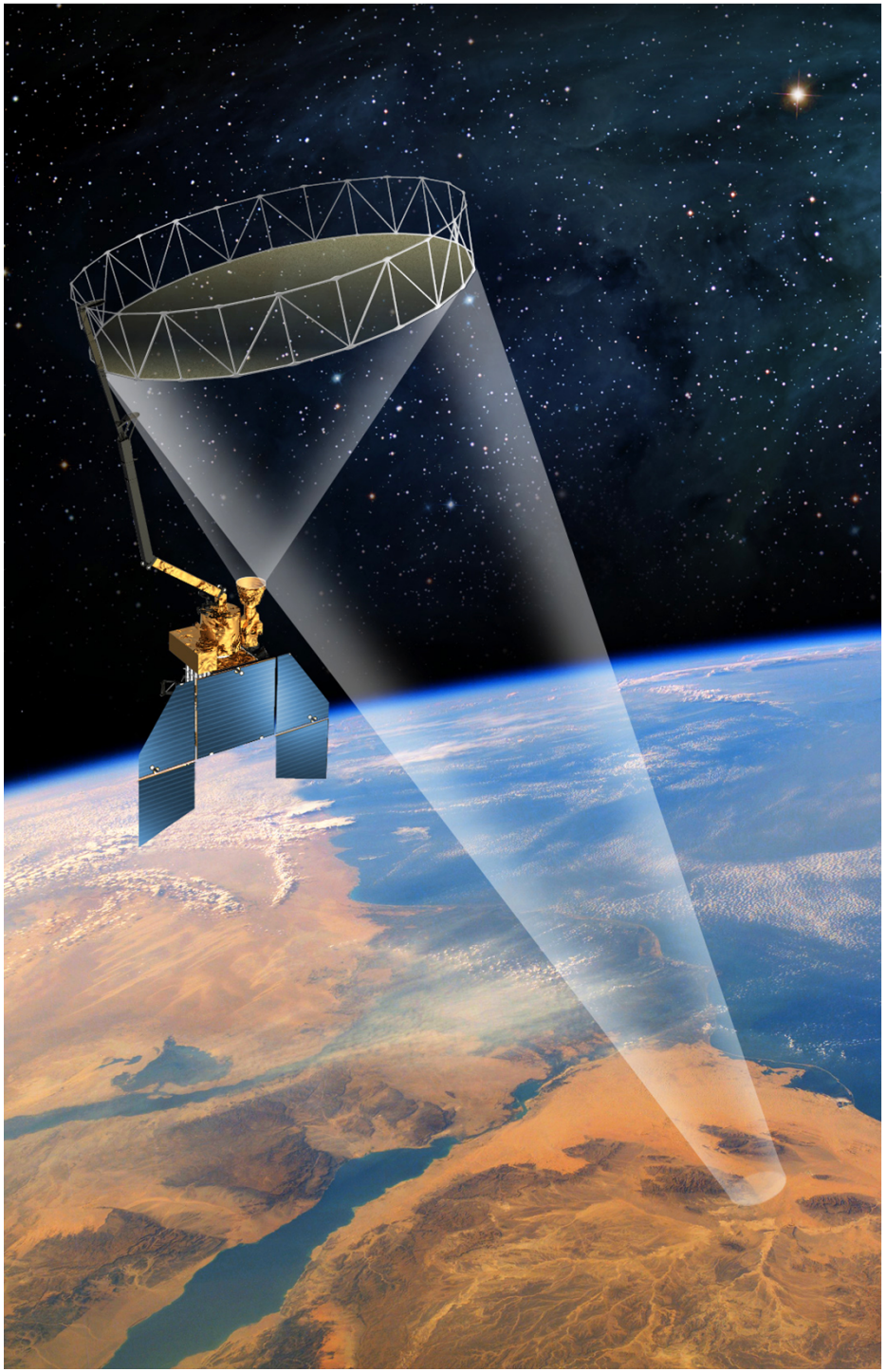


Tools for Incorporating ISO 19115 Metadata in Earth Science Data Systems

Barry Weiss
Hook Hua

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA

July 12, 2011

The SMAP mission has not been formally approved by NASA. The decision to proceed with the mission will not occur until the completion of the National Environmental Policy Act (NEPA) Process. Material in this document related to SMAP is for information purposes only.

Objective

- Ease the implementation and utilization of the ISO 19115 metadata standard into Earth Science Data Products
  - Initial goal - Enable the proposed Soil Moisture Active Passive (SMAP) mission to incorporate ISO metadata into data products.
  - Follow-up goal – Provide infrastructure and documented experience that enables other NASA sponsored teams to implement and incorporate ISO 19115 metadata into their data products and data systems.

Major Products

- This effort will generate:
  - A reusable common reader/writer API for accessing ISO 19115 metadata in ISO 19139 compliant XML file format.
  - A generic tool that provides direct and easy access to specific metad ata elements of interest that conform to the ISO 19115 standard and its ISO 19139 XML encodings.
  - A generic HDF5 tool to support the conversion from ISO 19139 comp liant XML file format to and from standard HDF5 metadata represent ation.
  - Technical and user oriented documentation of the ISO 19115 APIs a nd tools.

Implementation Plan

- Implement reader/writer APIs that provide access to the ISO metadata in both XML and HDF5 format.
- Generate an ISO compliant schema for proposed SMAP data products.
- Integrate the tools into the proposed SMAP Science Data System.
- Review the outcome
  - Review tools with potential users within and outside the proposed SMAP project for efficacy and ease of use.
  - Review the ISO schema with the Metadata Evolution for NASA Data Systems (MENDS)/ESDIS team to enable development of a “NASA convention” of ISO 19115.
- Iterate on the above process
- Generate the tool that enables exploration of ISO compliant metadata in XML format during the second implementation iteration.

Current Progress

- Began working in late May
- Reworked proposed SMAP metadata dataset schema
  - Restricted work to elements in ISO 19115 and 19115-2
  - Restricted work to dataset schema, series schema will follow
  - Removed vestiges of North American Profile
- Used UML tool to generate an initial SMAP XSD schema
- Identified reusable XSD Schema constructs that are commensurate with XML binding approach
- Assessed existing XML Binding tools for handling **expected variations** in XSD Schema model of metadata in ISO 19139

Major Challenges

- MENDS recommendations stipulate the need for a NASA usage convention
  - Anticipated metadata entries always appear in specific ISO elements
  - Flexible guidelines that enable each mission, data center and clearing house to identify and specify locally required values

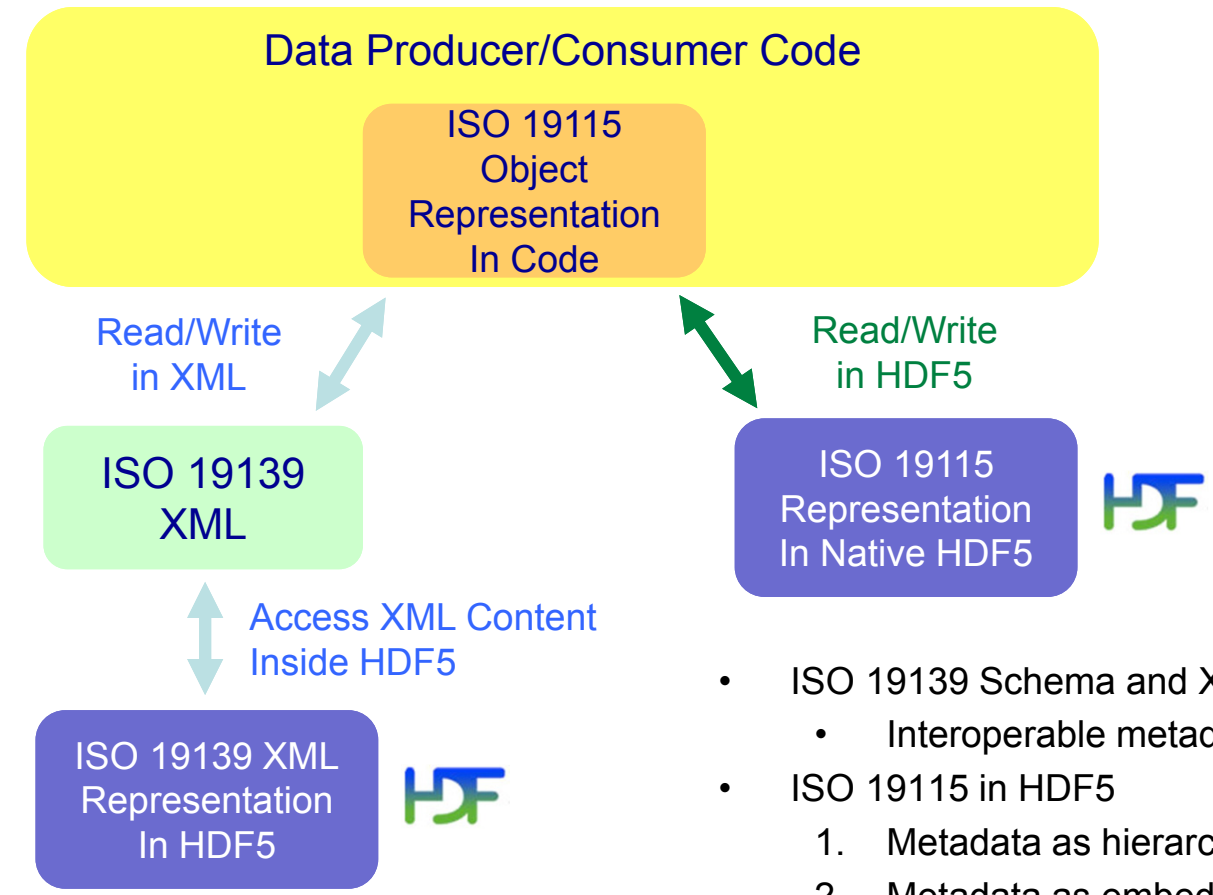
Effective effort requires a convention for ISO usage

- Within any given mission or entity, the specifications of the ISO schema will vary
  - Different data products from the same mission
  - Different missions serviced by the same organization

The API/tool must accommodate design variations with relative ease

Proposed Approach

Metadata Handling



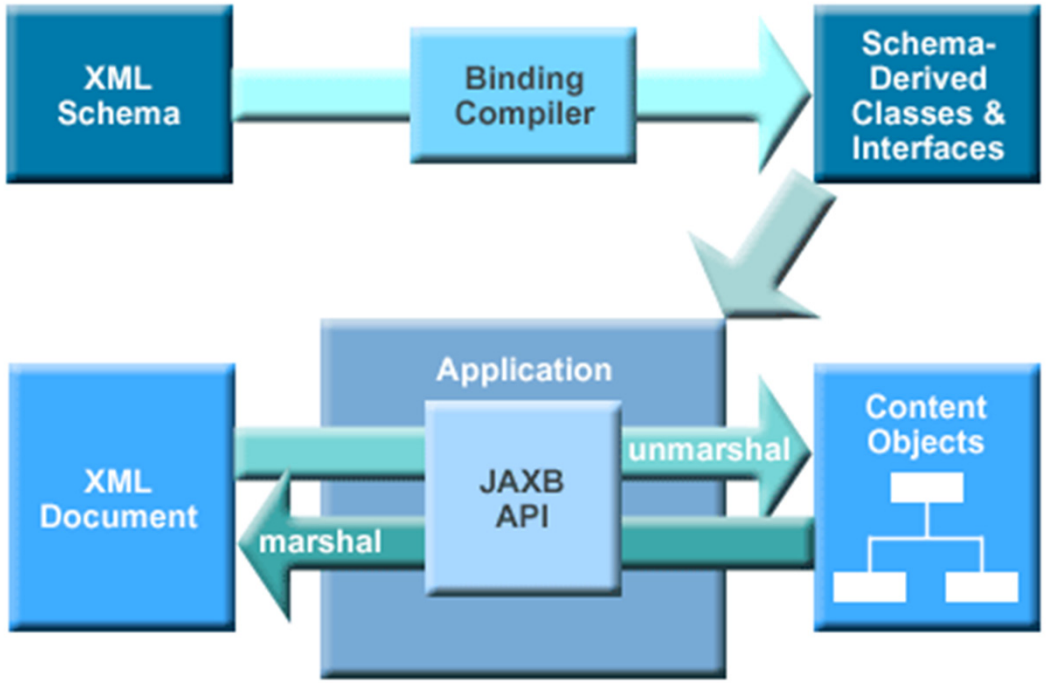
- ISO 19139 Schema and XML
  - Interoperable metadata interchange format
- ISO 19115 in HDF5
  1. Metadata as hierarchical group/attributes
  2. Metadata as embedded ISO 19139 XML stream
     - Compatible with legacy formats such as HDF4 and NetCDF3
- Conversion cross-walks are possible

XML Schema Definition (XSD)

- ISO 19139 codifies XML representation of ISO 19115
- XSD provides:
  - A full description of the XML structure
  - Specification of permissible values in an XML document
- Enables validation
  - Can be used to validate *the structure and the value* of an XML metadata instance
- Can leverage existing UML and XSD documents of ISO 19115

XML Data Binding

- Represents information in an XML document as an object in memory
  - No more XML parsing as in DOM or SAX
- Leverages the model in an XSD Schema to create classes and interfaces that adhere to the information structure defined by the schema.
- Enables serialization/deserialization of XML instances to/from code.
- Supports expected variations in the ISO model



Means of Representing ISO in XSD

- Identify XSD Schema constructs commensurate with XML binding approach
- Use built-in types
  - *xs:string*
  - *xs:decimal*
  - *xs:integer*
  - *xs:boolean*
  - *xs:date*
  - *xs:time*
  - xs:dateTime
  - xs:duration
- Define *Complex Elements* at the top-level
  - Follow the *"Venetian Blind"* approach to XSD layout

Representation of ISO in an XSD Schema

- Two potential approaches for representation of ISO code lists in XSD
  - *Restriction*/*Facets* provide improved validation
    - Restrictions are used to define acceptable values for XML elements or attributes. Restrictions on XML elements are called facets.
    - Restriction of values chosen from an enumeration
  - *Indicators*
    - Order indicators
      - *choice*: specifies that only one child element may appear
      - *sequence*: specifies that child elements must appear in a specific order
    - Occurrence indicators
      - maxOccurs
      - minOccurs

Assessment of XML Binding tools

- C++
  - **CodeSynthesis XSD**
    - An open-source, cross-platform W3C XML Schema to C++ data binding compiler.
  - **CodeSynthesis XSD/e: XML for Light-Weight C++ Applications**
    - An open-source, dependency-free XML Schema to C++ compiler for mobile, embedded, and light-weight applications
  - **Mel**
    - An open source C language XML Binding tool with *limited* XSD Schema support
- Java
  - **Java Architecture for XML Binding (JAXB)**
    - http://www.oracle.com/technetwork/articles/javase/index-140168.html

Conclusions

- Effort will meet the challenge for SMAP as the first of the Decadal Survey Missions
  - SMAP will be first NASA mission to implement ISO 19115.
  - Provides the necessary tools to implement ISO.
  - Decisions about standard use are still forthcoming. Need to converge as closely as possible with developing standard.
- Effort will encourage collaboration across NASA ESDIS
  - Select optimal technical approach to enable modifications to the metadata schema and reuse by other NASA organizations
  - Regularly review approach and implementation with outside organizations to ensure applicability

Backup

CodeSynthesis XSD

- An open-source, cross-platform W3C XML Schema to C++ data binding compiler.
- Free for non-commercial use
- Supports two XML Schema to C++ mappings:
  - in-memory C++/Tree
  - stream-oriented C++/Parser.
- http://codesynthesis.com/products/xsd/

```
<contact>
  <name>John Doe</name>
  <email>j@doe.com</email>
  <phone>555 12345</phone>
</contact>
```

```
auto_ptr<Contact> c = contact ("c.xml");
cout << c->name () << ", "
     << c->email () << ", "
     << c->phone () << endl;
```

CodeSynthesis XSD/e

- An open-source, dependency-free XML Schema to C++ compiler for mobile, embedded, and light-weight applications
- Small footprint
- XML parsing and serialization
- XML Schema validation
- XML data binding
- Mappings
  - C++/Hybrid in-memory mapping creates a light-weight, tree-like object model of the XML data
  - C++/Parser (for XML parsing)
  - C++/Serializer (for XML serialization)
- http://codesynthesis.com/products/xsde/

```
<object>
    <name>Red Hill</name>
    <position>
      <lat>-33.69</lat>
      <lon>18.83</lon>
    </position>
</object>
```

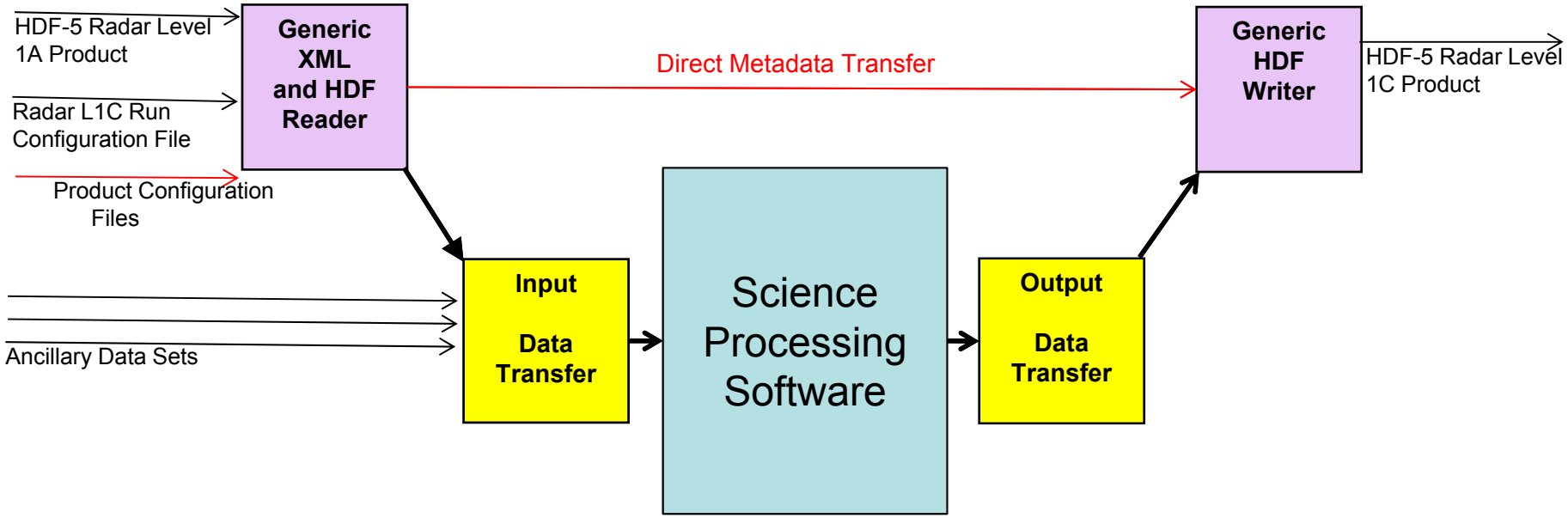
```
object* obj = ...; // Parse XML.
const char* name = obj->name ();
position& pos = obj->position ();
float lat = pos.lat ();
float lon = pos.lon ();

delete obj;
```

mel

- An open source C language XML Binding tool with *limited* XSD Schema support
- Produces
  - .h file containing C structures
  - .c file that provides the functions mentioned in the .h file
  - mellib.a to linked against code.
- http://xmel.sourceforge.net/

Mapping Requirement



- Many metadata values transfer directly from the input product to the output product
- XPaths provide a simple means to codify automated mapping
  - XPaths can be used to drive automated mapping calls.

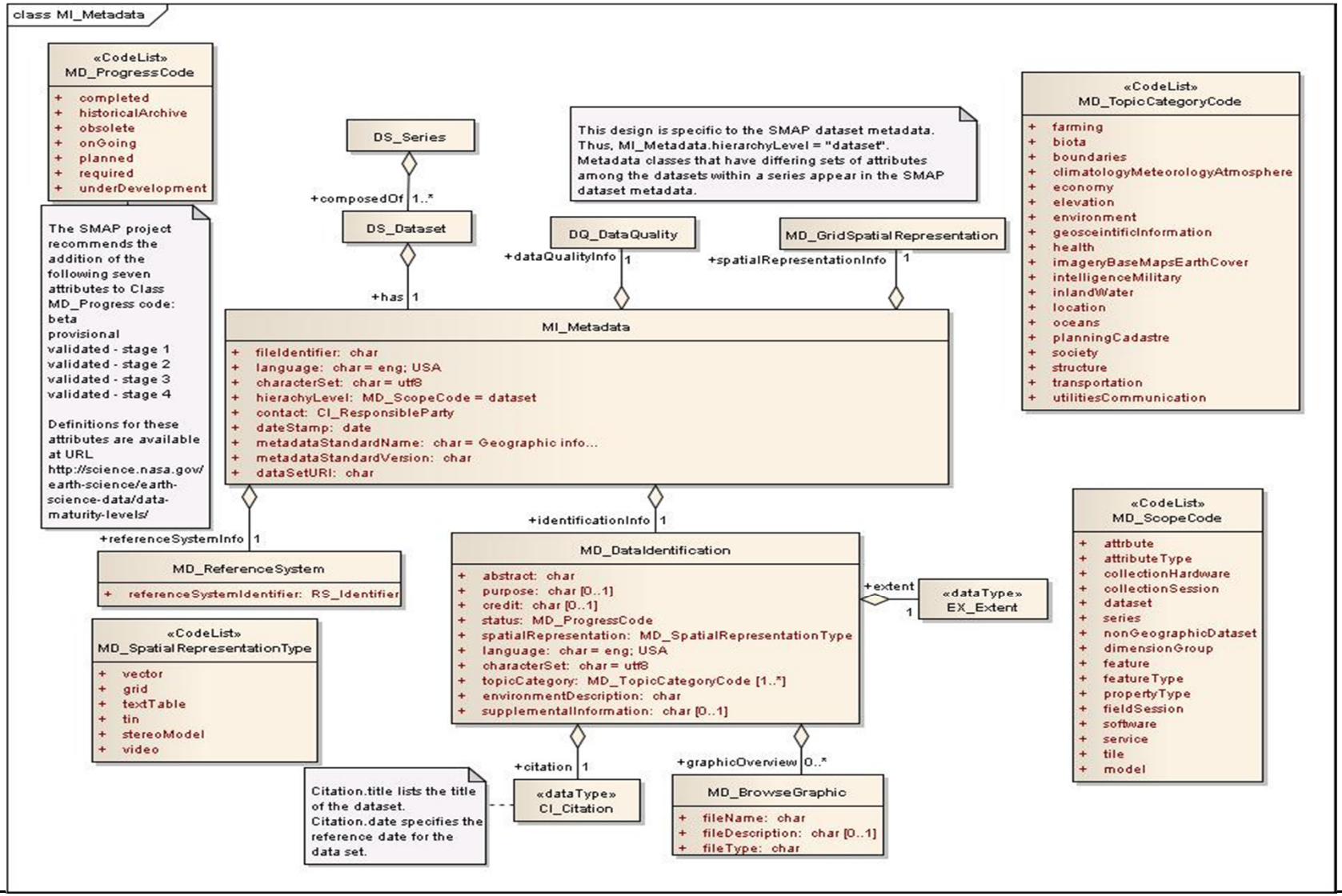
XPath

- An alternative approach to automatic data binding
- Benefits
  - Data binding code only needs tree structure knowledge of XML document
  - Does not require use of an XSD Schema
  - Binds specifically to needed elements
- Drawbacks
  - Without an XSD Schema, application can not validate XML element structure and types
  - Does not create an automated data model

Existing Authoring Tools

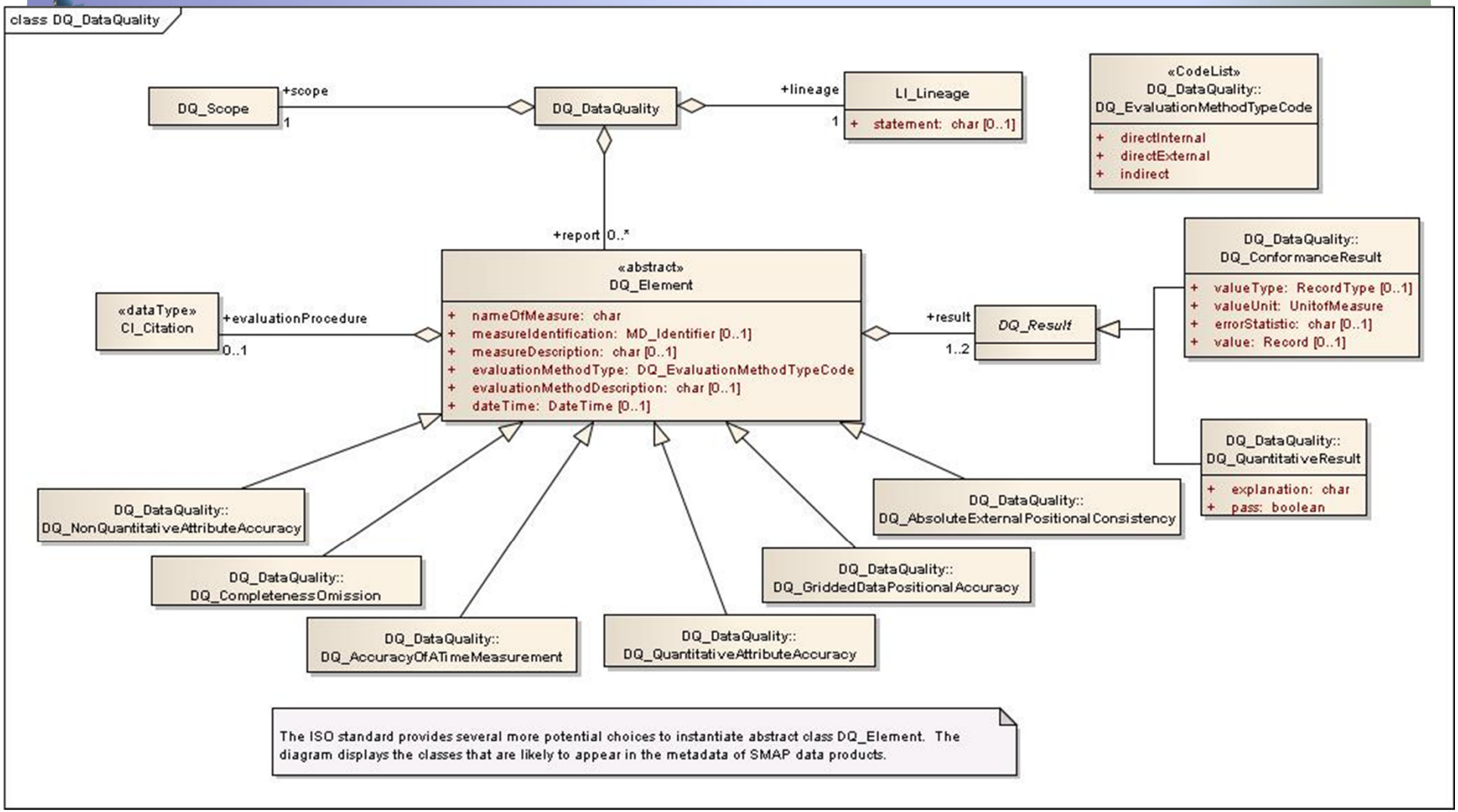
- GCMD DIF authoring tool
  - http://gcmd.nasa.gov/User/difguide/difman.html
- FGDC has authoring tools tkme and xtme
  - http://www.fgdc.gov/metadata/geospatial-metadata-tools

Updated ISO Schema UML Diagram – MI_Metadata

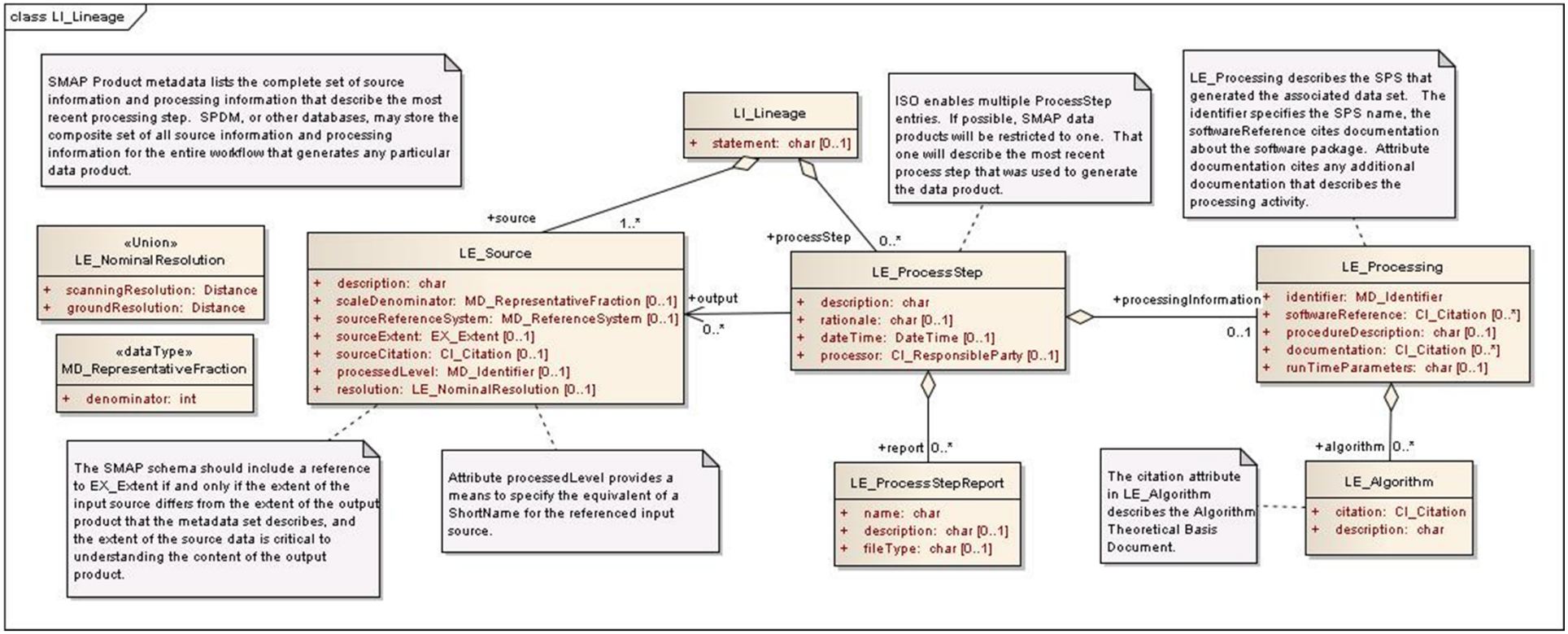


The SMAP mission has not been formally approved by NASA. The decision to proceed with the mission will not occur until the completion of the National Environmental Policy Act (NEPA) Process. Material in this document related to SMAP is for information purposes only.

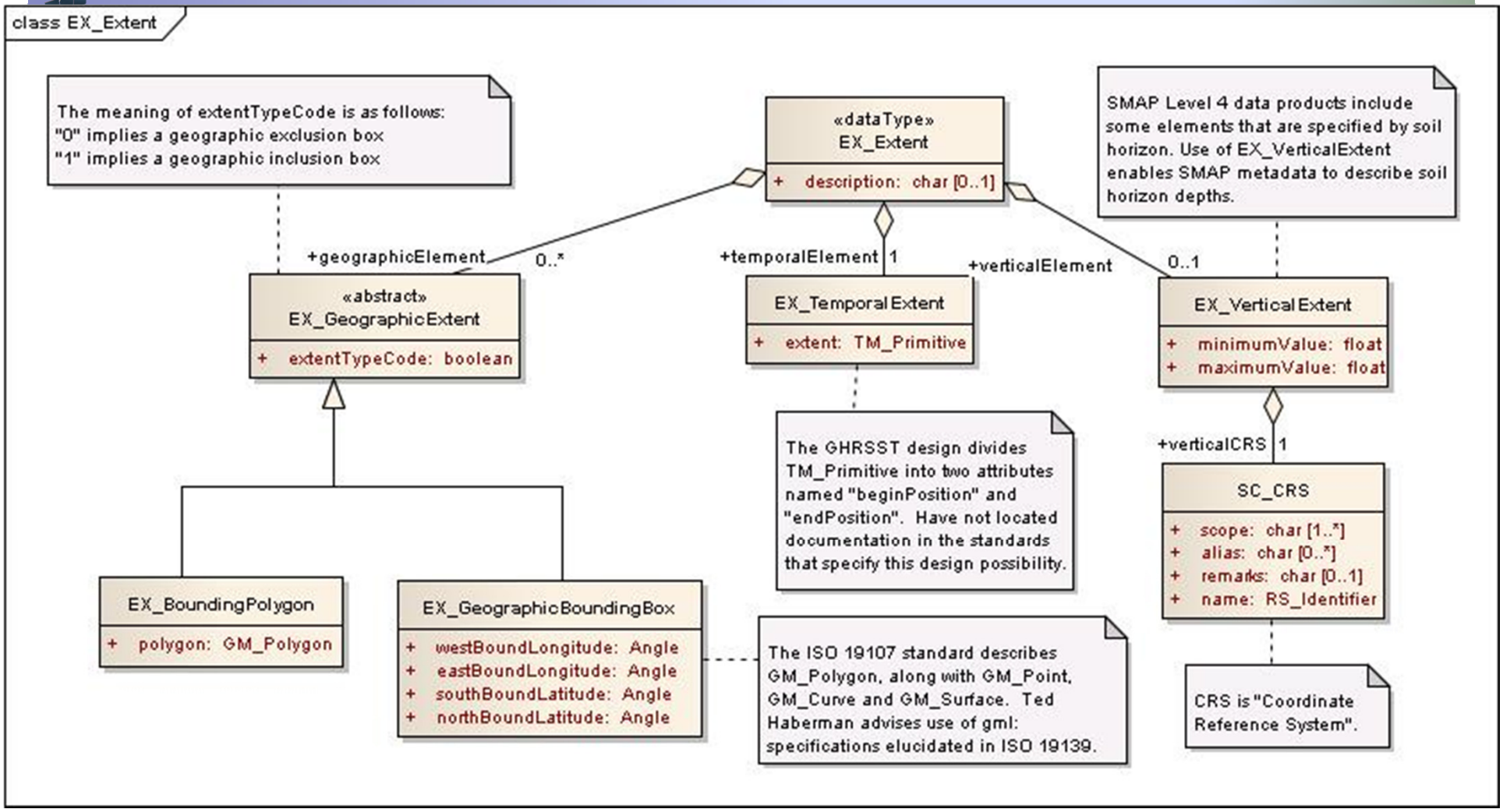
Updated ISO Schema UML Diagram – DQ_Quality



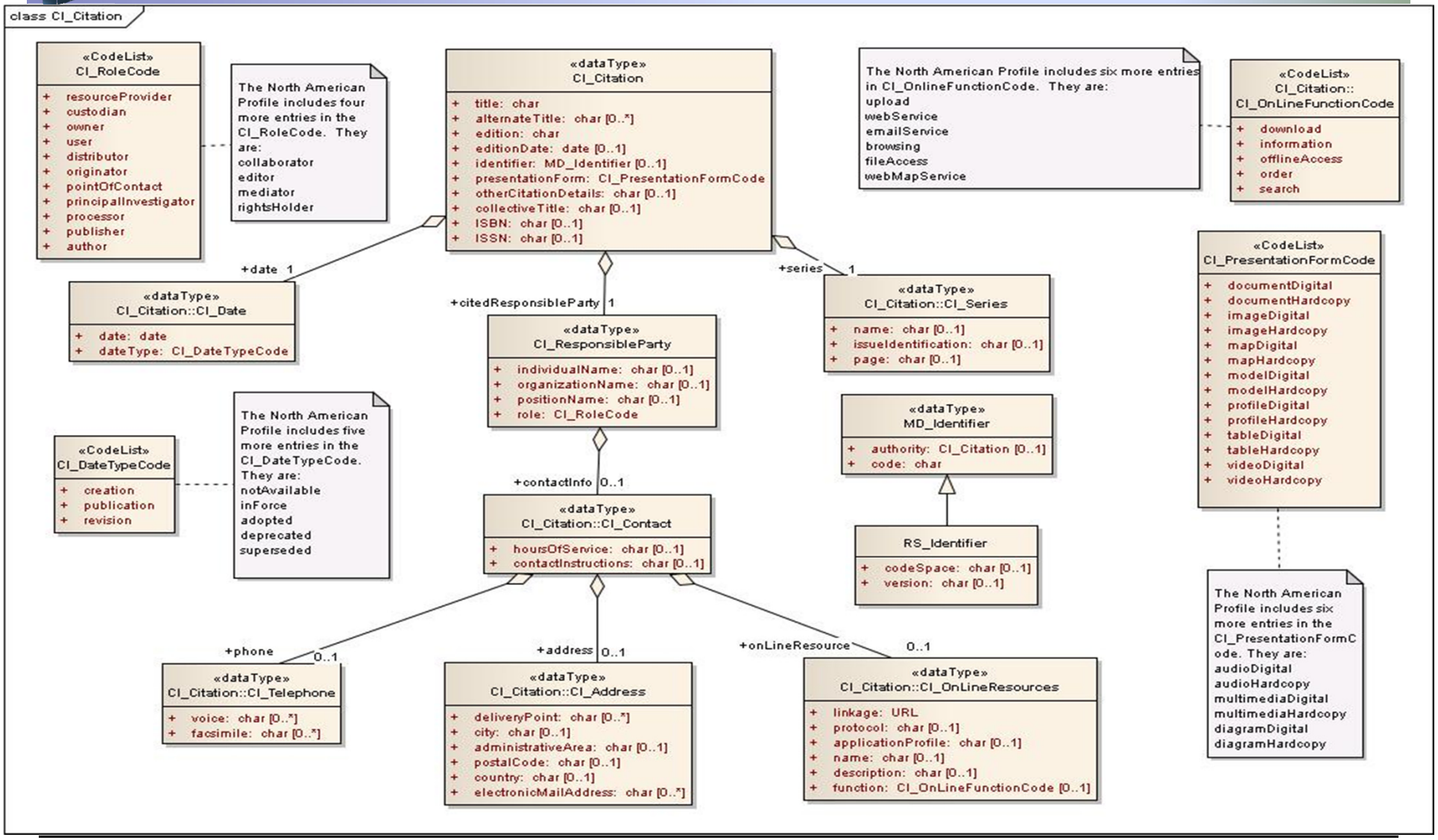
Updated ISO Schema UML Diagrams –LI_Lineage



Updated ISO Schema UML Diagram – EX_Extent



Updated ISO Schema UML Diagram – CI_Citation



Automatically Generated XSD

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:include/>
    <xs:complexType name="MI_Metadata">
        <xs:sequence>
            <xs:element name="fileIdentifier" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="language" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="characterSet" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="hierachyLevel" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="contact" type="CI_ResponsibleParty" minOccurs="1" maxOccurs="1"/>
            <xs:element name="dateStamp" type="date" minOccurs="1" maxOccurs="1"/>
            <xs:element name="metadataStandardName" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="metadataStandardVersion" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="dataSetURI" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="identificationInfo" type="MD_DataIdentification" minOccurs="1" maxOccurs="1"/>
            <xs:element name="dataQualityInfo" type="DQ_DataQuality" minOccurs="1" maxOccurs="1"/>
            <xs:element name="spatialRepresentationInfo" type="MD_GridSpatialRepresentation" minOccurs="1"
maxOccurs="1"/>
            <xs:element name="contact" type="CI_ResponsibleParty" minOccurs="1" maxOccurs="1"/>
            <xs:element name="referenceSystemInfo" type="MD_ReferenceSystem" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="MD_ScopeCode">
        <xs:sequence>
            <xs:element name="attrbute" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="attributeType" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="collectionHardware" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="collectionSession" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="dataset" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="series" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="nonGeographicDataset" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="dimensionGroup" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="feature" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="featureType" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="propertyType" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="fieldSession" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="software" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="service" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="tile" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="model" type="xs:string" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
```