

BACnet Ontology

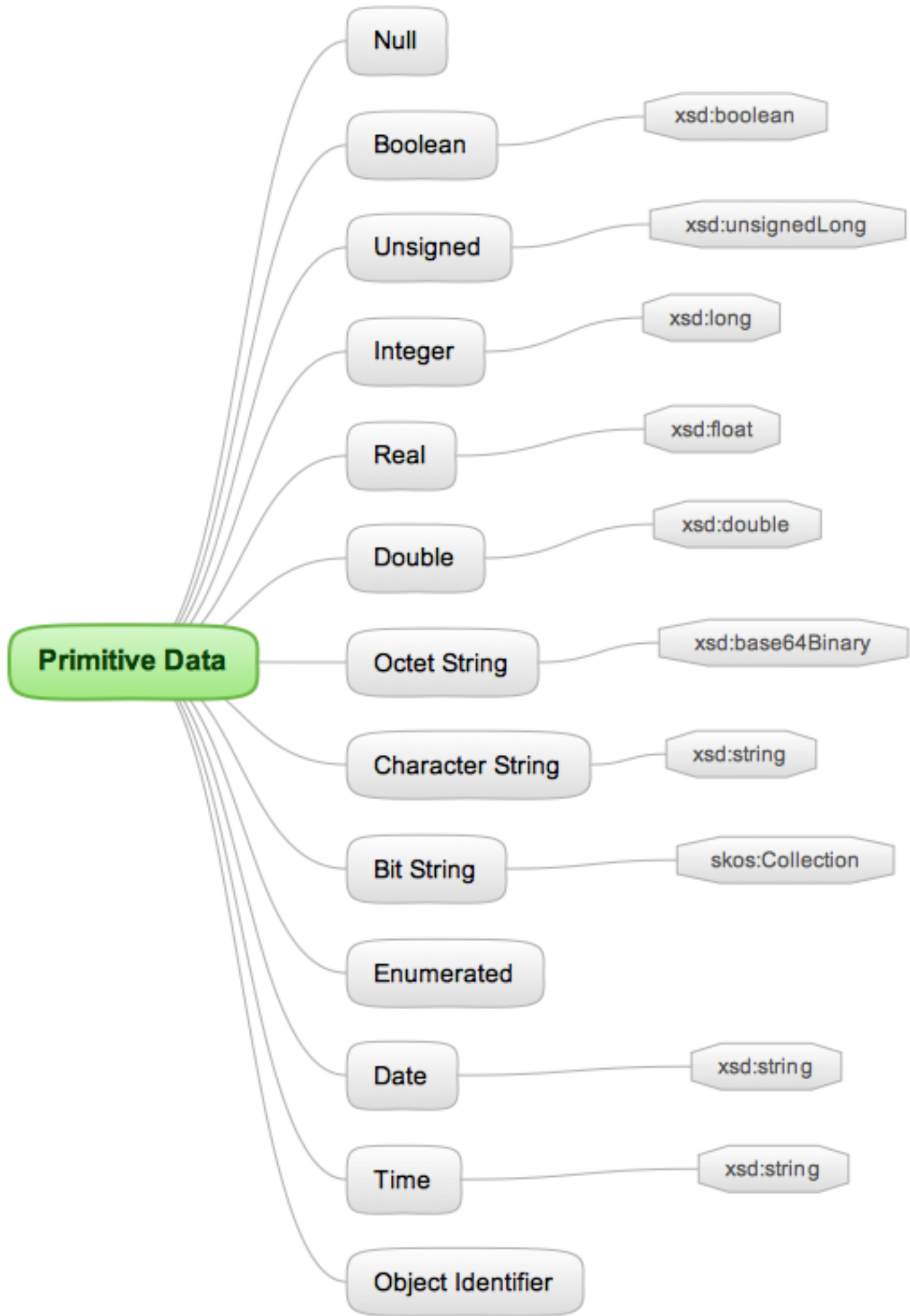
From text and ASN.1 to OWL

Top down? Bottom up?

- Top down:
 - Start with the BACnet lexicon defining key terms and definitions, eventually refining the formal description to something that matches the formal pieces
 - Joel says, “No formal topic mapping experience.”
- Bottom up:
 - Start with the ASN.1 formal definitions, re-writing each one in OWL terms, building up to something that can be used to describe the model
 - Joel says, “Hey, it’s just another programming language, and I have a Python library that has the ‘content’ of the ASN.1.”

Begin with The End in Mind

- What is the point? How should this be evaluated?
 - Provide URI's that precisely define BACnet concepts that can be referenced by other ontologies
 - Correctly map the BACnet model, both the pieces that are defined in ASN.1 and normative text
- Information Model or Data Model?
 - Joel says, "As I travel up and down the stack of concepts, it's not always clear. I'm probably mixing information model concepts into the data model, and using data model labels for information model concepts."

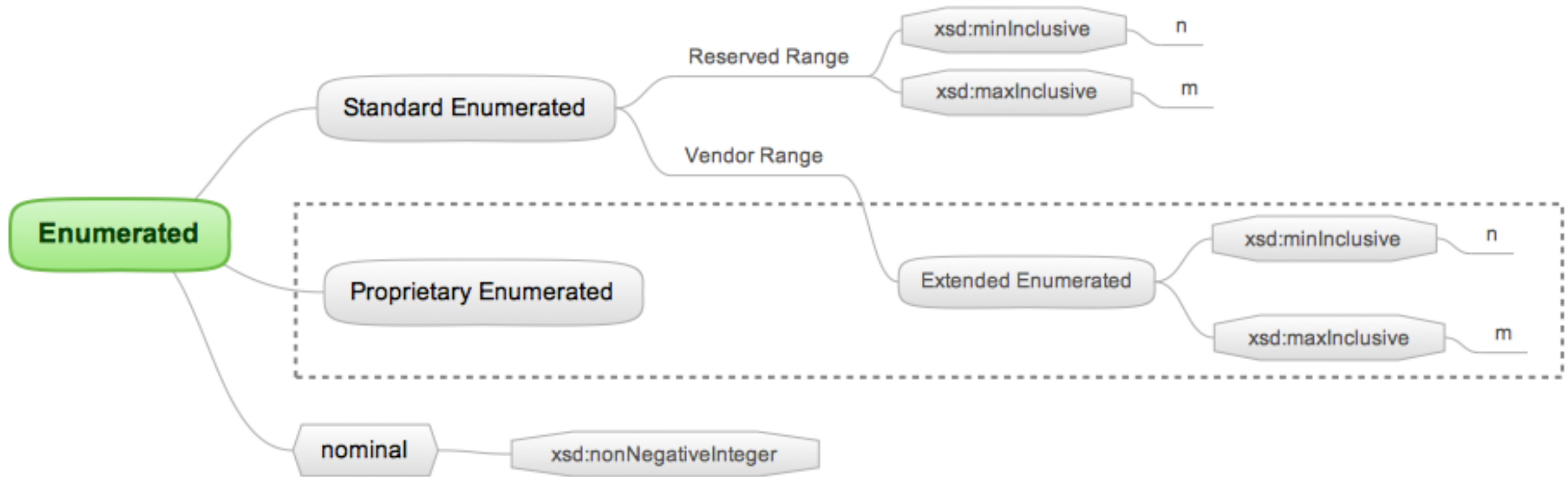


Null is Interesting



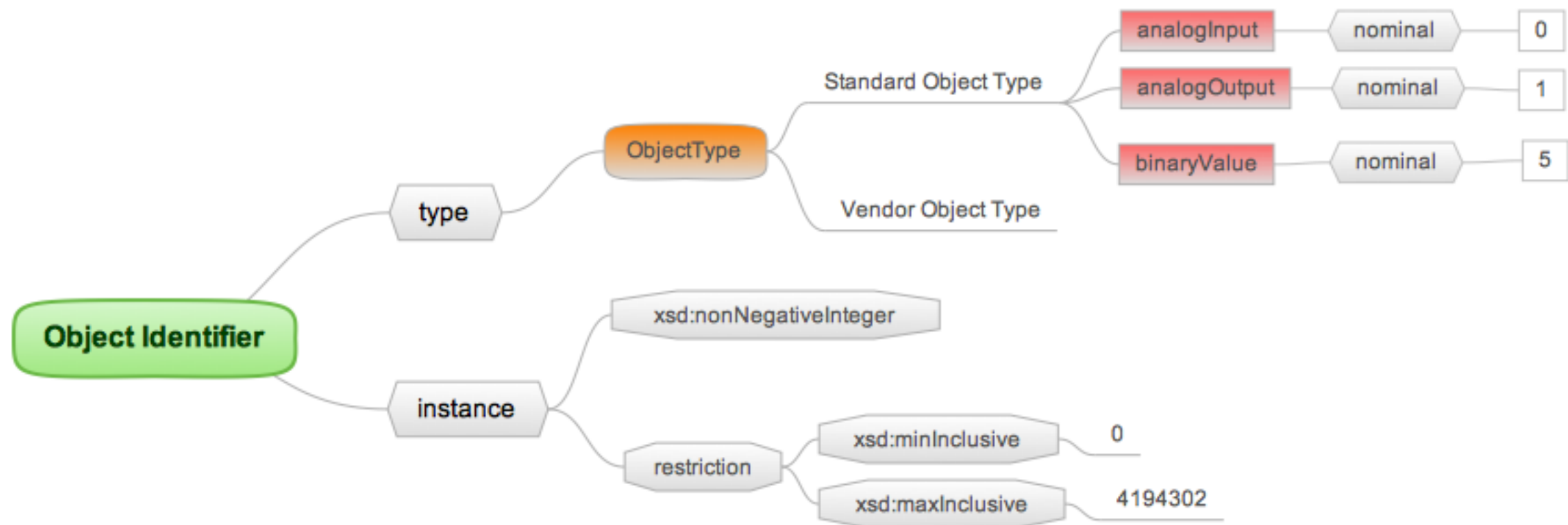
- Null is used as both a data type and as a special value, usually reserved for values that mean “this is not the droid you’re looking for”
- A “priority array” might have any number of values in it corresponding to conflicting algorithms, null would mean “I have no opinion.” “Highest” priority rules.

Enumerations are Interesting



- Enumerations are just like colors in C or Pascal; red=0, green=1, blue=2
- Some enumerations are defined in the standard, vendors can come up with their own
- Some enumerations are defined in the standard and vendors can extend them with their own values, as long as they are not in the range reserved for ASHRAE

Object Identifiers are Interesting



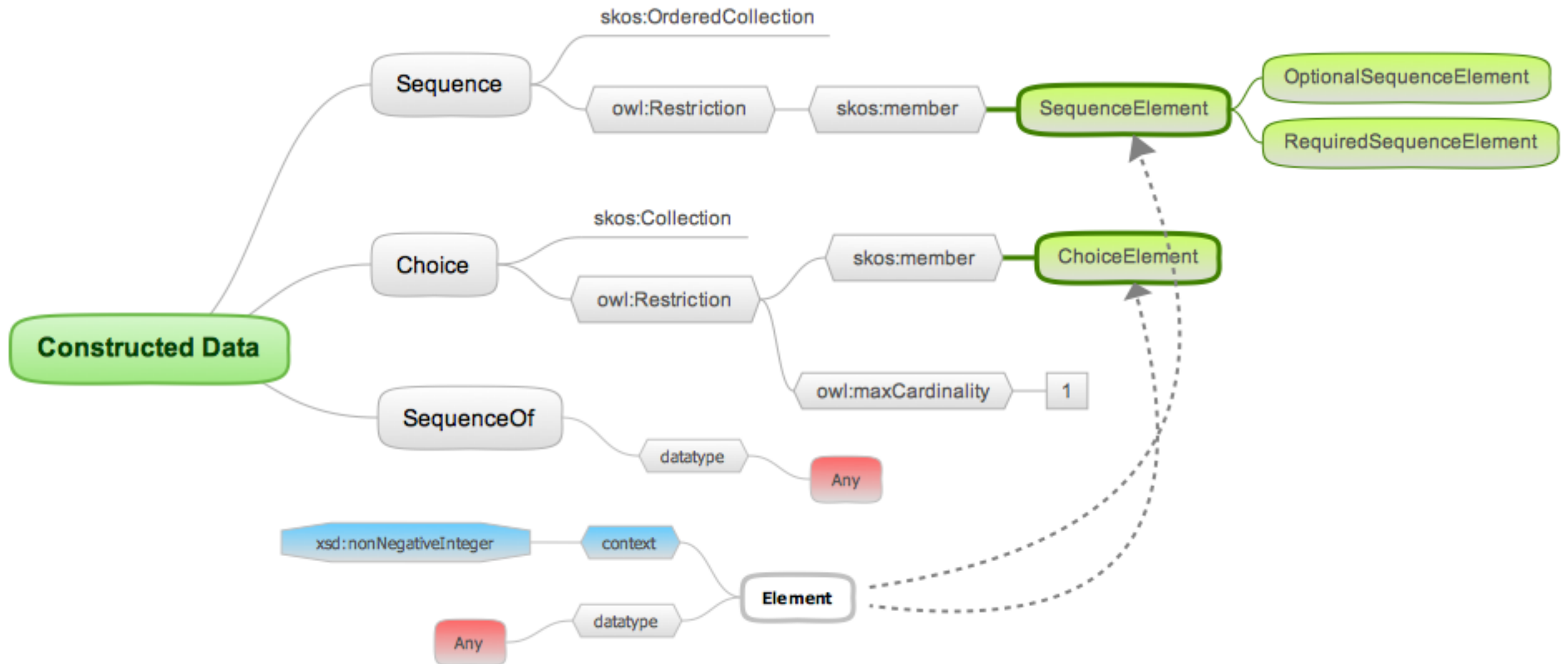
- Object identifiers are 32-bit packed structures with an object type and instance number
- The object type can be one of the standard types or a vendor type by using a value that is in the range reserved for extending the object type enumeration
- Note the use of owl:NamedIndividual's for enumerations

Bit Strings are Interesting



- Bit strings are collections of bits, very similar to colors; red=0, green=1, blue=2
- A bit string can have more than one bit set, otherwise BACnet would have defined it as an enumerated
- Bit "1" in some bit string is not the same bit as bit "1" in some other bit string
- Note the use of owl:NamedIndividual's for individual bits

Sequence's and Choice's



- A Sequence is analogous to a structure in C, a JSON object, or an element in Minimal XML
- Sequence elements can be optional or required, context tagged or not, and be any primitive or constructed data type

Sequence's and Choice's

- From this:

- AtomicReadFile-Request ::= SEQUENCE {
 fileIdentifier BACnetObjectIdentifier,
 accessMethod CHOICE {
 streamAccess [0] SEQUENCE {
 fileStartPosition INTEGER,
 requestedOctetCount Unsigned
 },
 recordAccess [1] SEQUENCE {
 fileStartRecord INTEGER,
 requestedRecordCount Unsigned
 }
 }
}

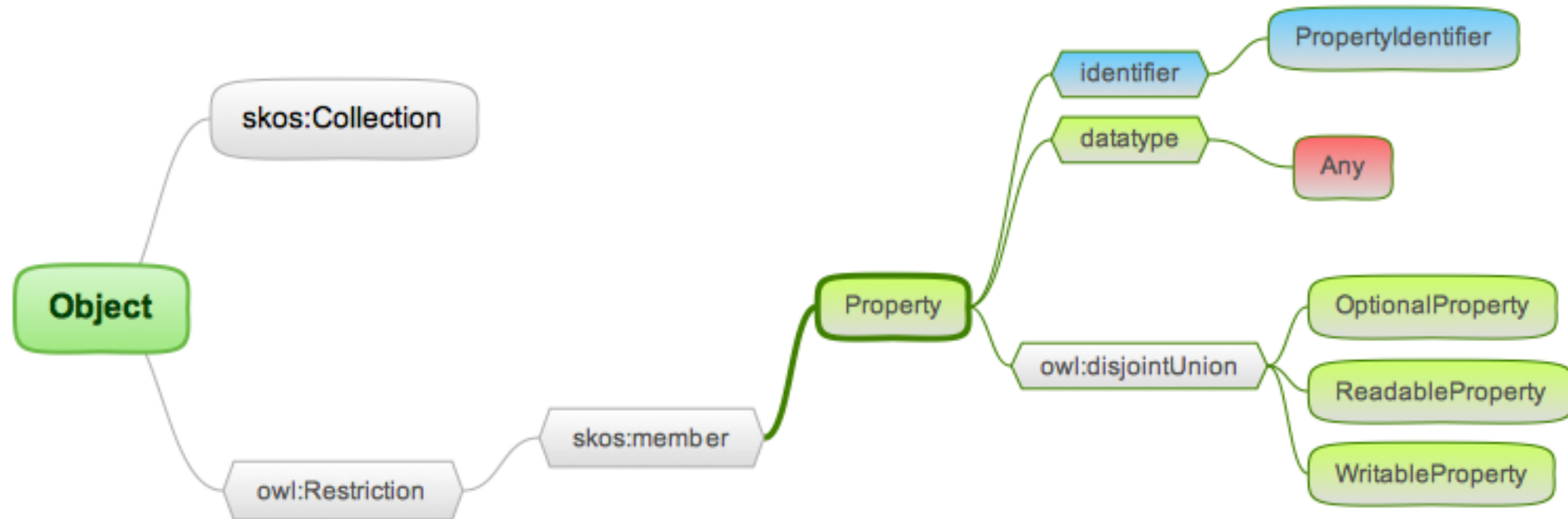
Sequence's and Choice's

- To this (JSON-LD):

- ```
{ "$type": "AtomicReadFileRequest",
 fileIdentifier: { type: "file", instance: 1 },
 accessMethod: {
 streamAccess: {
 fileStartPosition: 50,
 requestedOctetCount: 100
 }
 }
}
```

- Can structure elements be things in their own right, as well as properties of the structures they help describe?

# Objects and Properties



- Objects “have” properties
- All objects have some standard properties that are required
- Standard objects have additional required and optional properties
- Vendors can define their own properties for standard types (property identifiers are extended enumerations)
- Vendors can define their own types (object type identifiers are extended enumerations) which use standard properties or their own
- Property values can be any Primitive or Constructed data type.

# Cognitive Disconnect

- Saying what something *is*, what it *could be*, what it *cannot be*
- Subclass, domain, and range - they don't mean what you think they mean
- No canonical examples - simple statements found on the web have errors, can't be easily converted between formats
- Flat namespace - like going back to FORTRAN - modularize?

# Wrong tool? or wrong job?

- It's not clear that OWL is appropriate for data modeling
- It's not clear where the BACnet lexicon (with the normative text) stops (where the effort to ontologize it should stop) and the data model begins
- Effort continues to be worthwhile; highlights areas where my API needs to improve, something needs to be the anchor for BIM and sensor networks