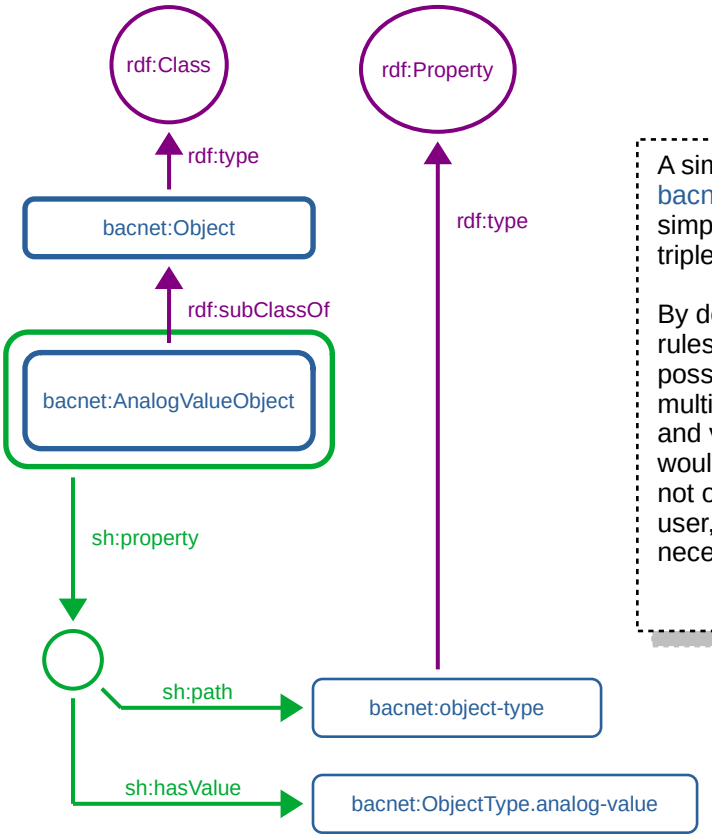
```
<ex:a> bacnet:object-type bacnet:ObjectType.analog-value .
```



BACnet enumerations are divided into two disjoint classes; those specified in the standard and those provided by a vendor. The standard enumerations are divided into two subclasses; those that are fixed and those that can be extended by a vendor.


rdf:Class
bacnet:Enumerated
An ObjectType (a.k.a. BACnetObjectType or “0-BACnetObjectType”) is a class that is a subtype of an enumeration that can be extended. It has a member class, which is the set of all named individuals of that enumeration. Each instance URI is constructed based on the enumeration and the ASN.1 name.
It is a single concept with two representations, a string that is the ASN.1 name and the value when it is encoded. Vendor object types are IRIs like cu:ObjectType-neuron, that are members of cu:ObjectType that includes/extends bacnet:ObjectType.
disjointWith
bacnet:ExtendedEnumerated
rdf:type
bacnet:ObjectTypeEnumerationValue
bacnet:ObjectType
bacnet:memberClass
rdf:type
"analog-value"
bacnet:ObjectType.analog-value
"2"^^xsd:nonNegativeInteger


rdf:Class
bacnet:Enumerated
rdf:type
rdf:type
rdf:subClassOf
bacnet:Object
rdf:subClassOf
bacnet:AnalogValueObject
sh:NodeShape
sh:property
sh:path
bacnet:object-type
sh:hasValue
bacnet:ObjectType.analog-value
The **Shapes Constraint Language** (SHACL) is used to validate that some “thing” that says it is an instance of a **bacnet:AnalogValueObject** then it must have a **bacnet:object-type** property having a value “analog-value” as being valid, other combinations are an error.
This is composed of two pieces, a sh:NodeShape that matches the “node” and an sh:PropertyShape that matches the predicate and value.
There can be additional rules that “fill in” one RDF statement if the other one doesn’t exist.

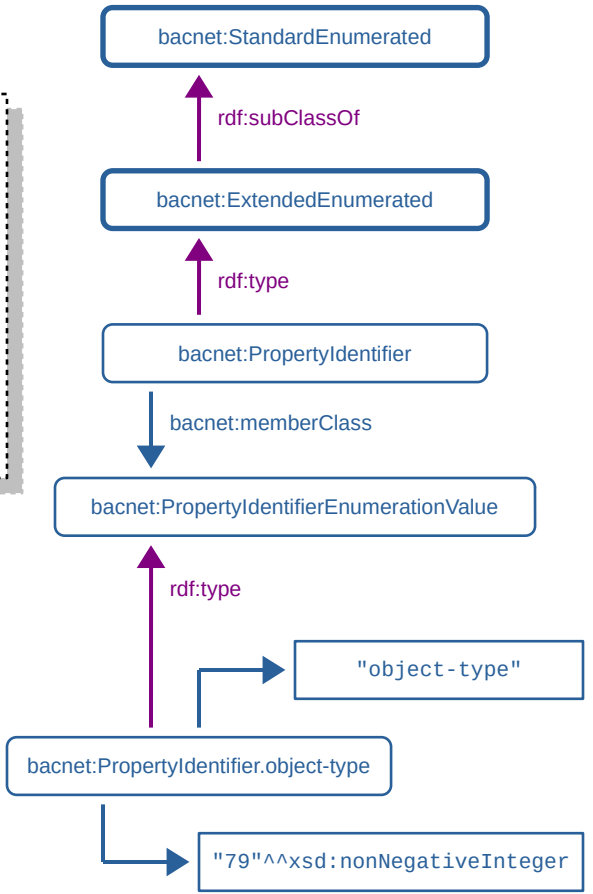


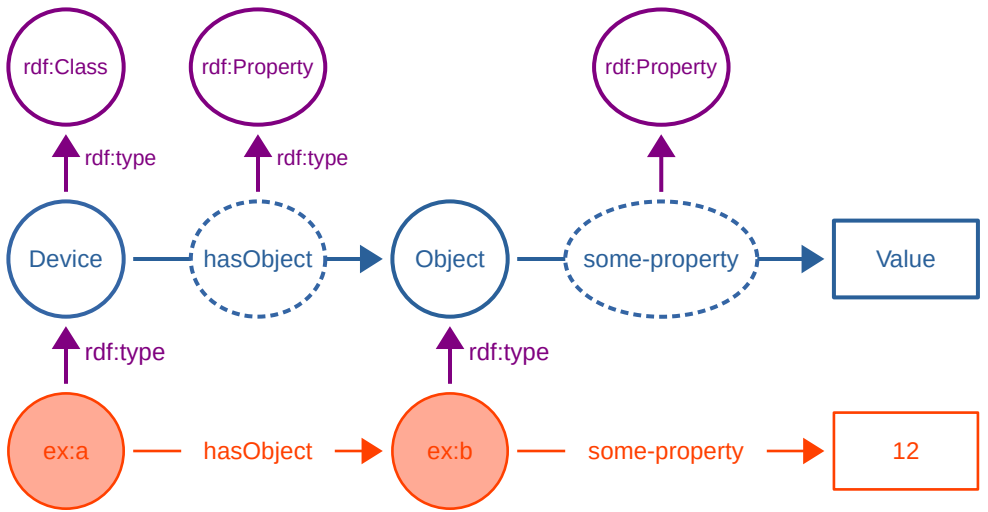
A similar design pattern is used for the bacnet:object-type, which in terms of RDF is simply a property (the predicate part of a triple).

By defining an RDF *data model* using SHACL rules, rather than an OWL *ontology*, it is possible to use the same property/predicate in multiple contexts (like **bacnet:present-value**) and validate different combinations of what would otherwise be the *domain* and *range*. It not only reduces the cognative load on the user, but significantly reduces the computation necessary to validate an instance of the model.

A PropertyIdentifier (a.k.a. BACnetPropertyIdentifier) is an enumeration and follows the same pattern as all other enumerations, it is also an instance of an extended enumerated.

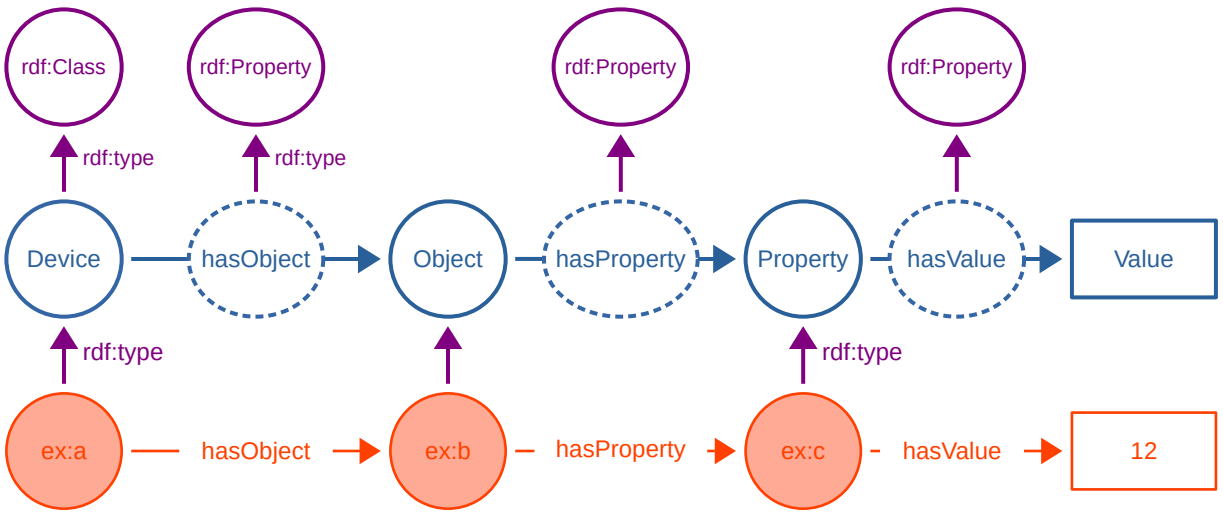
It is a single concept with two representations, a string that is the ASN.1 name and the value when it is encoded. Vendor property identifiers are IRIs like cu:PropertyIdentifier.synapse, that are instances of cu:PropertyIdentifierEnumerationValue.





This an **L1** model, one hop from the object to the value and the predicate matches the ASN.1 property name as expressed in CSML.

```
<bacnet://1234> bacnet:hasObject <bacnet://1234/analog-value,1> .
<bacnet://1234/analog-value,1> bacnet:some-property 12 .
```



This an **L2** model, two hops from the property through a Property to the value. In this case, `ex:c` would conceptually similar to `bacnet://1234/analog-value,1/some-property`

```
<ex:a> bacnet:hasObject <ex:b> .
<ex:b> bacnet:hasProperty <ex:c> .

<ex:c> bacnet:hasValue 12 .
<ex:c> a bacnet:Present_Value .
```

CSML

```
<?xml version="1.0" encoding="UTF-8"?>
<CSML xmlns="http://bacnet.org/csml/1.2" defaultLocale="en-US">
    <!--
        Define a test sequence with a single required null.
    -->
    <Definitions>
        <Sequence name="0-BACnetTestSequence">
            <Null name="null"/>
        </Sequence>
    </Definitions>
</CSML>
```

SHACL

```
@prefix bacnet: <http://data.ashrae.org/bacnet/2020#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .

bacnet:TestSequence a sh:NodeShape ;
    sh:property [
        a sh:PropertyShape ;
        sh:path bacnet:null ;
        sh:hasValue bacnet:Null ;
        sh:maxCount 1 ;
        sh:minCount 1 ;
        ] .
```

Ex

```
<ex:1> bacnet:null bacnet:Null .
```

CSML

```
<?xml version="1.0" encoding="UTF-8"?>
<CSML xmlns="http://bacnet.org/csml/1.2" defaultLocale="en-US">
    <!--
        Define a test sequence with a single optional null.
    -->
    <Definitions>
        <Sequence name="0-BACnetTestSequence">
            <Null name="null" optional="true"/>
        </Sequence>
    </Definitions>
</CSML>
```

SHACL

```
@prefix bacnet: <http://data.ashrae.org/bacnet/2020#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .

bacnet:TestSequence a sh:NodeShape ;
    sh:property [
        a sh:PropertyShape ;
        sh:path bacnet:null ;
        sh:hasValue bacnet:Null ;
        sh:maxCount 1 ;
        sh:minCount 0 ;
        ] .
```

Ex

```
<ex:1> bacnet:null bacnet:Null .
```

CSML

```
<?xml version="1.0" encoding="UTF-8"?>
<CSML xmlns="http://bacnet.org/csml/1.2" defaultLocale="en-US">
    <Definitions>
        <Sequence name="0-BACnetTestSequence">
            <Boolean name="boolean"/>
        </Sequence>
    </Definitions>
</CSML>
```

SHACL

```
@prefix bacnet: <http://data.ashrae.org/bacnet/2020#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

bacnet:TestSequence a sh:NodeShape ;
    sh:property [ a sh:PropertyShape ;
            sh:datatype xsd:boolean ;
            sh:maxCount 1 ;
            sh:minCount 1 ;
            sh:path bacnet:boolean ] .
```

Ex

```
<ex:1> bacnet:boolean true .
```

Turtle

```
@prefix bacnet: <http://data.ashrae.org/bacnet/2020#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

bacnet:TestSequence a sh:NodeShape ;
    sh:property [ a sh:PropertyShape ;
            sh:datatype xsd:boolean ;
            sh:maxCount 1 ;
            sh:minCount 1 ;
            sh:path bacnet:boolean ] .
```

XML

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:sh="http://www.w3.org/ns/shacl#"
>
  <rdf:Description rdf:nodeID="n195ff53ddf1b4e0a8d5b8b9c14890d86b1">
    <rdf:type rdf:resource="http://www.w3.org/ns/shacl#PropertyShape"/>
    <sh:datatype rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
    <sh:maxCount rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</sh:maxCount>
    <sh:minCount rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</sh:minCount>
    <sh:path rdf:resource="http://data.ashrae.org/bacnet/2020#boolean"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://data.ashrae.org/bacnet/2020#TestSequence">
    <rdf:type rdf:resource="http://www.w3.org/ns/shacl#NodeShape"/>
    <rdfs:subClassOf rdf:resource="http://data.ashrae.org/bacnet/2020#Sequence"/>
    <sh:property rdf:nodeID="n195ff53ddf1b4e0a8d5b8b9c14890d86b1"/>
  </rdf:Description>
</rdf:RDF>
```

CSML

```
<?xml version="1.0" encoding="UTF-8"?>
<CSML xmlns="http://bacnet.org/csml/1.2" defaultLocale="en-US">
    <Definitions>
        <Sequence name="0-BACnetTestSequence">
            <Unsigned name="unsigned"/>
        </Sequence>
    </Definitions>
</CSML>
```

SHACL

```
@prefix bacnet: <http://data.ashrae.org/bacnet/2020#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

bacnet:TestSequence a sh:NodeShape ;
    rdfs:subClassOf bacnet:Sequence ;
    sh:property [ a sh:PropertyShape ;
            sh:datatype xsd:nonNegativeInteger ;
            sh:maxCount 1 ;
            sh:minCount 1 ;
            sh:path bacnet:unsigned ] .
```

Ex

```
<ex:1> bacnet:unsigned “12”^^xsd:nonNegativeInteger .
```

CSML

```
<?xml version="1.0" encoding="UTF-8"?>
<CSML xmlns="http://bacnet.org/csml/1.2" defaultLocale="en-US">
    <Definitions>
        <Object name="0-AnalogInputObject">
               <Unsigned name="time-delay-normal" optional="true"
                  propertyIdentifier="356" value="10"
                  variability="operation-setting"
               />
        </Object>
    </Definitions>
</CSML>
```

SHACL

```
@prefix bacnet: <http://data.ashrae.org/bacnet/2020#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

bacnet:AnalogInputObject a rdfs:Class,
        sh:NodeShape ;
    rdfs:subClassOf bacnet:Object ;
    sh:property [ a sh:PropertyShape ;
            sh:hasValue "10"^^xsd:nonNegativeInteger ;
            sh:maxCount 1 ;
            sh:minCount 0 ;
            sh:path bacnet:time-delay-normal ] .
```

Ex

```
<ex:1> bacnet:time-delay-normal “10”^^xsd:nonNegativeInteger .
```

CSML

```
<?xml version="1.0" encoding="UTF-8"?>
<CSML xmlns="http://bacnet.org/csml/1.2" defaultLocale="en-US">
    <Definitions>
        <BitString name="0-BACnetDaysOfWeek">
            <NamedBits>
                <Bit name="monday" bit="0"/>
                <Bit name="tuesday" bit="1"/>
            </NamedBits>
        </BitString>
        <Sequence name="0-TestSequence1">
            <BitString name="some-bitstring" type="0-BACnetDaysOfWeek"/>
        </Sequence>
    </Definitions>
</CSML>
```

SHACL

```
bacnet:DaysOfWeek a rdfs:Class,
        sh:NodeShape ;
    bacnet:memberClass bacnet:DaysOfWeekNamedBits ;
    rdfs:subClassOf bacnet:Bitstring .

bacnet:DaysOfWeekNamedBits a rdfs:Class ;
    rdfs:subClassOf bacnet:NamedBits .

bacnet:DaysOfWeek.monday a bacnet:DaysOfWeekNamedBits ;
    bacnet:bit "0"^^xsd:nonNegativeInteger ;
    bacnet:name "monday" .

bacnet:DaysOfWeek.tuesday a bacnet:DaysOfWeekNamedBits ;
    bacnet:bit "1"^^xsd:nonNegativeInteger ;
    bacnet:name "tuesday" .
```

Ex

```
<ex:1> bacnet:some-bitstring ( bacnet:DaysOfWeek-monday ) .
```

CSML

```
<?xml version="1.0" encoding="UTF-8"?>
<CSML xmlns="http://bacnet.org/csml/1.2" defaultLocale="en-US">
    <Definitions>
        <BitString name="0-BACnetDaysOfWeek">
            <NamedBits>
                <Bit name="monday" bit="0"/>
                <Bit name="tuesday" bit="1"/>
            </NamedBits>
        </BitString>
        <Sequence name="0-TestSequence1">
            <BitString name="some-bitstring" type="0-BACnetDaysOfWeek"/>
        </Sequence>
    </Definitions>
</CSML>
```

SHACL

```
bacnet:TestSequence1 a sh:NodeShape ;
    rdfs:subClassOf bacnet:Sequence ;
    sh:property
        [ a sh:PropertyShape ;
            sh:maxCount 1 ;
            sh:minCount 1 ;
            sh:path bacnet:some-bitstring ] ;
    sh:xone (
        [ sh:property [
            sh:class bacnet:DaysOfWeekNamedBits ;
            sh:minCount 1 ;
            sh:path ( bacnet:some-bitstring [ sh:zeroOrMorePath rdf:rest ] rdf:first )
        ] ]
        [ sh:property [
            sh:hasValue () ;
            sh:minCount 1 ;
            sh:path bacnet:some-bitstring ] ]
        [ sh:property [
            sh:maxCount 0 ;
            sh:path bacnet:some-bitstring ] ]
        ) .
```

CSML

```
<?xml version="1.0" encoding="UTF-8"?>
<CSML xmlns="http://bacnet.org/csml/1.2" defaultLocale="en-US">
    <Definitions>
        <BitString name="0-BACnetDaysOfWeek">
            <NamedBits>
                <Bit name="monday" bit="0"/>
                <Bit name="tuesday" bit="1"/>
            </NamedBits>
        </BitString>
        <Sequence name="0-TestSequence1">
            <BitString name="some-bitstring" type="0-BACnetDaysOfWeek"/>
        </Sequence>
    </Definitions>
</CSML>
```

SHACL

```
bacnet:TestSequence1 a sh:NodeShape ;
    rdfs:subClassOf bacnet:Sequence ;
    sh:property
        [ a sh:PropertyShape ;
            sh:maxCount 1 ;
            sh:minCount 1 ;
            sh:path bacnet:some-bitstring ] ;
    sh:xone (
        [ sh:property [
            sh:class bacnet:DaysOfWeekNamedBits ;
            sh:minCount 1 ;
            sh:path ( bacnet:some-bitstring [ sh:zeroOrMorePath rdf:rest ] rdf:first )
        ] ]
        [ sh:property [
            sh:hasValue () ;
            sh:minCount 1 ;
            sh:path bacnet:some-bitstring ] ]
        [ sh:property [
            sh:maxCount 0 ;
            sh:path bacnet:some-bitstring ] ]
        ) .
```

CSML

```
<?xml version="1.0" encoding="UTF-8"?>
<CSML xmlns="http://bacnet.org/csml/1.2" defaultLocale="en-US">
    <Definitions>
        <Choice name="0-AtomicWriteFile-ACK">
            <Choices>
                <Integer name="file-start-position" contextTag="0"/>
                <Integer name="file-start-record" contextTag="1"/>
            </Choices>
        </Choice>
    </Definitions>
</CSML>
```

SHACL

```
bacnet:AtomicWriteFileACK a sh:NodeShape ;
    sh:xone ( [ sh:property [ a sh:PropertyShape ;
                        sh:datatype xsd:integer ;
                        sh:maxCount 1 ;
                        sh:minCount 1 ;
                        sh:path bacnet:file-start-position ] ]
              [ sh:property [ a sh:PropertyShape ;
                        sh:datatype xsd:integer ;
                        sh:maxCount 1 ;
                        sh:minCount 1 ;
                        sh:path bacnet:file-start-record ] ]
    ) .
```

Ex

```
<ex:1> bacnet:file-start-position 5 .
```

CSML

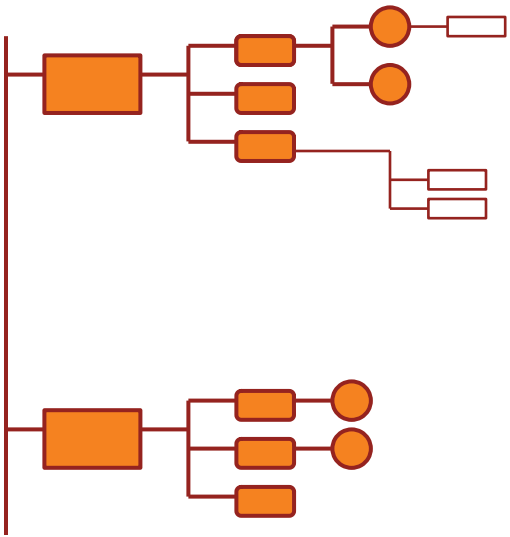
```
<?xml version="1.0" encoding="UTF-8"?>
<CSML xmlns="http://bacnet.org/csml/1.2" defaultLocale="en-US">
    <Definitions>
        <Choice name="TestChoice">
            <Choices>
                <Integer name="some-integer"/>
                <Sequence name="some-sequence">
                    <Null name="some-null"/>
                </Sequence>
            </Choices>
        </Choice>
    </Definitions>
</CSML>
```

SHACL

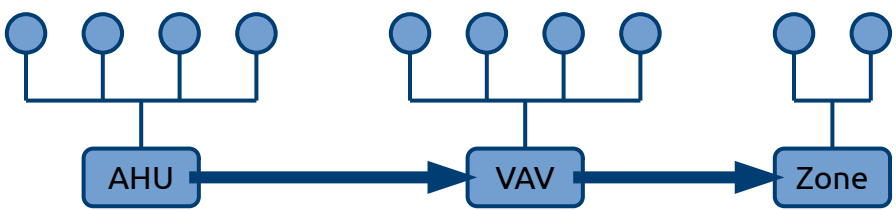
```
bacnet:TestChoice a sh:NodeShape ;
    sh:xone ( [ sh:property [ a sh:PropertyShape ;
                        sh:datatype xsd:integer ;
                        sh:maxCount 1 ;
                        sh:minCount 1 ;
                        sh:path bacnet:some-integer ] ]
              [ sh:property [ a sh:PropertyShape ;
                        sh:class bacnet:TestChoiceSomeSequence ;
                        sh:maxCount 1 ;
                        sh:minCount 1 ;
                        sh:path bacnet:some-sequence ] ]
    ) .
```

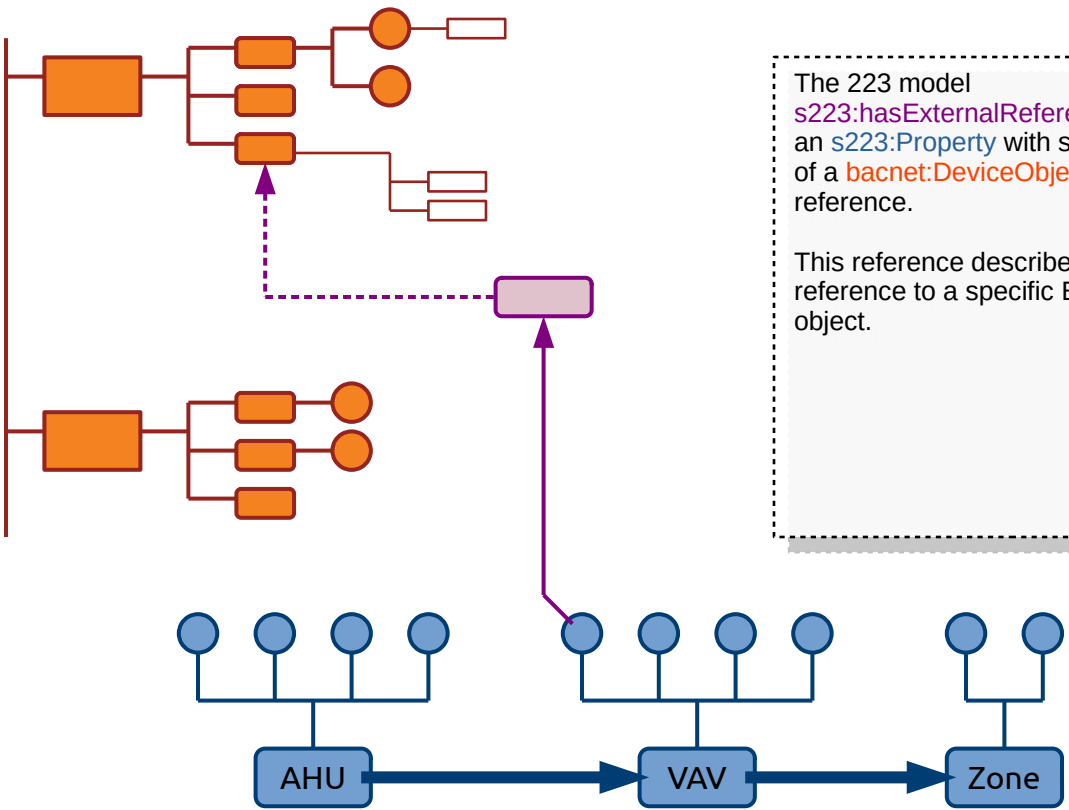
Ex

```
<ex:1> bacnet:some-sequence [ bacnet:some-null bacnet:Null ] .
```



Given a BACnet object with a bacnet:tags property referencing a list of class names, Haystack calls them marker tags.



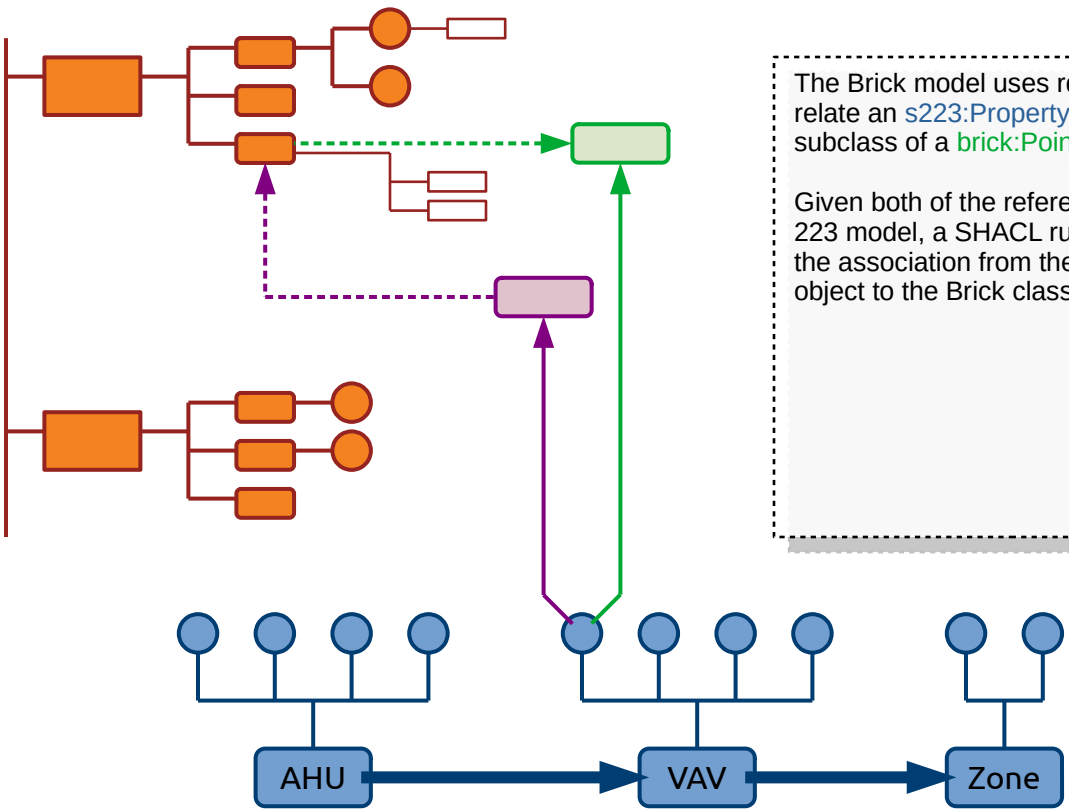

The 223 model s223:hasExternalReference relates an s223:Property with some version of a bacnet:DeviceObjectProperty reference.
This reference describes an implicit reference to a specific BACnet object.
AHU
VAV
Zone

CSML

```
<?xml version="1.0" encoding="UTF-8"?>
<CSML xmlns="http://bacnet.org/csml/1.2" defaultLocale="en-US">
    <Definitions>
        <Sequence name="0-BACnetDeviceObjectPropertyReference">
            <ObjectIdentifier name="object-identifier"/>
            <Enumerated name="property-identifier" type="0-BACnetPropertyIdentifier"/>
            <Unsigned name="property-array-index" optional="true"/>
            <ObjectIdentifier name="device-identifier" optional="true"/>
        </Sequence>
        <Object name="Property">
            <Sequence name="hasExternalReference"
                type="0-BACnetDeviceObjectPropertyReference"/>
        </Object>
    </Definitions>
</CSML>
```

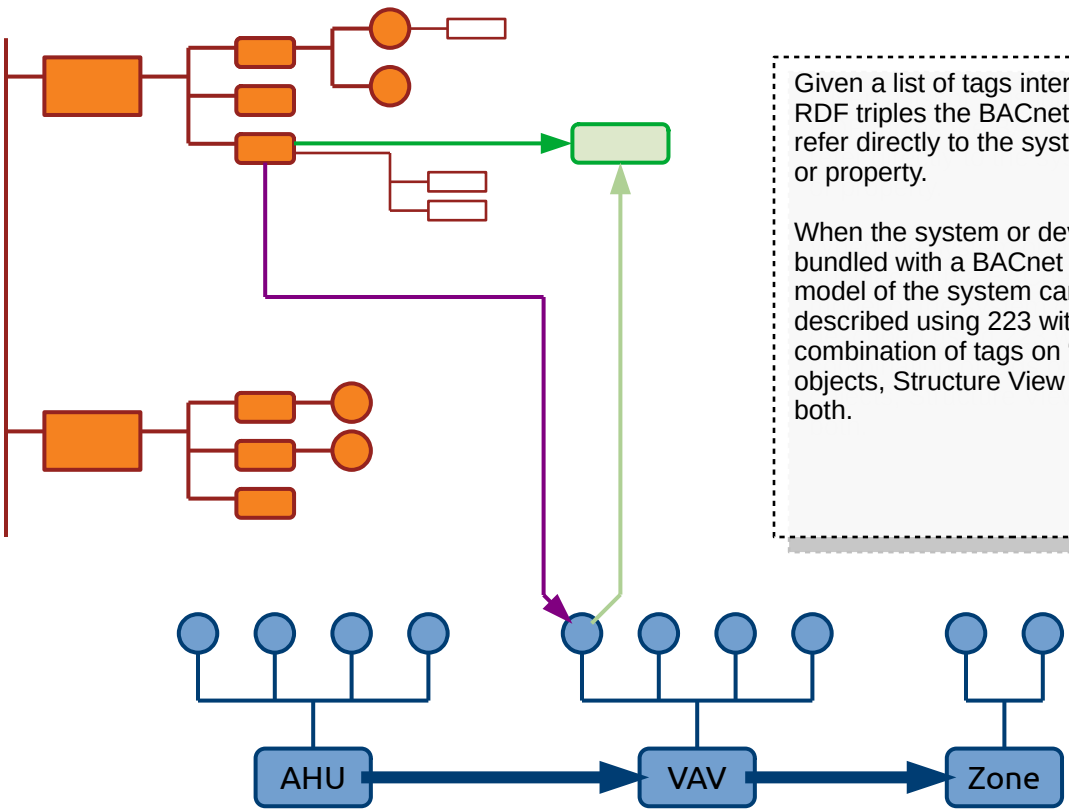
Ex

```
<ex:1> s223:hasExternalReference [
    a bacnet:DeviceObjectPropertyReference ;
    bacnet:device-identifier "device,1234" ;
    bacnet:object-identifier "analog-value,1" ;
    bacnet:property-identifier bacnet:PropertyIdentifier-present-value ;
    ] .
```



The Brick model uses rdf:type to relate an s223:Property with some subclass of a brick:Point.

Given both of the references in the 223 model, a SHACL rule can make the association from the BACnet object to the Brick class.



Given a list of tags interpreted as RDF triples the BACnet object can refer directly to the system, device, or property.

When the system or device is bundled with a BACnet controller, the model of the system can be described using 223 with a combination of tags on “regular” objects, Structure View Objects, or both.

CSML

```
<?xml version="1.0" encoding="UTF-8"?>
<CSML xmlns="http://bacnet.org/csml/1.2" defaultLocale="en-US">
    <Definitions>
        <Sequence name="0-BACnetNameValue">
            <String name="name" contextTag="0"/>
            <Any name="value" optional="true"
                comment="value is limited to primitve datatypes and BACnetDateTime"/>
        </Sequence>
        <Object name="0-AnalogValueObject">
            <Array memberType="0-BACnetNameValue" name="tags" optional="true"/>
        </Object>
    </Definitions>
</CSML>
```

Ex

```
<ex:1> a bacnet:AnalogValueObject ;
    bacnet:tags (
        [ bacnet:name "id" ; bacnet:value "a-0005" ]
        [ bacnet:name "cur" ]
        [ bacnet:name "his" ]
        [ bacnet:name "mixed" ]
        [ bacnet:name "air" ]
        [ bacnet:name "temp" ]
        [ bacnet:name "sensor" ]
        [ bacnet:name "point" ]
    ) .

_:a-0005 a bacnet:AnalogValueObject ;
    bacnet:tags (
        [ bacnet:name "brick:Mixed_Air_Temperature_Sensor" ]
    ) .
```