IMS Learning Resource XML Binding Specification

Final Specification
Version 1.1

About This Document

Title	IMS Learning Resource XML Binding Specification
Authors	Thor Anderson and Tom Wason
Version	1.1
Version Date	5th June, 2000
Status	**Final Specification.**
Summary	This document provides updated information regarding IMS Learning Resource XML Binding Specification.
Revision Information	5th June, 2000
Purpose	Defines the Learning XML Binding Specification
Document Location	http://www.imsproject.org/metadata/mdbindv1p1.html

Revision History

Version No.	Release Date	Comments
Final 1.0	August 20, 1999	The version 1.0 of the IMS Learning Resource XML Binding Specification Released
Final 1.1	May 5, 2000	All element names changed to lower case only. Comments on namespace use changed.

Table of Contents

Introduction

This document describes the XML binding for the IMS Learning Resource Meta-data Information Model. The model is based on the IEEE Learning Technology Standards Committee (LTSC) Learning Object Meta-data base document, plus modifications approved by the IMS Technical Board and submitted to IEEE. For links to the related IEEE documents, please see
http://www.imsproject.org/metadata/mdinfov1p1.html

XML Basics

The IEEE conceptual model for meta-data definitions is a hierarchy. Hierarchical models are convenient for representing data consisting of many elements and subelements. XML is perfectly suited for representing hierarchical models such as the IEEE LOM Base Document. An XML document is a hierarchy comprised of **elements** that have **contents** and **attributes**.

Elements

An element is a component of a document that has been identified in a way a computer can understand. Each element has a **tag name**. When a tag name is shown as "<tagname>", with less-than and greater-than symbols before and after the tag name, it serves as the **start-tag** to mark the beginning of an element. When that same tag name has a forward slash "/" added, it serves as an **end-tag** such as "</tagname>". An element may have contents between its start and end-tags, and may have one or more **attributes**. When an XML element has a start and end-tag (also called an **opening** and **closing tag)** with a common name, it is considered to be "well-formed" XML. The contents of an element are placed between the start and end-tags as shown below:

```
<tagname>contents</tagname>
```

Element Contents

An element may contain other elements, Parsed Character Data (PCDATA), Character Data (CDATA), or a mixture of PCDATA and elements. The allowable contents of an element are its content model. PCDATA really means any character string that does not contain elements. PCDATA is what the bulk of elements will use between their start and end-tags. CDATA is different in that it is a method for adding any character data that should not be processed. For example you could add some JavaScript code instructions using a CDATA section. A CDATA section tells the parser not to look for any markup until after it locates the end of the CDATA section.

Element Attributes

An attribute provides additional information about an element. Attributes are a way of attaching characteristics or properties to the elements of a document. An element may have more than one attribute and are contained within the start tag of an element. Attributes are represented by an attribute name followed by an equal sign and the attribute value in quotation marks:

```
<title><langstring lang="en-US">Sniffy the Virtual
Rat</langstring></title>
```

In this example, the title element contains another element, the langstring element. The langstring element has one attribute "lang", with the value "en-US" and the contents of the element being the string "Sniffy the Virtual Rat".

Element Names

Each element has a unique name, the tag name. XML is case-sensitive in its processing of tag names. The IMS Learning Resource Meta-data XML binding specification adheres to the following tag name rules:

- All tag names will conform to the rules for element naming as given within the XML Version 1.0 specification.
- Names beginning in "xml" in any case or mix of cases are not permitted.
- The IMS binding will use only **lower case** tag names. All element names in the IMS XML binding are to be in **lower case**. This will allow uniform machine conversion to a either case should the need arise.

- Element names may not include words reserved by the XML specification. These include:

  DOCTYPE

  ELEMENT

  ATTLIST

  ENTITY

  XML (in any case or combination of cases)

- Tag names defined by the IMS binding may not be redefined.

Document Type Definitions (DTD)

The *tag name*, *content model*, and *attributes* of elements are defined in a **Document Type Definition** (DTD) statement. This may exist as an external file or a block of text internal to an XML document. Internal DTDs are used to override elements defined in external DTD files, so an internal DTD should be used with care. The DTD defines the elements that may be used, and may define the contents of the elements.

This specification defines external DTDs with defined file names, specifically **IMS-MD01-lc.dtd** and **IMSCOR01-lc.dtd**. These file names represent the 1.1 version of the IMS Meta-data and the version 1.1 of the IMS CORE meta-data respectively. Some XML editors may make use of a DTD to help guide the developer in creating the proper elements at the proper locations in an XML file. Other developers will make use of DTDs to validate their XML documents to ensure their document is consistent with all of the element names and locations defined in the DTD. An XML document is **valid** if it has an associated document type declaration and if the document complies with the constraints expressed in it. Details of the construction of DTDs are outside the scope of this document, but links to the XML Version 1.0 specification and the IMS-MD01.dtd are included in the Appendices section of this document.

Declaring Element Contents

The information specifying the order and usage of allowable contents for an element are its content model. The content model is declared in a Document Type Definition or DTD (see below). The declaration of the content model is of the general form:

```
<!ELEMENT tagname (Content Model)>
```

The datetimeelement can again serve as an example of how an element is declared with its content model:

```
<!ELEMENT datetime(#PCDATA|extension)*>
```

The vertical bar character "|" indicates that the meta-data author may choose between the elements. The asterisk "*" after the content model means that the #PCDATA element and the extension element may be mixed or optionally interspersed with subelements. This definition of the datetimeelement's content model allows the following XML fragment to exist:

```
<datetime>1999-07-23
   <extension> <langstring lang="en">circa</langstring>
</extension>
</datetime>
```

Notice that the extension element is optional and was used in the example above. The XML specification provides more information about the details for creating and interpreting content models.

Declaring Element Attributes

An example of how the attributes for the element langstring are declared in a DTD is found below:

```
<!ELEMENT langstring (#PCDATA|extension)*>
   <!ATTLIST langstring
    lang CDATA #IMPLIED>
```

The first line declares that there is an element named "langstring" which is allowed to have PCDATA and extension elements as its contents. The second line begins with "!ATTLIST" to start an attribute list declaration for the langstring element. The word "lang" will serve as the attribute's name. The allowable value for this attribute must be of type CDATA.

At the end of the example above is the term #IMPLIED. It is at this location in the attribute declaration, where a default value for an attribute may be specified. It is also possible to use the keyword #REQUIRED which would force a lang value to be supplied and there would be no default value. In the example above, the #IMPLIED designation means that the DTD designer wants to allow users to omit the value for the attribute without forcing a particular default value.

Use of Attributes

Within the IMS XML binding, the use of attributes is reserved for information about the structure of, and source of terms in, the meta-data record. It is recommended that attributes not be used for information about the resource. This IMS XML binding specification uses only two element attributes (the "lang" attribute and the "type" attribute) in particular ways and for particular purposes.

lang:

This attribute specifies the human language of the contents of the element. It is only used as an attribute of the langstring element. The lang attribute may contain a two character language code followed by a two character country code. For example:

```
<otherplatformrequirements>
   <langstring lang="en-US">Will not run in browser.</langstring>
</otherplatformrequirements>
```

The codes for languages and countries are enumerated in the XML specification.

type:

This attribute specifies the type of string that may be used to identify the location of a learning resource as used in the Location element. The type attribute may be assigned the value of either "URI" or "TEXT". These values indicate whether the string used will be a simple textual description of where a resource is located or whether the string represents a resource available on the Internet with a specific address such as a URL. For example:

```
<technical>
   <format/>
   <size>1032353</size>
   <location type="URI">http://www.brookscole.com</location>
</technical>
```

The codes for languages and countries are enumerated in the XML specification.

Lists

The meta-data specification uses **listing** at multiple levels in the hierarchy. A list is a repetition of the contents of an element. In XML this is accomplished by repeating the containing element:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE record [
   <!ELEMENT record (greeting*)>
   <!ELEMENT greeting (#PCDATA)>
   ]>
<record>
   <greeting>Hello, world!</greeting>
   <greeting>How are you?</greeting>
</record>
```

In this example, the element "greeting" is repeated. Thus greeting is the **containing** element for the **repeated contents** of "Hello, world!" and "How are you?" The notation for repetitions of an element in a content model follows the XML specification. An asterisk (*) specifies that none or more repetitions of the element may be included in the XML instantiation. There are two main types of lists: **ordered** and **unordered**.

Ordered Lists

Repeating the listed element at its specific location in the XML structure creates an ordered list of contents. The order of the elements has significance as their placement in the XML file determines this. The following is an example of an XML fragment in which the educational element contains an ordered list of learningresourcetype (learning resource type) elements:

```
<educational>
<learningresourcetype>
   <langstring lang="en">Simulation</langstring>
</learningresourcetype>
<learningresourcetype>
   <langstring lang="en">Assessment</langstring>
   </learningresourcetype>
</educational>
```

Unordered Lists

Repeating the containing element at its specific location in the XML structure creates an unordered list of contents. The order of the repetitions has no significance. For example:

```
<general>
    <language>en_US</language>
    <language>fr_FR</language>
</general>
```

In this example, each new instance of a definition of a language requires that the language element be repeated. Whether an element list should be treated as ordered or unordered is specified by the IEEE Learning Object Meta-data (LOM) specification.

Namespaces

XML is designed to allow individuals to create their own element tag names. It soon became apparent that there could be problems if different DTDs were used in the same document and those DTDs had elements

using the same name. The XML Namespace recommendation proposal specifies a way to ensure that names from different DTDs can be identified in a single document.

The XML Namespace document provides more information about the flexible capabilities of namespaces. The W3C Recommendation for **Namespaces** (http://www.w3.org/TR/1999/REC-xml-names-19990114) does not specify how namespaces are to be used. The introductory abstract is as follows:

> "XML namespaces provide a simple method for qualifying element and attribute names used in Extensible Markup Language documents by associating them with namespaces identified by URI references."

The XML 1.0 standard does not specify how namespaces are to be processed. Currently there are two general approaches to namespaces:

1. Use to point to a specific encoding schema for machine interpretation, and
2. Use as a reference for uniqueness and possibly defnition (semantics).

These two approaches are not mutually exclusive. A namespace is applied as a prefix to an element or attribute name:

`<dc:subject>`.

The prefix of *dc:* is the qualifier, and must be defined elsewhere in the document. The user is directed to the W3C Namespace recommendation for more details on application. IMS does not specify how namespaces are to be resolved (semantically or for machine interpretation). Namespaces may point to DTD documents for validation.

An IMS XML binding document uses a default document IMS namespace of http://www.imsproject.org/metadata/. An example namespace declaration for the IMS is shown below.

Namespace declaration:

```
<record xmlns="http://www.imsproject.org/metadata/">
```

In order for any XML documents that are based upon this XML binding to find an associated DTD such as "IMS-MD01-lc.dtd", the IMS-MD01-lc.dtd file and the actual meta-data record must be placed in the same directory or the DTD must be found at the specified URL.

Always make sure to place the IMS-MD01-lc.dtd and the meta-data record in the same directory if you wish to validate the record locally.

Special Handling Requirements for Meta-data Elements

There are some common structures that are used more than once within the meta-data. The use of these common structures may facilitate the creation of common data storage structures in implementations. These structures have the suffix of "TYPE". interactivitytype is not a common structure, even though it ends in "TYPE". The types defined in the LOM and encoded in the XML are:

- LangStringType
- DateType

LangStringType

LangStringType denotes a string that is encoded for a specific language or other interpretable type. It is of the logical form:

```
LangStringType
  langstring
     language
     string
```

It is important to note how the logical form specified by the IEEE LOM appears to be different from the XML binding suggested in this document. In the suggested binding of this document, LangStringType is not an XML element. Rather langstring is an element with a "lang" attribute which is used to define the language of a string value:

```
<langstring lang="en">string value</langstring>
```

The **lang** attribute can contain both language and country codes as defined in the XML specification. Any element that contains a langstring element may contain multiple langstring elements with each one representing a different language

Each langstring within an element is considered to contain the same information, differing by language.

The XML 1.0 specification also allows the lang attribute to be assigned an arbitrary value that is agreed upon by parties in private use. These attributes are identified by the prefix "~~x~~" or "X-". This practice is strongly discouraged for IMS meta-data records.

DateType

DateType is a formatted date. There are two subelements representing the two different date formats used. Precise date and time information is formatted according to the ISO 8601 specification. Point in time and time duration information is captured in the DateTime element. More general date and time information is captured using the Description element. A DateType may contain values for both a DateTime and Description.

DateType has the logical structure of:

```
DateType
    datetime
    description
       LangStringType
```

```
langstring
   language
   string
```

It is important to note that, just as with the LangStringType, the logical form specified by the IEEE LOM for DateType appears to be different from the XML binding suggested in this document. In this binding, DateType is not an XML element. Rather *date* is an element with two subelements: *datetime* and *description*.

The datetime element makes use of the format dictated by the ISO8601 specification Dates are captured using the CCYY-MM-DD form while Time elements are specified as hh:mm:ss. There is also the ability to specify Time Zone Determination information by adding "+hh:mm" to indicate differences in time zones. Both the date and time value are combined using the capital "T" character to separate them. Three examples of this are found below:

Use *datetime* for precise dates and times. Use *description* for more general date time information.

```
<datetime>1999-07-26<datetime>
<datetime>1999-07-26T12:15:35<datetime>
<datetime>1999-07-26T12:15:35+01:00<datetime>
```

There is a version of the ISO8601 specification that is available through the W3C which also specifies how a date range is represented and how date and time duration is accurately represented. Readers may refer to the ISO8601 specification available at: http://www.w3.org/TR/NOTE-datetime for more detailed information on the usage of date and time values.

Language elements

Meta-data authors may specify a language that will be used as the default language for any langstring elements that are encountered. This is done by providing a value for the langstring element that is contained by the *metametadata* element. Each individual occurrence of langstring may override this default value by declaring a language and country code using the "lang" attribute.

The default language for a record can be specified in the *metametadata* category. Use individual *langstring* elements to override the default language.

TaxonPath elements

In most cases, the value of using the *taxonpath* element lies in the ability to locate the source of the taxonomy. If the *source* for a *taxonpath* is not provided or doesn't map to an existing, logical source then the element should contain something useful regarding how to location information about the taxonomy.

Always try to provide a *source* element when using the *taxonpath* element.

vCard elements

There are at least two elements in the IEEE LOM that require contributing entity information; elements **lifecycle.contribute.entity** and **metametadata.contribute.entity**. Both specify the vCard specification as the source for representing these elements' data.

When using only IMS Core elements, the **formatted name** or "FN" element from the vCard specification should contain the name of the individual contributing to the learning resource of metameta-data of the

resource. If a company, rather than an individual, contributed to the resource or resource metameta-data, the **organization** or "ORG" element from the vCard specification should be used. This is illustrated below:

```
   <centity>
   <vcard>
BEGIN:vCard
FN:Lotta Data
END:vCard
   </centity>
```

As far as most XML parsers are concerned, the information between the <vcard> and </vcard> tags is just a bunch of text. It is up to meta-data implementors to individually determine how they will process vCard information. The vCard specification allows for a rich set of information to be captured as the example below illustrates. The reader is directed to the "Using the vCard Specification" portion of this document and the vCard specification itself for more details regarding its usage.

Keywords

The elements *general.keywords* and *classification.keywords* are found in the IMS meta-data set. It is expected that the keywords describing a learning resource are likely to be provided in multiple languages. To accommodate this, the IMS XML binding suggests that keywords and short, keyword phrases be represented as separate langstring elements rather than a comma-delimited text string as in the example below:

Use multiple *langstring* elements to represent keywords and keyword phrases.

```
   <keywords>
      <langstring lang="en">operant conditioning</langstring>
      <langstring lang="en">psychology</langstring>
      <langstring lang="en">simulation</langstring>
      <langstring lang="en">program</langstring>
      <langstring lang="en">shaping</langstring>
      <langstring lang="en">mouse</langstring>
      <langstring lang="en">learn by doing</langstring>
   </keywords>
```

Extensibility

Some meta-data providers will find the current element set defined in the IMS meta-data as too restrictive to accomplish their meta-data purposes. To ensure meta-data extensibility, the IEEE LOM Base Document requires that there be no limit on potential extensions to the meta-data as long as the integrity of the specified meta-data is not impaired. An extension is the addition of information to an existing meta-data XML structure. There are two general ways to extend IMS meta-data:

1. One or more elements may be added to the structure using elements defined in the IEEE LOM Base Document; and
2. One or more elements may be added to the structure using elements that are not defined in the LOM specification.

These two types of extension are defined as:

1. Use of elements from the LOM: The LOM specification contains some elements with definitions that are not specific for any particular context. The context in which an element is placed provides specificity for its definition. These elements may be reused as long as the definition is not changed.
2. Use of elements not contained in the LOM specification: New elements may be introduced and used to extend the structure.

The XML binding does not inhibit either of these types of meta-data extension. The XML binding defines the Extension element as the element used for indicating where a set of extension elements can be found in the meta-data structure. In the IMSMD01.dtd file the extension element exists in every element's content model allowing every element to contain one or more extension elements. The only element without an extension capability is *identifier*, as it is a reserved word. The *extension* element's DTD declaration is:

Use the *extension* element to extend the meta-data structure.

```
<!ELEMENT extension ANY>
```

An example of the inclusion of extension in the content model of element coverage is:

```
<!ELEMENT coverage (langstring*, extension?)>
```

The use of the extension element is illustrated as follows:

```
<coverage>
   <langstring>1880-1900</langstring>
   <extension>
     <role>Date Range</role>
   </extension>
</coverage>
```

The contents, but not a content model, of an extension must be declared in an internal or external DTD. Many extensions can be created through the use of existing elements. Care must be used with internal DTDs, as they override external DTD declarations. The contents of an extension must obey the attribute and content models of the elements employed. New elements that duplicate the definitions of existing elements should not be introduced.

Prefacing the extension element with an appropriate namespace may usefully reference descriptions of extensions. For example, a group such as the Advanced Distributed Learning (ADL) initiative may wish to add the "adl" prefix to an extension element to uniquely identify ADL extensions. An example of this is show below:

```
<learningcontext>
   <langstring lang="en">Military Training</langstring>
      <extension adl:type="Topic">
         <title>Roman military tactics</title>
         <langstring lang="en">This example discusses how the Romans
           defined many aspects of modern warfare.
         </langstring>
      </extension>
</learningcontext>
```

This serves to note the entire extension structure. Extensions should always be added at the lowest point (farthest from the root element) in the hierarchy possible, to the degree that the structure defines the meaning of the extension.

Using the vCard Specification

The IMS XML binding uses the vCard specification wherever the Entity element is defined. An Entity, as far as IMS meta-data is concerned, represents a person or organization. The IMS binding uses the clear text form of the vCard specification. The vCard specification defines the **clear text** form as a "Simplegram". This is not intended as a complete description of the vCard coding; it is intended to provide some guidelines for simple cases. The vCard specification is located at http://www.imc.org/pdi/.

The vCard specification defines a set of **properties** that contain the information about an entity. The default character set encoding for the vCard is 7-bit US-ASCII. The default character set can be overridden for an individual property using the "CHARSET" property parameter. For example, the ISO 8859-8 or Latin/Hebrew character set used in an address is specified by:

```
ADR;CHARSET=ISO-8859-8:…
```

It is also possible to set the encoding for the entire record to another encoding. See the vCard specification for further instructions. The default language is "en-US", which may similarly be overridden for a property using the "language" property parameter. Property names are case insensitive.

The general form of the Simplegram vCard encoding is:

```
BEGIN:vcard
Items
END:vcard
```

Items is a list of items separated by a any valid **line ending protocol**. For example, in 7-bit ASCII, the Carriage Return (CR) character (ASCII decimal 13), the Line Feed character (LF) (ASCII decimal 10), the Carriage Return character followed by a Line Feed character (CRLF), or the Property Delimiter are line ending protocols. Property parameter substrings are delimited by the Field Delimiter, specified by the Semi-Colon (;) character (ASCII decimal 59). Each item is of the general form:

```
name:value A;value B CRLF
```

An example of an item with no substrings is the **formatted name** property, **FN**. **FN** is the full formatted name of a person:

```
FN:Mr. James Q. Smith, Jr.
```

Some items may have multiple properties or **parameter substrings**. For example, a person's name, **N**, can contain a Family Name, a Given Name, an Other Names, Prefix and a Suffix. The property value is a concatenation of the Family Name (first field), Given Name (second field), Additional Names (third field), Name Prefix (fourth field), and Name Suffix (fifth field) strings separated by the Field Delimiter ";".

```
N:Smith;James;Q.;Mr.;Jr
```

An unused substring, if not at the end of the list of substrings, is represented with a semicolon only:

```
N:Smith;James;Q.;;Jr
```

Vcards may be organized to contain groups. The grouping of a comment property with a telephone property is shown in the following example:

```
A.TEL;Home:+1-213-555-1234
A.NOTE:This is my vacation home.
```

Below are some commonly used vCard properties, with substrings. Separate lines should not be used for field substrings, but are used here for clarity:

Formatted Name:

```
FN:Text Value
```

Example:

```
FN:Dr. Thomas D. Wason, Sr.
```

Name:

```
N:Family (Sur) Name;
First (Given) Name;
Other Names;
Prefix;
Suffix CRLF
```

Example:

```
N:Wason;Thomas;D.;Dr.;Sr
```

Address:

The property value is a concatenation of the Post Office Address (first field), Extended Address (second field), Street (third field), Locality (fourth field, e.g., City), Region (fifth field, e.g., State), Postal (Zip) Code (six field), and Country (seventh field) strings:

```
ADR:P.O.Box;
: Extended Address
Street;
Locality;
Region;
Postal Code;
Country CRLF
```

Example:

```
ADR:;IMS Project;1421 Park Drive;Raleigh;North Carolina;27605-
1727;United States of America
```

Address Delivery Label:

```
LABEL: Text
```

Example:

```
LABEL;QUOTED-PRINTABLE:IMS Project=0A=
1421 Park Drive=0A=
Raleigh, NC 27605-1727=0A=
```

Note the use of the escaped line feed (=0A=). The property parameters are preceded by a semicolon (;) after the property name. They are optional, defining the uses of the Delivery Label.

Organization:

The property value is a concatenation of the Organization Name (first field), Organizational Unit (second field) strings. Additional positional fields, if specified, contain additional Organizational Units:

```
ORG:Organization Name;
   Organizational Unit[;
   Organizational Unit] CRLF
```

Example:

```
ORG:IMS Project;Meta Data Team
```

Electronic Mail:

```
EMAIL; Electronic Mail Type:email
```

Example:

```
EMAIL;INTERNET:twason@imsproject.org
```

Telephone:

```
TEL:telephone number
```

Example:

```
TEL:+1 919.839.8187
```

All of these previously described properties are included in the following example:

```
BEGIN:VCARD
N:Wason;Thomas;D.;Dr.;Sr.
FN:Thomas D. Wason, Ph.D.
ORG:IMS Project;Meta Data Team
ADR:;IMS Project;1421 Park Drive;Raleigh;North Carolina;27605-
1727;United States of America
TEL:+1 919.839.8187
EMAIL;INTERNET:twason@imsproject.org
LABEL;QUOTED-PRINTABLE:IMS Project=0A=
1421 Park Drive=0A=
Raleigh, NC 27605-1727=0A=
USA
END:VCARD
```

Sample Meta-data Record

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE record SYSTEM "http://www.imsproject.org/XML/IMS-MD01-lc.dtd">
<!-- Reference to master lowercase DTD at IMS site.  -->
<!--DOCTYPE record SYSTEM "IMS-MD01-lc.dtd" -->
<!-- Use this doc type declaration for references to a local dtd. -->
<record xmlns="http://www.imsproject.org/metadata/">
  <!-- 2000=05-12, Thomas D. Wason: Sniffy01-lc.xml.  A complete meta-data set.  A few empty fields.
Located at http://www.ims project.org/xml/Sniffy01-lc.xml -->
  <!-- The full IEEE - IMS Learning Object Meta-data in XML.  -->
  <metametadata>
    <catalogentry>
      <catlogue>IMS-Test</catlogue>
      <entry>
        <langstring>2000.000001</langstring>
      </entry>
    </catalogentry>
    <contribute>
      <role>
        <langstring lang="en">Author</langstring>
      </role>
      <centity>
        <!-- Simple vCard example -->
        <vcard>
BEGIN: vCard
N:Wason;Thomas;D.;Dr.;Sr.
FN:Thomas D. Wason, Ph.D.
ORG:IMS Project;Meta Data Team
ADR:;IMS Project;1421 Park Drive;Raleigh;North Carolina;27605-1727;United States of America
TEL:+1 919.839.8187
EMAIL;INTERNET:twason@imsproject.org
LABEL;QUOTED-PRINTABLE:IMS Project=0A=
1421 Park Drive=0A=
Raleigh, NC 27605-1727=0A=
USA
END: vCard
</vcard>
      </centity>
      <date>
        <datetime>1999-08-05</datetime>
      </date>
    </contribute>
    <metadatascheme>IEEELOM:1.0</metadatascheme>
    <language>en-US</language>
    <!-- English as default meta-data language.  -->
  </metametadata>
  <general>
    <title>
      <langstring lang="en-US">Sniffy The Virtual Rat</langstring>
      <langstring lang="fr">Sniffy Virtuel le Rat</langstring>
    </title>
    <catalogentry>
      <catlogue>ISBN</catlogue>
      <entry>
        <langstring>0-534-26702-5</langstring>
```

```
    </entry>
   </catalogentry>
   <language>en-US</language>
   <description>
    <!--English description-->
    <langstring lang="en-US">A computer program that enables students to explore the prinicples of shaping and partial reinforcement in operant conditioning, using a "virtual rat" named Sniffy. Each student learns by doing-conditioning his or her own rat-and experiences many benefits of animal experimentation but none of the drawbacks associated with using live animals.</langstring>
    <!--French Description -->
    <langstring lang="fr">Un programme machine qui permet &#224; des &#233;tudiants d'explorer les prinicples de la formation et du renfort partiel dans l'op&#233;rateur conditionnant, utilisant " un rat virtuel " a nomm&#233; Sniffy. Chaque &#233;tudiant apprend par le faire-conditioning ses propres rat-et &#233;prouve beaucoup d'avantages l'exp&#233;rimentation animale de mais aucune des inconv&#233;nients associ&#233;s &#224; utiliser les animaux vivants. </langstring>
   </description>
   <keywords>
    <!--English Keywords, unordered list-->
    <langstring lang="en">operant conditioning</langstring>
    <langstring lang="en">psychology</langstring>
    <langstring lang="en">simulation</langstring>
    <langstring lang="en">program</langstring>
    <langstring lang="en">shaping</langstring>
    <langstring lang="en">rat</langstring>
    <langstring lang="en">learn by doing</langstring>
   </keywords>
   <keywords>
    <!--French keywords, unordered list-->
    <langstring lang="fr">traitement d'op&#233;rateur</langstring>
    <langstring lang="fr">psychologie </langstring>
    <langstring lang="fr">simulation</langstring>
    <langstring lang="fr">programme</langstring>
    <langstring lang="fr">formation</langstring>
    <langstring lang="fr">rat</langstring>
    <langstring lang="fr">apprenez en faisant</langstring>
   </keywords>
   <coverage>
    <langstring/>
   </coverage>
   <structure>
    <langstring>Hierarchical</langstring>
   </structure>
   <aggregationlevel>2</aggregationlevel>
  </general>
  <lifecycle>
   <version>
    <langstring>4.5</langstring>
   </version>
   <status>
    <langstring lang="en">Final</langstring>
   </status>
   <!--Contains an unordered list of  contribute-->
   <contribute>
    <role>
     <langstring lang="en">Author</langstring>
    </role>
```

```
      <!--Contains an ordered list of centity-->
      <centity>
        <vcard>
BEGIN: vCard
FN:Lester Krames
END:vCard
</vcard>
      </centity>
      <centity>
        <vcard>
BEGIN:vCard
FN:Jeff Graham
END:vCard
</vcard>
      </centity>
      <centity>
        <vcard>
BEGIN:vCard
FN:Tom Alloway
END:vCard
</vcard>
      </centity>
      <date>
        <datetime>1995</datetime>
      </date>
    </contribute>
    <contribute>
      <role>
        <langstring lang="en">Technical Imple menter</langstring>
      </role>
      <centity>
        <vcard>
BEGIN:vCard
FN:Greg Wilson
END:vCard
</vcard>
      </centity>
    </contribute>
    <contribute>
      <role>
        <langstring lang="en">Publisher</langstring>
      </role>
      <centity>
        <vcard>
BEGIN:vCard
ORG:Brooks/Cole publishing;International Thomson Publishing Company
END:vCard
</vcard>
      </centity>
      <date>
        <datetime>1995</datetime>
      </date>
    </contribute>
  </lifecycle>
  <technical>
    <format/>
```

```
  <size>1032353</size>
  <location type="URI">http://www.brookscole.com</location>
  <requirements>
    <type>
      <langstring lang="en">Operating System</langstring>
    </type>
    <name>
      <langstring lang="en">MS-DOS</langstring>
    </name>
    <minimumversion>5.0</minimumversion>
    <maximumversion/>
  </requirements>
  <installationremarks>
    <langstring lang="en">Load from diskette</langstring>
  </installationremarks>
  <otherplatformrequirements>
    <langstring lang="en">Will not run in browser.</langstring>
  </otherplatformrequirements>
  <duration>
    <datetime>0000-00-00T01:20</datetime>
  </duration>
 </technical>
 <educational>
  <interactivitytype>
    <langstring>Active</langstring>
  </interactivitytype>
  <learningresourcetype>
    <langstring lang="en">Simulation</langstring>
  </learningresourcetype>
  <interactivitylevel>3</interactivitylevel>
  <semanticdensity>2</semanticdensity>
  <intendedenduserrole>
    <langstring lang="en">Learner</langstring>
  </intendedenduserrole>
  <!--May be an ordered list of intended user roles with the most relevant first.-->
  <learningcontext>
    <langstring lang="en">Secondary Education</langstring>
  </learningcontext>
  <typicalagerange>
    <langstring>12-99</langstring>
  </typicalagerange>
  <difficulty>2</difficulty>
  <typicallearningtime>
    <datetime>0000-00-00T03:00</datetime>
  </typicallearningtime>
  <description>
    <langstring lang="en">Interactive teaching of the concepts of operant conditioning.</langstring>
  </description>
  <language>en_US</language>
 </educational>
 <rights>
  <cost>
    <langstring lang="en">yes</langstring>
  </cost>
  <copyrightorotherrestrictions>
    <langstring>yes</langstring>
```

```
    </copyrightorotherrestrictions>
    <description>
      <langstring lang="en">Copyright 1995 Brooks Cole Publishing</langstring>
      <langstring lang="en">Contact publisher to purchase</langstring>
    </description>
  </rights>
  <relation>
    <kind>
      <langstring lang="en">IsReferencedBy</langstring>
    </kind>
    <resource>
      <!--identifier reserved for later use.  Do not use.-->
      <description>
        <langstring lang="en">Companion book of the same name.</langstring>
      </description>
    </resource>
  </relation>
  <annotation>
    <centity>
      <vcard>
BEGIN:vcard
FN:Stuart Sutton
ORG:GEM;Syracuse University
END:vcard</vcard>
    </centity>
    <date>
      <datetime>1999-06-10</datetime>
    </date>
    <description>
      <langstring lang="en-US">A useful simulation for the student new to concepts of behavioral
science.</langstring>
    </description>
  </annotation>
  <classification>
    <purpose>
      <langstring lang="en">Discipline</langstring>
    </purpose>
    <taxonpath>
      <source>Dewey</source>
      <!--Ordered list of Taxons-->
      <taxon>
        <id>300</id>
        <entry>
          <langstring lang="en">Social Sciences</langstring>
        </entry>
      </taxon>
      <taxon>
        <id>320</id>
        <entry>
          <langstring lang="en">Political Science</langstring>
        </entry>
      </taxon>
    </taxonpath>
    <taxonpath>
      <source>LCSH</source>
      <!--Ordered list of Taxons-->
```

```
      <taxon>
        <id>B</id>
        <entry>
          <langstring lang="en">Philosophy, Psychology, Religion</langstring>
        </entry>
      </taxon>
      <taxon>
        <id>F</id>
        <entry>
          <langstring lang="en">Psychology</langstring>
        </entry>
      </taxon>
      <taxon>
        <id>180</id>
        <entry>
          <langstring lang="en">Experimental Psychology</langstring>
        </entry>
      </taxon>
    </taxonpath>
    <description>
      <langstring lang="en-US">principles of operant conditioning</langstring>
    </description>
    <keywords>
      <langstring lang="en">operant conditioning</langstring>
      <langstring lang="en">psychology</langstring>
    </keywords>
  </classification>
</record>
```

Appendix

Additional Resources

IMS XML Documents

IMS_METADTAv1p1.dtd is located at:
http://www.imsproject.org/xml/IMS_METADATAv1p1.dtd

IMS_MDv1p1.xml is located at: http://www.imsproject.org/xml/IMS-MDv1p1.xml

IMS Meta-data Documents

The IMS Resource Meta-data Best Practice and Implementation Guide can be found at:
http://www.imsproject.org/metadata/mdbestv1p1.html

The IMS Learning Resource Meta-data Information Model document can be found at:
http://www.imsproject.org/metadata/mdinfov1p1.html

ISO/IEC 10646

ISO (International Organization for Standardization). ISO/IEC 10646-1993 (E). Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane. [Geneva]: International Organization for Standardization, 1993 (plus amendments AM 1 through AM 7).

Unicode

The Unicode Consortium. The Unicode Standard, Version 2.0. Reading, Mass.: Addison-Wesley Developers Press, 1996.

vCard

vCard Specification http://www.imc.org/pdi/

XML

XML Version 1.0 specification of the W3C: http://www.w3.org/TR/1998/REC-xml-19980210

XML Namespace Recommendation of W3C: http://www.w3.org/TR/1999/REC-xml-names-19990114