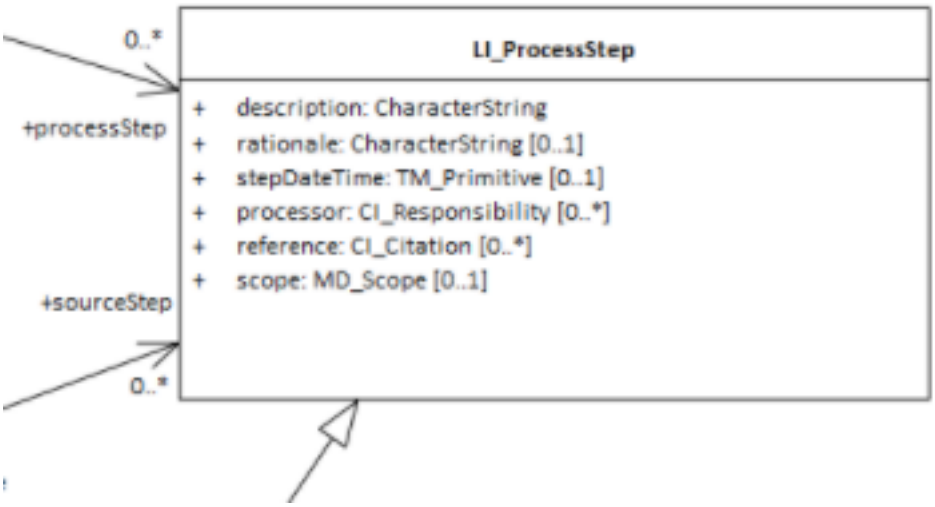
Metadata Concept Needs

Ted Habermann

Lineage rationale

The lineage metadata includes a list of processStep dictionaries. ISO 19115-1 processSteps include a rationale element that makes it possible to explain why a processStep was executed.



Including this in mdJSON requires adding an element to the processStep dictionary:

```
"processStep": [
        {
                "timePeriod": {
                        "startDateTime": "2020-10-28T06:00:00.718Z",
                        "endDateTime": "2020-10-28T06:00:00.536Z"
         },
                "description": "This is the first process step in the metadata lineage",
                "rationale": "This is why we did this."
                "stepId": "0",
                "processor": []
        },
```

The definition of this element is: a set of reasons or a logical basis for a course of action

Multiple party identifiers

Having identifiers for parties, i.e. people and organizations, is great. In fact, it is so great that we need to be able to have multiple identifiers for people and organizations. This takes care of the case when we need local id's for people but also need to include global identifiers in the metadata, e.g. ORCIDs or RORs.

The contact class already has contactId and this is a special identifier used by the system to connect contacts to metadata records. That special identifier could be kept as is and a list of identifier dictionaries could be added to the contact.

DataCite has a structure for nameIdentifiers that would look like:

```
"contact": [
  {
    "contactId": "individualID",
    "nameIdentifiers": [
      {
        "schemeUri": "https://orcid.org",
        "nameIdentifier": "https://orcid.org/0000-0003-3368-9046",
        "nameIdentifierScheme": "ORCID"
      }
    ],
    "isOrganization": true,
    "name": "individual",
    "memberOfOrganization": ["individualMemberOfID"],
  }
]
```

The existing mdJSON identifier class used for metadataIdentifier could also be used. This is probably preferable:

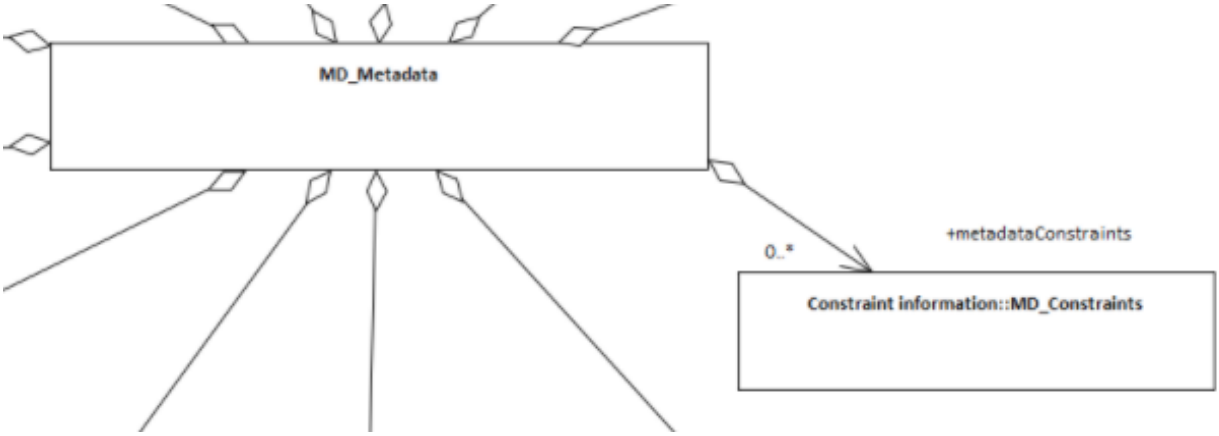
```
"contact": [
  {
    "contactId": "individualID",
    "Identifiers": [
      {
            "identifier": "Identifier",
            "namespace": "IdentifierNamespace",
            "description": "IdentifierDescription",
            "version": "IdentifierVersion"
            }
    ],
    "isOrganization": true,
```


ted@tedhabermann.com
https://orcid.org/0000-0003-3585

```
    "name": "individual",
    "memberOfOrganization": ["individualMemberOfID"],
  }
]
```

Metadata constraints

MdJSON includes a great constraint object in resourceInfo. In some cases, it is necessary to specify constraints on metadata that are different than constraints on the resource, e.g. open metadata for embargoed data. ISO 19115-1 includes a metadataConstraints object to accommodate this:

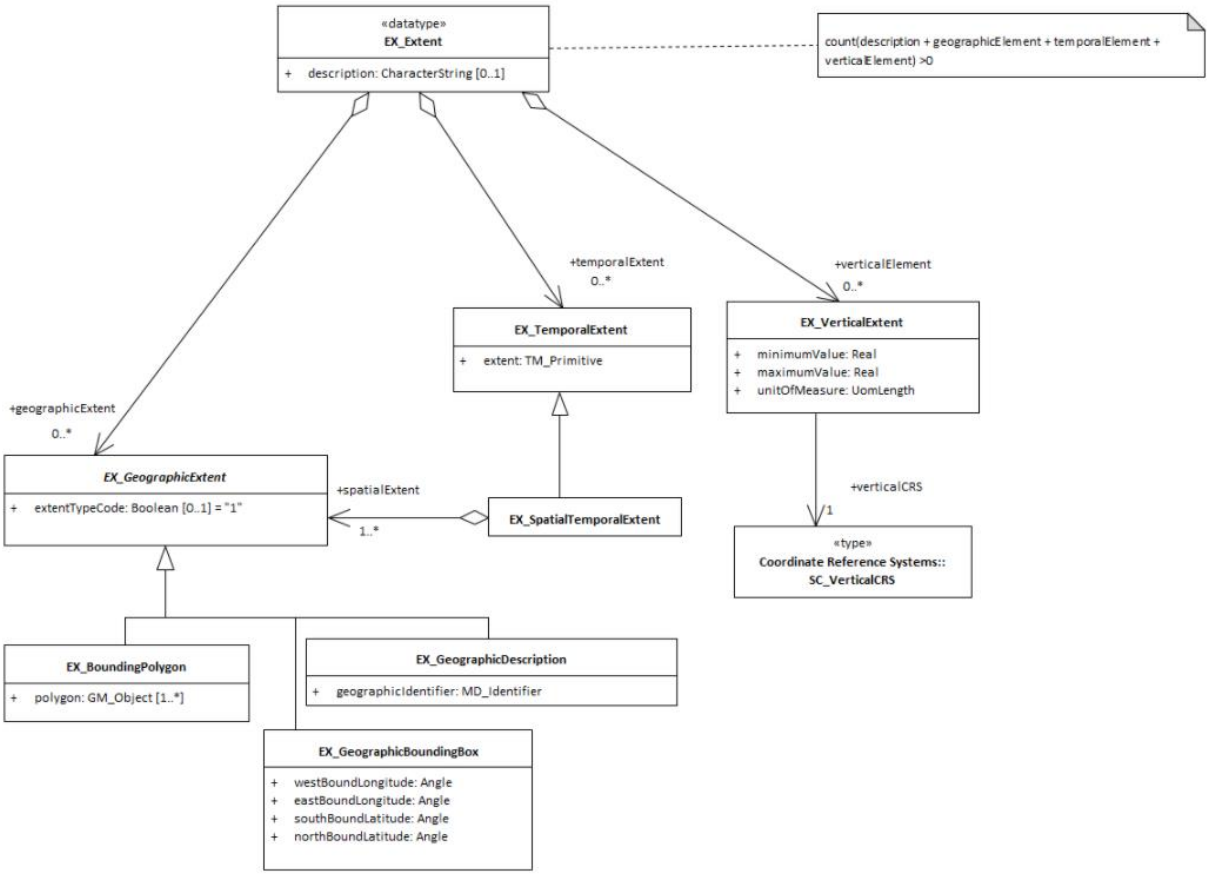


Adding the existing mdJSON constraint object to the metadataInfo dictionary would solve this problem.

Vertical extent

There are many situations, e.g. cores, ocean, ecological, and ocean observations where specifying a vertical extent is required for meaningful understanding. ISO 19115-1 handles this with a vertical element included in the EX_Extent. That class includes min and max values as well as units and a coordinate reference system.

ted@tedhabermann.com
https://orcid.org/0000-0003-3585



There are a few ways to handle this and I'm sure you have thought about it. I would tend to include this addition in the development of a single extent object that includes all three (spatial, temporal vertical). This is one of the best things done at in ISO 19115. The single and separate spatial/temporal objects is very limiting. It turns out that many metadata objects have extents: provenance, maintenance, quality information, responsibilities, sources. It would be a big plus if mdTools had this capability.

ted@tedhabermann.com
https://orcid.org/0000-0003-3585