Preserving State Government Digital Information
Core Legislative XML Schema Meeting

Minnesota Historical Society

Friday, October 29, 2008
Pillsbury Conference Room, Minnesota History Center
St. Paul, Minnesota

Meeting participants included Dan Dodge and Jolene Sather (Thomson-Reuters), Isaac Holmlund, Tim Orr, and Michele Timmons (Minnesota Revisor of Statutes Office), Nancy Hoffman, Bob Horton, Jennifer Jones, Charles Rodgers, Shawn Rounds, and Carol Kussmann (Minnesota Historical Society).

The following account is paraphrased.

Introduction:

Bob gave a short introduction to the meeting, including reviewing the materials that were sent in by Dan and Tim. Overall impression is that they will be very useful. Other institutions are developing similar tools, such as the Smithsonian Archives XML email preservation schema. He turned to the summary of the last meeting and reviewed the intended functions of the schema — data transmission & extraction, archive & access, and reuse. He also pointed out the metadata crosswalk containing the list of possible metadata elements we identified at that time and referred to Tim and Dan's documents. Discussion will consider how the entities would be incorporated in the final product to support the intended functionality.

Review of Tim's Handout: MN_bill_Metadata

Tim: This shows metadata that is accessible today from the Minnesota Revisor's Office (Revisor's Office), Shawn has added this information to the metadata crosswalk.

Shawn: The update to the crosswalk shows the metadata from this document as well as from Dan's file. The first column shows the standardized set developed for this project while the other columns show a comparison to metadata standards of other systems. In general they match up pretty well, a few from the Revisor's Office do not fit exactly but can be moved if necessary or added for specific projects, such as this one.

Review of Dan's Information: XML Wrapper

Dan: This shows how we could use a wrapper for content in a way that would meet our goal of defining a standard XML schema that would be the same for both senders and receivers. We previously abandoned the one size fits all method because it won't work. So to preserve data, we would have a standard wrapper schema plus a jurisdiction specific schema allowing adjustments or additions based on what is available from each state. The wrapper schema would require certain metadata. Validation would be done to ensure that the data transmitted was complete and met standards.

Going down the document, page one has the wrapper for the XML source where the real copy of bill and its XML data is preserved. There is a wrapper for an optional XHTML version if one is provided for browsing or offline printing. There is an optional wrapper for binary attachments. Each attachment would have its own wrapper so that they could be opened separately.

The second page has some examples. Participants would each need a namespace set-up in advance. This would avoid the problem of providers and receivers having to negotiate with each other every time data are transmitted. The XML Schema Definition (XSD) should stay the same; the wrapper should be very stable and unchanging. The *xml.source* should conform to the namespace and the schema syntax. Everything the parser needs should comply with the standards. The metadata elements, such as the *meta.identifier*, need to be determined and defined. This example is based on the Revisor's office metadata elements, which could have limitations because they are specific to Minnesota.

Tim: We might decide that some of our elements should go away.

Dan: Or they could be used as a fall back. At some point we will need to talk about special characters also. (More here that repeats points about separate wrappers for each binary instance so that they can be opened separately - and valid XHTML so that the bill text can be viewed in a browser)

Page three is the wrapper schema only. The top is the XSD import syntax. I used the Minnesota Revisor's Office namespace in the example. I put the schema with the XML to avoid cutting and pasting. The rest is schema definition and validation. I did not want to specify a set of elements since we will still need to define the required metadata, what the child elements might be, and specifics such as minimum and maximum occurrences.

Bob: We will need to match-up the metadata elements with the crosswalk and possibly add more.

Dan: Yes, this is just a prototype for where we want to go.

Bob: Once the crosswalk is agreed upon, then the metadata elements can be added to this prototype version.

Dan: The HTML code will come from the Revisor's Office.

Page four – attachments can be one or more binary elements. I didn't do much research on encoding because it was not the main focus of this document. ID may be another required element. It makes

sense to use existing standards and then adapt them as needed (there was general agreement on this point).

Bob: Do we need to scope out standards for every possible type of attachment?

Dan & Tim: No we should start with the formats applicable to the Revisers Office.

Dan: We can work on the wrapper first and then clarify the internal specifications later.

Tim: I have a question about the example on page two, where users are directed to the Revisors Office bill schema. The recipient will need to go get the bill schema right away, in case it changes later.

Dan: Yes or the recipient could link out to it.

Tim: The schema is not always available to be linked to.

Jolene: Could the schema be sent with the bill file?

Issac: The schema Uniform Resource Identifier (URI) is not always a Uniform Resource Locator (URL). It should be sent with the bill.

Jolene: Would it be difficult to send the schema with the bill?

Tim: Having many schemas complicates the process.

Dan: Hard coded namespace schema definitions could be tweaked to be relative and portable.

Tim: Jolene is right; each file needs to be self contained.

Dan: Right, without the schema, the bill files could become inaccessible like so many obsolete technologies and formats we can think of.

Bob: Data that is transmitted will need to be tested to ensure that it is readable and authentic and we will need to archive the schema with the bill file.

Tim: Who would be responsible for that?

Bob: What is easiest for the creator?

Tim: From our standpoint, it would be easier to have the recipient check it.

Bob: We don’t want to put the onus on the data provider.

Dan: How often does the schema change – how often would it need to be collected?

Bob & Shawn: It would need to be collected right away when the data is transmitted.

Bob: I have two questions at this point: First, can we use a wrapper for a batch of data, such as all the bills from a session? Second, can we use the wrapper to transmit legislative data?

Tim: Logically it makes sense, but if you run a batch process you encounter new problems, one being the size of the data, 10's of gigabytes perhaps. Then, how do you transmit all the data at once? Also, legislation and other information are represented in a variety of ways. Bundling will not solve that problem.

Issac: With additional layers of wrappers, the hierarchal data structure falls apart when you try to aggregate it. We would want to look at semantic solutions to achieve that. This would be a very good approach for the kind of data we have. It would be more difficult to encapsulate all the bills in a session.

Dan: Yes, and wrapping would require a set of secondary files relating all the additional parts to each other.

Bob: Perhaps something like Encoded Archival Description (EAD) that archivists use might be a useful tool for providing access to aggregated files. EAD is an XML standard for encoding finding aids. It resembles traditional box, folder, and item level descriptions. This could serve as a wrapper on a wrapper idea such as: Minnesota, the 87th legislative session, and specific bills within that session.

Jolene: Is EAD hierarchical?

> Charlie goes to get EAD information for people to look at, including,
> 1. EAD (Encoded Archival Description) Cookbook (EAD version 2002) Available online at http://archivists.org/saagroups/ead/ead2002cookbook.html),
> 2. Describing Archives: A Content Standard 2007. Society of American Archivists.
> 3. Two examples of EAD finding aids – including one from Governor Johnson's records from the MSA and another from the Cookbook cited above.

Bob: Going back to extraction – how do we transmit information from the Revisor's Office to the Historical Society (MHS)? What does the Revisor's Office have and what does MHS need to do with it to get it?

Tim: We currently have programs that bundle one doc at a time and FPT it to Thomson Reuters. But the wrapper on wrapper idea might be a different project. For now, we could get the info to you and you can do what you want with it.

Dan: We might want to define the semantics between the objects.

Nancy: Has anyone seen the Open Archives Initiative – Object Reuse and Exchange specifications (OAI-ORE)? This is a system for expressing just the kind of semantic relationships we are considering.

Bob: We can make a note to explore that. So, how do we get there from here? We have looked at Dan's document at the item / object level and we think that it is doable now, but we need to answer

the question of how to get extraction from Revisor's Office to MHS and is there a way to wrap something like EAD over it once get it?

Tim to Dan: Does the XHTML need to be there? Would all the children inherit the schema from the namespace? Some testing would need to be done.

Jolene: Who decides how the pieces fit together? Is it the responsibility of source/provider – does it fit into the relation element in crosswalk? How do these things relate to capture the process of a bill and its changes before it becomes a law?

Bob: The bills and revisions are linked online now. The modifications are listed and you are able to connect pieces.

Jolene: Is linking done with metadata?

Bob: Could we use the 'Relations' element for linking?

Jolene: Or, would we want to use the 'Sessions' element or one of the other elements for linking?

Bob: We would need to consistently populate the element used to link files. Are the files produced by the Revisor's Office already linked?

Issac: Not completely.

Jolene: How is the linking done by the Revisor's Office?

Tim: We use two systems. There are different ways of representing connections between data. Relating things is the function of a relational database. Linking documents is something different.

Bob: How does Thomson link related files?

Jolene: Currently this linking is done manually.

Isaac: The Revisor's Office uses different systems to get information on the web. The bill drafting system is used to produce the printed statutes. We have various methods of getting the statute information on the web, but the systems don't mesh.

Michele: The bill status system is in an Oracle database maintained by the Revisor's Office, but the data is updated by House and Senate staffers. Perhaps we could write a program to link some of the related information such as authors, but these are not currently part of the metadata that is collected in the bill drafting system.

Isaac: The only things in XML produced by the Revisor's Office are the bills.

Jolene: We get the data and add value with additional information such as metadata in order to create a useful product that we can sell.

Tim: In order to include all of the metadata elements we have discussed when a bill is transferred, we would have to add data from the bill status system. We would have to write a program to do this

Michele: But it is doable.

Bob: How do you connect the information in the bill status system with the bill drafting system; do you have an identifier?

Tim: No, we use a combination of several pieces of information including bill title, bill number, and session but this is not consistent. It is very contextual.

Issac: We do not have consistent unique identifiers for each bill, but it would be good to have them.

Dan: This makes the point that one size might not fit everyone for the processing schema either.

Bob: Let's try to track the implications in relation to our goals (writing on the board):

> First - we want to facilitate capture and interchange at the item level. The metadata elements would allow us to group items but we would have to use another application to do anything with them. Right now we can confirm this approach by testing and making any refinements based upon the results.
>
> Second - processing events. We may want to capture events as well as information, such as the actions recorded in the bill status database.
>
> Third – archiving and preservation, possibly using EAD to wrap collections of items. We may want to try using different types of wrappers.
>
> Fourth – access and reuse. We would need to use other schemas to achieve this. This area might need to be explored further.

Charlie returns with EAD documentation and examples listed above for the team to review.

Tim: Everything on the list looks all right except number two (processing events). That might be something we can do after we work on the other three items on the list.

Jolene: The way we choose to accomplish the objectives listed in number three (archiving and preservation), may inform the way we get to those listed in number one (capture and interchange).

Bob: The metadata from the crosswalk might help define metadata on for capture and interchange. Perhaps we need to add another column to current crosswalk for EAD elements.

Tim: A low tech pre-step to address the archiving function might be a hierarchical directory on the server.

Bob: So maybe we would want to capture the directory structure.

Dan: Using the archival description analogy, would a box equal a document?

Bob: A box would equate with all the bills in a session.

Dan: If the objective is to create archival description information, would that be done by the sender or the receiver? Regarding objective number three (archiving and preservation), we don't want to remix the information received from a source; we want to keep the file structure the same.

Bob: The box–folder–item structure would ideally mirror original creation order.

Shawn: Directory structure seems to be a good idea.

Tim: And to bundle files, we could just use a higher level in the directory structure.

Dan: Would the identifier be the main file name?

Tim: Yes, all the key identifiers are in the file name (bill title, bill number, session number).

Michele: (referring to the crosswalk) The engrossment number is a part of the name and not a separate element.

Dan: The test of identifiers is that each one is unique. If you have twelve ids you should have twelve different documents.

Shawn: So it would be part of the title.

Tim: Actually, it is part of the identifier.

Jolene: Senders wouldn't send the file structure though, they would just send a set of files so twelve nodes would equal twelve documents.

Tim: In our directory, each file is one document.

Dan: Sub-directories would create incorrect counts because each node would not correspond to a document.

Tim: The directory structure would serve as an organizational tool only.

Jolene: So you have a folder that holds the documents for a given Minnesota Legislature year and session. Would this end up in the metadata?

Tim: Yes it would.

Bob: We would need to test a bill to see if there are any kinks in the process.

Jolene: Minnesota Revisor's office could send a bill and we could figure out how to organize and authenticate it – to ensure that the file was complete and accurate. Then we would package it and provide something else.

Tim: We need a schema to do this.

Dan: I could do a proof of concept with a bill and schema from the Revisor's Office. This could be staged on our work environment. I would create everything manually the first time, and then figure out how to automate the process.

Jolene: Once Dan has wrapped a bill, the Revisor's Office should determine if everything worked correctly.

Issac: Dan would need the additional formats also.

Dan: Getting the XHTML first would work best.

Tim & Issac: We can give you everything we have, but we don't have the XHTML yet.

Dan: So regular HTML would be a binary attachment.

Issac: Yes and most states would probably have mime types other than XHTML.

Jolene: What are the XML wrapper elements?

Dan: They would be the elements listed in column one in the crosswalk.

Tim: Might there be a benefit to using an existing metadata model?

Shawn: Column two in the crosswalk is the Minnesota Recordkeeping Metadata System (MRMS) and it closely matches the schema elements that we came up with.

Jolene: It also has many elements that match those in the Dublin Core set.

Shawn: We could use the MRMS and add a 'Session' element.

Dan: Would that work for other jurisdictions?

Bob: One way we could find out is by asking our partners at the December 8th meeting we have planned. We could also have the California Digital Library (CDL) test our proposal. We could roll out a demonstration at the spring project site visits. Presentations at the National Conference of State Legislatures (NCSL), National Association of Legislative Information Technology (NALIT) (http://www.ncsl.org/nalit/) meeting would be good place to get comments. We can send someone to a conference and NALIT members would be equipped to provide practical comments. Other organizations' meetings, such as law librarians and National Conference of Commissioners on Uniform State Laws (NCUSL) (http://www.nccusl.org/Update/) may be venues where we might want to make presentations also.

Dan: This would be a good way to get feedback about potential problems.

Bob: We have already shown that people are interested in reuse, and preservation, and at the production level, everyone will benefit from this research. During the first round of state partner

visits it became clear that everyone in our project is using XML in some fashion when producing legislative documents.

Discussion of Metadata Elements (see results in the table below)

Element	Required or Optional	Additional Notes
Record Identifier	Required.	Must be unique in the system of origin, no format requirements
Title	Required	Includes Bill name
Type	Required	
Coverage (Jurisdiction)	Required and repeatable	Includes state information as well as which body of govt.
Agent	Required	Includes Publisher (required) and bill author (optional)
Date	Optional	Date made public, other dates
Session	Required	Includes session year and number as children
Description	Required and repeatable	Bill title, or other summaries as needed/desired
Subject	Optional but repeatable	Topics
Relation	Optional but repeatable	Companion Bill names, law chapters, versions of (engrossments). Items related to, not people.
Management History	Optional	
Rights Management	Required?	NEED TO DISCUSS FUTHER (Michele will check on a copyright statement with the legal staff)
Aggregation Level	Optional	
Format	Optional	
Language	Optional	
Function	Optional	
Mandate	Optional	
Use History	Optional	
Preservation History	Optional	
Disposal	Optional	
Location	Optional	
Items found originally on Revisor's Office list have		

been moved		
Body	Moved to Jurisdiction	
Engrossment	Moved to Relations	
Authors	Covered under agents	
Authors Legislative ID's	Under agent, but optional	
Governors Action	Add as an optional field.	
File Name and Page	DROP from list.	

Points made during the element review:

Need for a unique identifier that could be made up of pieces of information from other elements such as bill name, session year, as needed by the data provider.

Although Dublin Core recommends adding attributes to an element, (http://dublincore.org/documents/2003/04/02/dc-xml-guidelines/) MRMS uses sub-elements.

The header of bills includes additional information such as the initials of the person who drafted the legislation, but it was decided that none of it should be added to the final metadata set.

A discussion of capturing legislator identification numbers lead to the observation that these numbers are essentially URIs and that treating them as such would facilitate semantic functionality if this information was made available in a distributed environment such as the web.

A copyright statement will need to be drafted after further research with state legal staff.

Summary

Metadata – Shawn offered to rework the crosswalk and write-up the specification for the metadata based upon the discussion at the meeting and she will post it for comment.

Testing the wrapper – Tim will send Dan the XML version of the legislation and any other formats such HTML. Dan will wrap it and send a sample back to Tim for review before the actual "test" to ensure the validity in both environments. The test will be completed by the partners meeting on December 8th. The process will be documented in Basecamp.

An example of California legislation could be tested by Devon Shepherd after the Minnesota test is successful. The goal for completion would be the end of this year.

Minnesota material from an entire year could be managed at MHS as a proof of concept.

Below you will find the two documents discussed above…
MN-Bill metadata and the XML wrapper

Subject: Minnesota Legislative Metadata per bill – FIRST DRAFT
From: Tim Orr
Date: October 8, 2008

The metadata in Table 1 is readily available today for each bill in the Minnesota legislature. The metadata resides in 3 digital data sources.

1. Filename - the filenaming convention of the document
    a. An example filename is

H1812-5.xml

2. Document - the XML document (within filename)
    a. A portion of the XML file H1812-5.xml containing metadata follows,

```
<bill xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      engrossed="y"
       type="bill"
       xsi:noNamespaceSchemaLocation="http://revisor.leg.state.mn.us/xtend/bill.xsd">
<meta>
  <meta_val name="BIX" val="bix" />
  <meta_val name="RLANG" val="true" />
  <meta_val name="header" val="true" />
  <meta_val name="office" val="REVISOR" />
  <meta_val name="message" val="FIFTH ENGROSSMENT" />
  <meta_val name="initials" val="DI" />
  <meta_val name="print_date" val="HF1812" />
  <meta_val name="art_sec_footer" val="true" />
  <meta_val name="page_footer" val="true" />
</meta>
<document_info document_name="H1812-5" document_path="draft/RS/85/engrossment/" />
```

3. BSS DB - the Revisor's Bill Status System database
    a. An example of the metadata can be viewed at,

https://www.revisor.leg.state.mn.us/revisor/pages/search_status/status_detail.php?b=House&f=HF1812&ssn=0&y=2007

Table 1. Existing, Available Metadata per Bill

MN Recordkeeping Metadata Standard ?	Name	Definition	Obligation	Source	Dat Typ
Agent	publisher	Reproducer of a work intended for public consumption	Mandatory	Constant	strin
	body	Body that introduced bill	Mandatory	Filename, BSS DB	strin
Title, Record Identifier(1/3)	bill name	Name used by people.	Mandatory	BSS DB	strin
Type	type	Document type.	Mandatory	Filename, BSS DB	strin
	engrossment	The version of the bill after amendments were applied.	Optional	Filename	inte
	authors		Mandatory	BSS DB	strin (CS
	authors legislative ids	Unique ID for every legislator, maintained by LRL.	Optional	BSS DB	inte , (CS
Subject	topics	1 or more fixed phrases describing the document	Mandatory	BSS DB	strin (CS
Subject	short description	Phrase or 1-2 sentences.	Optional	BSS DB	strin
Subject	summary	House Research or Senate Counsel Summary	Optional		strin (UR
Relation	companion bill name	Identical bill introduced in the other body.	Optional	BSS DB	strin
Relation	law chapter	A sequential number assigned each bill that becomes law.	Optional		inte
	governor action	The governor's action taken on bills passed by the legislature	Optional	BSS DB	strin
	filename	Computer filename.	Mandatory	Filename	strin
	pages	Number of pages in the document when printed.	Optional	Document	inte
Date	date_time made public	Time when document was publically released	Mandatory	Filename	inte
Format	noNamespaceSchemaLocation	Schema describing the XML tags in the XML document	Mandatory	Document	strin
Coverage, Record	session year	Calendar year	Mandatory	BSS DB	inte

MN Recordkeeping Metadata Standard ?	Name	Definition	Obligation	Source	Dat Typ
Identifier(2/3)					
Coverage, Record Identifier(3/3)	special session number	Extra session held after regular session adjournment.	Mandatory	BSS DB	inte

---

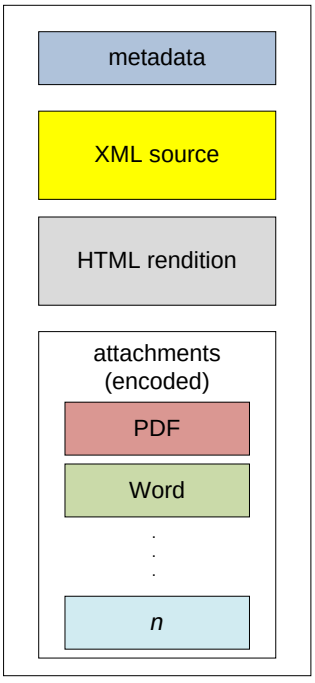
Conceptual Core Content Capture and Archival "Wrapper" Schema

Thomson Reuters
October 20, 2008

Overview

Using the existing core schema produced by the Revisor's IS staff, plus additional ideas generated during our September 12 meeting, we have drafted a conceptual schema that we believe illustrates what the XML structure could look like.

This diagram shows the basic document structure:



The entire XML instance has up to four main components:

- The first required component is the metadata instance. It contains required metadata essential to the interchange recipients.

- Another required component is the XML source, in the provider's native syntax. It uses an appropriate namespace prefix and identifier for each separate native schema.

- An optional component is an HTML rendition of the document being interchanged.

- Another optional component is a block of one or more attachments. These attachments are an encoded version of a binary format, such as PDF or Word or other proprietary delivery format. By encoding it (using Base64 or other "7-bit-safe" format) and enclosing it within a CDATA marked section, the provider may include other types of versions of the document.

Example XML

```
<?xml version="1.0" encoding="utf-8"?>
<document
 xmlns="http://www.digitalpreservation.gov/schemas/legislative"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xhtml="http://www.w3.org/1999/xhtml"
 xmlns:mn="http://revisor.leg.state.mn.us"
 xsi:schemaLocation=
   "http://www.digitalpreservation.gov/schemas/legislative coreschema_v2008-10-20.xsd
    http://revisor.leg.state.mn.us bill.xsd">
```

```
<metadata>
   <meta.identifier>H.F. No. 1812, 5th Engrossment - 85th Legislative Session
                   (2007-2008)</meta.identifier>
   <meta.provider>House of Representatives</meta.provider>
   <meta.subject>A bill for an act....</meta.subject>
   <meta.sponsor>Representative Smith</meta.sponsor>
   <meta.state.jurisdiction>MN</meta.state.jurisdiction>
   <meta tag="session" value="85"/>
   <meta tag="spec_sess_num" value="0"/>
   <meta tag="publisher" value="Revisor of Statutes"/>
   <meta tag="pub_phone" value="651.296.2868"/>
   <meta tag="pub_zipcode" value="55155"/>
   <meta tag="date_official" value="2008-06-02"/>
   <meta tag="date_xmit" value="2008-07-18"/>
</metadata>
```

```
<xml.source>
<mn:bill>namespace-prefixed XML for sample Minnesota bill document here</mn:bill>
</xml.source>
```

```
<html.rendition>
   <xhtml:HTML>
   <xhtml:HEAD>
   <xhtml:META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1252"/>
   <xhtml:TITLE>Home</xhtml:TITLE>
   </xhtml:HEAD>
   <xhtml:BODY LINK="#0000ff" VLINK="#800080">
   <xhtml:P>&amp;#160;</xhtml:P>
```

```
</xhtml:BODY>
</xhtml:HTML>
</html.rendition>
```

```
<attachments>
   <binary ID="thing1" encoding="Base64" source="application/pdf">
      <![CDATA[
         encoded data here
      ]]>
   </binary>
</attachments>
```

```
</document>
```

The metadata component has examples of metadata. It does not show all the required or optional metadata because the Crosswalk process has not been completed.

Draft Core Schema (2009-10-20)

A draft W3C schema has been developed to validate the sample document illustrated above. It includes a (temporary) namespace for the elements in the Minnesota Revisor's office Bill schema. This core schema and a "stubbed-out" version of the Bill schema (named "bill.xsd") are shown below.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://www.digitalpreservation.gov/schemas/legislative"
            xmlns="http://www.digitalpreservation.gov/schemas/legislative"
            xmlns:mn="http://revisor.leg.state.mn.us"
            elementFormDefault="qualified">

    <xsd:import namespace="http://revisor.leg.state.mn.us" schemaLocation="bill.xsd"/>

    <xsd:element name="document">
        <xsd:complexType>
            <xsd:sequence>
               <xsd:element ref="metadata" minOccurs="1" maxOccurs="1"/>
               <xsd:element name="xml.source" type="mn:legislative_bill"/>
               <xsd:element ref="html.rendition" minOccurs="0" maxOccurs="1"/>
               <xsd:element ref="attachments" minOccurs="0" maxOccurs="1"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="metadata">
        <xsd:complexType>
            <xsd:sequence>
               <xsd:element name="meta.identifier" type="xsd:string" minOccurs="1"/>
               <xsd:element name="meta.provider" type="xsd:string" minOccurs="1"/>
               <xsd:element name="meta.subject" type="xsd:string" minOccurs="1"/>
               <xsd:element name="meta.sponsor" type="xsd:string" minOccurs="1"/>
               <xsd:element name="meta.state.jurisdiction" type="xsd:string"
                    minOccurs="1"/>
               <xsd:element name="meta" maxOccurs="unbounded">
                  <xsd:complexType>
                     <xsd:attribute name="tag" type="xsd:string"/>
                     <xsd:attribute name="value" type="xsd:string"/>
                  </xsd:complexType>
               </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
```

```
    <xsd:element name="html.rendition">
        <xsd:complexType>
            <xsd:sequence>
               <xsd:any namespace="http://www.w3.org/1999/xhtml" processContents="skip"
                    maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="attachments">
        <xsd:complexType>
            <xsd:sequence>
               <xsd:element name="binary">
                  <xsd:complexType mixed="true">
                       <xsd:attribute name="ID" type="xsd:ID" use="required" />
                       <xsd:attribute name="encoding" use="required">
                           <xsd:simpleType>
                               <xsd:restriction base="xsd:NMTOKEN">
                                   <xsd:enumeration value="Base64"/>
                                   <xsd:enumeration value="US-ASCII"/>
                               </xsd:restriction>
                           </xsd:simpleType>
                       </xsd:attribute>
                       <xsd:attribute name="source" use="required">
                           <xsd:simpleType>
                               <xsd:restriction base="xsd:normalizedString">
                                   <xsd:enumeration value="application/eps"/>
                                   <xsd:enumeration value="application/msword"/>
                                   <xsd:enumeration value="application/pdf"/>
                                   <xsd:enumeration value="application/zip"/>
                                   <xsd:enumeration value="text/rtf"/>
                                   <xsd:enumeration value="text/plain"/>
                               </xsd:restriction>
                           </xsd:simpleType>
                       </xsd:attribute>
                   </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

</xsd:schema>
```

Placeholder Bill Schema

This is a temporary schema developed to test the core schema. In practice, each provider would maintain a schema for use by the receivers of the legislative XML data.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://revisor.leg.state.mn.us"
            xmlns="http://revisor.leg.state.mn.us"
            elementFormDefault="qualified">

   <xsd:complexType name="legislative_bill">
      <xsd:sequence>
         <xsd:element name="bill" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>

</xsd:schema>
```