

# BACnet Ontology Hackathon

From text and ASN.1 to OWL

# Project Plan

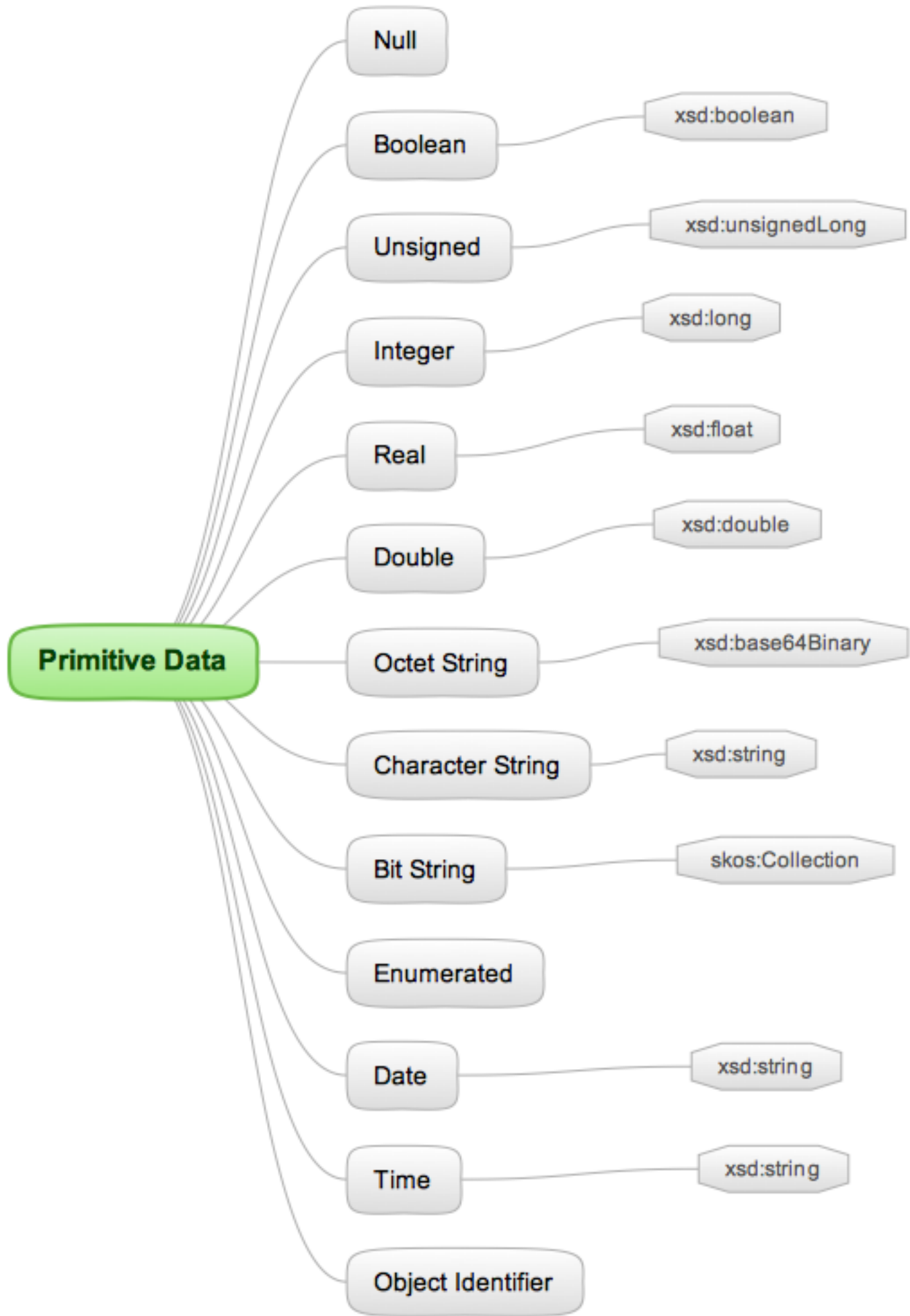
- Get a buy-in that a BACnet Ontology was worth doing, get some support
- Write some templates that take the Python library I've already written and generate OWL
- Feed the generated OWL into validators
- Spend a few hours interpreting error messages, if any
- Adjust the templates a bit, celebrate success

# Plan Slams into Reality

- Picking an OWL format harder than expected
  - too many to choose from; OWL Fn, OWL/XML, RDF/XML, Turtle
  - no clean round-trip, statements about individuals dropped
- Examples of fundamental CS constructs hard to find (atoms/literals, lists, sets, objects)
- Data modeling and set theory don't work well together
  - **You keep using [ subclass ], I do not think it means what you think it means ~ Inigo Montoya**

# Pre-Hackathon Progress

- Start with the simplest concepts, the “primitive data types”
- Most everything matches well
  - booleans, integers, strings are simple
  - binary data (blobs) not so bad
  - date/time can't be mapped (wildcard values) - s'OK
- Mental collision with named values (enumerations) and identifiers as data types

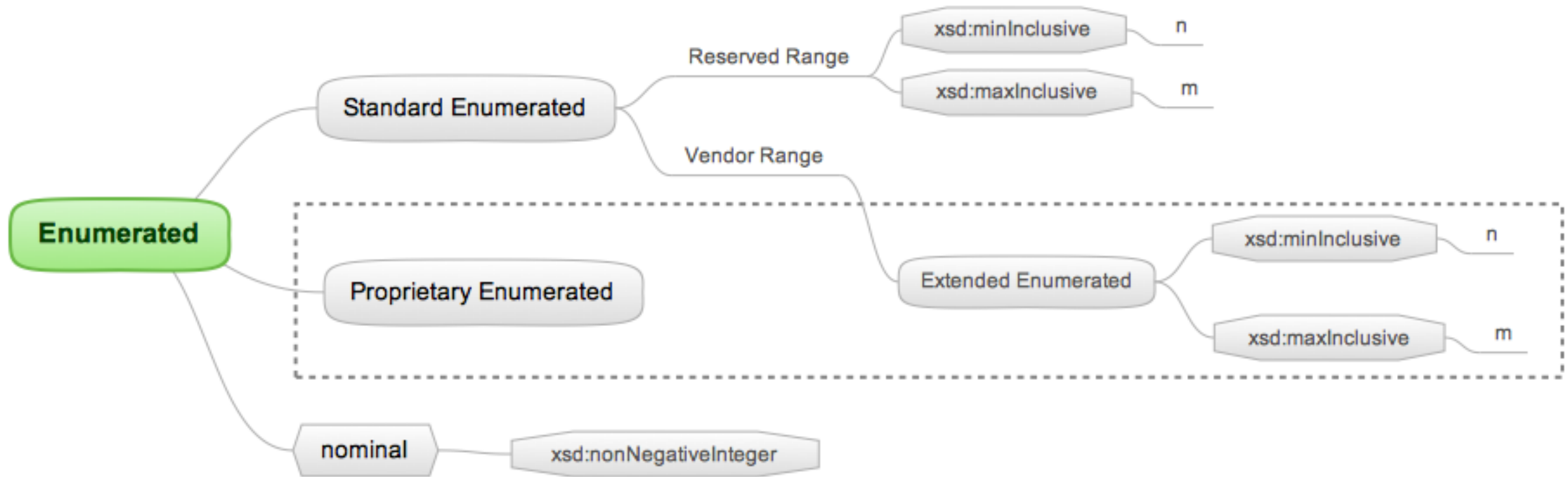


# Null is Interesting



- 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.
- Created a Null class and a null owl:NamedIndividual

# 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
- owl:NamedIndividual's for enumerations

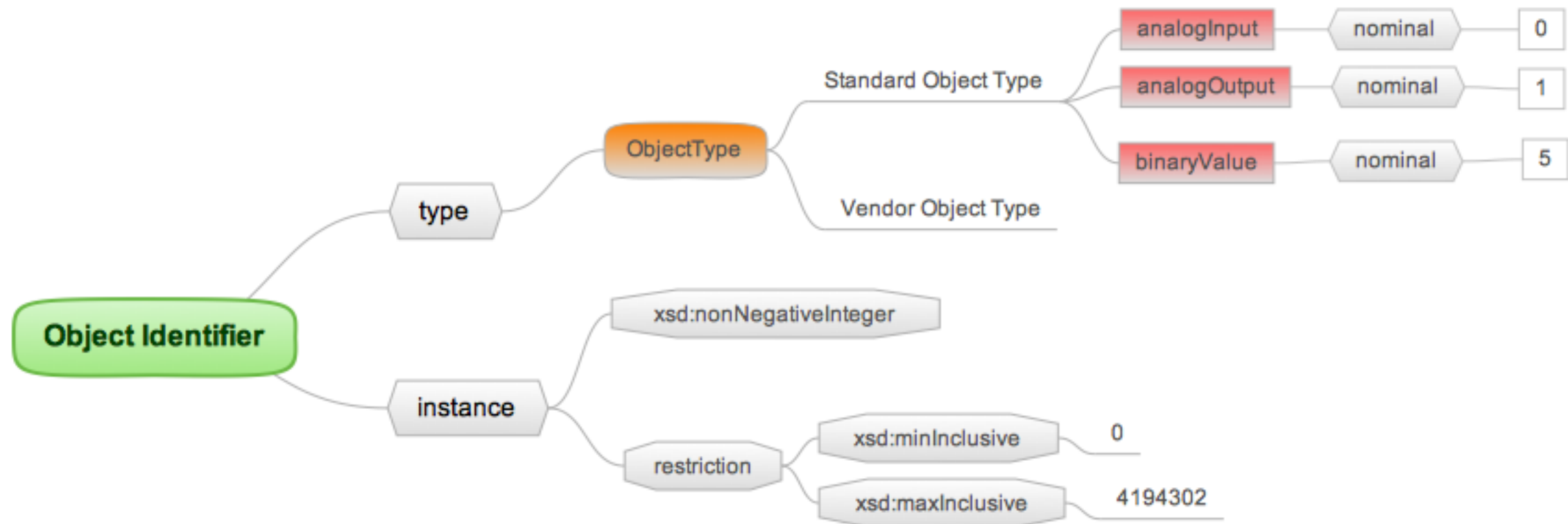
# Bit Strings are Interesting



- 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
- owl:NamedIndividual’s for individual bits



# Object Identifiers are Interesting

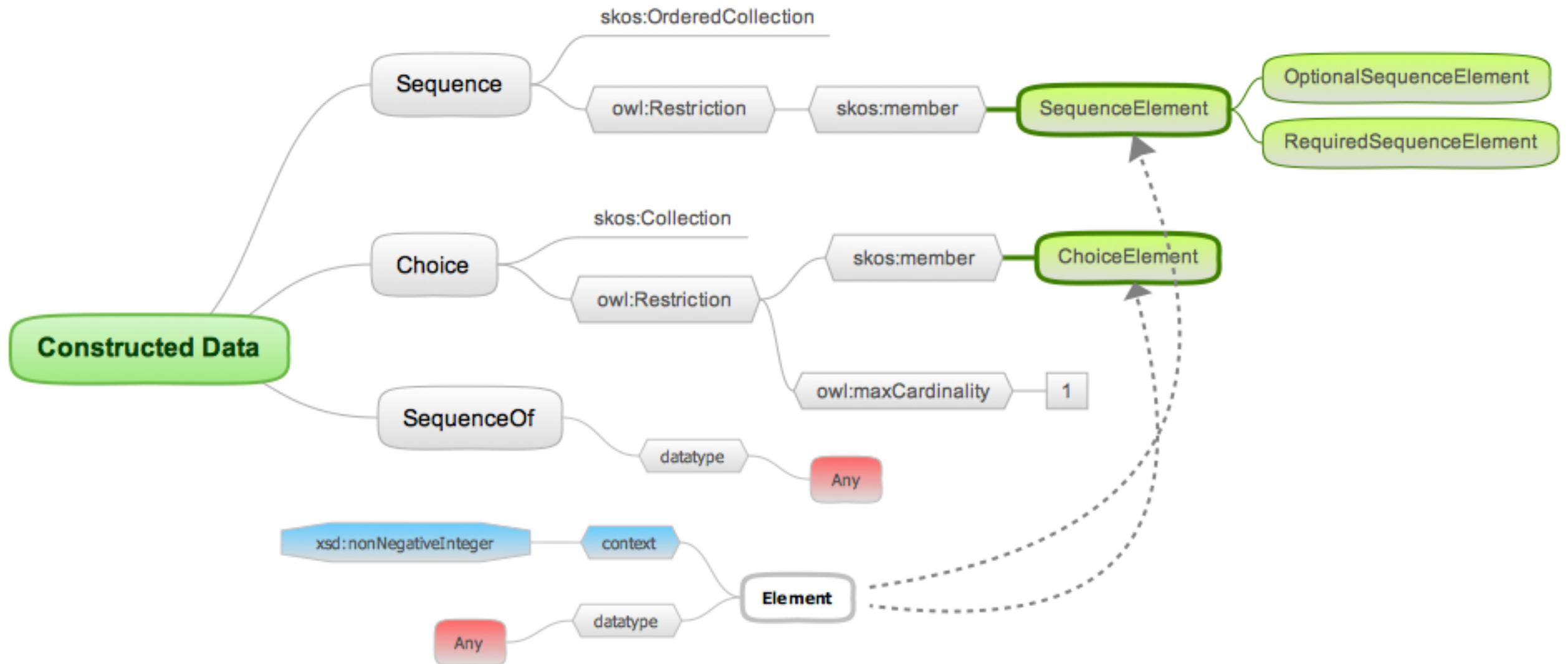


- Object identifiers are 32-bit packed structures with an object type and instance number
- The object type can be one of the standard types or a vendor type by using a value that is in the range reserved for extending the object type enumeration

# Hackathon Session

- Extracted pieces of Turtle and put them, along with design commentary, on project web site
- Reviewed the design with the team (Peter, Steve, Mike, Jacob)
- Identified many fundamental problems
  - mixing objects and datatypes
  - domains/ranges for restrictions
  - use SKOS for ordered collections and sets
- It only gets worse

# 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

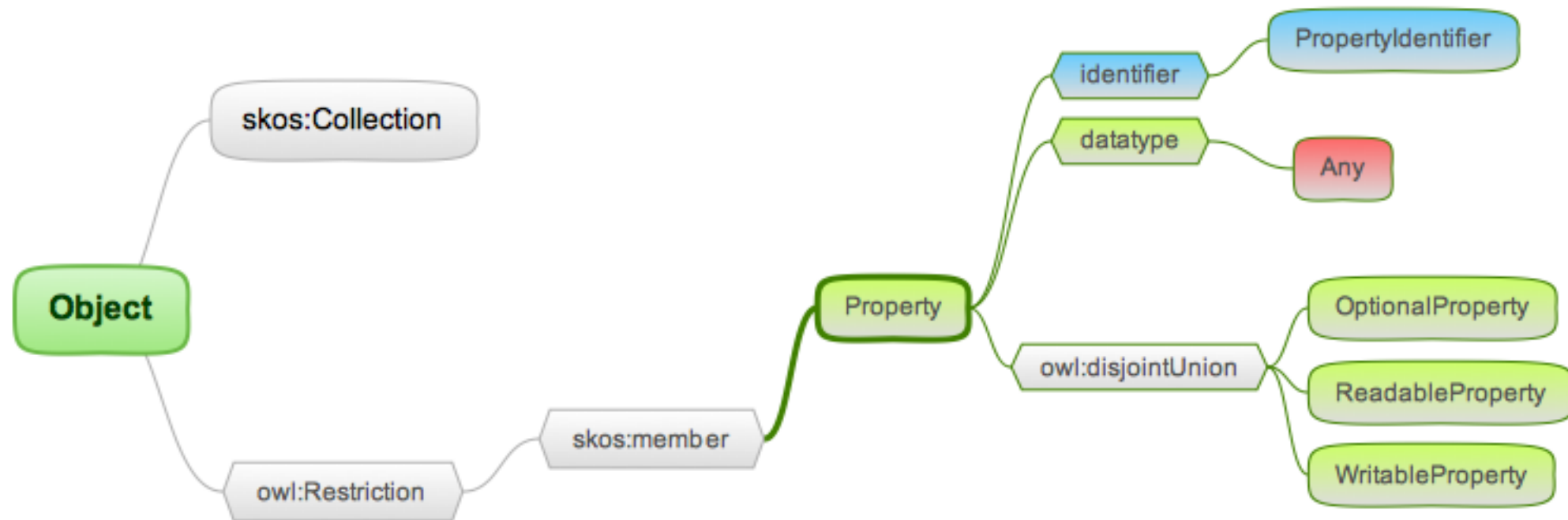
# Sequence's and Choice's

- 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

- ```
{ "$type": "AtomicReadFileRequest",  
  fileIdentifier: { type: "file", instance: 1 },  
  accessMethod: {  
    streamAccess: {  
      fileStartPosition: 50,  
      requestedOctetCount: 100  
    }  
  }  
}
```
- Can structured type 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 API needs to improve, something needs to be the anchor for BIM, building automation sensor networks, smart grid