

# CanCore $\Leftrightarrow$ WOO RuleML

David Hirtle and Zhong Wei Sun

Faculty of Computer Science  
University of New Brunswick  
Fredericton, NB, Canada  
*David.Hirtle@unb.ca, e28a1@unb.ca*

## Abstract

This paper presents a sample translation addressing the diversity of existing learning object metadata standards. Specifically, the bidirectional translation is between CanCore metadata and Weighted Object-Oriented RuleML, and is accomplished using XSLT. The overall aim of these translators is to demonstrate that interoperability is not precluded by lack of consensus among groups interested in LOM, and that, in fact, a representation of these metadata in WOO RuleML has additional benefits arising from Semantic Web tools.

## 1 Introduction

Metadata is an integral part of the next generation of the World Wide Web, the Semantic Web [BLHL01], and e-Learning is a major field of interest with respect to both the current web and its (envisioned) semantically-enhanced successor. A natural step forward for e-Learning is to combine metadata and learning objects (LOs), the resources required for online learning. This can only be successful on a large scale with a consistent metadata framework: indexing guidelines must be established and adhered to. The current XML-based standard, referred to as the Learning Object Metadata (LOM) standard [IEEE02], is not viewed as satisfactory by all interested parties. As a result, numerous organizations worldwide have developed their own specifications or simply modified existing ones.

A Canadian attempt to address the perceived problems has resulted in a set of recommendations known as CanCore [FFR03], a simplified yet detailed subset of the LOM standard aimed at promoting interoperability. Numerous LO repositories (LORs) are already being developed based (more or less) on these guidelines, the most local of which is known as CanLOM or knowledgeAGORA<sup>1</sup> [<http://www.knowledgeagora.com>]. These Canada-wide LORs have been networked as part of the eduSource project [<http://www.edusource.ca>].

---

<sup>1</sup>knowledgeAGORA is actually an e-marketplace for LOs, but it can also be considered a repository.

The variation among LOM efforts undermines the usefulness of the initiative as a whole. No consensus has been reached (or is envisioned in the near future) concerning an optimal metadata specification. This means that LOs will be indexed in different ways by different groups, and that large-scale consistency cannot be attained. But interoperability is still possible: translation among the diverse specifications aids in reconciling these differences.

Direct translation between specifications (CanCore, Dublin Core, uklomcore, etc.) would require many translators, but an intermediate format can drastically reduce this number. Weighted Object-Oriented RuleML [Bol03], a shared rule markup language, is particularly well suited for this role not only because of its flexibility, but also because of the advantages which accompany representation within a rule base:

1. Support for weights, facilitating agent matching.
2. Access to various tools such as the Weighted-Tree Similarity Algorithm [BBY03].
3. Derivations, transformations, and other general rule-related operations are possible in the same formalism.

As a neutral interchange format, Weighted Object-Oriented RuleML also facilitates the translation of this metadata to (and between) various other Semantic Web representations such as RDF, OWL and N3.

For this purpose, a bidirectional translation between CanCore guidelines and WOO RuleML has been designed and implemented. The mapping is complex but is well within the expressive power of XSLT (XSL Transformations), the language used for the translation. The WOO RuleML versions of CanCore metadata constitute generic reconstructions of non-generic markup, removing all domain-dependent vocabulary from the markup: hence their size increases. The mapping will be detailed (section 2), followed by an in-depth explanation of each translator's functionality (section 3 and section 4).

## 2 Mapping

The process of unifying WOO RuleML and CanCore was difficult because of several noteworthy challenges: discrepancies, datatypes, orderedness and weights.

### 2.1 Discrepancies

The version 1.9 CanCore guidelines [<http://www.cancore.ca/documents.html>] obviously represent a lot of work, but some details are still ambiguous. For instance, the `intendedEndUserRole` element is not indicated as being ordered, but its explanation suggests otherwise: "Principal user(s) for which this learning object was designed, *most dominant first*" (emphasis added). Also, at one point in the guidelines, the `interactivityLevel` element is listed as a CanCore recommendation, but it is later called redundant and thus to be avoided.

Of even greater concern, however, are discrepancies between CanCore and the XML Schema [Fal01] referenced by LOM documents found at knowledgeAGORA

[<http://adlib.athabascau.ca/catalog/xml/LOMv1p0/lom.xsd>]. For instance, the “How Robots Work” LOM file [[http://www.knowledgeagora.com/details/test.cfm?object\\_id=77168](http://www.knowledgeagora.com/details/test.cfm?object_id=77168)] is not even valid with respect to the indicated schema. Granted, validity problems against evolving schemas are difficult to prevent, but this is not an isolated case: many of the LOM files examined were not entirely valid. The following examples are taken directly from CanCore guidelines:

```
<duration>PT1H20M25S</duration>
<duration>
  <string language=en>between 20 and 25 minutes</string>
</duration>
```

The schema requires another duration element to be nested, and string descriptions to be within a description element:

```
<duration>
  <duration>PT1H20M25S</duration>
</duration>
<duration>
  <description>
    <string language=en>between 20 and 25 minutes</string>
  </description>
</duration>
```

Also, the schema defines, but does not currently use, the vocabulary enumerations listed in the guidelines. For example,

```
<aggregationLevel>
  <source>foo</source>
  <value>bar</value>
</aggregationLevel>
```

is permissible, despite this explicit enumeration within the schema:

```
<xs:simpleType name="aggregationLevelValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="1"/>
    <xs:enumeration value="2"/>
    <xs:enumeration value="3"/>
    <xs:enumeration value="4"/>
  </xs:restriction>
</xs:simpleType>
```

In order to specify the mapping, plausible assumptions had to be made with respect to these discrepancies, where helpful hints were obtained from the Teleeducation NB group.

## 2.2 Datatypes

All non-aggregate CanCore elements conform to a particular datatype. This information survives the translation by taking advantage of term typing in WOO RuleML. The mapping assumes the default type is “CharacterString”, for which no type is specified in WOO RuleML. For example, the CanCore element `id` translates into an untyped `ind`(ividual constant) surrounded by a metarole (`_r`) named “id”:

```
<id>971</id>
```

⇔

```
<_r n="id">
  <ind>971</ind>
</_r>
```

Elements of the “LangString” datatype are equivalent to one or more elements of the CharacterString datatype with a language explicitly identified, so the language information is maintained as a type in WOO RuleML with the individual CharacterStrings being contained with a (com)plex:

```
<title>
  <string language="en">How Robots Work</string>
  <string language="fr">Comment fonctionnent les robots</string>
</title>
```

⇔

```
<_r n="title">
  <plex>
    <_r n="language">
      <ind type="en">How Robots Work</ind>
    </_r>
    <_r n="language">
      <ind type="fr">Comment fonctionnent les robots</ind>
    </_r>
  </plex>
</_r>
```

The “DateTime” and “Duration” datatypes are similar and are thus handled in a similar fashion. The date element is of type `DateTime`, whereas the `typicalLearningTime` element is of type `Duration`:

```
<date>
  <dateTime>2002</dateTime>
  <description>
    ...
```

```

    </description>
</date>

⇔

<_r n="date">
  <plex>
    <_r n="dateTime">
      <ind type="dateTime">2002</ind>
    </_r>
    <_r n="description">
      ...
    </_r>
  </plex>
</_r>

```

and

```

<typicalLearningTime>
  <duration>PT1H20M25S</duration>
  <description>
    ...
  </description>
</typicalLearningTime>

⇔

<_r n="typicalLearningTime">
  <plex>
    <_r n="duration">
      <ind type="duration">PT1H20M25S</ind>
    </_r>
    <_r n="description">
      ...
    </_r>
  </plex>
</_r>

```

As previously mentioned, the “Vocabulary” datatype is not currently enforced (though it is implied), so no explicit type is used when translating to WOO RuleML.

Finally, some typing beyond CanCore recommendations is included to maximize the utility of the WOO RuleML representation. Specifically, entity elements are typed “vCard”; format elements, “mimeType”; and size elements, “nonNegativeInteger”. This additional expressivity could be useful for certain applications, and is enforced within the referenced schema (but not the guidelines document).

## 2.3 Orderedness

Of the approximately sixty CanCore-recommended elements, only the relative order of multiple occurrences of the following elements is significant: `contribute`, `entity`, `location`, `learningResourceType`, `intendedEndUserRole`, `taxon` and `keyword`<sup>2</sup>.

Ordered elements could be represented in WOO RuleML in many different ways. The method selected, however, is similar to how regular non-ordered CanCore elements are treated, resulting in a consistent overall mapping: an artificial container metarole with the postfix “\_list” is created, in which are found all instances of the ordered element, each postfixed with an underscore followed by its left-to-right (top down) position. This explicit positionalizing maintains the meaningful order of these elements. This particular strategy also facilitates weighting, since only metaroles can be weighted.

For example, two instances of the ordered entity element are translated as follows:

```
<entity>...1...</entity>
<entity>...2...</entity>

⇔

<_r n="entity_list">
  <plex>
    <_r n="entity_1">
      <ind type="vCard">...1...</ind>
    </_r>
    <_r n="entity_2">
      <ind type="vCard">...2...</ind>
    </_r>
  </plex>
</_r>
```

## 2.4 Weights

A WOO RuleML version of a CanCore element can be assigned importance relative to its siblings, commonly referred to as “weight”. This is accomplished by assigning a number between 0 and 1 to the `w` attribute of the role (`_r`) in WOO RuleML which corresponds to the actual CanCore element. The mapping gives an equal weight to all elements on a given level by default (except when order is significant), which can be later adjusted based on user preferences, for example. This creates a lot of possibilities in agent-matching applications.

In general, any given role’s weight (corresponding to a CanCore element’s weight) is equal to the reciprocal of the total number of children of the element’s parent (in other words, the number of siblings). For example, the two children of this description element each receive a weight of  $1/2 = 0.5$ :

---

<sup>2</sup>Only keyword elements which are children of classification placeholder element are ordered.

```

<description>
  <string language="en">A robot and a human being...</string>
  <string language="fr">Un robot et un etre humain...</string>
</description>

```

⇔

```

<_r n="description" w="...">
  <plex>
    <_r n="language" w="0.5">
      <ind type="en">A robot and a human being...</ind>
    </_r>
    <_r n="language" w="0.5">
      <ind type="fr">Un robot et un etre humain...</ind>
    </_r>
  </plex>
</_r>

```

If the description were specified in three languages, the weights would become  $1/3 = 0.(3)$ :

```

<description>
  <string language="en">A robot and a human being...</string>
  <string language="fr">Un robot et un etre humain...</string>
  <string language="es">Una robusteza y un humano...</string>
</description>

```

⇔

```

<_r n="description" w="...">
  <plex>
    <_r n="language" w="0.33333">
      <ind type="en">A robot and a human being...</ind>
    </_r>
    <_r n="language" w="0.33333">
      <ind type="fr">Un robot et un etre humain...</ind>
    </_r>
    <_r n="language" w="0.33334">
      <ind type="es">Una robusteza y un humano...</ind>
    </_r>
  </plex>
</_r>

```

For ordered cases, the calculations are more complicated. Weights for aggregate elements (postfixed with “\_list”) are based on the number of instances of the ordered element. For example, for a lifeCycle element with 1 (unordered) version and 2 (ordered) contribute elements, the weight for contribute\_list would be equal to

the number contribute elements (2) multiplied by the base weight of  $1/3 = 0.(3)$ , yielding  $2 * 0.(3) = 0.(6)$ :

```

<lifeCycle>
  <version>...</version>
  <contribute>...</contribute>
  <contribute>...</contribute>
</lifeCycle>

⇔

<_r n="lifeCycle" w="...">
  <plex>
    <_r n="version" w="0.33333">...</_r>
    <_r n="contribute_list" w="0.66667">
      <plex>
        <_r n="contribute_1" w="...">...</_r>
        <_r n="contribute_2" w="...">...</_r>
      </plex>
    </_r>
  </plex>
</_r>

```

The weights of individual instances of ordered elements are equal to the reciprocal of  $\text{sum}(1, \dots, n)$  (where  $n$  = number of siblings) multiplied by the element's right-to-left (bottom up) position. For example, since there are two contribute elements from the previous example, the weight for the first is  $1/\text{sum}(1,2) * (\text{r-l position}) = 1/3 * 2 = 0.(6)$ , whereas the weight for the second is  $1/\text{sum}(1,2) * (\text{r-l position}) = 1/3 * 1 = 0.(3)$ :

```

<_r n="contribute_list" w="0.66667">
  <plex>
    <_r n="contribute_1" w="0.66667">...</_r>
    <_r n="contribute_2" w="0.33333">...</_r>
  </plex>
</_r>

```

If, however, another contribute element were added, the weights would all shift:

```

<lifeCycle>
  <version> ... </version>
  <contribute> ... </contribute>
  <contribute> ... </contribute>
  <contribute> ... </contribute>
</lifeCycle>

⇔

```

```

<_r n="contribute_list" w="0.75">
  <plex>
    <_r n="contribute_1" w="0.5"> ... </_r>
    <_r n="contribute_2" w="0.33333"> ... </_r>
    <_r n="contribute_2" w="0.16667"> ... </_r>
  </plex>
</_r>

```

In this manner, the priority implicit within ordered elements is reinforced by assigning equidistant weights to the roles.

### 3 Translation from CanCore to WOO RuleML

With the mapping established, the actual transformation can occur. A useful tool for performing these transformations (and many other tasks related to XML) is XMLSPY [<http://www.altova.com/xmlspy>].

The language used, XML Stylesheet Language Transformations (XSLT) [Cla99], is based on templates which detail precisely how to handle node sets encountered while traversing an XML document structure. For example, the following template matches the document root, represented in XPath (as with traditional file paths) as “/”:

```

<xsl:template match="/">
  <rulebase>
    <fact>
      <_head>
        <atom>
          <_opr>
            <rel>lom</rel>
          </_opr>
          <xsl:apply-templates/>
        </atom>
      </_head>
    </fact>
  </rulebase>
</xsl:template>

```

This template indicates that the root is to be replaced by a rulebase element and its (deeply-nested) children, where templates matching subsequent nodes are activated by the line `<xsl:apply-templates/>`. The result of this template being applied to the abbreviated CanCore document

```

<lom>
  ...
</lom>

```

is the following deeply-nested WOO RuleML:

```

<rulebase>
  <fact>
    <_head>
      <atom>
        <_opr>
          <rel>lom</rel>
        </_opr>
        ...
      </atom>
    </_head>
  </fact>
</rulebase>

```

Templates can also be specified to match explicitly named elements:

```

<xsl:template match="general">
  <_r n="general" w="{1 div count(../*)}">
    <plex>
      <xsl:apply-templates/>
    </plex>
  </_r>
</xsl:template>

```

In this case, the `general` element is transformed into a roled `plex` with a weight of 1 divided by the number of its parent's (`..`) children (`/*`). Thus the CanCore document

```

<lom>
  <general>
    ...
  </general>
</lom>

```

expands to the following WOO RuleML:

```

<rulebase>
  <fact>
    <_head>
      <atom>
        <_opr>
          <rel>lom</rel>
        </_opr>
        <_r n="general" w="1">
          <plex>
            ...
          </plex>
        </_r>
      </atom>
    </_head>
  </fact>
</rulebase>

```

```

    </_head>
  </fact>
</rulebase>

```

Based on such template matching, complex transformations quickly develop:

```

<lom>
  <general>
    <identifier>
      <catalog>URI</catalog>
      <entry>http://www.robots.com</entry>
    </identifier>
    <title>
      <string language="en">How Robots Work</string>
    </title>
    <language>en</language>
    <keyword>
      <string language="x-none">Robotics</string>
    </keyword>
    <aggregationLevel>
      <source>LOMv1.0</source>
      <value>3</value>
    </aggregationLevel>
  </general>
  ...
</lom>

```

⇔

```

<rulebase>
  <fact>
    <_head>
      <atom>
        <_opr>
          <rel>lom</rel>
        </_opr>
        <_r n="general" w="...">
          <plex>
            <_r n="identifier" w="0.2">
              <plex>
                <_r n="catalog" w="0.5">
                  <ind>URI</ind>
                </_r>
                <_r n="entry" w="0.5">
                  <ind href="www.robots.com"/>
                </_r>
              </plex>
            </_r>
          </plex>
        </_r>
      </atom>
    </_head>
  </fact>
</rulebase>

```

```

        </plex>
    </_r>
    <r n="title" w="0.2">
        <plex>
            <r n="language" w="1">
                <ind type="en">How Robots Work</ind>
            </_r>
        </plex>
    </_r>
    <r n="languageOrNone" w="0.2">
        <ind type="en"/>
    </_r>
    <r n="keyword" w="0.2">
        <plex>
            <r n="language" w="1">
                <ind type="x-none">Robotics</ind>
            </_r>
        </plex>
    </_r>
    <r n="aggregationLevel" w="0.2">
        <plex>
            <r n="source" w="0.5">
                <ind>LOMv1.0</ind>
            </_r>
            <r n="value" w="0.5">
                <ind>3</ind>
            </_r>
        </plex>
    </_r>
</plex>
</_r>
...
</atom>
</_head>
</fact>
</rulebase>

```

Note that in the above situation, the five children of the general element (identifier, title, languageOrNone, keyword and aggregationLevel) each have a weight of  $1/5 = 0.2$ . Also note that this transformation is still highly abbreviated: the CanCore recommendation consists of 8 additional top-level elements. The transformation results of an entire CanCore file are included in the appendix.

To ensure that the translated result is in valid WOO RuleML, a validator from Brown University [<http://www.stg.brown.edu/service/xmlvalid>] is used. Only the following declaration need be added to the output:

```
<!DOCTYPE rulebase
```

```
SYSTEM "http://www.ruleml.org/dtd/0.85/urhornlog.dtd">
```

The increased complexity of this transformation from CanCore to WOO RuleML is a disadvantage of this representation (caused by the universality of WOO RuleML), but is more than offset by the advantages. For example, whole LORs could be updated at once using WOO RuleML rules (e.g. if a URL changes).

Another difficulty is the loss of namespace and XML Schema information during the translation to WOO RuleML, but this issue could be overcome in the future simply by embedding these details within a rule label.

## 4 Translation from WOO RuleML to CanCore

Translation of a WOO RuleML representation of LOM back to CanCore requires templates which perform the inverse of the previously discussed templates. For example, the template

```
<xsl:template match="/">
  <lom>
    <xsl:apply-templates select="//_r[@n='general']"/>
    <xsl:apply-templates select="//_r[@n='lifeCycle']"/>
    <xsl:apply-templates select="//_r[@n='metaMetadata']"/>
    <xsl:apply-templates select="//_r[@n='technical']"/>
    <xsl:apply-templates select="//_r[@n='educational']"/>
    <xsl:apply-templates select="//_r[@n='rights']"/>
    <xsl:apply-templates select="//_r[@n='relation']"/>
    <xsl:apply-templates select="//_r[@n='annotation']"/>
    <xsl:apply-templates select="//_r[@n='classification']"/>
  </lom>
</xsl:template>
```

takes the WOO RuleML

```
<rulebase>
  <fact>
    <_head>
      <atom>
        <_opr>
          <rel>lom</rel>
        </_opr>
        ...
      </atom>
    </_head>
  </fact>
</rulebase>
```

and condenses it into the following minimal CanCore file:

```
<lom>
  ...
</lom>
```

Similarly, the general element template

```
<xsl:template match="/rulebase/fact/_head/atom/_r[@n='general']">
  <general>
    <xsl:apply-templates select="plex/_r[@n='identifier']"/>
    <xsl:apply-templates select="plex/_r[@n='title']"/>
    <xsl:apply-templates select="plex/_r[@n='languageOrNone']"/>
    <xsl:apply-templates select="plex/_r[@n='description']"/>
    <xsl:apply-templates select="plex/_r[@n='keyword']"/>
    <xsl:apply-templates select="plex/_r[@n='aggregationLevel']"/>
  </general>
</xsl:template>
```

takes the WOO RuleML representation

```
<rulebase>
  <fact>
    <_head>
      <atom>
        <_opr>
          <rel>lom</rel>
        </_opr>
        <_r n="general" w="1">
          <plex>
            ...
          </plex>
        </_r>
      </atom>
    </_head>
  </fact>
</rulebase>
```

and reduces it to:

```
<lom>
  <general>
    ...
  </general>
</lom>
```

Its quite evident that WOO RuleML relies rather heavily on the use of attribute values, especially `_r`'s `n(ame)` attribute, which usually corresponds to the element's name.

XPath has a simple method of accessing attribute information: the @ character. For example, the following template, matching only roles named “language”, creates a string element with a language attribute equal to the value of ind’s type attribute:

```
<xsl:template match="plex/_r[@n='language']">
  <string language="{ind/@type}">
    <xsl:value-of select="ind"/>
  </string>
</xsl:template>
```

Thus,

```
<_r n="language" w="1">
  <ind type="en">How Robots Work</ind>
</_r>
```

becomes:

```
<string language="en">How Robots Work</string>
```

As can be seen in the previous two examples, weights are lost during the transformation from WOO RuleML back to a LOM file, because CanCore does not support such functionality.

Again, these are relatively small examples. The full results of a detailed transformation are included in the appendix.

## 5 Conclusion

The translation between CanCore metadata and WOO RuleML that has been designed and implemented is a step forward in achieving interoperability in the field of learning object metadata, where standards all too often diverge. The aim of this tool is to reconcile these differences using XSLT for inter-translation. A neutral interchange language greatly facilitates this process.

WOO RuleML is particularly well-suited for the role of intermediate format because of the potential related to representation within a rulebase. There are several disadvantages to take into consideration as well. First of all, the resulting WOO RuleML representation is verbose because of the language’s universal nature. This is unfortunate, but almost unavoidable like in other OO representation languages such as RDF. Another issue encountered is the loss of namespace and schema information, but this can be resolved simply by using a rule label in a future revision.

The current implementation of this bidirectional translation is not flawless, but it succeeds in its purpose quite well. First of all, the mapping is still subject to debate; it could be improved upon in the future, especially pending collaboration with those involved in the CanCore standard. An “optimal” mapping is difficult to envision, especially considering the flexibility of WOO RuleML. Optimization of the code itself would also be worthwhile, especially with respect to the handling of ordered elements. The additional features of the upcoming XSLT version 2.0 [[http:](http://)

[//www.w3.org/TR/xslt20](http://www.w3.org/TR/xslt20)] would be useful in this respect. Ideally, everyone would someday agree on a single standard and adhere to it unerringly, obviating such translations.

The problem is a large one, but this is a solid first step, assuming that additional translations are specified between WOO RuleML and other LOM standards. Currently, no other inter-standard translators are known to exist for LOM.

The translation could also have been implemented in Perl [<http://www.perl.com>], which uses powerful regular expressions capable of similar functionality. However, Perl would treat the documents as unstructured text, so some details implicitly taken care of by XSLT would have to be expressly handled, resulting in additional processing. It would be interesting to compare the translation speeds of Perl and XSLT. However, since the aim of this project is to promote standardization, the irony of ignoring a W3C standard for its implementation would be inescapable.

There is a lot of work underway in the field of metadata, of which LO metadata is only a subset. Future work could be based on developments in the area, and on those of the Semantic Web as a whole. The progress of the Rule Markup Initiative is also very important: new tools and rule applications will likely prove useful for metadata. Several possible improvements have already been mentioned, but even simple maintenance will also be required based on future versions of CanCore guidelines. Also, an XML Schema specific to CanCore should be created in the near future, so that the LOM from knowledgeAGORA is at least valid.

## References

- [BBY03] V. C. Bhavsar, H. Boley, and L. Yang. A Weighted-Tree Similarity Algorithm for Multi-Agent Systems in e-Business Environments. In Proc. Business Agents and the Semantic Web (BASeWEB) Workshop, pages 5372. NRC 45836, June 2003.
- [BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web: A New Form of Web Content that is Meaningful to Computers and Will Unleash a Revolution of New Possibilities. *Scientific American*, 284(5):34-43, May 2001.
- [Bol03] H. Boley. Object-Oriented RuleML: User-Level Roles, URI Grounded Clauses, and Order-Sorted Terms. In Proc. Rules and Rule Markup Languages for the SemanticWeb (RuleML-2003). Sanibel Island, Florida, LNCS 2876, Springer-Verlag, October 2003.
- [Cla99] J. Clark. XSL Transformations (XSLT) Version 1.0. Recommendation REC-xslt-19991116, W3C, November 1999.
- [Fal01] D. C. Fallside. XML Schema Part 0: Primer. Recommendation REC-xmlschema-0-2010502, W3C, May 2001.
- [FFR03] N. Friesen, S. Fisher, and A. Roberts. CanCore Guidelines for the Implementation of Learning Object Metadata. Athabasca University, 2003.

[IEEE02] Learning Technology Standards Committee. Draft Standard for Learning Object Metadata. IEEE 1484.12.1-2002. [<http://ltsc.ieee.org/doc/wg12/LOM.1484.12.1.v1.Final.Draft.pdf>]

## Appendix

This appendix consists of four parts: The first part A.1 includes the original CanCore document. CanCore to WOO RuleML Translator and WOO RuleML to CanCore Translator are contained in the second part A.2 and the third part A.3 respectively. The fourth part A.4 comprises the final CanCore document.

### A.1 Original CanCore Document [`cancore_robot_example.xml`]

```
<lom xmlns="http://ltsc.ieee.org/xsd/LOMv1p0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ltsc.ieee.org/xsd/LOMv1p0
http://adlib.athabascau.ca/catalog/xml/LOMv1p0/lom.xsd"
>
  <general>
    <identifier>
      <catalog>URI</catalog>
      <entry>http://www.howstuffworks.com/robot.htm</entry>
    </identifier>
    <title>
      <string language="en">How Robots Work</string>
      <string language="fr">Comment fonctionnent les robots</string>
    </title>
    <language>en</language>
    <language>fr</language>
    <description>
      <string language="en">A robot and a human being ... </string>
      <string language="fr">Un robot et un etre humain ... </string>
    </description>
    <keyword>
      <string language="x-none">Robotics</string>
    </keyword>
    <keyword>
      <string language="x-none">Artificial intelligence</string>
    </keyword>
    <keyword>
      <string language="x-none">Electronic control</string>
    </keyword>
    <aggregationLevel>
      <source>LOMv1.0</source>
      <value>3</value>
    </aggregationLevel>
  </general>
</lom>
```

```

    </aggregationLevel>
</general>
<lifeCycle>
  <version>
    <string language="en">limited edition</string>
    <string language="fr">edition limitee</string>
  </version>
  <contribute>
    <role>
      <source>CanCore v1.9</source>
      <value>Content Provider</value>
    </role>
    <entity>BEGIN: vCard ORG: HowStuffWorks.com END: vCard</entity>
    <entity>BEGIN: vCard ORG: Answers.com END: vCard</entity>
    <date>
      <dateTime>2002</dateTime>
      <description>
        <string language="en">second quarter of 2002</string>
        <string language="fr">le deuxieme quart de 2002</string>
      </description>
    </date>
  </contribute>
  <contribute>
    <role>
      <source>CanCore v1.9</source>
      <value>Graphic Designer</value>
    </role>
    <entity>
      BEGIN: vCard ORG: GraphicDesignGuys.com END: vCard
    </entity>
    <date>
      <dateTime>2002</dateTime>
      <description>
        <string language="en">second quarter of 2002</string>
        <string language="fr">le deuxieme quart de 2002</string>
      </description>
    </date>
  </contribute>
  <contribute>
    <date>
      <dateTime>2010</dateTime>
    </date>
  </contribute>
  <contribute>
    <date>
      <dateTime>2011</dateTime>
    </date>
  </contribute>

```

```

        </date>
    </contribute>
<contribute>
    <date>
        <dateTime>2012</dateTime>
    </date>
</contribute>
</lifeCycle>
<metaMetadata>
    <identifier>
        <catalog>CanLOM</catalog>
        <entry>77168</entry>
    </identifier>
    <contribute>
        <role>
            <source>CanCore v1.9</source>
            <value>Creator</value>
        </role>
        <entity>BEGIN: vCard ORG: CanLOM END: vCard</entity>
        <entity>BEGIN: vCard ORG: LOM END: vCard</entity>
        <date>
            <dateTime>2002</dateTime>
            <description>
                <string language="en">second quarter of 2002</string>
                <string language="fr">le deuxieme quart de 2002</string>
            </description>
        </date>
    </contribute>
<contribute>
    <role>
        <source>LOMv1.0</source>
        <value>validator</value>
    </role>
    <entity>BEGIN:VCARD\nVERSION:3.0\nN:Smith;FN:Mary
        \nORG:;\nEND:VCARD\n
    </entity>
    <date>
        <dateTime>2003-11-03</dateTime>
    </date>
</contribute>
<metadataSchema>CanCore v1.9</metadataSchema>
<metadataSchema>LOMv1.0</metadataSchema>
<language>en</language>
</metaMetadata>
<technical>
    <format>HTML</format>

```

```

<format>application/zip</format>
<size>430024</size>
<location>http://www.howstuffworks.com/robot.htm</location>
<location>http://sdf943892.onlinehome.us/robot</location>
<otherPlatformRequirements>
  <string language="en">HARDWARE: Internet-enabled PC or Mac.
    SOFTWARE: Netscape Navigator or Internet Explorer and email.
  </string>
</otherPlatformRequirements>
<duration>
  <duration>PT1H20M25S</duration>
  <description>
    <string language="en">depending on playback speed,
      between 20 and 25 minutes
    </string>
  </description>
</duration>
</technical>
<educational>
  <learningResourceType>
    <source>LOMv1.0</source>
    <value>Exercise</value>
  </learningResourceType>
  <intendedEndUserRole>
    <source>CanCore v1.9</source>
    <value>Learner</value>
  </intendedEndUserRole>
  <context>
    <source>CanCore v1.9</source>
    <value>Elementary Grade 8</value>
  </context>
  <typicalAgeRange>
    <string language="x-none">18-25</string>
  </typicalAgeRange>
  <typicalLearningTime>
    <duration>PT1H20M25S</duration>
    <description>
      <string language="en">For average students, this unit
        may take one full day of class time including a few
        hours of homework. Advanced students may need only
        a half-day in total.
      </string>
    </description>
  </typicalLearningTime>
  <language>en</language>
</educational>

```

```

<rights>
  <cost>
    <source>LOMv1.0</source>
    <value>No</value>
  </cost>
  <copyrightAndOtherRestrictions>
    <source>LOMv1.0</source>
    <value>Yes</value>
  </copyrightAndOtherRestrictions>
  <description>
    <string language="en">Some restrictions may apply.
      Contact publisher for details.
    </string>
  </description>
</rights>
<relation>
  <kind>
    <source>LOMv1.0</source>
    <value>references</value>
  </kind>
  <resource>
    <identifier>
      <catalog>ISSN</catalog>
      <entry>1191-8276</entry>
    </identifier>
  </resource>
</relation>
<annotation>
  <entity>BEGIN:VCARD\nVERSION:3.0\n Tremblay;Andrea.
    \nFN:Andrea Tremblay\nEND:VCARD
  </entity>
  <date>
    <dateTime>2003-04</dateTime>
  </date>
  <description>
    <string language="en">
      I spent a number of hours looking over and reviewing
      the information provided on this site. I found that
      the information provided would be helpful to both
      students and teachers. As an educator the activities
      and lesson ideas were very appealing. Children would
      be drawn in and would gain tremendous knowledge about
      the nervous system by it.
    </string>
  </description>
</annotation>

```

```

<classification>
  <purpose>
    <source>CanCore v1.9</source>
    <value>Discipline</value>
  </purpose>
  <taxonPath>
    <source>
      <string language="en">ERIC</string>
    </source>
    <taxon>
      <id>971</id>
      <entry>
        <string language="en">Computer Science</string>
      </entry>
    </taxon>
    <taxon>
      <id>972</id>
      <entry>
        <string language="en">Computers</string>
      </entry>
    </taxon>
    <taxon>
      <id>973</id>
      <entry>
        <string language="en">Internet</string>
      </entry>
    </taxon>
    <taxon>
      <id>974</id>
      <entry>
        <string language="en">Applied Sciences</string>
      </entry>
    </taxon>
    <taxon>
      <id>975</id>
      <entry>
        <string language="en">Technology</string>
      </entry>
    </taxon>
  </taxonPath>
  <keyword>
    <string language="en">educational technology</string>
  </keyword>
  <keyword>
    <string language="en">computer based instruction</string>
  </keyword>

```

```

    <keyword>
      <string language="en">computer</string>
    </keyword>
    <keyword>
      <string language="en">instruction</string>
    </keyword>
  </classification>
</classification>
  <purpose>
    <source>CanCore v1.9</source>
    <value>PedagogicType</value>
  </purpose>
  <taxonPath>
    <source>
      <string language="en">CanCore v1.9</string>
    </source>
    <taxon>
      <entry>
        <string language="en">Lesson</string>
      </entry>
    </taxon>
  </taxonPath>
</classification>
</lom>

```

## A.2 Cancore to WOO RuleML Translator [[cancore2wooruleml.xslt](#)]

```

<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:lom="http://ltsc.ieee.org/xsd/LOMv1p0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
exclude-result-prefixes="lom"
>
  <xsl:output method="xml" indent="yes"/>
  <!--
    bug: 0.06666... becomes 0.6667 using XSLT's format-number()
    with the following format string:
    <xsl:variable name="WEIGHT_FORMAT" select="'0.#####'"/>

    so, rounded manually using round(10000*x) div 100000
  -->
  <!--
    currently, use apply-templates to force order to be consistent
    for unordered elements
    e.g. identifier/catalog before identifier/entry (this isn't
    necessary and can easily be changed in the future by deleting

```

```

    select attributes ... would permit merging of many different
    templates)
-->
<!--
    currently, namespace and xml schema information is lost when
    translating to WOO RuleML this could be avoided by supplying a
    rule label
-->
<!--
    any given role's weight (corresponding to a CanCore element's
    weight) is computed by dividing 1 by the total number of siblings
    on that level (the parent's children)
    e.g. if lom has 1 of each of the 9 different basic aggregate
    elements (general, lifeCycle, etc.), then the weight of each of
    those children is 1/9 = 0.(1)
    (this is different for ordered elements - see the comments before
    the ordered template for details)
-->
<xsl:template match="/">
  <rulebase>
    <fact>
      <_head>
        <atom>
          <_opr>
            <rel>lom</rel>
          </_opr>
          <xsl:apply-templates/>
        </atom>
      </_head>
    </fact>
  </rulebase>
</xsl:template>

<!--
    matches all top-level aggregate elements:
    general (1)
    lifeCycle (2)
    metaMetadata (3)
    technical (4)
    educational (5)
    rights (6)
    relation (7)
    annotation (8)
    classification (9)
-->
<xsl:template match="/lom:lom/*">

```

```

    <_r n="{name(.)}" w="{round(100000*(1 div count(../*))) div 100000}">
      <plex>
        <xsl:apply-templates/>
      </plex>
    </_r>
  </xsl:template>
  <!--
    matches:
    general/identifier (1.1)
    metaMetadata/identifier (3.1)
    relation/resource/identifier (7.2.1)
  -->
  <xsl:template match="lom:identifier">
    <_r n="identifier" w="{round(100000*(1 div count(../*))) div 100000}">
      <plex>
        <xsl:apply-templates select="lom:catalog"/>
        <xsl:apply-templates select="lom:entry"/>
      </plex>
    </_r>
  </xsl:template>
  <!--
    matches:
    general/aggregationLevel (1.8)
    //contribute/role (2.3.1, 3.2.1)
    educational/context (5.6)
    rights/cost (6.1)
    rights/copyrightAndOtherRestrictions (6.2)
    relation/kind (7.1)
    classification/purpose (9.1)
  -->
  <xsl:template match="lom:aggregationLevel | lom:role | lom:context |
    lom:cost | lom:copyrightAndOtherRestrictions | lom:kind | lom:purpose">
    <_r n="{name(.)}" w="{round(100000*(1 div count(../*))) div 100000}">
      <plex>
        <xsl:apply-templates select="lom:source"/>
        <xsl:apply-templates select="lom:value"/>
      </plex>
    </_r>
  </xsl:template>
  <!--
    note that all source elements are of type CharacterString, except
    that from classification/taxonPath/source
  -->
  <!--
    note that all entry elements are of type CharacterString, except
    that from classification/taxonPath/taxon/entry
  -->

```

```

-->
<!--
  matches elements of type LangString:
  general/title (1.2)
  //description (1.4, 2.3.3.2, 3.2.3.2, 4.7.2, 5.9.2, 8.2.2, 8.3)
  general/keyword (1.5) [except classification/keyword]
  lifeCycle/version (2.1)
  technical/otherPlatformRequirements (4.6)
  educational/typicalAgeRange (5.7)
  classification/taxonPath/source (9.2.1)
  classification/taxonPath/taxon/entry (4.2.2.2)
-->
<xsl:template match="lom:title | lom:description |
  /lom/lom/lom:general/lom:keyword | lom:version |
  lom:otherPlatformRequirements | lom:typicalAgeRange |
  lom:classification/lom:taxonPath/lom:source |
  lom:classification/lom:taxonPath/lom:taxon/lom:entry">
  <r n="{name(.)}" w="{round(100000*(1 div count(../*))) div 100000}">
    <plex>
      <xsl:apply-templates/>
    </plex>
  </r>
</xsl:template>

<!--
  matches:
  all strings found in elements of type LangString
-->
<xsl:template match="lom:string">
  <r n="language" w="{round(100000*(1 div count(../*))) div 100000}">
    <ind type="{@language}"><xsl:value-of select="."/;></ind>
  </r>
</xsl:template>

<!--
  matches elements of type CharacterString:
  //identifier/catalog (1.1.1, 3.1.1, 7.2.1.1)
  //source [except classification/taxonPath/source]
  //value
  metaMetadata/metadataSchema (3.3)
  classification/taxonPath/taxon/id (9.2.2.1)
-->
<xsl:template match="lom:catalog | lom:source | lom:value |
  lom:metadataSchema | lom:id">
  <r n="{name(.)}" w="{round(100000*(1 div count(../*))) div 100000}">
    <ind><xsl:value-of select="."/;></ind>
  </r>

```

```

</xsl:template>

<!--
  matches:
    //identifier/entry (1.1.2, 3.1.2, 7.2.1.2)
    [except classification/taxonPath/taxon/entry]
-->
<xsl:template match="lom:entry">
  <_r n="entry" w="{round(100000*(1 div count(../*))) div 100000}">
  <!--
    if catalog is URI, URL, URN or PURL, embed
    text in href attribute of ind
  -->
  <xsl:choose>
    <xsl:when test="../lom:catalog = 'URI' or ../lom:catalog = 'URL'
      or ../lom:catalog = 'URN' or ../lom:catalog = 'PURL'">
      <ind href="{.}" />
    </xsl:when>
    <xsl:otherwise>
      <ind><xsl:value-of select="." /></ind>
    </xsl:otherwise>
  </xsl:choose>
  </_r>
</xsl:template>
<!--
  matches:
    general/language (1.3)
    metaMetadata/language (3.4)
    educational/language (5.11)
-->
<xsl:template match="lom:language">
  <!--
    language elements in general can also contain "none",
    though not in metaMetadata
  -->
  <xsl:choose>
    <xsl:when test="name(..) = 'general'">
      <_r n="languageOrNone"
        w="{round(100000*(1 div count(../*))) div 100000}">
      <ind type="{.}" />
      </_r>
    </xsl:when>
    <xsl:otherwise>
      <_r n="language"
        w="{round(100000*(1 div count(../*))) div 100000}">
      <ind type="{.}" />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

```

```

        </_r>
    </xsl:otherwise>
</xsl:choose>
</xsl:template>

<!-- general template for all ordered elements -->
<!--
weights for aggregate elements (postfixed with "_list") are based
on number of occurrences of ordered element
e.g. if lifeCycle has 1 version and 2 contribute elements, then the
role contribute_list would have weight 0.(6), which is equal to the
number of occurrences of contribute multiplied by the default weight:
2 * 0.(3) = 0.(6)
-->
<!--
the priority implicit with ordered elements is given more substance
by assigning equidistant weights to the roles
i.e. n = (number of ordered elements), w = (weights for each element,
ordered left to right):
n = 1, w = 1.0
n = 2, w = 0.(6), 0.(3)
n = 3, w = 0.4(9), 0.(3), 0.1(6)

this is achieved by dividing 1 by sum(1,...,n) where n is the number
of ordered elements and then multiplying this by the right to left
position of the leaves

e.g. 3 ordered location elements, so sum(1,...,3) is 1+2+3 = 6.
Dividing 1 by this number results in 0.1(6).
This is the "base weight" which is then multiplied by the position
number, shown below:

      0
     /|\
    / | \
   /  |  \
  3  2  1

So, from left to right, the first element has a weight of
3 * 0.1(6) = 0.4(9), the second element has a weight of
2 * 0.1(6) = 0.(3), and the last element has a weight of 0.1(6)
-->
<!--
matches first child of:
lifeCycle/contribute (2.3)
metaMetadata/contribute (3.2)

```

```

//entity (2.3.2,3.2.2) [except annotation/entity (8.1)]
technical/location (4.3)
educational/learningResourceType (5.2)
educational/intendedEndUserRole (5.5)
classification/taxonPath/taxon (9.2.2)
classification/keyword (9.4) [but not general/keyword (1.8)]
-->
<xsl:template match="lom:contribute[1] | lom:entity[1] |
lom:location[1] | lom:learningResourceType[1] |
lom:intendedEndUserRole[1] | lom:taxon[1] |
/lom:lom/lom:classification/lom:keyword[1]">
  <xsl:variable name="siblings_count"
    select="count(parent::node()/child::*[name()=name(current())])"/>
  <xsl:variable name="base_weight"
    select="1 div ($siblings_count * ($siblings_count + 1) div 2)"/>

  <!-- this is the first element, so set up the container role -->
  <r n="{concat(name(), '_list')}"
    w="{round(100000 * (1 div count(../*) * $siblings_count)) div 100000}">
    <plex>
      <r n="{concat(name(), '_1')}"
        w="{round(100000 * ($siblings_count * $base_weight)) div 100000}">
        <xsl:choose>
          <xsl:when test="name()='contribute'">
            <plex>
              <xsl:apply-templates select="lom:role"/>
              <xsl:apply-templates select="lom:entity"/>
              <xsl:apply-templates select="lom:date"/>
            </plex>
          </xsl:when>
          <xsl:when test="name()='entity'">
            <ind type="vCard"><xsl:value-of select="."/></ind>
          </xsl:when>
          <xsl:when test="name()='location'">
            <ind href="{.}" />
          </xsl:when>
          <xsl:when test="name()='learningResourceType'
            or name()='intendedEndUserRole'">
            <plex>
              <xsl:apply-templates select="lom:source"/>
              <xsl:apply-templates select="lom:value"/>
            </plex>
          </xsl:when>
          <xsl:when test="name()='taxon'">
            <plex>
              <xsl:apply-templates select="lom:id"/>
            </plex>
          </xsl:when>
        </xsl:choose>
      </r>
    </plex>
  </r>

```

```

        <xsl:apply-templates select="lom:entry"/>
    </plex>
</xsl:when>
<xsl:when test="name()='keyword'">
    <plex>
        <xsl:apply-templates select="lom:string"/>
    </plex>
</xsl:when>
<xsl:otherwise/>
</xsl:choose>
</_r>
<!-- for all subsequent elements -->
<xsl:for-each select="following-sibling::*[name()=name(current())]">
    <_r n="{concat(name(), '-', position() + 1)}"
        w="{round(100000 * ((last() - position() + 1)
            * $base_weight)) div 100000}">
    <xsl:choose>
        <xsl:when test="name()='contribute'">
            <plex>
                <xsl:apply-templates select="lom:role"/>
                <xsl:apply-templates select="lom:entity"/>
                <xsl:apply-templates select="lom:date"/>
            </plex>
        </xsl:when>
        <xsl:when test="name()='entity'">
            <ind type="vCard"><xsl:value-of select="."/></ind>
        </xsl:when>
        <xsl:when test="name()='location'">
            <ind href="{.}" />
        </xsl:when>
        <xsl:when test="name()='learningResourceType'
            or name()='intendedEndUserRole'">
            <plex>
                <xsl:apply-templates select="lom:source"/>
                <xsl:apply-templates select="lom:value"/>
            </plex>
        </xsl:when>
        <xsl:when test="name()='taxon'">
            <plex>
                <xsl:apply-templates select="lom:id"/>
                <xsl:apply-templates select="lom:entry"/>
            </plex>
        </xsl:when>
        <xsl:when test="name()='keyword'">
            <plex>
                <xsl:apply-templates select="lom:string"/>

```

```

        </plex>
        </xsl:when>
        <xsl:otherwise/>
    </xsl:choose>
</_r>
</xsl:for-each>
</plex>
</_r>
</xsl:template>

<!--
do nothing because the first instance of this element
looks ahead and handles all others
-->
<!--
(this template pattern is less specific and thus of
lower priority, so the other template is matched first)
-->
<xsl:template match="lom:contribute | lom:entity | lom:location |
lom:learningResourceType | lom:intendedEndUserRole | lom:taxon |
lom:keyword"/>

<!--
matches:
annotation/entity (8.1) [not ordered as it is elsewhere]
-->
<xsl:template match="lom:annotation/lom:entity">
    <_r n="entity" w="{round(100000*(1 div count(../*))) div 100000}">
        <ind type="vCard"><xsl:value-of select="."/></ind>
    </_r>
</xsl:template>

<!--
matches elements of type dateTime:
lifeCycle/contribute/date (2.3.3)
metaMetadata/contribute/date (3.2.3)
annotation/date (8.2)
-->
<xsl:template match="lom:date">
    <_r n="{name(.)}" w="{round(100000*(1 div count(../*))) div 100000}">
        <plex>
            <xsl:apply-templates select="lom:dateTime"/>
            <xsl:apply-templates select="lom:description"/>
        </plex>
    </_r>
</xsl:template>
<!--

```

```

    matches elements of type duration:
    technical/duration (4.7)
    educational/typicalLearningTime (5.9)
-->
<xsl:template match="lom:duration | lom:typicalLearningTime">
  <r n="{name(.)}" w="{round(100000*(1 div count(../*))) div 100000}">
    <plex>
      <xsl:apply-templates select="lom:duration"/>
      <xsl:apply-templates select="lom:description"/>
    </plex>
  </r>
</xsl:template>

<!--
  matches:
  lifeCycle/contribute/date/dateTime (2.3.3.1)
  metaMetadata/contribute/date/dateTime (3.2.3.1)
  annotation/date/dateTime (8.2.1)
  technical/duration/duration (4.7.1)
  educational/typicalLearningTime/duration (5.9.1)
-->
<xsl:template match="lom:dateTime | lom:duration/lom:duration |
  lom:typicalLearningTime/lom:duration">
  <r n="{name(.)}" w="{round(100000*(1 div count(../*))) div 100000}">
    <ind type="{name(.)}"><xsl:value-of select="."/></ind>
  </r>
</xsl:template>

<!--
  matches elements of type mimeType:
  technical/format (4.1)
-->
<xsl:template match="lom:format">
  <r n="format" w="{round(100000*(1 div count(../*))) div 100000}">
    <ind type="mimeType"><xsl:value-of select="."/></ind>
  </r>
</xsl:template>

<!--
  matches elements of type nonNegativeInteger:
  technical/size (4.2)
-->
<xsl:template match="lom:size">
  <r n="size" w="{round(100000*(1 div count(../*))) div 100000}">
    <ind type="nonNegativeInteger"><xsl:value-of select="."/></ind>
  </r>
</xsl:template>

```

```

<!--
  matches:
  relation/resource (7.2)
-->
<xsl:template match="lom:resource">
  <r n="resource" w="{round(100000*(1 div count(../*))) div 100000}">
    <plex>
      <xsl:apply-templates/>
    </plex>
  </r>
</xsl:template>

<!--
  matches:
  classification/taxonPath (9.2)
-->
<xsl:template match="lom:taxonPath">
  <r n="taxonPath" w="{round(100000*(1 div count(../*))) div 100000}">
    <plex>
      <xsl:apply-templates select="lom:source"/>
      <xsl:apply-templates select="lom:taxon"/>
    </plex>
  </r>
</xsl:template>
</xsl:stylesheet>

```

### A.3 WOO RuleML to CanCore Translator [wooruleml2cancore.xslt]

```

<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
>
  <xsl:output method="xml" indent="yes"/>
  <!--
  issues:
  hard-coded handling of ordered elements
  reincorporate namespace and schema info from rule label?
-->
  <xsl:template match="/rulebase">
    <lom>
      <xsl:apply-templates select="//_r[@n='general']"/>
      <xsl:apply-templates select="//_r[@n='lifeCycle']"/>
      <xsl:apply-templates select="//_r[@n='metaMetadata']"/>
      <xsl:apply-templates select="//_r[@n='technical']"/>
      <xsl:apply-templates select="//_r[@n='educational']"/>
    </lom>
  </xsl:template>

```

```

        <xsl:apply-templates select="//_r[@n='rights']"/>
        <xsl:apply-templates select="//_r[@n='relation']"/>
        <xsl:apply-templates select="//_r[@n='annotation']"/>
        <xsl:apply-templates select="//_r[@n='classification']"/>
    </lom>
</xsl:template>
<xsl:template match="/rulebase/fact/_head/atom/_r[@n='general']">
    <general>
        <xsl:apply-templates select="plex/_r[@n='identifier']"/>
        <xsl:apply-templates select="plex/_r[@n='title']"/>
        <xsl:apply-templates select="plex/_r[@n='languageOrNone']"/>
        <xsl:apply-templates select="plex/_r[@n='description']"/>
        <xsl:apply-templates select="plex/_r[@n='keyword']"/>
        <xsl:apply-templates select="plex/_r[@n='aggregationLevel']"/>
    </general>
</xsl:template>
<xsl:template match="/rulebase/fact/_head/atom/_r[@n='lifeCycle']">
    <lifeCycle>
        <xsl:apply-templates select="plex/_r[@n='version']"/>
        <xsl:apply-templates select="plex/_r[@n='contribute_list']"/>
    </lifeCycle>
</xsl:template>
<xsl:template match="/rulebase/fact/_head/atom/_r[@n='metaMetadata']">
    <metaMetadata>
        <xsl:apply-templates select="plex/_r[@n='identifier']"/>
        <xsl:apply-templates select="plex/_r[@n='contribute_list']"/>
        <xsl:apply-templates select="plex/_r[@n='metadataSchema']"/>
        <xsl:apply-templates select="plex/_r[@n='language']"/>
    </metaMetadata>
</xsl:template>
<xsl:template match="/rulebase/fact/_head/atom/_r[@n='technical']">
    <technical>
        <xsl:apply-templates select="plex/_r[@n='format']"/>
        <xsl:apply-templates select="plex/_r[@n='size']"/>
        <xsl:apply-templates select="plex/_r[@n='location_list']"/>
        <xsl:apply-templates select="plex/_r[@n='otherPlatformRequirements']"/>
        <xsl:apply-templates select="plex/_r[@n='duration']"/>
    </technical>
</xsl:template>
<xsl:template match="/rulebase/fact/_head/atom/_r[@n='educational']">
    <educational>
        <xsl:apply-templates select="plex/_r[@n='learningResourceType_list']"/>
        <xsl:apply-templates select="plex/_r[@n='intendedEndUserRole_list']"/>
        <xsl:apply-templates select="plex/_r[@n='context']"/>
        <xsl:apply-templates select="plex/_r[@n='typicalAgeRange']"/>
        <xsl:apply-templates select="plex/_r[@n='typicalLearningTime']"/>
    </educational>
</xsl:template>

```

```

        <xsl:apply-templates select="plex/_r[@n='language']"/>
    </educational>
</xsl:template>
<xsl:template match="/rulebase/fact/_head/atom/_r[@n='rights']">
    <rights>
        <xsl:apply-templates select="plex/_r[@n='cost']"/>
        <xsl:apply-templates select="plex/_r[@n='copyrightAndOtherRestrictions']"/>
        <xsl:apply-templates select="plex/_r[@n='description']"/>
    </rights>
</xsl:template>
<xsl:template match="/rulebase/fact/_head/atom/_r[@n='relation']">
    <relation>
        <xsl:apply-templates select="plex/_r[@n='kind']"/>
        <xsl:apply-templates select="plex/_r[@n='resource']"/>
    </relation>
</xsl:template>
<xsl:template match="/rulebase/fact/_head/atom/_r[@n='annotation']">
    <annotation>
        <xsl:apply-templates select="plex/_r[@n='entity']"/>
        <xsl:apply-templates select="plex/_r[@n='date']"/>
        <xsl:apply-templates select="plex/_r[@n='description']"/>
    </annotation>
</xsl:template>
<xsl:template match="/rulebase/fact/_head/atom/_r[@n='classification']">
    <classification>
        <xsl:apply-templates select="plex/_r[@n='purpose']"/>
        <xsl:apply-templates select="plex/_r[@n='taxonPath']"/>
        <xsl:apply-templates select="plex/_r[@n='keyword_list']"/>
    </classification>
</xsl:template>
<xsl:template match="plex/_r[@n='identifier']">
    <identifier>
        <xsl:apply-templates select="plex/_r[@n='catalog']"/>
        <xsl:apply-templates select="plex/_r[@n='entry']"/>
    </identifier>
</xsl:template>
<xsl:template match="plex/_r[@n='title']">
    <title>
        <xsl:apply-templates select="plex/_r[@n='language']"/>
    </title>
</xsl:template>
<xsl:template match="plex/_r[@n='language']">
    <xsl:choose>
        <xsl:when test="ind[@type='languageOrNone'] |
            ind[@type='language']">
            <language>

```

```

        <xsl:value-of select="ind"/>
    </language>
</xsl:when>
<xsl:otherwise>
    <string language="{ind/@type}">
        <xsl:value-of select="ind"/>
    </string>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
<xsl:template match="plex/_r[@n='languageOrNone'] |
_r[@n='metaMetadata']/plex/_r[@n='language'] |
_r[@n='educational']/plex/_r[@n='language']">
    <language>
        <xsl:value-of select="ind/@type"/>
    </language>
</xsl:template>
<xsl:template match="_r[@n='description']">
    <description>
        <xsl:apply-templates select="plex/_r[@n='language']"/>
    </description>
</xsl:template>
<xsl:template match="_r[@n='keyword']">
    <keyword>
        <xsl:apply-templates select="plex/_r[@n='language']"/>
    </keyword>
</xsl:template>
<xsl:template match="plex/_r[@n='aggregationLevel']">
    <aggregationLevel>
        <xsl:apply-templates select="plex/_r[@n='source']"/>
        <xsl:apply-templates select="plex/_r[@n='value']"/>
    </aggregationLevel>
</xsl:template>
<xsl:template match="plex/_r[@n='version']">
    <version>
        <xsl:apply-templates select="plex/_r[@n='language']"/>
    </version>
</xsl:template>
<xsl:template match="plex/_r[@n='contribute_list']">
    <xsl:apply-templates select="plex/_r[@n='contribute_1']"/>
    <xsl:apply-templates select="plex/_r[@n='contribute_2']"/>
    <xsl:apply-templates select="plex/_r[@n='contribute_3']"/>
    <xsl:apply-templates select="plex/_r[@n='contribute_4']"/>
    <xsl:apply-templates select="plex/_r[@n='contribute_5']"/>
</xsl:template>
<xsl:template match="_r[@n='entity_list']">

```

```

        <xsl:apply-templates select="plex/_r[@n='entity_1']"/>
        <xsl:apply-templates select="plex/_r[@n='entity_2']"/>
    </xsl:template>

    <xsl:template match="_r[@n='contribute.1']">
        <contribute>
            <xsl:apply-templates select="plex/_r[@n='role']"/>
            <xsl:apply-templates select="plex/_r[@n='entity_list']"/>
            <xsl:apply-templates select="plex/_r[@n='date']"/>
        </contribute>
    </xsl:template>
    <xsl:template match="_r[@n='contribute.2']">
        <contribute>
            <xsl:apply-templates select="plex/_r[@n='role']"/>
            <xsl:apply-templates select="plex/_r[@n='entity_list']"/>
            <xsl:apply-templates select="plex/_r[@n='date']"/>
        </contribute>
    </xsl:template>
    <xsl:template match="_r[@n='contribute.3']">
        <contribute>
            <xsl:apply-templates select="plex/_r[@n='role']"/>
            <xsl:apply-templates select="plex/_r[@n='entity_list']"/>
            <xsl:apply-templates select="plex/_r[@n='date']"/>
        </contribute>
    </xsl:template>
    <xsl:template match="_r[@n='contribute.4']">
        <contribute>
            <xsl:apply-templates select="plex/_r[@n='role']"/>
            <xsl:apply-templates select="plex/_r[@n='entity_list']"/>
            <xsl:apply-templates select="plex/_r[@n='date']"/>
        </contribute>
    </xsl:template>
    <xsl:template match="_r[@n='contribute.5']">
        <contribute>
            <xsl:apply-templates select="plex/_r[@n='role']"/>
            <xsl:apply-templates select="plex/_r[@n='entity_list']"/>
            <xsl:apply-templates select="plex/_r[@n='date']"/>
        </contribute>
    </xsl:template>
    <xsl:template match="_r[@n='location.1']">
        <location>
            <xsl:value-of select="ind/@href"/>
        </location>
    </xsl:template>
    <xsl:template match="_r[@n='location.2']">
        <location>

```

```

        <xsl:value-of select="ind/@href"/>
    </location>
</xsl:template>
    <xsl:template match="_r[@n='entity_1']">
        <entity>
            <xsl:apply-templates select="ind[@type='vCard']"/>
        </entity>
    </xsl:template>
    <xsl:template match="_r[@n='entity_2']">
        <entity>
            <xsl:apply-templates select="ind[@type='vCard']"/>
        </entity>
    </xsl:template>
    <xsl:template match="_r[@n='entity']">
        <entity>
            <xsl:apply-templates select="ind[@type='vCard']"/>
        </entity>
    </xsl:template>

    <xsl:template match="_r[@n='id']">
        <id>
            <xsl:value-of select="ind"/>
        </id>
    </xsl:template>
    <xsl:template match="plex/_r[@n='metadataSchema']">
        <metadataSchema>
            <xsl:value-of select="ind"/>
        </metadataSchema>
    </xsl:template>
    <xsl:template match="plex/_r[@n='format']">
        <format>
            <xsl:value-of select="ind"/>
        </format>
    </xsl:template>
    <xsl:template match="plex/_r[@n='size']">
        <size>
            <xsl:value-of select="ind"/>
        </size>
    </xsl:template>
    <xsl:template match="plex/_r[@n='location_list']">
        <xsl:apply-templates select="plex/_r[@n='location_1']"/>
        <xsl:apply-templates select="plex/_r[@n='location_2']"/>
    </xsl:template>
    <xsl:template match="plex/_r[@n='otherPlatformRequirements']">
        <otherPlatformRequirements>
            <xsl:apply-templates select="plex/_r[@n='language']"/>
        </otherPlatformRequirements>
    </xsl:template>

```

```

    </otherPlatformRequirements>
</xsl:template>
<xsl:template match="plex/_r[@n='duration']">
  <duration>
    <xsl:if test="plex/_r/@n='duration'">
      <duration>
        <xsl:value-of select="plex/_r/ind"/>
      </duration>
    </xsl:if>
    <xsl:value-of select="ind"/>
    <xsl:apply-templates select="plex/_r[@n='description']"/>
  </duration>
</xsl:template>
<xsl:template match="plex/_r[@n='dateTime']">
  <xsl:value-of select="ind"/>
</xsl:template>
<xsl:template match="plex/_r[@n='learningResourceType_list']">
  <xsl:apply-templates select="plex/_r[@n='learningResourceType_1']"/>
  <xsl:apply-templates select="plex/_r[@n='learningResourceType_2']"/>
  <xsl:apply-templates select="plex/_r[@n='learningResourceType_3']"/>
</xsl:template>
<xsl:template match="_r[@n='learningResourceType_1']">
  <learningResourceType>
    <xsl:apply-templates select="plex/_r[@n='source']"/>
    <xsl:apply-templates select="plex/_r[@n='value']"/>
  </learningResourceType>
</xsl:template>
<xsl:template match="_r[@n='learningResourceType_2']">
  <learningResourceType>
    <xsl:apply-templates select="plex/_r[@n='source']"/>
    <xsl:apply-templates select="plex/_r[@n='value']"/>
  </learningResourceType>
</xsl:template>
<xsl:template match="_r[@n='learningResourceType_3']">
  <learningResourceType>
    <xsl:apply-templates select="plex/_r[@n='source']"/>
    <xsl:apply-templates select="plex/_r[@n='value']"/>
  </learningResourceType>
</xsl:template>
<xsl:template match="plex/_r[@n='intendedEndUserRole_list']">
  <xsl:apply-templates select="plex/_r[@n='intendedEndUserRole_1']"/>
  <xsl:apply-templates select="plex/_r[@n='intendedEndUserRole_2']"/>
  <xsl:apply-templates select="plex/_r[@n='intendedEndUserRole_3']"/>
</xsl:template>
<xsl:template match="_r[@n='intendedEndUserRole_1']">
  <intendedEndUserRole>

```

```

        <xsl:apply-templates select="plex/_r[@n='source']"/>
        <xsl:apply-templates select="plex/_r[@n='value']"/>
    </intendedEndUserRole>
</xsl:template>
<xsl:template match="_r[@n='intendedEndUserRole_2']">
    <intendedEndUserRole>
        <xsl:apply-templates select="plex/_r[@n='source']"/>
        <xsl:apply-templates select="plex/_r[@n='value']"/>
    </intendedEndUserRole>
</xsl:template>
<xsl:template match="_r[@n='intendedEndUserRole_3']">
    <intendedEndUserRole>
        <xsl:apply-templates select="plex/_r[@n='source']"/>
        <xsl:apply-templates select="plex/_r[@n='value']"/>
    </intendedEndUserRole>
</xsl:template>
<xsl:template match="plex/_r[@n='keyword_list']">
    <xsl:apply-templates select="plex/_r[@n='keyword_1']"/>
    <xsl:apply-templates select="plex/_r[@n='keyword_2']"/>
    <xsl:apply-templates select="plex/_r[@n='keyword_3']"/>
    <xsl:apply-templates select="plex/_r[@n='keyword_4']"/>
</xsl:template>
<xsl:template match="_r[@n='keyword_1']">
    <keyword>
        <xsl:apply-templates select="plex/_r[@n='language']"/>
    </keyword>
</xsl:template>
<xsl:template match="_r[@n='keyword_2']">
    <keyword>
        <xsl:apply-templates select="plex/_r[@n='language']"/>
    </keyword>
</xsl:template>
<xsl:template match="_r[@n='keyword_3']">
    <keyword>
        <xsl:apply-templates select="plex/_r[@n='language']"/>
    </keyword>
</xsl:template>
<xsl:template match="_r[@n='keyword_4']">
    <keyword>
        <xsl:apply-templates select="plex/_r[@n='language']"/>
    </keyword>
</xsl:template>
<xsl:template match="plex/_r[@n='taxon_list']">
    <xsl:apply-templates select="plex/_r[@n='taxon_1']"/>
    <xsl:apply-templates select="plex/_r[@n='taxon_2']"/>
    <xsl:apply-templates select="plex/_r[@n='taxon_3']"/>

```

```

        <xsl:apply-templates select="plex/_r[@n='taxon_4']"/>
        <xsl:apply-templates select="plex/_r[@n='taxon_5']"/>
    </xsl:template>
    <xsl:template match="_r[@n='context']">
        <context>
            <xsl:apply-templates select="plex/_r[@n='source']"/>
            <xsl:apply-templates select="plex/_r[@n='value']"/>
        </context>
    </xsl:template>
    <xsl:template match="plex/_r[@n='typicalAgeRange']">
        <typicalAgeRange>
            <xsl:apply-templates select="plex/_r[@n='language']"/>
        </typicalAgeRange>
    </xsl:template>
    <xsl:template match="plex/_r[@n='typicalLearningTime']">
        <typicalLearningTime>
            <xsl:apply-templates select="plex/_r[@n='duration']"/>
            <xsl:apply-templates select="plex/_r[@n='description']"/>
        </typicalLearningTime>
    </xsl:template>
    <xsl:template match="_r[@n='source']">
        <source>
            <xsl:value-of select="ind"/>
            <xsl:apply-templates select="plex/_r[@n='language']"/>
        </source>
    </xsl:template>
    <xsl:template match="_r[@n='value']">
        <value>
            <xsl:value-of select="ind"/>
        </value>
    </xsl:template>
    <xsl:template match="plex/_r[@n='cost']">
        <cost>
            <xsl:apply-templates select="plex/_r[@n='source']"/>
            <xsl:apply-templates select="plex/_r[@n='value']"/>
        </cost>
    </xsl:template>
    <xsl:template match="plex/_r[@n='copyrightAndOtherRestrictions']">
        <copyrightAndOtherRestrictions>
            <xsl:apply-templates select="plex/_r[@n='source']"/>
            <xsl:apply-templates select="plex/_r[@n='value']"/>
        </copyrightAndOtherRestrictions>
    </xsl:template>
    <xsl:template match="plex/_r[@n='kind']">
        <kind>
            <xsl:apply-templates select="plex/_r[@n='source']"/>

```

```

        <xsl:apply-templates select=" plex/_r[@n='value']"/>
    </kind>
</xsl:template>
<xsl:template match="_r[@n='resource']">
    <resource>
        <xsl:apply-templates select=" plex/_r[@n='identifier']"/>
        <xsl:apply-templates select=" plex/_r[@n='description']"/>
    </resource>
</xsl:template>
<xsl:template match="_r[@n='date']">
    <date>
        <xsl:apply-templates select=" plex/_r[@n='dateTime']"/>
        <xsl:apply-templates select=" plex/_r[@n='description']"/>
    </date>
</xsl:template>
<xsl:template match="plex/_r[@n='purpose']">
    <purpose>
        <xsl:apply-templates select="plex/_r[@n='source']"/>
        <xsl:apply-templates select="plex/_r[@n='value']"/>
    </purpose>
</xsl:template>
<xsl:template match="plex/_r[@n='entry']">
    <entry>
        <xsl:value-of select="ind"/>
        <xsl:value-of select="ind/@href"/>
        <xsl:apply-templates select=" plex/_r[@n='language']"/>
    </entry>
</xsl:template>
<xsl:template match="plex/_r[@n='taxon_1']">
    <taxon>
        <xsl:apply-templates select=" plex/_r[@n='id']"/>
        <xsl:apply-templates select=" plex/_r[@n='entry']"/>
    </taxon>
</xsl:template>
<xsl:template match="plex/_r[@n='taxon_2']">
    <taxon>
        <xsl:apply-templates select=" plex/_r[@n='id']"/>
        <xsl:apply-templates select=" plex/_r[@n='entry']"/>
    </taxon>
</xsl:template>
<xsl:template match="plex/_r[@n='taxon_3']">
    <taxon>
        <xsl:apply-templates select=" plex/_r[@n='id']"/>
        <xsl:apply-templates select=" plex/_r[@n='entry']"/>
    </taxon>
</xsl:template>

```

```

<xsl:template match="plex/_r[@n='taxon_4']">
  <taxon>
    <xsl:apply-templates select=" plex/_r[@n='id']"/>
    <xsl:apply-templates select=" plex/_r[@n='entry']"/>
  </taxon>
</xsl:template>
<xsl:template match="plex/_r[@n='taxon_5']">
  <taxon>
    <xsl:apply-templates select=" plex/_r[@n='id']"/>
    <xsl:apply-templates select=" plex/_r[@n='entry']"/>
  </taxon>
</xsl:template>
<xsl:template match="plex/_r[@n='taxonPath']">
  <taxonPath>
    <xsl:apply-templates select=" plex/_r[@n='source']"/>
    <xsl:apply-templates select=" plex/_r[@n='taxon_list']"/>
  </taxonPath>
</xsl:template>
<xsl:template match="plex/_r[@n='catalog']">
  <catalog>
    <xsl:value-of select="ind"/>
  </catalog>
</xsl:template>
<xsl:template match="plex/_r[@n='id']">
  <id>
    <xsl:value-of select="ind"/>
  </id>
</xsl:template>
<xsl:template match="_r[@n='role']">
  <role>
    <xsl:apply-templates select=" plex/_r[@n='source']"/>
    <xsl:apply-templates select=" plex/_r[@n='value']"/>
  </role>
</xsl:template>
<xsl:template match="ind[@type='vCard']">
  <xsl:value-of select="."/>
</xsl:template>
<xsl:template match="plex/_r[@n='dateTime']">
  <dateTime>
    <xsl:value-of select="ind"/>
  </dateTime>
</xsl:template>
</xsl:stylesheet>

```

#### A.4 Final CanCore Document [wooruleml2cancore\_output.xml]

```
<?xml version="1.0" encoding="UTF-8"?>
<lom>
  <general>
    <identifier>
      <catalog>URI</catalog>
      <entry>http://www.howstuffworks.com/robot.htm</entry>
    </identifier>
    <title>
      <string language="en">How Robots Work</string>
      <string language="fr">Comment fonctionnent les robots</string>
    </title>
    <language>en</language>
    <language>fr</language>
    <description>
      <string language="en">A robot and a human being ... </string>
      <string language="fr">Un robot et un etre humain ... </string>
    </description>
    <keyword>
      <string language="x-none">Robotics</string>
    </keyword>
    <keyword>
      <string language="x-none">Artificial intelligence</string>
    </keyword>
    <keyword>
      <string language="x-none">Electronic control</string>
    </keyword>
    <aggregationLevel>
      <source>LOMv1.0</source>
      <value>3</value>
    </aggregationLevel>
  </general>
  <lifeCycle>
    <version>
      <string language="en">limited edition</string>
      <string language="fr">edition limitee</string>
    </version>
    <contribute>
      <role>
        <source>CanCore v1.9</source>
        <value>Content Provider</value>
      </role>
      <entity>BEGIN: vCard ORG: HowStuffWorks.com END: vCard</entity>
      <entity>BEGIN: vCard ORG: Answers.com END: vCard</entity>
    </contribute>
    <date>
```

```

        <dateTime>2002</dateTime>
        <description>
            <string language="en">second quarter of 2002</string>
            <string language="fr">le deuxieme quart de 2002</string>
        </description>
    </date>
</contribute>
<contribute>
    <role>
        <source>CanCore v1.9</source>
        <value>Graphic Designer</value>
    </role>
    <entity>
        BEGIN: vCard ORG: GraphicDesignGuys.com END: vCard
    </entity>
    <date>
        <dateTime>2002</dateTime>
        <description>
            <string language="en">second quarter of 2002</string>
            <string language="fr">le deuxieme quart de 2002</string>
        </description>
    </date>
</contribute>
<contribute>
    <date>
        <dateTime>2010</dateTime>
    </date>
</contribute>
<contribute>
    <date>
        <dateTime>2011</dateTime>
    </date>
</contribute>
<contribute>
    <date>
        <dateTime>2012</dateTime>
    </date>
</contribute>
</lifeCycle>
<metaMetadata>
    <identifier>
        <catalog>CanLOM</catalog>
        <entry>77168</entry>
    </identifier>
</contribute>
<role>

```

```

        <source>CanCore v1.9</source>
        <value>Creator</value>
    </role>
    <entity>BEGIN: vCard ORG: CanLOM END: vCard</entity>
    <entity>BEGIN: vCard ORG: LOM END: vCard</entity>
    <date>
        <dateTime>2002</dateTime>
        <description>
            <string language="en">second quarter of 2002</string>
            <string language="fr">le deuxieme quart de 2002</string>
        </description>
    </date>
</contribute>
<contribute>
    <role>
        <source>LOMv1.0</source>
        <value>validator</value>
    </role>
    <entity>BEGIN:VCARD\nVERSION:3.0\nN:Smith;FN:Mary\nORG:;\n
        END:VCARD\n
    </entity>
    <date>
        <dateTime>2003-11-03</dateTime>
    </date>
</contribute>
<metadataSchema>CanCore v1.9</metadataSchema>
<metadataSchema>LOMv1.0</metadataSchema>
<language>en</language>
</metaMetadata>
<technical>
    <format>HTML</format>
    <format>application/zip</format>
    <size>430024</size>
    <location>http://www.howstuffworks.com/robot.htm</location>
    <location>http://sdf943892.onlinehome.us/robot</location>
    <otherPlatformRequirements>
        <string language="en">HARDWARE: Internet-enabled PC or Mac.
            SOFTWARE: Netscape Navigator or Internet Explorer and email.
        </string>
    </otherPlatformRequirements>
    <duration>
        <duration>PT1H20M25S</duration>
    <description>
        <string language="en">depending on playback speed,
            between 20 and 25 minutes
        </string>
    </description>

```

```

        </description>
    </duration>
</technical>
<educational>
    <learningResourceType>
        <source>LOMv1.0</source>
        <value>Exercise</value>
    </learningResourceType>
    <intendedEndUserRole>
        <source>CanCore v1.9</source>
        <value>Learner</value>
    </intendedEndUserRole>
    <context>
        <source>CanCore v1.9</source>
        <value>Elementary Grade 8</value>
    </context>
    <typicalAgeRange>
        <string language="x-none">18-25</string>
    </typicalAgeRange>
    <typicalLearningTime>
        <duration>PT1H20M25S</duration>
        <description>
            <string language="en">For average students, this unit
                may take one full day of class time including a few
                hours of homework. Advanced students may need only
                a half-day in total.
            </string>
        </description>
    </typicalLearningTime>
    <language>en</language>
</educational>
<rights>
    <cost>
        <source>LOMv1.0</source>
        <value>No</value>
    </cost>
    <copyrightAndOtherRestrictions>
        <source>LOMv1.0</source>
        <value>Yes</value>
    </copyrightAndOtherRestrictions>
    <description>
        <string language="en">Some restrictions may apply. Contact
            publisher for details.
        </string>
    </description>
</rights>

```

```

<relation>
  <kind>
    <source>LOMv1.0</source>
    <value>references</value>
  </kind>
  <resource>
    <identifier>
      <catalog>ISSN</catalog>
      <entry>1191-8276</entry>
    </identifier>
  </resource>
</relation>
<annotation>
  <entity>BEGIN:VCARD\nVERSION:3.0\n Tremblay;Andrea.\n
    FN:Andrea Tremblay\nEND:VCARD
  </entity>
  <date>
    <dateTime>2003-04</dateTime>
  </date>
  <description>
    <string language="en">
      I spent a number of hours looking over and reviewing
      the information provided on this site. I found that the
      information provided would be helpful to both students
      and teachers. As an educator the activities and lesson
      ideas were very appealing. Children would be drawn in
      and would gain tremendous knowledge about the nervous
      system by it.
    </string>
  </description>
</annotation>
<classification>
  <purpose>
    <source>CanCore v1.9</source>
    <value>Discipline</value>
  </purpose>
  <taxonPath>
    <source><string language="en">ERIC</string></source>
    <taxon>
      <id>971</id>
      <entry><string language="en">Computer Science</string></entry>
    </taxon>
    <taxon>
      <id>972</id>
      <entry><string language="en">Computers</string></entry>
    </taxon>
  </taxonPath>

```

```

    <taxon>
      <id>973</id>
      <entry><string language="en">Internet</string></entry>
    </taxon>
    <taxon>
      <id>974</id>
      <entry><string language="en">Applied Sciences</string></entry>
    </taxon>
    <taxon>
      <id>975</id>
      <entry><string language="en">Technology</string></entry>
    </taxon>
  </taxonPath>
  <keyword>
    <string language="en">educational technology</string>
  </keyword>
  <keyword>
    <string language="en">computer based instruction</string>
  </keyword>
  <keyword>
    <string language="en">computer</string>
  </keyword>
  <keyword>
    <string language="en">instruction</string>
  </keyword>
</classification>
<classification>
  <purpose>
    <source>CanCore v1.9</source>
    <value>PedagogicType</value>
  </purpose>
  <taxonPath>
    <source><string language="en">CanCore v1.9</string></source>
    <taxon>
      <entry><string language="en">Lesson</string></entry>
    </taxon>
  </taxonPath>
</classification>
</lom>

```