XML Encoding of the North American Data Model

By NADM Data Interchange Technical Team
Eric Boisvert[1], Bruce R. Johnson[2],
Peter N. Schweitzer[2], Martin Anctil[1]

[1]Geological Survey of Canada-Québec Division
Natural Resources Canada
880 Chemin Ste-Foy
Québec, Qc, G1S 2L2
Canada
Telephone : (418) 654-3705
Fax : (418) 654-2615
e-mail: {eboisver, manctil}@nrcan.gc.ca

[2]U.S. Geological Survey
National Center, MS 908
12201 Sunrise Valley Drive
Reston, VA 20192
Telephone: (703) 648-6051
Fax: (703) 648-6383
e-mail: {bjohnson, pschweitzer}@usgs.gov

INTRODUCTION

The North America Data Model (NADM) is composed of a series of initiatives to create a common set of tools and technologies to manipulate geological information in the digital realm. One of these initiatives is to define a standard interchange format to allow easy exchange of information between systems and tools.

Extensible Markup Language (XML) is a formalism to encode domain-specific information (such as chemistry, biology, recipes, geology, etc) into a structured document. XML encoding must follow certain rules to be both 'well formed' and 'valid'. Although creating a 'well formed' XML document is relatively simple, a 'valid' document must comply to the domain specific rules (for example., those for geology). The first draft of a conceptual data model for geology and geologic maps has been worked out by the North American Data Model Steering Committee's Data Model Design Team (NADM-DMDT)(2003). This conceptual model expresses the rules to which the science of geology adheres. This model will be translated into a set of XML document construction rules called an XML schema, which will be used to assess the validity of XML-encoded documents containing NADM-compliant datasets. The Data Interchange Technical Team (DITT) is working to develop this XML schema.

ACKNOWLEDGMENTS

Special thanks to Serge J. Paradis (Geological Survey of Canada) and David R. Soller (United States Geological Survey) for suggestions that greatly improved the manuscript.

DATA INTERCHANGE TECHNICAL TEAM (DITT)

The Data Interchange Technical Team (DITT) is one of the technical teams composing NADM. Their mandate is, among other things, to (see <http://geology.usgs.gov/dm/steering/teams/interchange/interchange.txt> for charter):

- Develop standardized formats and mechanisms for exchanging digital geologic map databases
- Facilitate exchange of digital geologic map content between various implementations of the NADM

Various technical solutions are available to achieve this task, but considering the current trend in information technology, every path seems to involve XML. In the fall of 2002, at the Geological Society of America (GSA) meeting in Denver, the DITT started working on

the encoding of the conceptual model developed by the NADM DMT team. For this purpose, XML has quickly been elected as the technology of choice. This option was discussed early in the modelling process and the current technological trend only reinforced this option.

WHAT IS XML?

XML is a plain text file structured using 'tags', or '*mark-ups*', that organise the information contained in a document according to a set of predefined rules. Tags can be created to accommodate a domain and the rules to organise them can also be defined to fit the domain requirements.

XML technology has many advantages; it offers a general approach for structuring information in a document and is sophisticated and well adapted to the web environment. Lots of tools are available to manipulate XML and a growing community of XML enthusiasts, rooted in the *Open Source* movement, provides sufficient support (a search for XML on *Google* <http://www.google.com> returns 20 million pages !). XML is truly a multi-platform, multi-vendor, multi-programming language and even has its own transformation language (XSL / XSLT). Many emerging technologies are based on XML (for instance, the whole Microsoft Office XP suite) and it is at the root of the "Web Services" revolution.

Markup Language

A marked-up document is a file where important pieces of information are flagged to allow a human reader or a machine to quickly locate it, or to explicitly document it. For example, this small paragraph from Drewes (1998):

Dacitic Vent Breccia (Miocene)—*Light-medium-gray, finely porphyritic dacitic rock containing inclusions of Jurassic or Proterozoic granite and Jurassic rhyolite (welded tuff?) as much as 20 m in diameter. The subcircular outcrop mass of breccia probably is a volcanic vent or throat. A halo of strongly saussuritized rock 0.3–0.5 km wide*

makes sense for anyone who has a bit of geological background. One can easily locate the geological ages hidden in the text. A machine (or someone who has no formal knowledge in geology) cannot extract this information from the text because this piece of information is not explicitly identified. A software would need a complete thesaurus of age names. A simpler approach is to flag this information directly in the document:

```
Dacitic Vent Breccia (<age>Miocene</
age>)—Light-medium-gray, finely
porphyritic dacitic rock containing
inclusions of <age>Jurassic</age>
or <age>Proterozoic</age> granite
and <age>Jurassic</age> rhyolite
(welded tuff?) as much as 20 m in
diameter. The subcircular outcrop
mass of breccia probably is a volcanic
vent or throat. A halo of strongly
saussuritized rock 0.3–0.5 km wide
```

<Age> and </Age> are respectively opening and closing tags that flag subsets of the document and assign the enclosed words a specific interpretation; the tags identifies Miocene as an age. Better yet, attributes can be added to the tags to enhance the tag usability. For example, lowerBound and upperBound attributes contain the ages in millions of years before present, delimiting the stratigraphic age.

```
Dacitic Vent Breccia (<Age lowerBound="-
23.8" upperBound="-5.3">Miocene</Age>)—
Light-medium-gray, finely porphyritic
dacitic rock containing inclusions of
<Age lowerBound="-206" upperBound="-
144">Jurassic</Age> or <Age lowerBound="-
2500" upperBound="-543">Proterozoic</
Age> granite and <Age lowerBound="-206"
upperBound="-144">Jurassic</Age> rhyolite
(welded tuff?) as much as 20 m in diameter.
The subcircular outcrop mass of breccia
probably is a volcanic vent or throat. A
halo of strongly saussuritized rock 0.3–0.5
km wide
```

A second level of information structure is given by the organization of the tags themselves. In the previous example, the tags are scattered loosely in the text, but XML documents can be organised in a more strict arrangement of tags. For example,

```
<Unit name="Dacitic Vent Breccia">
<Age lowerBound = "-23.8" upperBound="-
    5.3">Miocene</age>
<Rock Name = "Dacite">
<Color>Light Medium Gray</Color>
<Texture>finely porphyritic</Texture>
<Constituent Role = "Inclusion">
    <Rock Name = "Granite">
    <Age lowerBound="-206" upperBound="-
        144">Jurassic</Age>
    </Rock>
    <Rock Name = "Granite">
    <Age lowerBound="-2500" upperBound="-
        543">Proterozoic</Age>
```

```
    </Rock>
    <Rock Name = "Rhyolite">
    <Age lowerBound="-206" upperBound="-
       144">Jurassic</Age>
    </Rock>
</Constituent>
<Genesis>Volcanic vent or throat</Genesis>
</Rock>
</Unit>
```

This is a very different document since there is no free text, only tags with attributes representing a described rock. One important aspect here is that `<Texture>` tags are embedded into the `<Rock>` tag. This implies that the textural description is tied to its `<Rock>` container; this is critical since the texture applies to the Dacite unit and not any other rock body.

This latter example is still readable by a human. The content of the XML document can be divided into small pieces of information easily handled by the computer. The class of softwares that transform XML documents into chunks of usable information is called **parser**, and many are available for free on the Internet. For anyone who writes softwares using XML, the parser will take care of all the details of loading, analysing and, more important, **validating** the XML document (we'll see what a valid document is in the next section). All this would need to be developed from scratch if any other format structure were chosen.

XML SCHEMAS

There are no predefined tags in XML. They, with their rules, must be defined by a group wishing to encode a specific domain. There are XML encodings to describe chemistry (CML), mathematic formula (MathML), geographic features (GML), poetry (XML Poetry) and even recipes (RecipeXML, formally known as DESSERT=Document Encoding and Structuring Specification for Electronic Recipes). Luckily, geoscience has not been left out; XMML (eXploration Mining Markup Language) is being developed in Australia to address mining industry requirements (see <http://www.ned.dem.csiro.au/XMML/>). Finally, our own effort is directed toward defining a mark-up language to describe geological maps.

As previously mentioned, a usable XML document must be both 'well formed' and 'valid'. A **well formed** document is a document that follows the basic rules that **all** XML-based markup documents must follow.

Rules include, but are not limited to:

1. When a tag is opened, it must be closed. For example, `<MYTAG>` must be followed by a `</MYTAG>`. (or if the tag does not carry any content, a valid shortcut is `<MYTAG/>`)
2. Tags must be nested, this means that tags must closed in the reversed order or opening. `<A><B></B></A>` is well formed while `<A><B></A></B>` is not.
3. Tags are case sensitive , so `<MyTag>` is not the same tag as `<MYTAG>`, therefore `<MYTAG> . . . </MyTag>` is not a well formed structure.

For an XML document to be **valid**, it must follow the domain specific rules. These rules can be defined using two mechanisms: 1) Document Type Definition (DTD) which is becoming used less often or restricted to some specialised tasks, and 2) XML Schema, which is itself an XML document that follows its own domain rules (the business of describing other domains). The goals of the schema creator are to distil from a domain a set of rules and encode them into an XML schema. The process is not straightforward and many schemas can produce similar XML documents describing the same domain but using a different approach. XML schema specification can be found from the W3C website <http://www.w3.org/XML/Schema>, but books on XML schemas, such as Duckett and others (2001), are great helpers.

A simple domain rule such as 'A Rock is made of at least one 'Mineral' can be translated in XML as 'The `<Rock>` tag must enclose one or more `<Mineral>` tags'. In XML Schema jargon, this says that `<Rock>` is a *complex* tag, enclosing a *sequence* of `<Mineral>` tags that appear at least once, up to an unspecified number of times. This rule indicates that the following XML document is **valid**:

```
<Rock>
<Mineral/>
<Mineral/>
</Rock>
```

while this one is not:

```
<Rock>
</Rock>
```

because at least one `<Mineral>` tag is required.

NADM CONCEPTUAL MODEL (NADM-C1)

For the *DITT*, we were lucky to have the NADM-specified domain for geologic maps already described by the DMDT. This team has worked for the last 2 years to develop the rules that describe geology and geological maps. The output of their effort is an extensive diagram,

called NADM-C1 (North American Data Model Steering Committee’s Data Model Design Team, 2003) showing principal geological features and how they relate to each other (the schematic for NADM-C1 (NADM Conceptual Model 1.0) is available from <http://geology.usgs.gov/dm/steering/teams/design>). Even if some aspects of the model are still being debated, a version 1.0 should be available soon.

The DITT must convert this information (see fig. 1 for a simplified example of a portion of the NADM-C1 diagram) into a format that is a suitable XML schema. As we pointed out, XML documents have their own sets of requirements and constraints. Therefore, decisions have to be made to ensure consistency in the conversion process. XML schema provide enough rope to hang any designer and no schematisation approach is ‘better’ that the other.

Many XML encoding styles exist. For example, some designers prefer the use of attributes,

```
<Rock Name = ”Granite”>
<Mineral/>
</Rock>
```

while some argue that we should avoid them and replace them by tags.

```
<Rock>
<Name>Granite</Name>
<Mineral/>
</Rock>
```

Both alternatives work, but they impose different constraints. Options must be evaluated and decisions must be made on the style to be used. This is critical to ensure consistency of the XML documents.

The simplified UML diagram of a portion of NADM-C1 (fig. 1) can be read as follows:

*A **CompoundMaterial** is a kind of **EarthMaterial,** which is a kind of GeologicConcept, that is composed of at least one other **EarthMaterial** but not limited to one. A GeologicConcept must be associated with least one Name, but can have more than one, and can also be associated with many **Descriptions,** but this is optional. A **CompoundMaterial** must be associated with one and only one **ConsolidationDegree**. A **Rock** is a kind of **CompoundMaterial;** therefore, it is also composed of other **EarthMaterials** and is also associated with a **ConsolidationDegree**. And since a CompoundMaterial is a GeologicConcept, a Rock is a GeologicConcept as well and must have at least one name, and potentially some descriptions. A Mineral is a kind of **EarthMaterial** (and is not a composition of **EarthMaterials**) . . .*

and so on.

We can tell from the diagram that a **Rock** is made of other **Rocks**, **Minerals**, **Fluids** and **Glasses**. All these components forming a **Rock** are optional, but at least one component must appear. For example, a quartzite is essentially made of only quartz mineral; a porous conglomerate is a mixture of rocks, mineral, possibly glass, voids and even fluids.

An XML schema translation of those rules is shown in figure 2. This figure has been made with XML Spy (<http://www.altova.com>); it simplifies the reading of

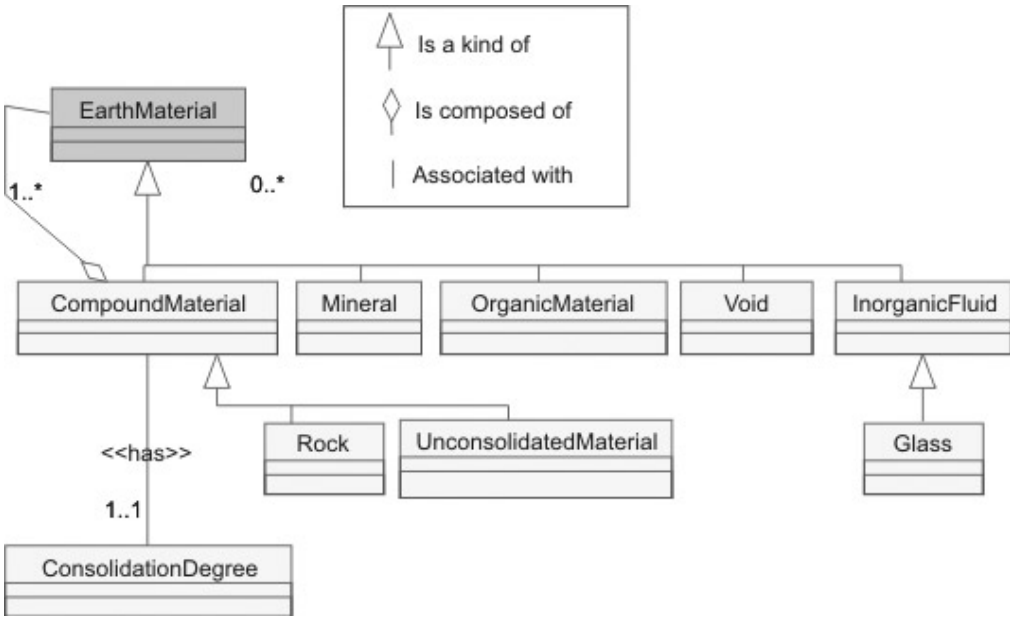

Figure 1. Excerpt of NADM-C1 Diagram showing a Compound Material made of other EarthMaterial.

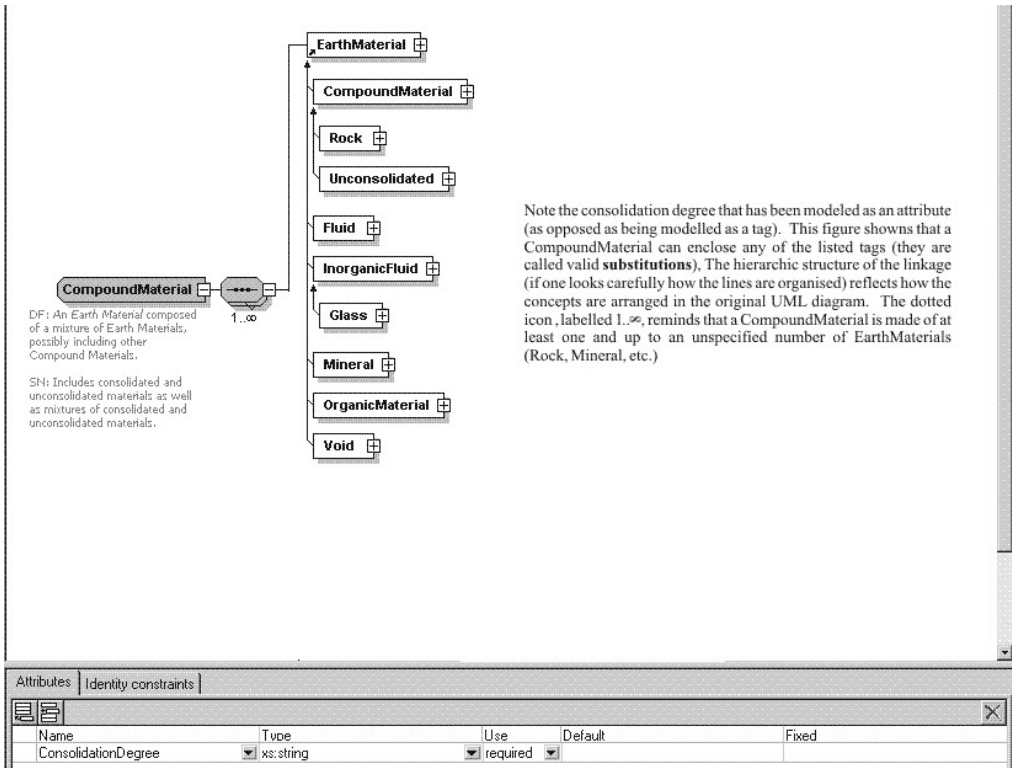

Figure 2. Translation of figure 1 in an XML schema using XMLSpy.

XML schemas by representing tags and relations between them using an easy to understand schematic. For instance, the following document describing a compound material (remember, Rock is a CompoundMaterial) would validate against the XML schema depicted in figure 2:

```
<Rock ConsolidationDegree = "lithified">
    <Name> Granite </Name>
    <Mineral>
        <Name> Biotite <Name/>
    </Mineral>
    <Mineral>
        <Name> Quartz <Name/>
    </Mineral>
</Rock>
```

But this next example would not be a valid NADM-C1 document because according to the NADM-C1 diagram, **Fluids** cannot contain **Minerals**.

```
<InorganicFluid>
    <Name>Water</Name>
    <Mineral>
        <Name>Salt</Name>
        <Name>Halite</Name>
    </Mineral>
</InorganicFluid>
```

The correct way to define it would be a mixture of Water and Mineral (not water *containing* mineral).

```
<CompoundMaterial ConsolidationDegree =
    "fluid">
    <Name>Salty Water</Name>
    <InorganicFluid>
        <Name>Water</Name>
    </InorganicFluid>
    <Mineral>
        <Name>Salt</Name>
        <Name>Halite</Name>
    </Mineral>
</CompoundMaterial>
```

XML AS AN INTERCHANGE MECHANISM

The real benefit of XML encoding is the ease with which one can manipulate the document. In addition to the existing programming tools used to develop applications, another set of tools are available to transform XML documents from one schema to another. XSLT (eXtensible Stylesheet Language Transformations) is a specification to encode transformation rules (also called a 'Style Sheet') to convert an XML schema into another XML schema. XSLT is not restricted to XML transformation; in fact, an interesting application is the transformation of XML documents to a any text based document (for example, a series of SQL command). But we still have to keep in mind that XSLT has been designed for XML.

Transforming dynamically from one schema to

another is the root of a technique called ‘mediator-wrapper’, where small softwares perform small translation tasks. A **wrapper** is a small application that translates a source dataset into an XML document or the other way around. The mediator takes several XML documents and manipulates them to create a new XML document suitable for another wrapper to handle and turn into something useful. Some mediators are merely transport mechanisms that do not transform the source XML but just convey it to a destination wrapper. Figure 3 shows such a mediator-wrapper architecture that transfers geological information from one database structure to another. The first wrapper transforms a subset of a database into a NADM-XML document, which is sent to another wrapper that extracts the information it needs to fill its own database. The first wrapper has no knowledge of the destination format, nor does the destination wrapper know anything about the source database. Each participating database must be able to translate between its own structure into NADM-XML and vice and versa.

Problems arise when we want to use the XML document to export a subset of a database. NADM-C1 is highly recursive; ‘things’ are described against other things, which in turn are also described using other ‘things’. XML on the other hand is *sequential*. It is constructed from top to bottom, with tags nested into tags. Where does the document stop? Should it contain ALL related data? It could potentially export the complete database by following the linkage between geological ‘things’. This forces us to think ahead on how to reference something outside a XML document. There are mechanisms in XML to point to something outside of the current document (Xpath, XPointer), but this mechanism assumes that the pointed elements are also inside the XML document. But in our case, the information might (and probably will) reside in a database. The actual schema, which is still in draft, is an encoding of the conceptual model, as if a document should contain all related data; the next step is to design those pointer mechanisms.

CONCLUSION

Encoding XML schema from the conceptual geologic data model is quite a challenge. Beside the substantial learning curve of schema design and the number of decisions that must be made to achieve a clean design, NADM-C1 is a complex model that reflects the complexity of geology. This complexity has an impact on the design of the schema. Lots of ‘design patterns’ and rules of thumb are available from the XML community. But as Alan Kay once said, “Simple things should be simple. Complex things should be possible” (Lipkie and others, 1982), and the payoff of this work should be a greater usability of geoscience and improved interoperability.

REFERENCES

Drewes, Harald, 1998, Geologic map of the Bartlett Mountain quadrangle, Pima and Santa Cruz Counties, Arizona: U.S. Geological Survey Miscellaneous Geologic Investigations Map I-2624, scale 1:24000.

Duckett, Jon, Griffin, Oliver, Mohr, Stephen, Francis, Wtokes-Rees, Ian, Williams, Kevin, Cagle, Kurt, Ozu, Nikola and Tennison, Jeni, 2001, Professional XML Schemas:

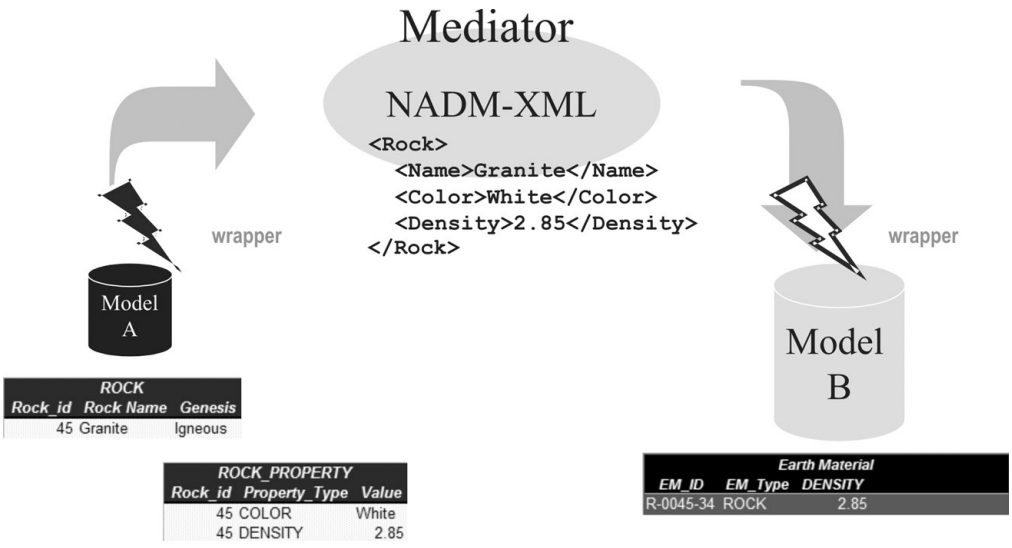

Figure 3. Scenario of an exchange of data between two databases of different structure using mediator-wrapper technology.

Birmingham, UK., Wrox Press ltd, 693 pages.

North American Data Model Steering Committee's Data Model Design Team, 2003, Proposed North American Standard Conceptual Data Model for Geologic Map Information. <http://geology.usgs.gov/dm/steering/teams/design/>

Lipkie, D. E., Evans, S. R., Newlin, J. K., and Weissman, R. L., Star graphics: An object-oriented implementation, *in* Proceedings, SIGGRAPH'82: Computer Graphics 16 (3), July 1982, pp. 115-124.

APPENDIX

Open source Movement:	<http://www.opensource.org/>
Leading XML portal:	<http://www.xml.org/>
W3C xml specifications	
XML:	<http://www.w3.org/XML/>
XSLT:	<http://www.w3.org/TR/xslt>
Schema:	<http://www.w3.org/XML/Schema>
XMML	<http://www.ned.dem.csiro.au/XMML/>
NADM-C1	<http://geology.usgs.gov/dm/steering/teams/design/NADM-C1.0/NADMC1_0.pdf>
XMLSpy:	<http://www.altova.com>