9-11-2015

The KF Modified Linked Data Project

F. Tim Knight
Osgoode Hall Law School of York University

Sarah Sutherland
CanLII

Follow this and additional works at: http://digitalcommons.osgoode.yorku.ca/librarians

Part of the Cataloging and Metadata Commons

Good morning everyone. Welcome to our panel discussion: "Navigating Linked Open Data." We'll start by talking about a couple of linked data projects we've been working on / and then we'll invite *you* to participate in the discussion.

We hope you'll be willing to *share* some of your thoughts and experiences engaging with linked data processes / and help us work through some of *our* assumptions and expectations about how linked data might work in the library environment. However, if you find you're not feeling especially chatty this morning we've also prepared a few talking points that we can introduce to get the conversation going.

I'm going to start things off by talking a bit about the KF Modified Linked Data project that my colleague Sarah Sutherland and I have been working on. And then I'll hand it over to Christina Harlow who'll describe her work using linked open data and a local controlled vocabulary to expand library authorities.

Outline:

A very brief history of KF Modified

Goals and objectives of the linked data project

Project phases

Where we are

What's left to do

I'll begin with a very brief history of the KF Modified classification scheme: where it came from and why it's still a popular choice for Canadian law libraries.

Then I'll run through the goals and objectives of the project, the various project phases and what we've been able to accomplish so far. And I'll end with what we still need to do in order to complete this project.

So, a brief history of the KF Modified classification scheme …

Brief History of KF Modified

"The problem is the rate of growth in the production of written literature. In both this country and throughout the rest of the world the rate of increase in the amount of information that is published each year is simply staggering. The growth is so phenomenal that some sober observers are beginning to talk about the 'monster of literacy that is ... engulfing us.'"

Layman E. Allen, Logic, Law and Dreams, 1959

This quote, from a rather interesting article by Layman E. Allen, who was at Yale University at the time, is referring to what he called the "librarians big problem" and talks about the state of legal publishing in 1959. He wrote … }}} The "monster of literacy" is an interesting turn of phrase that Allen attributes to the then VP of Research at Bell Telephone, William Oliver Baker.

So this was about 10 years *before* the first "official" law classification scheme became available. That was the 1968 draft of American federal law, dubbed the Library of Congress KF schedule. And, at the time, there was no standard classification available for Canadian law libraries and the Library of Congress didn't provide a classification scheme for Canadian law until 1976. So the "librarians problem" was how to handle this "monstrous" growth in publishing using the various, and largely alphabetical, "in-house" classification systems in use across the country.

When that 1968 draft came out a small group of law librarians, led by Shih-Sheng Hu, working at the University of Manitoba Law Library, took that draft KF schedule and modified it in a way that reflected the subject-oriented systems that were already in use in Canadian law libraries.

Today the Library of Congress has a fully developed law classification system but KF Modified has continued to be developed and is still a standard used in many contemporary law libraries in Canada.

LCC Classed First by Jurisdiction

Resources scattered throughout the collection
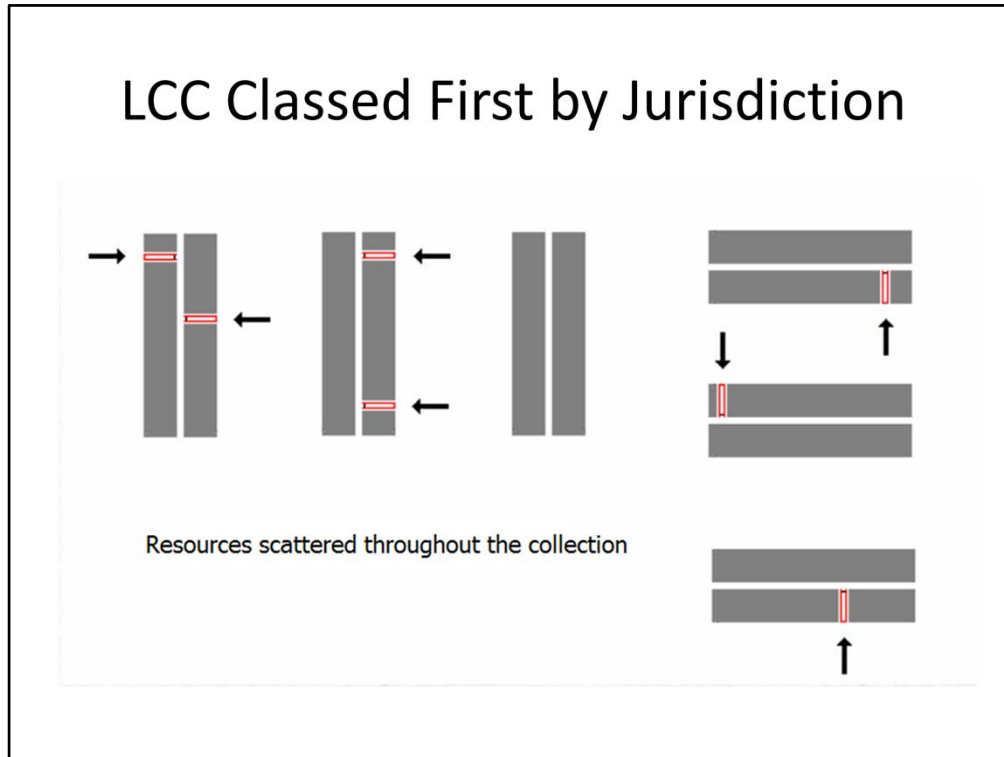
One of the reasons KF Modified has *endured* is the positive effect it seems to have had on the browsability of legal collections. The Library of Congress Classification provides a classification schedule for each of the various legal systems. Which, for the library user, means that books on criminal law, for example, would appear in different places throughout the library. Resources on criminal law in Canada, Ontario, England, Australia, etc. would be assigned their appropriate classification number based on the particular jurisdiction, ~~i.e. KE, KEO, KD or KU~~.

However, the nature of *common* law is that it essentially feeds off of itself. The law looks back and relies on previous judicial decisions that have established a legal precedent in that area of the law. As a result, precedents in all common law jurisdictions may be worth considering depending on the legal context.

KF Modified Classed First by Subject

Resources grouped together by subject
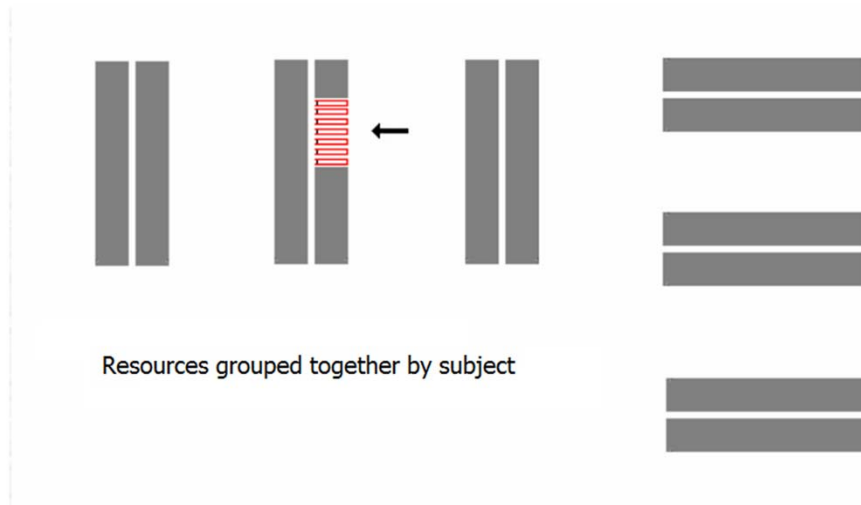
So, for lawyers working in common law jurisdictions it made some sense to group legal subjects together so that potentially related legal precedents could be more easily consulted.

KF Modified was designed to do just that by dividing common law resources first by *subject* and *then* by jurisdiction. This approach also drew on the experience of law library users familiar with the characteristics of the subject oriented systems that had been developed locally in law libraries. Another possible reason why KF Modified took hold and remains a popular classification standard today … at least in Canada.

Project Goals

- develop KF Modified as a linked data classification scheme
- bonus: internet accessible version of KF Modified
- end up with a methodology for the somewhat technically inclined librarian working on their own

The primary goal of this project was obviously to create a version of the KF Modified scheme that could, as linked data, play a role in organizing legal resources in the semantic web. It *seemed* like a small manageable chunk of work, something that could be done in a reasonable amount of time, and it offered opportunities to learn more about the technology and processes involved in creating linked data. After all, theory is one thing but practical application is a whole different story.

Aside from that basic goal the project also meant we'd be able to publish a web-based version of KF Modified. This would make it easier to manage and edit, easier for law cataloguers to use, and make it a potentially more attractive option to consider for common law libraries working *outside* of Canada.

And finally, we hoped that we could contribute a methodology for creating and using linked data that might benefit other technically minded librarians who find themselves without the necessary skills or resources to develop a full scale linked data project.

As somebody suggested in notes from the recent LODLAM Summit in Sydney, one of the many obstacles to getting projects off the ground is a perceived division between developers with the technical skills and researchers and subject specialists who realize the potential of this work but don't know how to get started.

So far my research has led me down many lovely garden paths … Which is why I'm very grateful for conferences like this one which provide wonderful opportunities to get together and crosspolinate, if you will … *but* I digress.

Project Phases

- develop XML schema appropriate for KF Modified
- convert print KF Modified to XML
- consider LCC linked data service and use of SKOS
- convert XML to HTML and RDF using XSLT
- test sample MARC record set using KF Modified
- publish KF Modified as linked data

Back to our project …

We outlined the main project development phases like this.
- Develop an XML schema appropriate for KF Modified classification
- Convert the print version of KF Modified to XML
- Review and consider the Library of Congress Classification linked data service and their use of SKOS
- Convert the XML version of KF Modified to HTML and RDF using XSLT
- Extract, convert and reconcile a MARC record set using the linked data version of KF Modified
- And, finally, make the project publicly available

Project Phases

- **develop XML schema appropriate for KF Modified**
- convert print KF Modified to XML
- consider LCC linked data service and use of SKOS
- convert XML to HTML and RDF using XSLT
- test sample MARC record set using KF Modified
- publish KF Modified as linked data

The first thing we *had* to do was to get KF Modified out of the legacy word processing software and into something more useful like XML. We decided on XML because it provides a lot of flexibility and allows the possibility of converting to a variety of other formats, for example maintaining print, HTML, or RDF versions.

However, / developing an XML schema that accommodated *all* of the quirky print conventions that cataloguers take for granted / proved to be a major challenge. Human readers can see something on a page and fairly easily interpret and understand what's intended by what they find there. But translating these nuances into something machine *actionable* was something else.

Take for example, this section from the original print version. This shows an internal table that can be applied later for classification of particular professions. There are actually two tables here: Table 'A' for single class numbers and Table 'B' applied to Cutter numbers.

If the profession is represented by a class number you'd add .3 for resources on malpractice. If it's represented by a Cutter number you add 3 to that Cutter.

So malpractice of engineers would be classed at KF 2928.3 …

… and dental malpractice would be classed at KF 2910 D33.

Converting to XML

```xml
<KFClassNumber>
    <KFNumber GD="false" form="false">
        <noClass />
        <classNumberCaption>Particular professions</classNumberCaption>
    </KFNumber>
    <KFTable>
    <KFTable>
        <tableName>Table B, Cutter number</tableName>
        <tableCutter>
            <cutterNumber>.x</cutterNumber>
            <cutterCaption>General</cutterCaption>
            <cutterForm>VII</cutterForm>
        </tableCutter>
        <tableCutter>
            <cutterNumber>.x1</cutterNumber>
            <cutterCaption>Licensing. Certification</cutterCaption>
            <cutterForm>VII</cutterForm>
        </tableCutter>
        <tableCutter>
            <cutterNumber>.x2</cutterNumber>
            <cutterCaption>Professional ethics</cutterCaption>
            <cutterForm>VII</cutterForm>
        </tableCutter>
        <tableCutter>
            <cutterNumber>.x3</cutterNumber>
            <cutterCaption>Malpractice. Liability</cutterCaption>
            <cutterForm>VII</cutterForm>
        </tableCutter>
    </KFTable>
```

So representing Table B in XML might look something like this.

After a lot of testing, sample coding and trial and error, we settled on an XML schema we could use.

XML Schema

```
<xs:element name="KFModified">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="KFClassification" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="KFConcept" type="conceptHierarchy" />
                        <xs:element name="KFClassGroup" minOccurs="1">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="KFClassNumber" maxOccurs="unbounded">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element name="KFNumber" type="classification" />
                                                <xs:element name="KFTable" type="table" minOccurs="0" maxOccurs="3" />
                                                <xs:element name="KFCutter" minOccurs="0" maxOccurs="unbounded" >
                                                    <xs:complexType>
                                                        <xs:sequence>
                                                            <xs:element name="cutterHeader" minOccurs="0" maxOccurs="1" />
                                                            <xs:element name="cutterHeaderNote" minOccurs="0" maxOccurs="3" />
                                                            <xs:choice>
                                                                <xs:element name="xCutter" type="cutter" minOccurs="0" maxOccurs="unbounded" />
                                                                <xs:element name="primaryCutter" type="cutter" minOccurs="0" maxOccurs="unbounded" />
                                                                <xs:element name="secondaryCutter" type="cutter" minOccurs="0" maxOccurs="unbounded" />
                                                                <xs:element name="tertiaryCutter" type="cutter" minOccurs="0" maxOccurs="unbounded" />
                                                            </xs:choice>
                                                        </xs:sequence>
                                                    </xs:complexType>
                                                </xs:element>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```
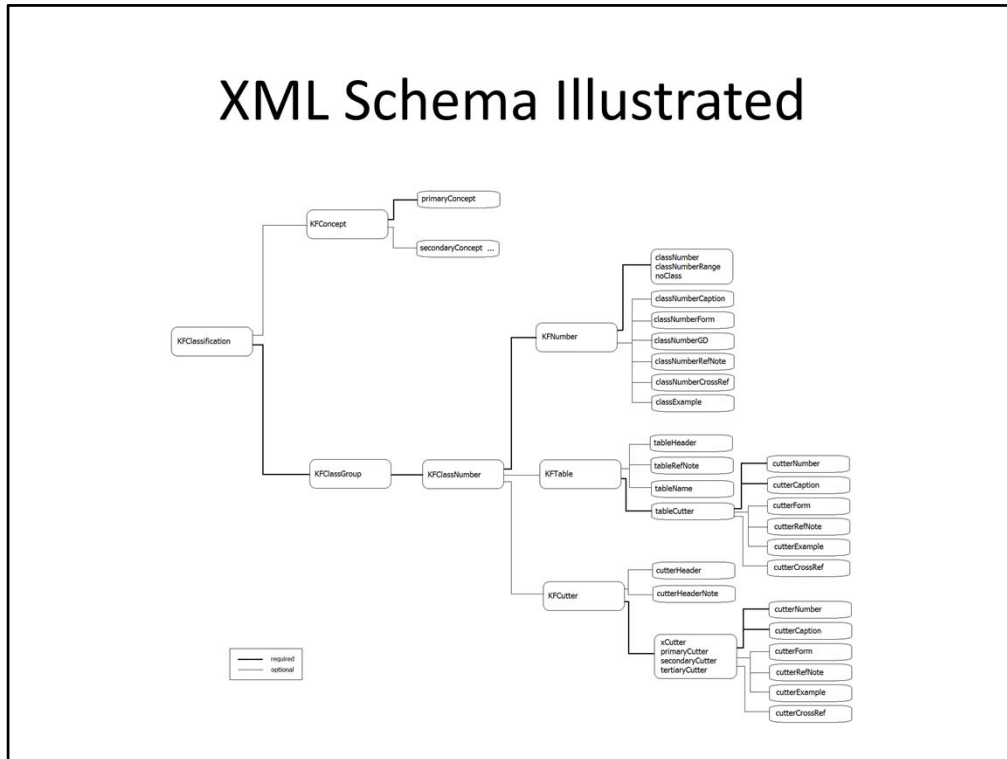
This is the main part of the schema, which you probably can't see very well here. I'll post these slides and the schema over on the KF Modified Blog so you can take a closer look if you're interested.

KF Modified Concepts

```xml
<xs:complexType name="conceptHierarchy">
    <xs:sequence>
        <xs:element name="primaryConcept">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="xs:string">
                        <xs:attribute name="indentUnit" type="xs:nonNegativeInteger" use="required"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
        <xs:element name="secondaryConcept" minOccurs="0" maxOccurs="15" >
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="xs:string">
                        <xs:attribute name="indentUnit" type="xs:nonNegativeInteger" use="required"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

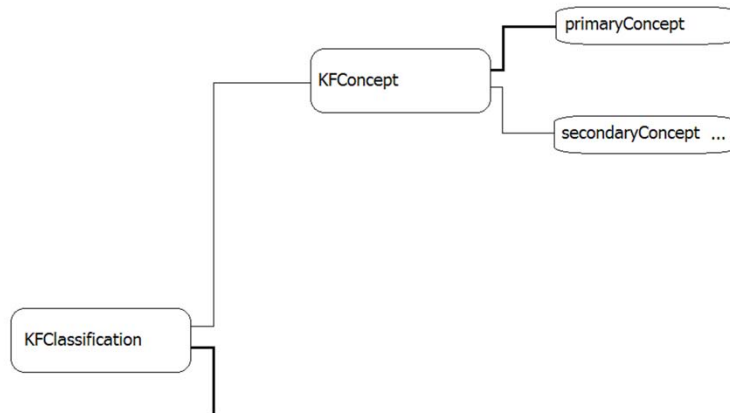The main part of the schema refers to a couple of XML "complex types" … this one handles the KF Modified concepts …
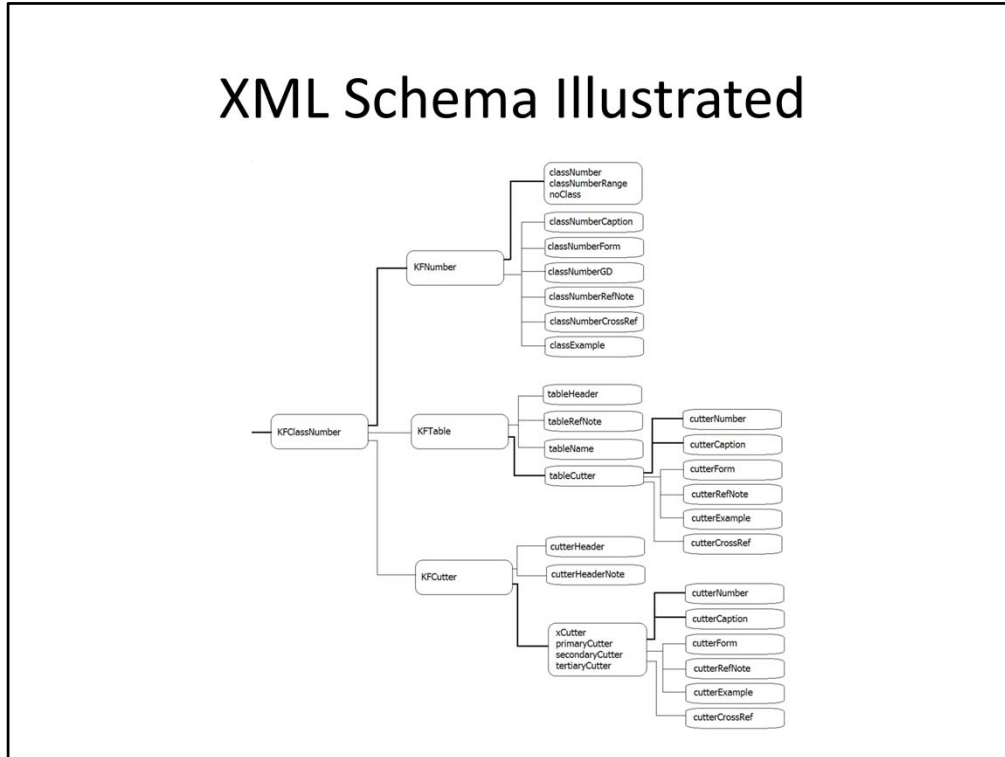
... and these the various components of the KF Modified classification number.

The schema can be illustrated something like this.

With the relatively simple KF Modified concepts …

XML Schema Illustrated

… and the more detailed KF Classification Number that consists of combinations of the Class number, Tables and Cutter numbers.

Project Phases

- develop XML schema appropriate for KF Modified
- **convert print KF Modified to XML**
- consider LCC linked data service and use of SKOS
- convert XML to HTML and RDF using XSLT
- test sample MARC record set using KF Modified
- publish KF Modified as linked data

With the schema in place we were able to start converting the word processing file and coding some XML.

The original plan was to accomplish this primarily using some regular expression magic. Our first pass yielded us something like this, which was not nearly as useful as we had hoped it would be. Ultimately, the variability in the original print version meant that we had to start coding XML manually.

The blow was softened a bit by creating a collection of macros that could fairly quickly express the various XML elements, but this still meant we were only able to fully code a short excerpt of the schedule: from KF 4480 to KF 8200.

Fortunately, and gratefully, Galen Charlton at Equinox Software, the "open source in library" folks, has recently volunteered to assist us with this final phase of this conversion process and he's made some great progress on the conversion so far.

Project Phases

- develop XML schema appropriate for KF Modified
- convert print KF Modified to XML
- **consider LCC linked data service and use of SKOS**
- convert XML to HTML and RDF using XSLT
- test sample MARC record set using KF Modified
- publish KF Modified as linked data

The next thing to do, on our way toward an expression of KF Modified classification in RDF/XML, was to review the work done at the Library of Congress Classification linked data service.

LCC Linked Data Service

http://id.loc.gov/authorities/classification/K.html

This is certainly a logical place to explore but we discovered that their linked data implementation was a little more complex than we thought we would likely need. LC reference their own authority databases through MADS and MODS. We would of course like to take advantage of that work and those connections, but we thought we could do that by mapping our data elements to their use of SKOS.

Mapping XML to SKOS

XML Elements Mapped to SKOS

Elements/*Types*	Subelements	SKOS	XSLT Notes
KFModified			
KFClassification			
	KFConcept		
conceptHierarchy	primaryConcept	skos:prefLabel; skos:altLabel; rdfs:label	skos:prefLabel if below don't exist
	secondaryConcept	skos:prefLabel; skos:altLabel; rdfs:label	skos:prefLabel if below don't exist; concatenated if statement
	KFClassGroup		
	KFClassNumber		
	KFNumber		
	classNumber	rdf:Description about; skos:notation	if exists statement
classification	classNumberCaption	skos:prefLabel; skos:altLabel; rdfs:label	skos:prefLabel if below don't exist; concatenated if statement
	classNumberForm		
	classNumberGD		kfmod ns?
	classNumberRefNote	skos:scopeNote	
	classNumberCrossRef	skos:related	
	classExample	skos:example	
	classNumberRange	rdf:Description about; skos:notation	
classification	classNumberCaption	skos:prefLabel; skos:altLabel; rdfs:label	skos:prefLabel if below don't exist; concatenated if statement

So after considering the LC's implementation we began mapping our XML elements to SKOS. This is the beginning of a chart showing the first draft of the SKOS mapping.

Project Phases

- develop XML schema appropriate for KF Modified
- convert print KF Modified to XML
- consider LCC linked data service and use of SKOS
- **convert XML to HTML and RDF using XSLT**
- test sample MARC record set using KF Modified
- publish KF Modified as linked data

In order to accomplish the planned conversions, the next part of the project meant a better understanding of XSLT and what it can do. HTML is something we may offer to cataloguers as an alternative to KF Modified in print so that seemed a logical starting point.

I began working through Jeni Tennison's book, 'Beginner XSLT 2.0' a resource I'd recommend if you're also starting to learn about XSLT.

Transforming to HTML

```xml
<xsl:template match="KFClassGroup/KFClassNumber">
    <xsl:choose>
        <xsl:when test="KFNumber/classNumber">
            <p><span class="classNumber"><xsl:value-of select="KFNumber/classNumber" /></span><span class="caption"><xsl:value-of select=
            "KFNumber/classNumberCaption" />
                <xsl:if test="KFNumber/classNumberForm"> (<xsl:value-of select="KFNumber/classNumberForm" />)</xsl:if></span></p>
            <p><xsl:if test="KFNumber/classNumberRefNote"><span class="refNote"><xsl:value-of select="KFNumber/classNumberRefNote" />
            </span></xsl:if></p>
            <p><xsl:if test="KFNumber/classNumberCrossRef"><span class="crossRef"><xsl:value-of select="KFNumber/classNumberCrossRef" />
            </span></xsl:if></p>
        </xsl:when>
        <xsl:when test="KFNumber/classNumberRange">
            <p><span class="classNumber"><xsl:value-of select="KFNumber/classNumberRange" /></span><span class="caption"><xsl:value-of
            select="KFNumber/classNumberCaption" />
                <xsl:if test="KFNumber/classNumberForm"> (<xsl:value-of select="KFNumber/classNumberForm" />)</xsl:if></span></p>
            <p><xsl:if test="KFNumber/classNumberRefNote"><span class="refNote"><xsl:value-of select="KFNumber/classNumberRefNote" />
            </span></xsl:if></p>
            <p><xsl:if test="KFNumber/classNumberCrossRef"><span class="crossRef"><xsl:value-of select="KFNumber/classNumberCrossRef" />
            </span></xsl:if></p>
        </xsl:when>
        <xsl:otherwise>
            <p><xsl:if test="KFNumber/classNumberRefNote"><span class="refNote"><xsl:value-of select="KFNumber/classNumberRefNote" />
            </span></xsl:if></p>
            <p><xsl:if test="KFNumber/classNumberCrossRef"><span class="crossRef"><xsl:value-of select="KFNumber/classNumberCrossRef" />
            </span></xsl:if></p>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:if test="KFCutter">
```
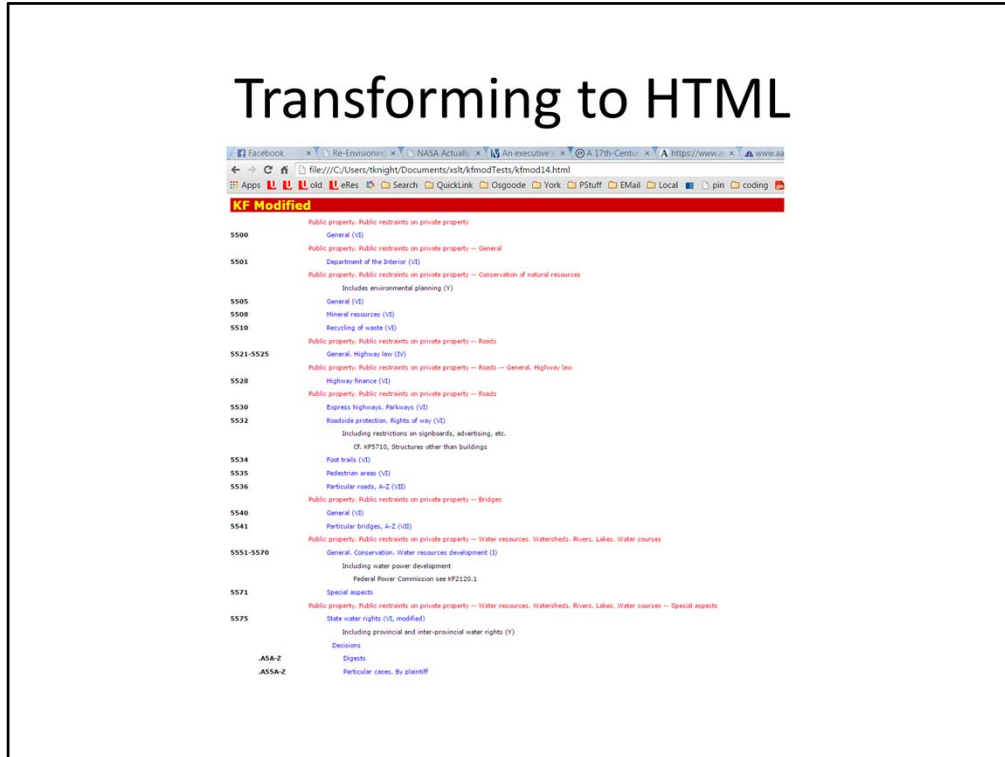
After some more trial and error and a number of coding experiments, I settled on this script that was able to transform a small excerpt of the KF Modified XML into HTML.

This is an excerpt of the XSLT. Again probably too small for you to read here.

And here's what the transformation to HTML looks like …

Transforming to SKOS

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="http://kfmod.ca/KF6617.D4">
    <skos:prefLabel xmlns:skos="http://www.w3.org/2004/02/skos/core#">Diesel fuel</skos:prefLabel>
    <skos:closeMatch> "http://id.loc.gov/authorities/classification/KF6617.D4"
    <rdfs:label xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">Public finance--National
      revenue--Particular sources of revenue--Taxation--Particular taxes--Indirect taxes. Sales or turnover taxes--Excise taxes.
      Taxes on transactions--Particular commodities, services, transactions--Motor fuels--Particular fuels--Other fuels, A-Z--Diesel
      fuel</rdfs:label>
    <skosxl:altLabel xmlns:skosxl="http://www.w3.org/2008/05/skos-xl#">
      <rdf:Description>
        <rdf:type rdf:resource="http://www.w3.org/2008/05/skos-xl#Label"/>
        <skosxl:literalForm>Public finance--National revenue--Particular sources of revenue--Taxation--Particular taxes--
          Indirect taxes. Sales or turnover taxes--Excise taxes. Taxes on transactions--Particular commodities, services,
          transactions--Motor fuels--Particular fuels--Other fuels, A-Z--Diesel fuel</skosxl:literalForm>
      </rdf:Description>
    </skosxl:altLabel>
    <skosxl:altLabel xmlns:skosxl="http://www.w3.org/2008/05/skos-xl#">
      <rdf:Description>
        <rdf:type rdf:resource="http://www.w3.org/2008/05/skos-xl#Label"/>
        <skosxl:literalForm>Diesel fuel--Taxation</skosxl:literalForm>
      </rdf:Description>
    </skosxl:altLabel>
    <skos:broader rdf:resource="http://kfmod.ca/KF6617.A-KF6617.Z" xmlns:skos="http://www.w3.org/2004/02/skos/core#"/>
    <skos:narrower rdf:resource="http://kfmod.ca/KF6617.D4A515-KF6617.D4A52" xmlns:skos="http://www.w3.org/2004/02/skos/core#"/>
    <skos:narrower rdf:resource="http://kfmod.ca/KF6617.D4A7-KF6617.D4Z9" xmlns:skos="http://www.w3.org/2004/02/skos/core#"/>
    <skos:narrower rdf:resource="http://kfmod.ca/KF6617.D4A29-KF6617.D4A3" xmlns:skos="http://www.w3.org/2004/02/skos/core#"/>
    <skos:notation xmlns:skos="http://www.w3.org/2004/02/skos/core#">KF6617.D4</skos:notation>
    <skos:inScheme rdf:resource="http://kfmod.ca" xmlns:skos="http://www.w3.org/2004/02/skos/core#"/>
    <skos:altLabel xmlns:skos="http://www.w3.org/2004/02/skos/core#">Public finance--National revenue--Particular sources of
      revenue--Taxation--Particular taxes--Indirect taxes. Sales or turnover taxes--Excise taxes. Taxes on transactions--Particular
      commodities, services, transactions--Motor fuels--Particular fuels--Other fuels, A-Z--Diesel fuel</skos:altLabel>
    <skos:altLabel xmlns:skos="http://www.w3.org/2004/02/skos/core#">Diesel fuel--Taxation</skos:altLabel>
  </rdf:Description>
</rdf:RDF>
```

Getting to something like this RDF/XML representation with SKOS is proving to be again more of a challenge.

I've had some success using scripts that Eric Hanson posted in a project he was involved with at North Carolina State University Libraries. They've made their project files available and I'd recommend his article, "A Beginner's Guide to Creating Library Linked Data" that outlines the steps they used to create their organization name linked data project.

Project Phases

- develop XML schema appropriate for KF Modified
- convert print KF Modified to XML
- consider LCC linked data service and use of SKOS
- convert XML to HTML and RDF using XSLT
- **test sample MARC record set using KF Modified**
- publish KF Modified as linked data

In anticipation of the test reconciliation against the KF Modified linked data service a MARC record set of about 100 records has been extracted.

Converting MARC to RDF

Step 1: extract MARC records from ILS
Step 2: clean and remove unwanted local tags, e.g. XX9, 596, etc.
Step 3: convert to MarcXML using MarcEdit*
Step 4: retrieve LC XSLT style sheets from <http://www.loc.gov/standards/marcxml/>
    You'll need: **MARC21slim2RDFDC.xsl** which also requires **MARC21slimUtils.xls**
        <http://www.loc.gov/standards/marcxml/xslt/MARC21slim2RDFDC.xsl>
        <http://www.loc.gov/standards/marcxml/xslt/MARC21slimUtils.xsl>
Step 5: put all files into the same directory (note assuming use of Ubuntu here)
Step 6: process file using style sheets with one of the following commands:

```
        xsltproc MARC21slim2RDFDC.xsl conversionTestFile-cleaned.xml | cat > test.rdf

        xsltproc MARC21slim2RDFDC.xsl conversionTestFile-cleaned.xml -o test-file-1.rdf
```

* probably a good idea to run this file through an XML validator, e.g.
<http://www.w3schools.com/xml/xml_validator.asp>

Part of that process includes the conversion of the MARC records to RDF.

This process uses MarcEdit to convert to MarcXML and a couple of stylesheets available from the Library of Congress.

Project Phases

- develop XML schema appropriate for KF Modified
- convert print KF Modified to XML [partial]
- consider LCC linked data service and use of SKOS
- convert XML to HTML and RDF using XSLT
- test sample MARC record set using KF Modified
- publish KF Modified as linked data

This has been our progress to date. We'd hoped to be a little further along with the project before this conference rolled around, but we hope sharing these experience has been useful.

We still have a ways to go. We need to complete conversion to XML and then work out the best way to convert to RDF/XML using SKOS. Then we need to set the classification scheme up as a linked data service that we can reconcile our data against.

Selected References

- **Allen, Layman E.** "Logic, Law and Dreams." *Law Library Journal* 52 (1959): 131.
- **Hanson, Eric M.** "A Beginner's Guide to Creating Library Linked Data: Lessons from NCSU's Organization Name Linked Data Project." *Serials Review* 40.4 (2014): 251–258.
- **Knight, F. Tim.** The KF Modified Blog https://kfmod.wordpress.com/
- **Library of Congress.** LCC Class K
- http://id.loc.gov/authorities/classification/K.html
- **NCSU Libraries.** Organization Name Linked Data https://www.lib.ncsu.edu/ld/onld/
- **Tennison, Jeni.** Beginning XSLT 2.0: From Novice to Professional. Berkeley, CA: Apress, 2005.

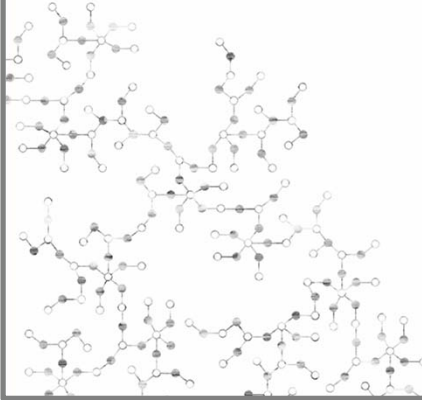Here are some references …

Thanks for your interest!

F. Tim Knight
Osgoode Hall Law School Library, York University
tknight@osgoode.yorku.ca   @freemoth

Sarah Sutherland
Canadian Legal Information Institute (CanLII)
ssutherland@canlii.org   @parallaxinfo

I'll post these slides on the KF Modified blog.

And I'll had this over to Christina …