Universal Learning Format (ULF)

Technical Specification

Version 1.0

Copyright

Trademarks

Contacting Saba

Telephone	(650) 696-3840
Fax	(650) 696-1773
Mailing Address	2400 Bridge Parkway Redwood Shores, CA 94065-1166 USA
Web	http://www.saba.com

Contents

Chapter 1

Overview of Universal Learning Format

This guide provides detailed documentation for Universal Learning Format, a modular set of XML-based formats developed by Saba for capturing various types of e-learning data, including online learning content, catalogs of learning resources, certification libraries, competency libraries, and learner information.

This chapter provides an overview of Universal Learning Format and introduces its principal components.

Specifically, it consists of the following topics:

Overview of Universal Learning Format

Choosing to describe your learning-based resources in any one preferred file format carries the risk of vendor lock-in and obsolescence. To minimize this risk, it is important to select a format that adheres closely to standards. However, ongoing efforts to develop learning standards are splintered and highly specialized and do not yield a comprehensive solution.

To address this problem Saba has developed the Universal Learning Format, a set of XML-based formats for capturing various types of e-learning data, including content, catalog, certification, competency, and learner profile information.

Universal Learning Format is intended for use by authors and integrators of learning-based resources to describe learning data in a format that allows universal portability, including import into Saba Learning Enterprise using the XML Bulk Import utility.

The formats comprising Universal Learning Format are based on and can be mapped (using style sheets) to and from IMS, ADL, IEEE LTSC, Dublin Core, and other specifications. Saba strongly believes in the value of open standards and is committed to maintaining compliance with all industry standards.

The table below lists and describes the component formats of Universal Learning Format:

Table 1-1: *Components of Universal Learning Format*

Format	Usage
Catalog Format	Use to import online course content
Learning Content Format	Use to import competency library definitions
Competency Format	Use to import certification library definitions
Certification Format	Use to import catalog entries to populate the Learning Catalog
Profile Format	Use to import learner profile data Saba provides two Profile formats, one for importing profile data for internal employees and another for importing profile data for clients.

The following diagram illustrates the architecture of Universal Learning Format:

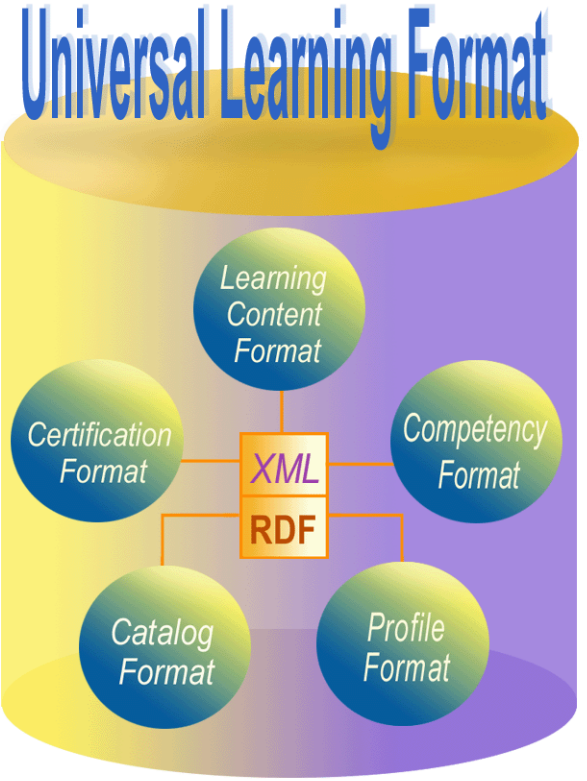

Figure 1-1: *Universal Learning Format*

Catalog Format

Catalog Format represents all information necessary to exchange catalogs of learning resources, focusing in particular on the data needed to locate and discover learning resources. It represents all catalog information as XML/RDF metadata, and incorporates several existing metadata standards, including:

- **IMS Learning Objects Metadata (LOM)** — A set of metadata elements for describing learning resources. IMS is a standards body focused on Internet-based learning technology. Its metadata specification is a subset of compatible work by the IEEE Learning Technology Standards Committee.
- **Dublin Core** — A set of metadata elements defined by a body of experts in library and bibliographic science. Dublin Core defines a set of 14 attributes, such as **title** and **author**, which are broadly applicable to practically any document on the web.
- **vCard** — A set of metadata elements defined by the Internet Engineering Task Force (IETF) as a standard for representing information about people and organizations such as that which is profiled on a common business card.

Catalog Format also addresses business needs not met by IMS metadata alone. It specifies the full set of data needed for online learning catalogs by defining price and ordering information, schedules and locations, and provided competencies.

By incorporating existing standards, Catalog Format ensures broad compatibility with both LOM data and the variety of RDF and Dublin Core tools for working with web metadata. By addressing critical unspecified areas, it captures all the information needed to create an effective model for publishing catalogs of learning resources. As a result, by employing Catalog Format, a learning provider can create a single course catalog that accomplishes all of the following:

- Establishes relationships with competency libraries for performance tracking
- Enables searches on critical metadata such as copyrights, authorship, keywords, and technical requirements
- Enables ordering of products using a variety of delivery modes and currencies
- Ensures compatibility with standard web search engines

For more information on Catalog Format, see “Catalog Format Specification”.

Learning Content Format

Learning Content Format (LCF) is an interchange format for online learning content. Several standards related to online content and courses are currently in the process of being defined, including IMS Packaging Format, IMS Question & Test, and ADL Course Structure Format. LCF adopts these standards and consolidates their best features into a stable and comprehensive format for describing online learning content.

LCF defines the structure and contents of online educational courses and content. It unifies all the information required for defining online learning, including:

- Flexible table of contents definitions
- Local and remote references to content resources
- Course structure and objectives
- Assessments for a variety of purposes, including tests, evaluations, and surveys

With LCF, a learning provider can structure learning resources as reusable chunks, then flexibly recombine them to create new and modified courses.

For more information on Learning Content Format, see “Learning Content Format Specification”.

Competency Format

Competency Format is an interchange format for competency-related information. A competency is a skill, knowledge, or behavior that can be measured, calculated, acquired, specified, or tested. In order to maximize human performance, organizations need to effectively track and manage competencies held or required by people throughout the extended enterprise. Successfully modeling an organization's competencies typically requires the ability to reuse and refine existing competency models.

Competency Format defines the structure of competency categories, levels, profiles, and behaviors. It can represent competency models across a variety of roles and industries, and allows competencies to be associated with other resources, such as learning offerings described in Catalog Format. By using Competency Format, a learning provider can create a web-based competency library that can be easily referenced and reused by a variety of online training.

For more information on Competency Format, see "Competency Format Specification".

Certification Format

Certification Format is an interchange format for certification-related information. A certification is a group of learning offerings that a learner must complete in order to gain a certification or be qualified in a particular technology, educational area, or field. Organizations need to track learner certifications for a variety of reasons, ranging from standard business procedures to customer and legal requirements. The ability to define and reuse standard certification models is critical to ensuring this compliance. Certification Format addresses the need for a standard way to exchange certification-related information.

Certification Format defines the structure of certification paths. A certification path represents the set of educational offerings and competencies needed to achieve a given certification. By using Certification Format, a learning provider can create a web-based certification library that easily integrates with other online formats, including catalogs and competencies.

For more information on Certification Format, see “Certification Format Specification”.

Profile Format

Profile Format is an XML-based representation for describing learner profile information. Learner profiles comprise a variety of data about learners, including personal and job information, learning history, goals and plans, and held competencies and certifications. Profile Format captures this information in an XML-based format using RDF to define metadata for describing learners. Profile Format incorporates several existing metadata standards, including the Dublin Core and vCard, which ensures compatibility with existing person/profile descriptions.

By employing Profile Format to describe the learners in a system, learning providers can extend their learning management architecture to support all of the following:

- Searches of critical learner metadata such as name, title, role, learning results, and held competencies and certifications
- Tracking the learning history of individual learners
- Assignment of competencies (with proficiency levels) and certifications to learners
- Assignment of learning goals to learners and tracking of progress towards fulfillment of those goals
- Creation of distributed profiles, where portions of a learner's profile are provided by different sources
- Compatibility with standard web search engines

For more information on Profile Format, see "Profile Format Specification".

Integration of Universal Learning Formats

The formats defined by Universal Learning Format integrate to provide a full range of capabilities for associating learning offerings with competencies and certifications, updating learner profiles, and querying all types of e-learning data.

Specifically, the interdependencies between the XML formats include:

- Catalog Format provides a metadata representation for "cataloging" information described in Learning Content Format, Competency Format, and Certification Format documents. Catalog entries in a Catalog Format document can refer to a learning resource, that is, a Learning Content Format document, a competency library, that is, a Competency Format document, or a certification path, that is, a Certification Format document. Thus, Catalog Format provides a consistent way of describing common metadata about all types of learning resources.
- You can embed properties from the Competency RDF schema within a Catalog Format document in order to define the competencies provided by a particular learning offering. This enables users to search for all courses providing a specific competency.
- Profile Format provides a metadata representation for associating information described in Catalog Format, Competency Format, and Certification Format documents with a learner profile. You can associate completed and planned learning with a learner profile by referencing a Catalog Format document. You can associate held certifications with a learner profile by referencing a Certification Format document. You can associate held competencies with a learner profile by referencing a Competency Format document.
- Learning Content Format, Competency Format, and Certification Format all support a **metadataRef** attribute that references a catalog entry in a Catalog Format document. This provides a mechanism for associating a learning offering, competency, or certification with metadata for that object.
- Additionally, Certification Format can reference learning offerings (defined in a Catalog Format document), competencies (defined in a Competency Format document), or other certifications as options for completing a step along the path to achieve the certification.

The following diagram represents the network of interdependencies among the XML formats defined by the Universal Learning Format:

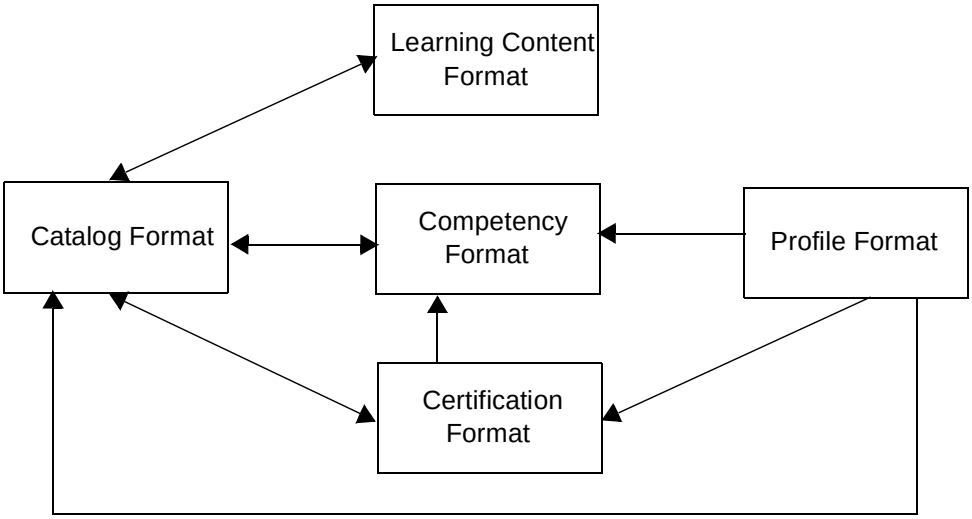

Figure 1-2: *Interdependencies Among Universal Learning Format Documents*

Catalog Format Specification

This document is the technical specification for Catalog Format, the component of Universal Learning Format designed for describing catalog information.

It includes the following sections:

Introduction

Catalog Format is an XML-based representation for describing catalog information. Catalog Format defines a set of elements used to classify the learning resources available for purchase or request through a learning catalog. It assumes that each learning resource is described using a unique identifier—either a URL or ID value—and has a set of property/value pairs that fully describes the resource. The properties supported by Catalog Format encompass both the information needed to search for particular learning resources and the information needed to order or register for the learning resources.

Usage

Catalog Format provides a solution for exchanging catalog descriptions about learning resources and querying course metadata. It also ensures that your catalog descriptions are compatible with any learning management system that supports either Catalog Format or any of its constituent industry standard formats.

Specifically, Catalog Format provides the following capabilities:

- **Build catalog descriptions** — Content providers can use Catalog Format to describe product catalogs, including metadata for prices, schedules, provided competencies, etc., to ensure maximum portability across systems.
- **Import catalog descriptions** — Content providers can import product catalogs described in Catalog Format into any learning management system that supports either Catalog Format or any of its constituent industry standard formats.
- **Exchange catalog descriptions** — Content providers can transport product catalogs described in Catalog Format from one learning management system to another using Catalog Format as the exchange format.
- **Support queries on catalog information** — Learners can search for information about learning resources described using Catalog Format. For example, a learner can search for all courses containing the word “Java” in the course title or course description, or a learner could search for all courses providing the “Java programming skills” competency.

Structure

Catalog Format is an XML-based representation designed to capture the information associated with an e-learning product catalog. A class hierarchy is used to classify the learning resources available for purchase through the product catalog.

Learning Objects Metadata (LOM)

Catalog Format is based on the Instructional Management Systems (IMS) interpretation of IEEE's standard for learning metadata known as Learning Objects Metadata (LOM). LOM is a standard for describing the relevant features of learning resources for search and discovery. Catalog Format fully supports LOM, while extending it with additional attributes needed to support e-learning catalogs.

IMS splits the LOM data model into two subsets:

- **Core LOM —** subset of the IEEE LOM proposal that consists of the most commonly used elements
- **SEL LOM —** remaining elements of the IEEE proposal

Catalog Format incorporates all of the properties defined in the IMS "Core LOM" data model and some of the properties defined in the "SEL LOM" group.

Resource Description Framework (RDF)

Catalog Format employs Resource Description Framework (RDF), an XML-based standard for defining metadata to describe web-based resources. Support for RDF makes it possible to define a set of unique RDF properties and merge these properties with properties defined in existing standards, such as LOM. This makes it possible for Catalog Format to represent the full set of data in an e-learning product catalog, as well as provide added value through tying features such as price lists and competencies back to the course descriptions. The consistent use of RDF also supports a unified mechanism for manipulating and querying this merged metadata.

Catalog Format Schemas

The data model used to describe resources in Catalog Format is defined in a set of external RDF schemas. Catalog Format documents must reference these schemas in an XML namespace in order to validate the properties they describe.

For example, in order to use properties from the Dublin Core schema, a Catalog Format document must define an XML namespace that references the URL of the Dublin Core schema, as follows:

```
xmlns:dc="http://purl.org/dc/elements/1.1/"
```

The following table lists and describes the RDF schemas used by Catalog Format:

Table 1: *Catalog Format Schemas*

Schema	Description
RDF	RDF is a standard framework for describing and interchanging metadata. The **RDF** schema represents a set of properties for describing web resources. It has the following URL: http://www.w3.org/1999/02/22-rdf-syntax-ns# The **RDFS** schema represents the meaning, characteristics, and relationships of the RDF properties. It has the following URL: http://www.w3.org/TR/1999/PR-rdf-schema-19990303# For more information on RDF and RDFS, see: http://www.w3.org/TR/REC-rdf-syntax/
Dublin Core	**Dublin Core** is a standard RDF schema comprising fifteen “core” elements that represent essential aspects related to the description of resources. A subset of the IMS metadata elements maps directly to the Dublin Core. The **Dublin Core Qualifiers** schema extends the descriptions of the fifteen “core” elements through the use of qualification and substructure. The Dublin Core schema has the following URL: http://purl.org/dc/elements/1.1/ The Dublin Core Qualifiers schema has the following URL: http://purl.org/dc/qualifiers/1.0/ For more information on support for Dublin Core within RDF, see: http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/
vCard	**vCard** is a standard RDF schema comprising elements that represent information about people and organizations such as that which is profiled on a business card. Some of the IMS metadata elements are vCard properties, which are represented using an RDF mapping. The vCard schema has the following URL: http://www.imc.org/vCard/3.0# Fore more information on support for vCard within RDF, see: http://www.dstc.edu.au/Research/Projects/rdf/draft-iannella-vcard-rdf-00.txt
Offerings	**Offerings** is an RDF schema designed to represent the hierarchy of courses, classes, and other learning resources available for purchase through an e-learning catalog. The Offerings schema has the following URL: http://www.saba.com/RDF/offering10.rdf To view the Offerings RDF schema, see Offerings Schema.

Table 1: *Catalog Format Schemas*

Schema	Description
IMS	**IMS** is an RDF schema designed to represent a subset of the IMS metadata elements. Saba's IMS schema has the following URL: http://www.saba.com/RDF/ims1.0.rdf To view the IMS RDF schema, see IMS Schema.
Scheduling	**Scheduling** is an RDF schema designed to represent information necessary for scheduling classes. The Scheduling schema has the following URL: http://www.saba.com/RDF/schedule10.rdf To view the Scheduling RDF schema, see Scheduling Schema.
Price List	**Price List** is an RDF schema designed to represent pricing and delivery-related information. The Price List schema is based on existing eCommerce specifications. Pricing information can be directly associated with Offering elements, or it can be defined in separate elements that reference the Offerings. The Price List schema has the following URL: http://www.saba.com/RDF/pricing10.rdf To view the Price List RDF schema, see Price List Schema.
Competency	**Competency** is an RDF schema designed to represent information about provided competency levels. The Competency schema has the following URL: http://www.saba.com/RDF/competency10.rdf To view the Competency RDF schema, see Competency Schema. **Note** For detailed information about the Competency RDF Schema, see the Competency Format Specification document.

Description Element

A Catalog Format document is an RDF document that contains one or more **Description** elements. Each **Description** element contains a unique identifier and a set of property/value pairs that describe a learning resource. These properties can draw from any of the Catalog Format RDF schemas.

Each **Description** element has an attribute that unambiguously identifies the learning resource being described. This attribute can be either of the following:

- **about**
- **id**

See below for more detailed information on using these identifier attributes.

The **Description** element can also include the **xml:lang** attribute for specifying the language in which the metadata description is authored. The **xml:lang** attribute contains the ISO 639/RFC 1766 language code with an optional geographic identifier, such as **en** for English, or **fr** for French.

Using the “about” Attribute

The **about** attribute specifies the URL of the resource, for example:

```
<rdf:Description about=”http://www.saba.com/courses/java101”
xml:lang=”en-US”>
...
</rdf:Description>
```

The **about** attribute is useful when the metadata description applies to a resource defined in another file, such as a WBT launched via a URL or a WBT defined as a Learning Content Format document. You can also define metadata descriptions for resources such as competency libraries and certification paths. In this case the **about** attribute references the URL for a Competency Format or Certification Format document.

You should AVOID using the **about** attribute to identify a learning resource when:

- the resource has multiple locations (such as multiple language versions of a course)
- the metadata description has other objects pointing to it (other objects can be an LCF document, a Competency Format document, a Certification Format document, or another metadata description).

Note Catalog Format documents generated for data defined in Saba, either for export or for use in metadata searches, can use the **about** attribute to indicate the internal ID of the described offering.

Using the “id” Attribute

The **id** attribute specifies a unique ID for the resource, for example:

```
<rdf:Description id=”JavaCourse” xml:lang=”en-US”>
...
</rdf:Description>
```

The **id** attribute is useful when the metadata description applies to a resource with multiple locations (such as multiple language versions of a course) OR when the metadata description is referenced by other objects (such as an LCF document, a Competency Format document, a Certification Format document, or another metadata description), in which case the referencing object requires the presence of an ID to be used as a handle.

For example, you would use the **id** attribute for a metadata description that is referenced by an LCF document using the **metadataRef** attribute. Competency Format and Certification Format documents can also use the **metadatRef** attribute to reference metadata descriptions. You might also use the **id** attribute to reference another metadata description that is an equivalent.

Note When you use the **id** attribute with the **Description** element, you must provide the URL for the learning resource in the IMS **location** property.

Class Hierarchy

Catalog Format defines a class hierarchy for classifying resources. Every resource described in Catalog Format must be associated with a class in the class hierarchy.

The class to which a resource belongs defines the following:

- the relationship of the resource to other resources
- the properties that can be used to describe the resource

The class hierarchy is defined in the **Offerings** RDF schema, which represents the relationships between classes and the inheritance logic used by the class hierarchy. To view the Offerings RDF schema, see Offerings Schema.

Other Catalog Format schemas define properties and associate those properties with classes in the class hierarchy. In this way the class to which a resource belongs determines the properties that can be used to describe it. For example, the properties for describing location and date information apply only to resources belonging to the **Event** class.

The table below lists and describes the classes defined in the Catalog Format class hierarchy:

Class	Description
Resource	**Resource** is the most general class. It is defined by the RDF schema specification itself. A Resource can have any Dublin Core properties as well as some certain IMS properties. All other classes inherit from Resource. The URL for the Resource class is: http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource
Offering	An **Offering** is any Resource that is available for purchase. All offerings can use any IMS properties and can have associated price lists and associated competencies. The URL for the Offering class is: http://www.saba.com/RDF/offering10.rdf#offering
Event	An **Event** is a subclass of Offering that has a specific date and location, such as a class. Events are purchased via registration. Event is equivalent to an Instructor Led Training (ILT). The URL for the Event class is: http://www.saba.com/RDF/offering10.rdf#event
Inventory	**Inventory** is a subclass of Offering that is physically delivered. The URL for the Inventory class is: http://www.saba.com/RDF/offering10.rdf#inventory
Online	**Online** is a subclass of Offering that is delivered online. Online is equivalent to a Web-Based Training (WBT). The URL for the Online class is: http://www.saba.com/RDF/offering10.rdf#online
Virtual Class	**Virtual Class** is a subclass of Online that is purchased via registration and delivered at a specific time. The URL for the Virtual Class class is: http://www.saba.com/RDF/offering10.rdf#virtualclass
Competency	**Competency** is a subclass of Offering that represents a competency library. The URL for the Competency class is: http://www.saba.com/RDF/offering10.rdfcompetency
Certification	**Certification** is a subclass of Offering that represents a certification path. The URL for the Certification class is: http://www.saba.com/RDF/offering10.rdf#certification

Class	Description
Package	**Package** is a subclass of Resource that groups a set of offerings into a single unit. **Note** The **Package** class must contain the **contents** property (see "Defining Content Packaging" on page 10). The URL for the Package class is: http://www.saba.com/RDF/offering10.rdf#package

Note The **Competency** and **Certification** classes allow standard metadata, such as authorship and relationships, to be associated with imported certification and competency libraries.

Specifying Class Type

Resources described in Catalog Format have a property called **type**, which comes from the RDF schema. The **rdf:type** property specifies the class within the Catalog Format class hierarchy to which the resource belongs.

To assign a resource to a particular class, specify the URL for that class, as defined in the Offerings RDF schema, for example:

```
<rdf:type resource="http://www.saba.com/RDF/offering10.rdf#offering"/>
```

or

```
<rdf:type resource="http://www.saba.com/RDF/offering10.rdf#event"/>
```

Assigning a resource to a class defines the relationship of the resource to other resources and the properties you can use to describe the resource.

Defining Offering Acronyms

The **Offerings** schema defines a property called **acronym**. You can use this property to specify a default acronym for any learning resource in the Offering class. The acronym cascades to all instances of the offering.

Defining Content Packaging

The **Offerings** schema defines a property called **contents**, which you can use in combination with the **Package** class to specify the resources contained in a package.

The **Package** class provides a convenient mechanism for grouping several offerings together. Grouping offerings into a package allows them to share a single price list, a scheduling template, a set of provided competencies, or other common properties. For example, you might want to package together three courses and sell them at a special 3-for-2 price. Defining a package lets you do this.

To create a package, you define a resource with class type **Package**. The resource must contain the **contents** property, whose value is an RDF Bag containing other learning resources. For example:

```
<rdf:Description id="Package1" xmlns:offering="http://www.saba.com/RDF/
offering10.rdf#">
  <dc:title>Gold Customers</dc:title>
  <rdf:type resource="http://www.saba.com/RDF/offering10.rdf#package"/>
  <offering:contents>
    <rdf:Bag>
      <rdf:li resource="#JAVA100"/>
      <rdf:li resource="#JAVA101"/>
    </rdf:Bag>
  </offering:contents>
</rdf:Description>
```

Summary of Catalog Format Properties

The table below lists and describes the properties defined by Catalog Format:

Table 2: *Catalog Format Properties (page 1 of 3)*

Schema	Property	Description	Applies To
RDF (rdf)	type	Class of resource	Resource
Dublin Core (dc)	title	Title	Resource
	identifier	Unique ID	Resource
	language	Language of resource	Resource
	description	Description	Resource
	creator	Creator	Resource
	publisher	Publisher	Resource
	contributor	Contributor	Resource
	date	Date	Resource
	format	Delivery format	Resource
	rights	Copyright information	Resource
	subject	Keywords	Resource
	type	Type of resource	Resource
	relation	Relationships	Resource
Dublin Core Qualifiers (dcq)	relationType	Equivalent, Prerequisite, InstanceOf, or Replaces	dc:relation
Offering (offering)	acronym	Acronym for learning resource	Offering
	contents	Contents of a package	Package

Table 2: *Catalog Format Properties (page 2 of 3)*

Schema	Property	Description	Applies To
IMS Metadata (ims)	catalogue	Source for identifier	Resource
	contribute	Identifies contributor	Resource
	version	Version of resource	Resource
	date	Date of contribution	ims:contribute
	role	Contributor's role	ims:contribute
	metadataScheme	RDF/LOM-1.0	Resource
	location	Location of resource	Resource
	cost	Is the resource free?	Resource
	copyright	Is the resource copyrighted?	Resource
	classification	Classification of resource	Offering
	purpose	Purpose of this classification	ims:classification
	description	Description of this classification	ims:classification
	keywords	Keywords for this classification	ims:classification
	status	Status of resource	Resource
	interactivity	Interactivity level	Offering
	target	Target audience	Offering
	difficulty	Difficulty level	Offering
	source	Source of taxonomy	ims:classification
	taxon	Taxonomy path	ims:classification
	learningTime	Typical learning time	Offering

Table 2: *Catalog Format Properties (page 3 of 3)*

Schema	Property	Description	Applies To
Scheduling (schedule)	location	Physical location	Event
	startDate	Start date	Event, Virtual Class
	template	Session template	Event, Virtual Class
Price List (price)	pricelist	List of Prices	Offering
	price	Price	Offering price:pricelist
	startDate	Start date for price	price:price
	endDate	End date for price	price:price
	currency	Currency for price	price:price
	quantityRange	Range of quantities	price:price
	min	Minimum quantity	price:quantityRange
	max	Maximum quantity	price:quantityRange
	method	Selling method	price:price
	target	Target of price list	price:price
Competency (competency)	competency	Provided competency	Resource

Note Since Catalog Format is expressed as an RDF document, you can associate any valid RDF property with a catalog entry, as long as you reference the RDF schema from which it derives. This openness provides a straightforward path for future extensibility.

Dublin Core Schema

Dublin Core is a standard metadata schema comprising fifteen "core" elements that represent essential data related to the description of resources on the web.

For more information on the Dublin Core, see the Dublin Core home page at:
http://purl.org/dc/

For more information on support for Dublin Core within RDF, see:
http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/

Catalog Format supports all fifteen Dublin Core properties, which are listed and briefly described in the table below:

Table 3: *Dublin Core Properties*

Property	Description
Title	Specifies the name given to the resource.
Creator	Specifies the name of the entity responsible for creating the resource.
Subject	Specifies the name of the topic of the resource.
Description	Specifies a description of the content of the resource.
Publisher	Specifies the name of the entity responsible for making the resource available.
Contributor	Specifies the name of an entity responsible for contributing to the content of the resource.
Date	Specifies a date associated with the life cycle of the resource. Uses ISO 8601format.
Type	Specifies the name of the genre to which the resource belongs.
Format	Specifies the physical or digital manifestation of the resource.
Identifier	Specifies a unique identifier by which the resource can be referenced.
Source	Specifies the identifier for the resource from which the current resource is derived.
Language	Specifies the language in which the content of the resource is authored. Uses ISO 639/RFC 1766 format.

Table 3: *Dublin Core Properties*

Property	Description
Relation	Specifies the identifier for a related resource.
Coverage	Specifies the extent or scope of the content of the resource.
Rights	Specifies information about the rights held in and over a resource.

vCard Schema

vCard is a metadata standard for representing information about people and organizations such as that which is profiled on a business card.

For more information about the vCard standard, see:
http://www.imc.org/pdi/

Fore more information on support for vCard within RDF, see:
http://www.dstc.edu.au/Research/Projects/rdf/draft-iannella-vcard-rdf-00.txt

Catalog Format supports the use of vCard properties to describe an entity (a person or organization). This occurs with the following Dublin Core properties:

- **dc:creator**
- **dc:contributor**
- **dc:publisher**

Catalog Format supports the full set of properties defined in the vCard specification. The table below lists and briefly describes a few of the more commonly used vCard properties:

Table 4: *Commonly Used vCard Properties*

Property	Description
FN	Specifies the person's full name.
N	Contains a series of substructure properties that define pieces of the person's name. The substructure properties include: • **Family** — family name • **Given** — given name • **Other** — additional name • **Prefix** — honorific prefix • **Suffix** — honorific suffix

Table 4: *Commonly Used vCard Properties*

Property	Description
ORG	Contains a series of substructure properties that define pieces of the organization's identification. The substructure properties include: • **Orgname** — name of the organization • **Orgunit** — name of the organizational unit
NICKNAME	Specifies the person's nickname.
TITLE	Specifies the person's title.

IMS Schema

Catalog Format is based on the IMS interpretation of IEEE's standard for learning metadata known as Learning Objects Metadata (LOM). LOM is a standard for describing the relevant features of learning resources for search and discovery.

To view the IMS LOM specification, see:
http://www.imsproject.org/metadata/

To view the full IEEE LOM specification, see:
http://ltsc.ieee.org/doc/wg12/LOM-WD3.htm

Catalog Format incorporates all of the properties defined in the IMS "Core LOM" data model as well as a subset of the optional elements known as "SEL IMS".

The following sections describe the implementation of these IMS data models within Catalog Format and the mapping of the IMS properties to RDF.

The IMS RDF schema has the following URL:
http://www.saba.com/RDF/ims10.rdf

To view the IMS RDF schema, see IMS Schema.

IMS Core Properties

The table below summarizes the mapping of the IMS “Core LOM” properties to RDF

Table 5: *Mapping of IMS “Core LOM” Properties to RDF*

IMS Core Property			RDF Property
Group	**Number**	**Name**	
General	1.2	Title	dc:title
	1.3.1	CatalogEntry.Catalogue	ims:catalogue
	1.3.2	CatalogEntry.Entry	dc:identifier
	1.4	Language	dc:language
	1.5	Description	dc:description
Lifecycle	2.1	Version	ims:contribute + ims:version
	2.3.1	Contribute.Role	ims:contribute + ims:role
	2.3.2	Contribute.Entity	dc:creator, dc:publisher, or dc:contributor
	2.3.3	Contribute.Date	dc:date or ims:contribute + ims:date
MetaMetaData	3.4	MetadataScheme	ims:metadataScheme
	3.5	Language	**xml:lang** attribute of Description
Technical	4.1	Format	dc:format
	4.3	Location	**about** attribute of Description or ims:location
Rights	6.1	Cost	ims:cost
	6.2	CopyrightandOtherRestrictions	ims:copyright
	6.3	Description	dc:rights

Table 5: *Mapping of IMS "Core LOM" Properties to RDF*

IMS Core Property			RDF Property
Group	**Number**	**Name**	
Classification	9.1	Purpose	ims:classification + ims:purpose
	9.3	Description	ims:classification + ims:description
	9.4	Keywords	dc:subject or ims:classification + ims:keywords

The table below lists and describes the IMS "Core LOM" properties defined in the Catalog Format schema:

Table 6: *IMS "Core LOM" Properties (page 1 of 5)*

Property	Description
Title (1.2)	Specifies the name of the learning resource. Corresponds directly to the Dublin Core **title** property.
CatalogEntry (1.3)	Specifies a unique designation for the learning resource. Consists of two substructure properties: • Catalogue — name of the catalog source (e.g., ISBN or Ariadne) • Entry — catalog designation (this is typically the part number for the resource) The **CatalogEntry.Catalogue** property is represented by the IMS **catalogue** property. The **CatalogEntry.Entry** property directly corresponds to the Dublin Core **identifier** property.
Language (1.4)	Specifies the language in which the learning resource is authored. Uses the ISO 639/RFC 1766 language code with an optional geographic identifier, such as **en** for English, or **fr** for French. Corresponds directly to the Dublin Core **language** property. **Note** You can use the RDF Alternative construct to specify multiple language values.

Table 6: *IMS "Core LOM" Properties (page 2 of 5)*

Property	Description
Description (1.5)	Specifies a text description for the contents of the learning resource. Corresponds directly to the Dublin Core **description** property.
Version (2.1)	Specifies an identifier to represent the version or edition of the learning resource. Represented by the IMS **version** property.

Contribute (2.3)	Identifies a person or organization contributing to the learning resource. A resource can have multiple contributors.
	The **Contribute** property consists of three substructure properties:
	• **Role** — identifies the type of contributor. **Contribute.Role** accepts any value, such as the following common values:
	  • **author** — specifies the author of a learning resource
	  • **contributor** — specifies a contributor to a learning resource
	  • **publisher** — specifies the publisher of a learning resource
	  • **vendor** — specifies the vendor of a learning resource
	• **Entity** — describes the contributor. **Contribute.Entity** can be a person or an organization.
	• **Date** — dates the contribution
	Each of the **Contribute** substructure properties maps to:
	• the IMS **contribute** property, and
	• one of three Dublin Core properties:
	  • **creator**
	  • **publisher**
	  • **contributor**
	The mapping uses the following algorithm:
	• **Role** — **Contribute.Role** maps to the IMS **contribute** property.
	• **Entity** — **Contribute.Entity** maps to one of the Dublin Core properties, based on the value of the corresponding **Contribute.Role** property, as follows:
	  • when **Contribute.Role** = **author, Contribute.Entity** directly corresponds to the Dublin Core **creator** property
	  • when **Contribute.Role** = **publisher**, **Contribute.Entity** directly corresponds to the Dublin Core **publisher** property
	  • for any other value of **Contribute.Role**, **Contribute.Entity** directly corresponds to the Dublin Core **contributor** property

Table 6: *IMS "Core LOM" Properties (page 3 of 5)*

Property	Description

Contribute (2.3)
(Cont'd)
	• **Date** — **Contribute.Date** maps to the Dublin Core **date** property or the IMS **contribute** property, based on the value of the corresponding **Contribute.Role** property, as follows:
	  • when **Contribute.Role** = **publisher** or **author**, **Contribute.Date** directly corresponds to the Dublin Core **date** property
	  • for all other values of **Contribute.Role**, **Contribute.Date** maps to the IMS **contribute** property
	The IMS **contribute** property has the following substructure properties:
	• IDREF — identifies the Dublin Core property it matches
	• role — corresponds to **Contribute.Role**
	• date — corresponds to **Contribute.Date**
	The Dublin Core **creator**, **publisher**, and **contributor** properties can be a single value or an RDF Sequence.
	• The values for these properties are described using an RDF mapping of the vCard standard.
	• These properties can also have an **id** attribute.

MetadataScheme (3.4)	Identifies the structure used for the metadata. Represented by the IMS **metadataScheme** property with a fixed value of **RDF/LOM-1.0**.
Language (3.5)	Specifies the language in which the metadata description is authored (as opposed to the language of the learning resource). Uses the ISO 639/RFC 1766 language code with an optional geographic identifier, such as **en** for English, or **fr** for French. Represented by the **xml:lang** property on the enclosing **Description** element.
Format (4.1)	Specifies the delivery format of the learning resource. Value can be: • a standard MIME type (e.g., **video**, **mpeg**, **text/html**, **text/xml** etc.) • a physical inventory delivery type (**book**, **floppy disk**, **CD-ROM**, **video tape**) • **non-digital** Corresponds directly to the Dublin Core **format** property.

Table 6: *IMS "Core LOM" Properties (page 4 of 5)*

Property	Description

Location (4.3)	Specifies the URL of the resource.
	Typically, this information is represented using the **about** attribute of the **Description** element. However, when the **Description** element identifies a resource by the **id** attribute rather than the **about** attribute, you must use the **location** property to specify the URL of the resource. This occurs in the following cases:
	• The resource has multiple locations, such as multiple language versions of the same course.
	• The resource is referenced by other objects, which require the presence of an ID to be used as a handle.
	The **Location** property consists of one or more URLs listed in order of preference. Its value is expressed as a single value, an RDF Sequence, or an RDF Alternative, for example:
	`<ims:location>`
	`  <rdf:Alt>`
	`  <rdf:li xml:lang="en" rdf:resource=`
	`  "http://www.saba.com/courses/java101-en"/>`
	`  <rdf:li xml:lang="it" rdf:resource=`
	`  "http://www.saba.com/courses/java101-it"/>`
	`  </rdf:Alt>`
	`</ims:location>`

Cost (6.1)	Contains a boolean value used to specify whether the use of the resource requires payment. Represented by the IMS **cost** property. Value can be **true** or **false**, for example: `<ims:cost>true</ims:cost>`
CopyrightAndOther Restrictions (6.2)	Contains a boolean value used to specify whether the resource has copyrights or other restrictions on its use. Represented by the IMS **copyright** property. Value can be **true** or **false**, for example: `<ims:copyright>true</ims:copyright>`
Description (6.3)	Specifies a text description of the copyright held on the resource. Corresponds directly to the Dublin Core **rights** property. Its value can be a literal or an RDF Bag.

Table 6: *IMS "Core LOM" Properties (page 5 of 5)*

Property	Description
Classification (9)	Contains a set of properties describing various characteristics of the resource. A classification can include any information used to categorize the resource, such as a subject area or learning goal. The **Classification** property consists of three substructure properties: • **Purpose** — specifies the name of a particular characteristic of the learning resource • **Description** — specifies a description of the purpose • **Keywords** — specifies a list of keywords associated with the characteristic. Multiple keywords should be separated by commas and listed in order of relevance When the value of **Classification.Purpose** is either **Subject** or **Discipline**, **Classification.Keywords** directly corresponds to the Dublin Core **subject** property. For all other values of **Classification.Purpose**, Classification is represented by the IMS **classification** property containing the substructure properties **purpose**, **description**, and **keywords**.
Extension	Contains custom information added to a metadata description. Catalog Format uses the standard XML namespace mechanism to capture custom information. You can represent custom information by creating a namespace and defining custom properties using an appropriate RDF schema.
Langstring	Specifies the language in which text is authored. Uses the ISO 639/RFC 1766 language code with an optional geographic identifier, such as **en** for English, or **fr** for French. Represented by an **xml:lang** attribute on the property in question. **Note** You can use the RDF Alternative construct to specify multiple language values.

IMS “SEL LOM” Properties

The table below summarizes the mapping of the IMS “SEL LOM” properties to RDF:

Table 7: *Mapping of IMS Core Elements to RDF*

IMS SEL Property			RDF Property
Group	**Number**	**Name**	
LifeCycle	2.2	Status	ims:status
Educational	5.1	InteractivityType	ims:interactivity
	5.2	LearningResourceType	dc:type
	5.6	LearningContext	ims:target
	5.8	Difficulty	ims:difficulty
	5.9	TypicalLearningTime	ims:learningtime
Relation	7.1,7.2	Relation	dc:relation + rdf:value, dcq:relationType
Classification	9.2	TaxonPath	ims:classification + ims:source, ims:taxon

The table below lists and describes the IMS “SEL LOM” properties defined in the Catalog Format schema:

Table 8: *IMS “SEL LOM” Properties (page 1 of 4)*

Property	Description
Status (2.2)	Specifies the status of the learning resource. Value can be: • **Draft** • **Final** • **Revised** • **Unavailable** Represented by the IMS **status** property.
Interactivity Type (5.1)	Specifies the degree of interactivity provided by this learning resource. Value can be: • **Expositive** — resource in which information flows from the resource to the learner. Expositive documents are typically used for learning-by-reading, for example essays, video clips, all kinds of graphical material and hypertext documents. • **Active** — resource in which information flows from the learner to the resource. Active documents are typically used for learning-by-doing, for example simulations, questionnaires and exercises. • **Mixed** • **Undefined** Represented by the IMS **interactivity** property.
Learning ResourceType (5.2)	Specifies the type of learning resource. Accepts any value. Common values include exercise, simulation, diagram, exam, graph, etc. Corresponds directly to the Dublin Core **type** property. The value of the **type** property can be expressed as a single value or an RDF Sequence.
Learning Context (5.6)	Specifies the typical user type for the learning resource, such as university, professional development, manager, learner, etc. Represented by the IMS **target** element. The value of the **target** property can be expressed as a single value or an RDF Sequence.

Table 8: *IMS "SEL LOM" Properties (page 2 of 4)*

Property	Description
Difficulty (5.8)	Specifies the relative difficulty of the learning resource, on a scale of 0-4. Value can be: • **0** — very easy • **1** — easy • **2** — medium • **3** — difficult • **4** — very difficult Represented by the IMS **difficulty** element.
Typical Learning Time (5.9)	Specifies the approximate or typical amount of time it takes to work with the resource. Uses ISO 8601 format. For example, to indicate a duration of one hour thirty minutes, use **PT1H30M**. Represented by the IMS **learningTime** element. **Note** Can contain an IMS **description** property that further describes the time. In this case, the **learningTime** value must be in an **rdf:value** property.
Relation (7)	Specifies the relationship of this resource to other resources. Directly corresponds to the Dublin Core **relation** property. The **Relation** property consists of two substructure properties: • **value** (from the RDF schema) — identifies the related resource • **relationType** (from the Dublin Core Qualifiers schema) — identifies the type of relationship The **relationType** property accepts the following values: • **Equivalent** — indicates equivalent resources, such as a book and a course that provide the same learning content • **Prerequisite** — indicates a resource that is a prerequisite for the resource being described • **InstanceOf** — indicates inheritance, such as a class that is an instance of a course • **Replaces** — indicates that one resource replaces another. For example, you can use this value to indicate a replacement certification for a discontinued certification.

Table 8: *IMS "SEL LOM" Properties (page 3 of 4)*

Property	Description

Relation
(Cont'd)
	The following example describes two resources, a course and a book, which are defined to have an **equivalent** relationship:
	`<rdf:Description about="JAVACOURSE">`
	`<dc:identifier>JAVA-101</dc:identifier>`
	`<rdf:type resource=`
	`"http://www.saba.com/RDF/offering10.rdf#offering"/`
	`>`
	`</rdf:Description>`
	`<rdf:Description id="JAVABOOK">`
	`<dc:identifier>JAVA-101T</dc:identifier>`
	`<rdf:type resource=`
	`"http://www.saba.com/RDF/`
	`offering10.rdf#inventory"/>`
	`<dc:relation rdf:parseType="Resource">`
	`<dcq:relationType>Equivalent</dcq:relationType>`
	`<rdf:value rdf:resource="#JAVACOURSE"/>`
	`</dc:relation>`
	`</rdf:Description>`

TaxonPath (9.2)	Specifies a taxonomy source and taxonomic path to be used for classifying the learning resource. **Note** Unless you are working with a pre-defined taxonomic system, Saba recommends following the Saba Learning Exchange taxonomy, which you can view at: http://www2.saba.com/exchange The **TaxonPath** property consists of two substructure properties: • **Source** — specifies the name of the taxonomy you are using to classify the resource. If you are using the Saba Learning Exchange taxonomy, the **source** property should have the value **Saba Exchange**. • **Taxon** — specifies one or more classification categories to describe the taxonomic path for the resource, moving from general to more specific, for example: `Industry/ Technology/ Programming/ HTML` The taxonomy path is typically expressed as an RDF Sequence. **TaxonPath** is represented by the IMS properties **source** and **taxon**, which are substructure properties of the IMS **Classification** property.

Table 8: *IMS "SEL LOM" Properties (page 4 of 4)*

Property	Description

TaxonPath (9.2) (Cont'd)	For example:
	`<rdf:Description id="HTML101">`
	`<dc:title>Introduction to HTML</dc:title>`
	`<rdf:type resource=`
	`"http://www.saba.com/RDF/offering10.rdf#offering"/`
	`>`
	`<ims:classification rdf:parseType=”Resource”>`
	`<ims:source>Saba Taxonomy</ims:source>`
	`<ims:taxon>`
	`<rdf:Seq>`
	`<rdf:li>Industry</rdf:li>`
	`<rdf:li>Technology</rdf:li>`
	`<rdf:li>Programming</rdf:li>`
	`<rdf:li>HTML</rdf:li>`
	`</rdf:Seq>`
	`</ims:taxon>`
	`</ims:classification>`
	`</rdf:Description>`

Scheduling Schema

Scheduling is an RDF schema defined by Catalog Format to represent the information necessary for scheduling classes, such as time and location. The properties defined in the Scheduling schema apply only to instances of the **Event** class.

The Scheduling schema has the following URL:
http://www.saba.com/RDF/schedule10.rdf

To view the Scheduling RDF schema, see Scheduling Schema.

The table below lists and describes the properties defined in the Scheduling schema:

Property	Description
Location	Specifies the physical location of an event.
StartDate	Specifies the date the event begins. Uses ISO 8601 format.

Property	Description
Template	Contains a text value for a session template used to specify the day and time the class is given, for example: **MWF9-5**
MinimumLearners	Specifies the minimum number of learners that are required for enrollment in a class for the class to be held.
MaximumLearners	Specifies the maximum number of learners that can be enrolled in a class.

For example, the following fragment of a Catalog Format document defines an offering called "JAVACOURSE" and two classes "JAVACOURSE100" and "JAVACOURSE101" that are instances of that offering:

```
<rdf:Description about="JAVACOURSE">
   <dc:identifier>JAVA-101</dc:identifier>
   <rdf:type resource="http://www.saba.com/RDF/
offering10.rdf#offering"/>
</rdf:Description>

<rdf:Description id="JAVACOURSE100">
   <rdf:type resource="http://www.saba.com/RDF/offering10.rdf#event"/>
   <dc:relation rdf:parseType="Resource">
      <dcq:relationType>InstanceOf</dcq:relationType>
      <rdf:value rdf:resource="#JAVACOURSE"/>
   </dc:relation>
   <schedule:location>Redwood Shores, CA</schedule:location>
   <schedule:startDate>1999-12-13</schedule:startDate>
   <schedule:template>MWF 9-5</schedule:template>
   <schedule:minimumlearners>10</schedule:minimumlearners>
   <schedule:maximumlearners>25</schedule:maximumlearners>
</rdf:Description>

<rdf:Description id="JAVACOURSE101">
   <rdf:type resource="http://www.saba.com/RDF/offering10.rdf#event"/>
   <dc:relation rdf:parseType="Resource">
      <dcq:relationType>InstanceOf</dcq:relationType>
      <rdf:value rdf:resource="#JAVACOURSE"/>
   </dc:relation>
   <schedule:location>Redwood Shores, CA</schedule:location>
   <schedule:startDate>1999-12-20</schedule:startDate>
   <schedule:template>MWF 9-5</schedule:template>
   <schedule:minimumlearners>3</schedule:minimumlearners>
   <schedule:maximumlearners>20</schedule:maximumlearners>
```

```
    </rdf:Description>
```

Price List Schema

Price List is an RDF schema defined by Catalog Format to represent pricing and delivery-related information and associate this information with catalog offerings.

The Price List schema has the following URL:
http://www.saba.com/RDF/pricing10.rdf

To view the Price List RDF schema, see Price List Schema.

You can define pricing information in either of two ways:

- directly within the associated catalog offerings in a Catalog Format document
- in a separate Price List document that references catalog offerings

Defining pricing information in a separate Price List document allows for maximum flexibility and reuse. However, defining the information directly within a Catalog Format document provides added convenience.

Note If you define a separate Price List document that references a particular resource description, you can override some or all of the price list information for that resource by respecifying those properties in the resource description.

The table below lists and describes the properties defined in the Price List schema:

Property	Description
PriceList	When present, **pricelist** is the core element of the Price List schema. However, it is an optional property. The **pricelist** element can contain multiple **price** elements. The **pricelist** element can apply to any instance of the Offering class.
Price	When **pricelist** is not present, **price** is the core element of the Price List schema. All descriptive properties are contained within the **price** element. The **price** element can be a single value expressed as a simple price, in which case it uses the default system currency and is always orderable. Or, it can be a complex price with amount, currency, and availability information, in which case it is expressed as a structured value. When the **price** element is expressed as a structured value, the RDF **value** property is used to specify the unit price. The RDF **value** property can be expressed as a single value or as a collection of values, using an RDF Bag, Sequence, or Alternative. The **price** element can apply to the **pricelist** element, or it can apply to any instance of the Offering class.

Property	Description
StartDate	Specifies the start date for date range during which the price is effective. Uses ISO 8601 format. If only a single price is specified, startDate specifies the initial date when this resource can be ordered.
EndDate	Specifies the end date for date range during which the price is effective. Uses ISO 8601 format.
Currency	Specifies the currency value for this price. Uses the 3-character ISO 4217 format.
QuantityRange	Specifies the range of quantities to which this price applies. It is a structured value containing **Min** and **Max** attributes.
Method	Specifies the selling method for catalog offerings associated with the price list. The supported methods are: • **register** • **ship** • **rent** • **download** • **checkout** (used for items in a library)
Target	Specifies the URL of the offering to which the price applies. The URL can point to an entire Catalog Format document, in which case it applies to all resources described in the document. Or, it can refer to a specific resource description within a document, using the syntax: *URL#resource_description_id* The **target** property can be a expressed as a single value or an RDF Bag that refers to multiple offerings.

Simple Price Example

The following example expresses a simple price:

```
<price:price>500</price:price>
```

Multiple Currency Example

The following example represents a price expressed in multiple currencies. Notice that **startDate** and **endDate** apply to the entire price list in this example:

```
<Description id="BASICPRICE">
   <price:price rdf:parseType="Resource">
      <rdf:value>
         <rdf:Bag>
            <rdf:li parseType="Resource">
               <rdf:value>200</rdf:value>
               <price:currency>usd</price:currency>
            </rdf:li>
            <rdf:li parseType="Resource">
               <rdf:value>130</rdf:value>
               <price:currency>gbp</price:currency>
            </rdf:li>
         </rdf:Bag>
      </rdf:value>
      <price:method>register</price:method>
      <price:startDate>1999-01-01</price:startDate>
      <price:endDate>1999-12-31</price:endDate>
      <price:target rdf:resource=
      "http://www.saba.com/catalog/Microsoft.rdf#JAVA101"/>
   </price:price>
</Description>
```

Complex Price List Example

The following example represents a full Price List document containing a complex price list with multiple ranges and dates:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:price="http://www.saba.com/RDF/pricing10.rdf">
   <rdf:Description id="PRICE001">
      <price:price rdf:parseType=”Resource”>
         <rdf:value>
            <rdf:Bag>
               <rdf:li parseType="Resource">
                  <rdf:value>200</rdf:value>
                  <price:startDate>1999-01-01</price:startDate>
                  <price:endDate>1999-05-01</price:endDate>
                  <price:range rdf:parseType="Resource">
                     <price:min>1</price:min>
                     <price:max>20</price:max>
                  </price:range>
               </rdf:li>
```

```
                  <rdf:li parseType="Resource">
                     <rdf:value>200</rdf:value>
                     <price:startDate>1999-01-01</price:startDate>
                     <price:endDate>1999-05-01</price:endDate>
                     <price:range rdf:parseType="Resource">
                        <price:min>21</price:min>
                     </price:range>
                  </rdf:li>
                  <rdf:li parseType="Resource">
                     <rdf:value>200</rdf:value>
                     <price:startDate>1999-05-02</price:startDate>
                     <price:endDate>1999-10-01</price:endDate>
                     <price:range rdf:parseType="Resource">
                        <price:min>1</price:min>
                        <price:max>20</price:max>
                     </price:range>
                  </rdf:li>
                  <rdf:li parseType="Resource">
                     <rdf:value>200</rdf:value>
                     <price:startDate>1999-05-02</price:startDate>
                     <price:endDate>1999-10-01</price:endDate>
                     <price:range rdf:parseType="Resource">
                        <price:min>21</price:min>
                     </price:range>
                  </rdf:li>
               </rdf:Bag>
            </rdf:value>
            <price:method>ship</price:method>
            <price:target>
               <rdf:Bag>
                  <rdf:li resource=
                  "http://www.saba.com/catalog/Microsoft.rdf#JAVA101T"/>
                  <rdf:li resource=
                  "http://www.saba.com/catalog/Microsoft.rdf#COM101T"/>
               </rdf:Bag>
            </price:target>
         </price:price>
      </rdf:Description>
   </rdf:RDF>
```

Catalog Format Examples

IMS Core Standard Example

The following Catalog Format document represents the information defined in the **SniffyCr.xml** file used as the standard example for the XML binding of the IMS Core Metadata (found at http://www.imsproject.org/xml/SniffyCr.xml). It also augments this information with price and taxonomy data.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- An RDF mapping of the SniffyCr.xml file used as an example of Core
IMS Metadata -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:ims="http://www.saba.com/RDF/ims10.rdf"
xmlns:price="http://www.saba.com/RDF/pricing10.rdf"
xmlns:vCard="http://imc.org/vCard/3.0#">
   <rdf:Description about="http://www.brookscole.com/programs/sniffy"
      xml:lang="en-US">
      <ims:metadataScheme>RDF/LOM-1.0</ims:metadataScheme>
      <dc:title>Sniffy The Virtual Rat</dc:title>
      <ims:catalogue>ISBN</ims:catalogue>
      <dc:identifier>0-534-26702-5</dc:identifier>
      <dc:language>en-US</dc:language>
      <dc:description>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.</dc:description>
      <ims:version>4.5</ims:version>
      <dc:creator rdf:ID="001">
         <rdf:Seq>
            <rdf:li parseType="Resource">
            <vCard:FN>Lester Krames</vCard:FN>
            <vCard:N rdf:parseType="Resource">
               <vCard:Family>Krames</vCard:Family>
            </vCard:N>
            </rdf:li>
            <rdf:li parseType="Resource">
            <vCard:FN>Jeff Graham</vCard:FN>
```

```
                <vCard:N rdf:parseType="Resource">
                   <vCard:Family>Graham</vCard:Family>
                </vCard:N>
                </rdf:li>
                <rdf:li parseType="Resource">
                <vCard:FN>Tom Alloway</vCard:FN>
                <vCard:N rdf:parseType="Resource">
                   <vCard:Family>Alloway</vCard:Family>
                </vCard:N>
                </rdf:li>
             </rdf:Seq>
          </dc:creator>
          <ims:contribute rdf:parseType="Resource" rdf:idref="001">
             <ims:role>Author</ims:role>
             <ims:date>1995</ims:date>
          </ims:contribute>
          <dc:contributor rdf:parseType="Resource" rdf:id="002">
             <vCard:FN>Greg Wilson</vCard:FN>
             <vCard:N rdf:parseType="Resource">
                <vCard:Family>Wilson</vCard:Family>
             </vCard:N>
          </dc:contributor>
          <ims:contribute rdf:idref="002">
             <ims:role>Technical Implementer</ims:role>
          </ims:contribute>
          <dc:publisher rdf:parseType="Resource">
             <vCard:ORG rdf:parseType="Resource">
                <vCard:Orgname>Brooks/Cole publishing;International Thomson
Publishing Company</vCard:Orgname>
             </vCard:ORG>
          </dc:publisher>
          <dc:date>1995</dc:date>
          <ims:cost>true</ims:cost>
          <ims:copyright>true</ims:copyright>
          <dc:rights>
             <rdf:Bag>
                <rdf:li>Copyright 1995 Brooks Cole Publishing</rdf:li>
                <rdf:li>Contact publisher to purchase</rdf:li>
             </rdf:Bag>
          </dc:rights>
          <dc:subject>
             <rdf:Seq>
                <rdf:li>operant conditioning</rdf:li>
                <rdf:li>psychology</rdf:li>
             </rdf:Seq>
          </dc:subject>
```

```
        <ims:classification rdf:parseType=”Resource”>
           <ims:purpose>Skill Level</ims:purpose>
           <ims:description>Skill level required to employ this program
              </ims:description>
           <ims:keywords>Beginner</ims:keywords>
        </ims:classification>
        <ims:classification rdf:parseType=”Resource”>
           <ims:purpose>Group</ims:purpose>
           <ims:description>The category of this resource
              </ims:description>
           <ims:keywords>Software</ims:keywords>
        </ims:classification>
        <ims:classification rdf:parseType="Resource">
           <ims:source>Saba Taxonomy</ims:source>
           <ims:taxon>
              <rdf:Seq>
                 <rdf:li>Science</rdf:li>
                 <rdf:li>Physiology</rdf:li>
              </rdf:Seq>
           </ims:taxon>
        </ims:classification>
        <price:price rdf:parseType="Resource">
           <rdf:value>100</rdf:value>
           <price:startDate>1999-10-01</price:startDate>
        </price:price>
        <ims:learningTime>PT3H</ims:learningTime>
     </rdf:Description>
</rdf:RDF>
```

Multiple Instance Example

The following Catalog Format document illustrates the use of the Offerings hierarchy to represent several instances of the same learning offering.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xml:lang="en"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcq="http://purl.org/dc/qualifiers/1.0/"
xmlns:vCard="http://imc.org/vCard/3.0#"
xmlns:ims="http://www.saba.com/RDF/ims10.rdf"
xmlns:price="http://www.saba.com/RDF/pricing10.rdf"
xmlns:schedule="http://www.saba.com/RDF/schedule10.rdf"
xmlns:ewp="http://www.saba.com/RDF/competency10.rdf">
```

```
<rdf:Description id="JAVA100" xml:lang="en-US">
   <ims:metadataScheme>RDF/LOM-1.0</ims:metadataScheme>
   <dc:title>Introduction to Java</dc:title>
   <dc:description>Introduction to Java concepts.</dc:description>
   <dc:subject>Java</dc:subject>
   <rdf:type resource="http://www.saba.com/RDF/
offering10.rdf#offering"/>
   <ims:version>1.0</ims:version>
   <!-- competency provided by this course -->
   <ewp:competency>
      http://www.saba.com/competencies/programming.xml#Java.Beginner
   </ewp:competency>
</rdf:Description>

<rdf:Description id="JAVA101" xml:lang="en-US">
   <ims:metadataScheme>RDF/LOM-1.0</ims:metadataScheme>
   <dc:title>Intermediate Java</dc:title>
   <dc:description>In-depth programming with Java.</dc:description>
   <rdf:type resource="http://www.saba.com/RDF/
offering10.rdf#offering"/>
   <ims:version>1.0</ims:version>
   <dc:relation rdf:parseType="Resource">
         <!-- Introduction to Java is a prerequisite -->
         <dcq:relationType>Prerequisite</dcq:relationType>
         <rdf:value rdf:resource="#JAVA100"/>
   </dc:relation>
</rdf:Description>

<!-- offerings that inherit from JAVA100 -->

<!-- an English-language textbook -->

<rdf:Description id="JAVA101T-EN">
   <dc:title>The Java Tutorial</dc:title>
   <dc:identifier>1-55612-818-3</dc:identifier>
   <dc:language>en</dc:language>
   <rdf:type resource="http://www.saba.com/RDF/
offering10.rdf#inventory"/>
   <dc:relation rdf:parseType="Resource">
      <dcq:relationType>InstanceOf</dcq:relationType>
      <rdf:value rdf:resource="#JAVA101"/>
   </dc:relation>
</rdf:Description>

<!-- a French-language textbook -->
```

```
<rdf:Description id="JAVA101T-FR" xml:lang="fr">
   <dc:title>Le Cours d'instruction De Java</dc:title>
   <dc:identifier>1-55612-818-4</dc:identifier>
   <dc:language>fr</dc:language>
   <rdf:type resource="http://www.saba.com/RDF/
offering10.rdf#inventory"/>
   <dc:relation rdf:parseType="Resource">
      <dcq:relationType>InstanceOf</dcq:relationType>
      <rdf:value rdf:resource="#JAVA101T-EN"/>
   </dc:relation>
</rdf:Description>

<!-- a web-based training. The URL references a Learning Content Format
document. -->

<rdf:Description about="http://www.saba.com/courses/java/java101.xml">
   <dc:format>text/xml</dc:format>
   <rdf:type resource="http://www.saba.com/RDF/offering10.rdf#online"/>
   <dc:relation rdf:parseType="Resource">
      <dcq:relationType>InstanceOf</dcq:relationType>
      <rdf:value rdf:resource="#JAVA101"/>
   </dc:relation>
</rdf:Description>

<!-- location-based training classes -->

<rdf:Description id="JAVA101-C">
   <rdf:type resource="http://www.saba.com/RDF/offering10.rdf#event"/>
   <dc:relation rdf:parseType="Resource">
      <dcq:relationType>InstanceOf</dcq:relationType>
      <rdf:value rdf:resource="#JAVA101"/>
   </dc:relation>
</rdf:Description>

<rdf:Description id="JAVA101-C1">
   <dc:relation rdf:parseType="Resource">
      <dcq:relationType>InstanceOf</dcq:relationType>
      <rdf:value rdf:resource="#JAVA101-C"/>
   </dc:relation>
   <schedule:location>Redwood Shores, CA</schedule:location>
   <schedule:startDate>2000-01-10</schedule:startDate>
   <schedule:template>MWF 9-5</schedule:template>
   <schedule:minimumlearners>10</schedule:minimumlearners>
   <schedule:maximumlearners>25</schedule:maximumlearners>
</rdf:Description>
```

```
<rdf:Description id="JAVA101-C2">
   <dc:relation rdf:parseType="Resource">
      <dcq:relationType>InstanceOf</dcq:relationType>
      <rdf:value rdf:resource="#JAVA101-C"/>
   </dc:relation>
   <schedule:location>Denver, CO</schedule:location>
   <schedule:startDate>2000-01-17</schedule:startDate>
   <schedule:template>MWF 9-5</schedule:template>
   <schedule:minimumlearners>3/schedule:minimumlearners>
   <schedule:maximumlearners>20</schedule:maximumlearners>
</rdf:Description>

<!-- price list for these offerings -->

<rdf:Description id="TEXTBOOK_PRICE">
   <price:price rdf:parseType=”Resource”>
      <rdf:value>
         <rdf:Bag>
            <rdf:li parseType="Resource">
               <rdf:value>50</rdf:value>
               <price:currency>usd</price:currency>
            </rdf:li>
            <rdf:li parseType="Resource">
               <rdf:value>325</rdf:value>
               <price:currency>frf</price:currency>
            </rdf:li>
         </rdf:Bag>
      </rdf:value>
      <price:method>ship</price:method>
      <price:target>
         <rdf:Bag>
            <rdf:li rdf:resource="#JAVA101T-EN"/>
            <rdf:li rdf:resource="#JAVA101T-FR"/>
         </rdf:Bag>
      </price:target>
      </price:price>
</rdf:Description>

<rdf:Description id="CLASS_PRICE">
   <price:price rdf:parseType=”Resource”>
      <rdf:value>350</rdf:value>
      <price:method>register</price:method>
      <price:target rdf:resource="#JAVA101-C"/>
   </price:price>
</rdf:Description>
```

```
</rdf:RDF>
```

Learning Content Format Specification

This document is the technical specification for Learning Content Format (LCF), the component of Universal Learning Format designed for describing online learning content.

It includes the following sections:

Introduction

LCF is an XML-based representation for capturing and exchanging online learning resources.

Content

LCF can be used to represent the following two types of learning content:

- **Course structure** — defines a hierarchical lesson plan for a course with references to text-based content and executable units of content. This is equivalent to the Course Structure Format defined by ADL, based on the AICC CMI content model.
- **Online learning content** — defines the contents of a single, executable online course, including local and remote content and questionnaires. This is equivalent to the Content Packaging Specification defined by IMS, which is also the basis for Microsoft Learning Resource Interchange. It also encompasses a subset of the assessment types defined by IMS Question and Test Interoperability; a type of "questionnaire" indicates this usage.

  Specifically, LCF supports the following types of online learning content:

  - static text content
  - URL references to external content
  - questionnaires
  - objectives

This flexible structure supports several different types of learning resources using a consistent format, including structured courses that group together a sequence of individual WBTs, a single WBT with embedded HTML or other multimedia content, and online questionnaires or assessments.

Usage

LCF provides an XML-based solution for capturing and exchanging online learning content. Specifically, it provides the following capabilities:

- **Assemble content information** — Content developers can assemble online courses using LCF to ensure maximum portability across systems
- **Import content information** — Learning providers can import online courses described in LCF into any learning management system that supports either LCF or any of its constituent industry standard formats
- **Exchange content information** — Learning providers can transport online courses from one learning management system to another using LCF as the exchange format
- **Launch content information** — Learners can launch online courses dirrectly from an LCF source document, which provides a decoupling of content from the database.

LCF Schemas

Learning Content Format is supported by the following XML-based representations:

Schema	Description	Where To Find
LCF DTD	Defines the format for describing learning content.	The LCF DTD has the following URL: http://www.saba.com/XML/content10.dtd To view the LCF DTD, see "Learning Content Format DTD".
LCF XML Schema	Defines the format for describing learning content.	To view the LCF XML Schema, see "LCF XML Schema".

Note The remainder of this document defines the specifications for the Learning Content Format DTD.

LCF DTD Detailed Description

LCF defines two types of elements:

- Structure Elements
- Content Elements

An LCF document places the structure information at the top-most level, followed by the content information. This separation of content from structure makes it possible to change them independently.

Structure Elements

LCF defines the following structure elements:

Table 9: *LCF Structure Elements*

Element	Description	Attributes
lcf	Root element containing all content elements for a particular resource. It contains a Table of Contents, followed by zero or more objectives, followed by one or more content elements.	Content Attributes type
toc	Defines the Table of Contents for the learning content.	
objective	Defines the objectives of the described courses and classes. Objectives can be associated with individual content elements.	**id** name description metadataRef lang (See description on page 7.)

Content Elements

LCF defines the following content elements::

Table 10: *LCF Structure Elements*

Element	Description	Attributes
text	Defines static text for display as a content screen.	Content Attributes
learningObject	Contains a URL pointer to an external source of content.	Content Attributes **href**
questionnaire	Identifies a test or other form of assessment. A questionnaire consists of optional introductory text, followed by one or more multiple choice, true/false, and essay questions.	Content Attributes type passingScore resumable displayType randomize warningTime
multipleChoice	Defines a question with multiple choice answers.	Content Attributes Question Attributes
trueFalse	Defines a question whose answer is either true or false.	Content Attributes Question Attributes **answer**
essay	Defines the text of an open-ended essay question. You can also use it to represent shorter fill-in-the-blank style questions.	Content Attributes Question Attributes

Content Attributes

All LCF content elements share a common set of content attributes. The table below lists and describes these content attributes:

Table 11: *Content Attributes*

Attribute	Description
id	Standard XML identifier for the content element. Must be unique within the document. **Note** The **id** attribute must begin with an alphabetic character. This attribute is required.
name	Specifies a human-readable identifier for the content element. This attribute is optional.
description	Specifies a text description of the content element. This attribute is optional.
metadataRef	URL pointer to an external Catalog Format document containing metadata for the content element. This attribute is optional.
objectiveRef	Reference to one or more objectives associated with the content element. Attribute type is IDREFS. This attribute is optional.
lang	Standard XML attribute for specifying the language in which the content element is authored. Contains the ISO 639/RFC 1766 language code with an optional geographic identifier, such as **en** for English, or **fr** for French. Attribute type is NMTOKEN. This element is optional.
timeAllowed	Specifies the time limit, in minutes, a specific learner is allowed to view the content element. This is primarily intended for questions on tests but may also be applied to other content elements. This attribute is optional.
accessCount	Specifies the number of times a given learner can access the content. This attribute is optional.
isVisible	Indicates whether or not the content is displayed. Value can be **true** or **false**. The default value is **true**.

Table 11: *Content Attributes*

Attribute	Description
startDate	Specifies the start date for optional date range within which a specific learner is allowed to view the content element. Uses ISO8601 format. This attribute is optional.
endDate	Specifies the end date for optional date range within which a specific learner is allowed to view the content element. Uses ISO8601 format. This attribute is optional.

Question Attributes

LCF question elements share a common set of question attributes. The table below lists and describes these question attributes:

Table 12: *Question Attributes*

Attribute	Description
points	Specifies the point value for the question. This attribute is optional.
questionnaire	Reference to the questionnaire with which the question is associated. Attribute type is IDREFS. **Note** By associating questions with a questionnaire, you identify the questions on which properties of the questionnaire (such as **passingScore**, **scoreband**, **resumable**, and **randomize**) will act. This attribute is optional.
title	Specifies a title for the question. This attribute is optional.

Lang Attribute

The **lang** attribute is common to many elements and subelements in an LCF document.

lang is a standard XML attribute for specifying the language in which the element is authored. It contains the ISO 639/RFC 1766 language code with an optional geographic identifier, such as **en** for English, or **fr** for French. Attribute type is NMTOKEN.

The **lang** attribute is always optional.

lcf

The **lcf** element is the root element in an LCF document and contains all content elements for a particular resource. It contains a Table of Contents, followed by objectives, followed by one or more content elements.

Subelements of **lcf** include:

- **toc**
- **objective**
- **text**
- **learningObject**
- **questionnaire**
- **multipleChoice**
- **trueFalse**
- **essay**

The **lcf** element is always required and occurs exactly once in every LCF document.

lcf Subelements

The **lcf** element includes the following subelements:

- exactly one **toc** element
- zero or more **objective** elements
- one or more **content** elements

lcf Attributes

The **lcf** element can have the following attributes:

- any of the Content Attributes (**id** attribute is required)
- a **type** attribute, which provides a hint as to the type of content contained in the file. Value for the **type** attribute can be any of the following:

Value	Description
class	The file defines a single, launchable class. This is directly analogous to an IMS Content Packaging file, where each **learningObject** is an external reference to a physical resource. The default value is **class**.
course	The file defines a course structure. This is directly analogous to an ADL Structure Course Format file, where each **learningObject** element is an **au**.

Value	Description
questionnaire	The file contains an online questionnaire only.

toc

The **toc** element defines a Table of Contents structure for the learning content. This structure consists of TOC items that reference content elements within the document. Items can be nested to convey TOC structure.

The **toc** element occurs exactly one time per **lcf** instance.

toc Subelements

Subelements of **toc** include:

Table 13: *toc Subelements*

Element	Description	Attributes
item	Defines the TOC items that comprise the TOC structure. Items can be nested using the **item** subelement, which is recursive. You can use nested items to convey TOC structure. The **item** parent element occurs one or more times per **toc** instance. The **item** subelement occurs zero or more times per parent **item** instance.	title ref

toc Attributes

Attributes specific to the **item** element include:

Attribute	Description
title	Specifies the the title of the TOC item. This attribute is optional.
ref	Reference to the ID of a content element within the document. This attribute is optional.

objective

The **objective** element defines the objectives for the courses and content elements described in an LCF document. Typically, an objective is a summary of the purpose and scope of a course or content element and includes a statement of the skills and knowledge to be acquired by the learner. Objectives can be associated with content elements.

The **objective** element consists of a required **id** attribute and optional **name** and **description** attributes. It can include nested objectives to convey a structure with sub-objectives.

The **objective** element occurs zero or more times per **lcf** instance.

objective Subelements

Subelements of **objective** include:

Table 14: *toc Subelements*

Element	Description	Attributes
objective	Specifies an objective for the content. Objectives can be nested using the **objective** sublement, which is recursive. Occurs zero or more times per **objective** instance. Subelements of **objective** include: **objective**	**id** name description metadataRef lang (See description on page 7.)

objective Attributes

Attributes specific to the **objective** element include:

Attribute	Description
id	Standard XML identifier for the objective. Must be unique within the document. **Note** The **id** attribute must begin with an alphabetic character. This attribute is required.
name	Specifies a title for the objective. This attribute is optional.

Attribute	Description
description	Specifies a text description for the objective. Typically contains information about the purpose and scope of a course or content element. This attribute is optional.
metadataRef	URL pointer to an external document containing metadata for the objective. This attribute is optional.

text

The **text** element defines static text for display as a content screen.

The **text** element occurs zero or more times per **lcf** instance.

learningObject

The **learningObject** element contains a URL pointer to an external source of learning content.

Note The formats supported for **learningObject** references are implementation-specific, although support for common web formats such as HTML documents and JPEG images is recommended.

The **learningObject** element occurs zero or more times per **lcf** instance.

Attributes specific to the **learningObject** element include:

Attribute	Description
href	Contains the URL pointer to an external source of content. This attribute is required.

questionnaire

The **questionnaire** element identifies a test or other form of assessment. A questionnaire typically contains multiple choice, true/false, and essay questions, but the structure of LCF enables them to contain other content as well, such as introductory text screens or even embedded tests. For example, to embed a subsection within a questionnaire, you can place another questionnaire within it using the TOC. Questionnaires may also contain one or more scoreband elements that define ranges of score results.

LCF provides two ways to organize the contents of a questionnaire:

- **Fixed Order** — For tests with a fixed number and order of questions, you can use the TOC structure to specify the exact order in which questions are presented.
- **Random Order** — For tests where you want to present the questions in random order, you can set a numeric value for the **randomize** attribute of the questionnaire to indicate the number of questions to include. The specified number of questions will display in random order.

The **questionnaire** element occurs zero or more times per **lcf** instance.

Subelements of **questionnaire** include:

Element	Description	Attributes
scoreband	Defines a range of score results. The **scoreband** element consists of numeric lower and upper bound attributes as well as an optional feedback text element. Subelements of **scoreband** include: **text** Occurs zero or more times per **questionnaire** instance.	**id** **lowerBound** **upperBound**
text (subelement of scoreband)	Specifies feedback text associated with the scoreband. Occurs zero or one times per **scoreband** instance.	Content Attributes

The attributes specific to the **questionnaire** element include:

Attribute	Description
type	Specifies the classification of a specific questionnaire (the context in which it is used). Value can be: • **test** • **survey** • **assessment** The default value is **test**.
passingScore	Defines the passing score for a test. **Note** You should use the **passingScore** attribute when there are just two possible results for the test—pass and fail. For more complex test results schemes, use the **scoreband** subelement.
resumable	Specifies whether a test can be paused and resumed at a later time. Value can be: • **true** • **false**
displayType	Specifies whether the questions of a test should be combined on a single screen. Value can be: • **together** • **separate** The default value is **separate**.
warningTime	Specifies the number of minutes before the end of a test to display a warning.
randomize	When present, the **randomize** attribute indicates that the questions on a test should be presented in random order. Its value specifies the number of questions to appear on the test.

Attributes specific to the **scoreband** element include:

Attribute	Description
id	Standard XML identifier for the scoreband. Must be unique within the document. **Note** The **id** attribute must begin with an alphabetic character. This element is required.
lowerBound	Specifies a numeric lower bound for the scoreband. This element is required.
upperBound	Specifies a numeric upper bound for the scoreband. This element is required.

multipleChoice

The **multipleChoice** element defines a question with multiple choice answers. You must identify one of the choices as the correct answer. You can define optional hints.

The **multipleChoice** element occurs zero or more times per **lcf** instance.

Subelements of **multipleChoice** include:

Element	Description	Attributes
question	Contains the multiple choice question. The question consists of one or more subelements, each of which can be either **text** or **learningObject**. Occurs one or more times per **multipleChoice** instance.	
hints	Contains one or more hints. Subelements of **hints** include: **hint** Occurs zero or one times per **multipleChoice** instance.	type
hint (subelement of hints)	Defines the text of a hint. Occurs one or more times per **hints** instance.	lang (See description on page 7.)

Element	Description	Attributes
choice	Defines the text of an answer to the question. Occurs one or more times per **multipleChoice** instance. **Note** At least one choice must have a value of **true** for the **answer** attribute.	answer

Attributes specific to the **hints** element include:

Attribute	Description
type	Contains a value that indicates how hints are presented. Value can be: • **incremental** — display all hints up to and including the current request • **multilevel** — display only the hint element corresponding to the number of times a hint has been requested The default value is **multilevel**.

Attributes specific to the **choice** element include:

Attribute	Description
answer	Contains a value that indicates whether the choice is the correct answer. Value can be: • **true** • **false** The default value is **false**. **Note** At least one choice must have a value of **true**.

trueFalse

The **trueFalse** element defines a question whose answer is either true or false. You must identify the correct answer. You can define optional hints.

The **trueFalse** element occurs zero or more times per **lcf** instance.

Subelements of **trueFalse** include:

Element	Description	Attributes
question	Contains the true/false question. The question consists of one or more subelements, each of which can be either **text** or **learningObject**. Occurs one or more times per **trueFalse** instance.	
hints	Contains one or more hints. Subelements of **hints** include: **hint** Occurs zero or one times per **trueFalse** instance.	type (see description on page 15)
hint (subelement of hints)	Defines the text of a hint. Occurs one or more times per **hints** instance.	lang (See description on page 7.)

Attributes specific to the **trueFalse** element include:

Attribute	Description
answer	Contains a value that indicates whether the answer to the question is true or false. Value can be: • **true** • **false** This is a required attribute.

essay

The **essay** element defines the text of an open-ended essay question. You can also use it to represent shorter fill-in-the-blank style questions. You can define optional hints.

The **essay** element occurs zero or more times per **lcf** instance.

Subelements of **essay** include:

Element	Description	Attributes
question	Contains the essay question. The question consists of one or more subelements, each of which can be either **text** or **learningObject**. Occurs one or more times per **essay** instance.	
hints	Contains one or more hints. Subelements of hints include: **hint** Occurs zero or one times per essay instance.	type (see description on page 15)
hint (subelement of hints)	Defines the text of a hint. Occurs one or more times per hints instance.	lang (See description on page 7.)

Comparing LCF to Learning Industry Standards

Learning Content Format integrates the key features of learning standards proposed by organizations such as ADL, IMS, AICC, and Microsoft into a single unified format for defining the structure and content of online learning.

The following sections describe the relationship of LCF to some of the key learning industry standards.

ADL

Advanced Distributed Learning (ADL) has defined a specification called the Sharable Courseware Object Reference Model (SCORM), which unifies several learning standards. Specifically, SCORM defines an XML-based course interchange format called Course Structure Format (CSF), which is based on AICC's CMI content model. It also includes a runtime environment based on AICC standards and a metadata mapping based on IMS and IEEE recommendations.

Like ADL's CSF, Learning Content Format (LCF) can be used to define the structure and contents of an online course. Both models include support for course structure, references to external content, and the definition of objectives and metadata. The principal CSF elements map to LCF as follows:

CSF Element	Corresponding LCF Element
block	toc
au	learningObject
objective	objective
externalMetadata	metadataRef

Some of the differences between CSF and LCF include:

- CSF uses hierarchically nested blocks to define the structure of content, while LCF extracts this information into a separate TOC.
- LCF supports a subset of the information in the CSF **au** and **objective** elements.
- CSF uses the XML binding of IMS metadata, while LCF uses Catalog Format's RDF binding of IMS metadata.

IMS

Instruction Management Systems (IMS) has defined two specifications that overlap with aspects of Learning Content Format (LCF). These specifications are:

- Content Packaging
- Question and Test Interoperability

Content Packaging

IMS Content Packaging is a specification for defining the contents and structure of an online course. Identical to the IMS Content Packaging specification is Microsoft's Learning Resource iNterchange (LRN) specification, which serves as the foundation of its "eLearn" initiative.

Like the IMS Content Packaging specification and Microsoft's Learning Resource iNterchange, Learning Content Format (LCF) can be used to define the structure and contents of an online class. Both support references to external content, both define a TOC structure for organizing content, and both define a metadata mapping. The principal LRN elements map to LCF as follows:

LRN Element	Corresponding LCF Element
tableofcontents	toc
file	learningObject
metadata	metadataRef

Some of the differences between LRN and LCF include:

- LRN supports external packages.
- LRN requires that all dependent resources be listed.
- LCF incorporates support for questionnaires.
- LCF supports additional attributes for describing the learning content, including **timeAllowed**, **startDate**, and **endDate**.
- LRN uses the XML binding of IMS metadata, while LCF uses Catalog Format's RDF binding of IMS metadata.

Question & Test

The IMS Question and Test Interoperability (QTI) specification defines an XML-based format for representing test questions and responses.

Like QTI, Learning Content Format (LCF) can be used to represent questions and tests. LCF defines a subset of QTI that represents the most commonly used question types. The principal QTI elements map to LCF as follows:

QTI Element	Corresponding LCF Element
render_choice	trueFalse multipleChoice
render_fib	essay
section	toc
assessment	questionnaire
itemfeedback	hints

Some of the differences between QTI and LCF include:

- QTI employs a single question element with an attribute that specifies the type of question. LCF employs different elements for the supported question types (multiple choice, true/false, and essay).
- LCF only supports some of the QTI formats. It does not support **text selection**, **drag-and-drop**, **object pair matching**, **object ordering**, **sliders**, and **image hotspot**.
- LCF uses the TOC structure to define the sequencing of questions, while QTI uses a test section approach.
- QTI defines its own metadata format, while LCF uses Catalog Format's RDF binding of IMS metadata.

LCF Examples

Below are two examples of LCF documents.

- Course with Inline Content and Test Questions
- Course with Objectives

Course with Inline Content and Test Questions

The following is an LCF document capturing online content for an Astronomy course.

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- LCF document used to represent a LOL class. References external
HTML content. -->

<!DOCTYPE lcf SYSTEM "http://www.saba.com/xml/content10.dtd">
<lcf id="CNTNT001" name="PG1CE1" type="class">
   <toc>
      <item ref="CNTNT002"/>
      <item ref="CNTNT003">
         <item ref="CNTNT004"/>
         <item ref="CNTNT005"/>
         <item ref="CNTNT006"/>
         <item ref="CNTNT007"/>
         <item ref="CNTNT008"/>
         <item ref="CNTNT009"/>
      </item>
      <item ref="CNTNT010">
         <item ref="CNTNT011"/>
         <item ref="CNTNT012"/>
      </item>
      <item ref="TEST001">
         <item ref="CNTNT013"/>
         <item ref="QUEST001"/>
         <item ref="QUEST002"/>
         <item ref="QUEST003"/>
      </item>
   </toc>
   <text id="CNTNT002" name="Setup Information">
Successful participation in this class will require:
- an Internet connection
- Apple QuickTime 3.0
   </text>
```

```
   <text id="CNTNT003" name="Astronomy Outline">
Astronomy Course
This astronomy information was developed by Margaret Lezniewski.
Objectives of this course include:
- Mastery of Orbital Issues
- Familiarity with Gravitional Issues
- Familiarity with Velocity and Brightness
   </text>
   <learningObject id="CNTNT004"
      name="Introduction"
      href="file://c:/content/sample_1_introduction.html"/>
   <learningObject id="CNTNT005"
      name="Elliptical Orbit: Effect"
      href="file://c:/content/sample_2_ellipticalOrbitEffect.html"/>
   <learningObject id="CNTNT006"
      name="Elliptical Orbit: Math"
      href="file://c:/content/sample_3_ellipticalOrbitMath.html"/>
   <learningObject id="CNTNT007"
      name="Tilt Effect"
      href="file://c:/content/sample_4_titlEffect.html"/>
   <learningObject id="CNTNT008" name="Space Launch Video"
      href="file://c:/content/sample_delta.qtw"/>
   <learningObject id="CNTNT009" name="Conclusion"
      href="file://c:/content/sample_5_Summation.html"/>
   <text id="CNTNT010" name="References">
Other References on the Web:
Click the links in the Navigation Bar to visit related URLs.
   </text>
   <learningObject id="CNTNT011" name="Sky &amp;amp; Telescope Magazine"
      href="http://www.skypub.com"/>
   <learningObject id="CNTNT012" name="The Learning Channel"
      href="http://tlc.com"/>
   <questionnaire id="TEST001" name="Assessment Test" type="test"
      resumable="false">
      <scoreband id="TEST001_FAIL" lowerBound="0"
         upperBound="10.0"/>
      <scoreband id="TEST001_PASS" lowerBound="20.0"
         upperBound="30.0"/>
   </questionnaire>
```

```
   <text id="CNTNT013">
Test for Astronomy:
Click the links in Navigation Bar to begin.
   </text>
   <trueFalse id="QUEST001" name="Question 1" accessCount="1"
      points="10.0" answer="true" questionnaire="TEST001">
      <question>
         <text id="CNTNT014">The Earth's sun is a star.</text>
      </question>
   </trueFalse>
   <multipleChoice id="QUEST002" name="Question 2" accessCount="1"
      points="10.0" questionnaire="TEST001">
      <question>
         <text id="CNTNT015">The planets travel around the sun in a
            an __________</text>
      </question>
      <choice answer="true">ellipse</choice>
      <choice answer="false">oval</choice>
      <choice answer="false">circle</choice>
      <hints>
         <hint>Rhymes with "eclipse."</hint>
      </hints>
   </multipleChoice>
   <essay id="QUEST003" name="Question 3" accessCount="1" points="10.0"
      questionnaire="TEST001">
      <question>
         <text id="CNTNT016">Please give a brief description of how
            the earth's axis plays a crucial role in the analemma
            phenomenon.</text>
      </question>
   </essay>
</lcf>
```

Course with Objectives

The following is an LCF document illustrating ADL-compatible course and objective information.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- LCF document used to describe a course structure.  References
individual LOL course descriptions. -->
<!DOCTYPE lcf SYSTEM "http://www.saba.com/xml/content10.dtd">
<lcf id="COURSE001" name="Mastering Astronomy" type="course">
   <toc>
      <item ref="CLASS001"/>
      <item ref="CLASS002"/>
      <item ref="CLASS003"/>
   </toc>

   <objective id="OBJ1"
      name=”Beginner”
      description=”Able to use a telescope”/>

   <objective id="OBJ2"
      name=”Advanced”
      description=”Able to repair a telescope”/>

   <objective id="OBJ3"
      name=”Expert”
      description=”Able to construct a telescope”/>

   <learningObject id="CLASS001"
      name="Introduction to Astronomy"
      objectiveRef="OBJ1"
      href="http://www.saba.com/classes/astronomy1.xml"/>
   <learningObject id="CLASS002"
      name="Intermediate Astronomy"
      objectiveRef="OBJ2"
      href="http://www.saba.com/classes/astronomy2.xml"/>
   <learningObject id="CLASS003"
      name="Expert Astronomy" objectiveRef="OBJ3"
      href="http://www.saba.com/classes/astronomy3.xml"/>
</lcf>
```

Competency Format Specification

This document is the technical specification for Competency Format, the component of Universal Learning Format designed for describing competency-related information.

It includes the following sections:

Introduction

Competency Format is an XML-based representation for capturing competency-related information.

Usage

Competency Format provides an XML-based solution for exchanging competency-related information. It also ensures that your competency libraries are compatible with any learning management system that supports Competency Format.

Specifically, Competency Format provides the following capabilities:

- **Define competency libraries** — Competency model authors can define competency libraries using Competency Format to ensure maximum portability across systems
- **Import competency libraries** — Competency managers can import competency libraries described in Competency Format into any learning management system that supports Competency Format
- **Exchange competency libraries** — Competency managers can transport competency libraries described in Competency Format from one learning management system to another using Competency Format as the exchange format
- **Associate catalog offerings with competencies** — Content providers can use the Competency RDF Schema to associate catalog descriptions defined using Catalog Format with the competencies they provide.
- **Support queries on competencies** — Learners can search for competency-related information. For example, a learner can use competency metadata to search for all courses that provide a specific competency.

Competency Format Schemas

Competency Format is supported by the following XML-based representations:

Schema	Description	Where To Find
Competency Format DTD	Defines the format for describing competencies and competency libraries.	The Competency Format DTD has the following URL: http://www.saba.com/XML/competency10.dtd To view the Competency Format DTD, see "Competency Format DTD".
Competency Format XML Schema	Defines the format for describing competencies and competency libraries.	To view the Competency Format XML Schema, see "Competency Format XML Schema".
Competency Format RDF Schema	Defines the format for describing competency states (e.g., competencies held, provided, etc.).	The Competency Format RDF Schema has the following URL: http://www.saba.com/RDF/competency10.rdf To view the Competency Format RDF Schema, see Competency RDF Schema.

Note The remainder of this document defines the specifications for the Competency Format DTD and Competency Format RDF Schema.

Competency Format DTD

The Competency DTD defines the format for describing competency libraries. A competency library is a set of competencies that can be associated with individuals, job descriptions, or learning resources. Each competency consists of a unique description and a set of one or more pre-defined proficiency levels. The Competency DTD also provides support for defining groups of related competencies and references to external metadata.

To view the Competency Format DTD, see "Competency Format DTD".

The following is an example of a simple competency library in Competency Format:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE competencyLibrary
   SYSTEM "http://www.saba.com/xml/competency10.dtd">

<competencyLibrary xml:lang="en">
   <competency id="Java">
      <name>Java programming</name>
      <description>
         Ability to write object-oriented software in Java.
      </description>
      <level id="Java.Beginner">
         <name>Beginner</name>
         <description>Can compile a Java file</description>
         <evidence>Executes javac command.</evidence>
      </level>
      <level id="Java.Expert">
         <name>Expert</name>
         <description>Thinks in Java bytecode</description>
         <evidence>Implements a JVM.</evidence>
         <evidence>Edits Java bytecode.</evidence>
      </level>
   </competency>
</competencyLibrary>
```

Proficiency Levels

The Competency DTD defines two types of support for proficiency levels:

1. A competency library can have a definition for one set of shared proficiency levels – specific numeric values and descriptive information that are applied to all competencies by default.
2. Individual competencies can also define their own proficiency levels, which override the shared proficiency levels.

Hence, the following permutations are possible:

- A competency library has a single set of shared proficiency levels that apply to all competencies.
- A competency library has no shared proficiency levels, and each competency defines its own proficiency levels.
- A competency library has a set of shared proficiency levels that is applied to all competencies by default, but some competencies override the shared proficiency levels with their own proficiency levels.

Vendors who wish to define more than one set of shared levels should author multiple competency library documents, with one set of levels per document.

The following example demonstrates the use of competency groups and shared proficiency levels:

```
<competencyLibrary>
   <group id="Programming">
      <name>Programming</name>
   </group>
   <sharedLevels>
      <level id="Beginner" value="1.0">
         <name>Beginner</name>
         <description>Can compile "Hello World".</description>
      </level>
      <level id="Experienced" value="2.0">
         <name>Experienced</name>
         <description>
            Can write a fully-functional application.</description>
      </level>
      <level id="Expert" value="3.0">
         <name>Expert</name>
         <description>
            Can implement a language compiler.</description>
      </level>
   </sharedLevels>
   <competency id="Java" groupRef="Programming">
      <name>Java programming</name>
      <description>
         Ability to write object-oriented software in Java.
      </description>
   </competency>
   <competency id="C" groupRef="Programming">
      <name>C programming</name>
      <description>Ability to write software in C.</description>
   </competency>
</competencyLibrary>
```

Competency DTD Elements

The table below lists and describes the elements defined in the Competency DTD:

Table 15: *Competency DTD Elements (page 1 of 3)*

Element	Description	Attributes
competencyLibrary	The **competencyLibrary** element is the root element in a Competency Format document. It defines one or more competencies and can include an optional set of competency group definitions and an optional set of shared proficiency level definitions. The **competencyLibrary** element is always required and occurs exactly once in every Competency Format document. Subelements of **competencyLibrary** include: **group** **sharedLevels** **competency**	metadataRef lang
group (subelement of competencyLibrary)	Defines a group for categorizing competencies. You can define multiple groups and nest groups inside other groups. Then, you can associate competencies with the various groups you have defined. This makes it possible to design a group structure that reproduces your taxonomy (classification system) for competencies. **Note** Unless you are working with a pre-defined taxonomy for competencies, Saba recommends following the Saba Learning Exchange taxonomy, which you can view at: http://www2.saba.com/exchange/ Competency groups can be nested using the **group** sublement, which is recursive. Occurs one or more times per **competencyLibrary** instance. Subelements of **group** include: **name** **description** **group**	**id**

Table 15: *Competency DTD Elements (page 2 of 3)*

Element	Description	Attributes
sharedLevels (subelement of competencyLibrary)	Defines a set of shared proficiency levels. Shared proficiency levels are applied by default to all competencies defined in the document. However, any competency can override the shared proficiency levels. Occurs zero or one times per **competencyLibrary** instance. Subelements of **sharedLevels** include: **level**	
competency (subelement of competencyLibrary)	Contains information about a competency. By default, competencies inherit the proficiency levels defined using the **sharedLevels** element. You can override this default by defining proficiency levels specific to each competency. **Note** If the **sharedLevels** element is not defined, you must specify one or more proficiency levels for each competency. Occurs one or more times per **competencyLibrary** instance. Subelements of **competency** include: **name** **description** **level**	**id** groupref
name (subelement of competency, group, and level)	Defines the name of the competency, competency group, or proficiency level. Occurs exactly one time per **competency** or **group** instance. Occurs zero or one times per **level** instance.	
description (subelement of competency, group, and level)	Defines a text description for the competency, competency group, or proficiency level. Occurs zero or one times per **competency**, **group**, or **level** instance.	

Table 15: *Competency DTD Elements (page 3 of 3)*

Element	Description	Attributes
level (subelement of sharedLevels, competency)	Contains information about a proficiency level. Occurs one or more times per **sharedLevels** instance and zero or more times per **competency** instance. Subelements of **level** include: **name** **description** **evidence**	**id** value
evidence (subelement of level)	Defines an evidence description for the proficiency level. Typically, this description includes information about the characteristics exhibited by someone holding this proficiency level. Occurs zero or more times per **level** instance.	

The table below lists and describes the attributes defined in the Competency DTD:

Table 16: *Competency DTD Attributes*

Attribute	Description
metadataRef	URL pointer to an external Catalog Format document containing metadata for the competency library. This attribute is optional.
lang	Standard XML attribute for specifying the language in which the content element is authored. Contains the ISO 639/RFC 1766 language code with an optional geographic identifier, such as **en** for English, or **fr** for French. Attribute type is NMTOKEN. This attribute is optional.
id	Standard XML identifier. Must be unique within the document. This attribute is required.
groupref	Reference to a competency group within the document (using the **id** attribute). This attribute is optional.
value	Defines a numeric value for the proficiency level. This attribute is optional.

Competency Format RDF Schema

The Competency Format RDF Schema defines the metadata format for representing competency-related information in order to associate it with learning offerings defined in a Catalog Format document.

Specifically, the Competency Format RDF Schema provides the following capabilities:

- **Associate catalog offerings with competencies** -- Content providers can use the Competency RDF Schema to associate learning offerings defined in a Catalog Format document with the competencies they provide.
- **Support queries on competencies** -- Users can search for competency-related information. For example, you can use Competency Format metadata to find all courses providing a specific competency.

To view the Competency Format RDF Schema, see Competency RDF Schema.

The table below lists and describes the properties defined in the Competency Format RDF Schema:

Table 17: *Competency Format RDF Schema Properties*

Property	Description
Competency	Specifies a set of one or more competencies and proficiency levels associated with a learning offering. The **competency** property is a literal or RDF Bag. Typically, its value is a URL pointer to a competency ID in a competency library defined using Competency Format and published as a web document. This decouples competency profiles from competency libraries while ensuring that all necessary data is still accessible. For convenience, it is also possible for the **competency** property to contain just the ID value of the proficiency level. This is most useful in a scenario where you are generating competency metadata for the purpose of querying. Assuming a fixed set of pre-imported competencies, specifying just the ID value allows subsequent queries to be simpler.
Attained	Specifies the date a held competency was attained. Uses ISO 8601 format.
Provider	Specifies the competency provider that provided a held proficiency level.
Details	Specifies any further information about how a held proficiency level was attained.

The following is a simple example of a competency provider. It is a Catalog Format document that also incorporates the competency attribute.

```
<Description about="http://www.saba.com/AllAboutJava/J21H">
   <ims:metadataScheme>RDF/LOM-1.0</ims:metadataScheme>
   <dc:title>Teach Yourself Java in 21 Hours</dc:title>
   <ewp:competency>
      http://www.saba.com/competencies/programming#Java.Expert
   </ewp:competency>
</Description>
```

Multiple competencies can be represented in a Bag structure:

```
<Description about="http://www.saba.com/people/sally_brown">
   <ewp:competency>
   <rdf:Bag>
      <rdf:li>XML.Intermediate</rdf:li>
      <rdf:li>HTML.Advanced</rdf:li>
   </rdf:Bag>
   </ewp:competency>
</Description>
```

Certification Format Specification

This document is the technical specification for Certification Format, the component of Universal Learning Format designed for describing certification-related information.

It includes the following sections:

Introduction

Certification Format is an XML-based representation for capturing certification-related information.

Usage

Certification Format provides an XML-based solution for exchanging certification-related information. It also ensures that your certification libraries are compatible with any learning management system that supports Certification Format.

Specifically, Certification Format provides the following capabilities:

- **Define certification libraries** — Certification managers can define certification libraries using Certification Format to ensure maximum portability across systems
- **Import certification libraries** — Certification managers can import certification libraries described in Certification Format into any learning management system that supports Certification Format
- **Exchange certification libraries** — Certification managers can transport certification libraries described in Certification Format from one learning management system to another using Certification Format as the exchange format
- **Import the results of certification testing** — Certification agencies can use the Certification Results Format to define the results of certification testing maximum portability across systems.
- **Associate catalog offerings with certifications** — Certification providers can use the Certification RDF Schema to associate catalog descriptions defined using Catalog Format with the certifications they provide.
- **Support queries on certifications** — Learners can search for certification-related information. For example, a learner can use certification metadata to search for all courses that provide a specific certification.

Certification Format Schemas

Certification Format is supported by the following XML-based representations:

Schema	Description	Where To Find
Certification Format DTD	Defines the format for describing certifications and certification libraries.	The Certification Format DTD has the following URL: http://www.saba.com/XML/certification10.dtd To view the Certification Format DTD, see "Certification Format DTD".
Certification Format XML Schema	Defines the format for describing certifications and certification libraries.	To view, the Certification Format XML Schema, see "Certification Format XML Schema".
Certification Results DTD	Defines the format for describing certification testing results.	The Certification Results DTD has the following URL: http://www.saba.com/XML/certification_result10.dtd To view the Certification Results DTD, see "Certification Results Format DTD".

Note The remainder of this document defines the specifications for the Certification Format DTD and the Certification Results DTD.

Certification Format DTD

The Certification Format DTD defines the format for describing a library of certifications that can be associated with individuals, job descriptions, or learning resources.

A certification is a set of pre-defined paths, where each path consists of a specific sequence of steps (learning modules) that must be completed in order to certify knowledge or mastery of a specific skill set. Learning modules consist of a set of learning interventions, one or more of which must be fulfilled in order to complete the learning module. Learning interventions are typically specific courses or other learning offerings, although they may also refer to specific competency levels or actions to be taken.

The following diagram illustrates the structure of a certification:

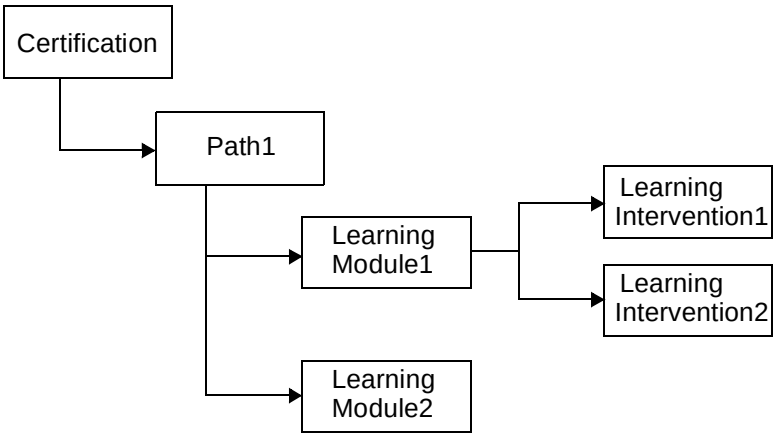

Figure 1: *Diagram of Certification Structure*

To view the Certification Format DTD, see “Certification Format DTD”.

The following is an example of a very simple certification defined in Certification Format:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE certificationLibrary SYSTEM
   "http//www.saba.com/xml/certification10.dtd">

<certificationLibrary>
   <certification id="MSCD" name="Microsoft Certified Developer"
   startDate="1999-01-01" endDate="2001-01-01">
      <path>
         <module name="Architecture 1">
            <option href="http://www.saba.com/offerings.rdf#WINARCH1"/>
         </module>
         <module name="Architecture 2">
            <option href="http://www.saba.com/offerings.rdf#WINARCH2"/>
         </module>
         <module name="Compiler">
            <option href="http://www.saba.com/offerings.rdf#VJ60"/>
            <option href="http://www.saba.com/offerings.rdf#VC60"/>
         </module>
         <module name="Database">
            <option href="http://www.saba.com/offerings.rdf#MSSQL"/>
         </module>
      </path>
   </certification>
</certificationLibrary>
```

Each option specifies a URL referencing a specific offering, such as a course or orderable product, defined in a Catalog Format document.

As illustration, the following is a fragment of the Catalog Format document referenced by the above example:

```
<?xml version="1.0" encoding="UTF-8"?>

<rdf:RDF xml:lang="en"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:ims="http://www.saba.com/RDF/ims10.rdf" >

   <rdf:Description ID="WINARCH1">
      <ims:metadataScheme>RDF/LOM-1.0</ims:metadataScheme>
      <dc:title>Introduction to Windows Architecture</dc:title>
      <dc:identifier>WINARCH1</dc:identifier>
      <dc:description>
         Introductory concepts of Windows NT architecture.
      </dc:description>
   </rdf:Description>
```

```
   <rdf:Description ID="MSSQL">
      <ims:metadataScheme>RDF/LOM-1.0</ims:metadataScheme>
      <dc:title>Microsoft SQL Server 7.0</dc:title>
      <dc:identifier>MSSQL</dc:identifier>
      <dc:description>
         Administration of Microsoft SQL Server.</dc:description>
   </rdf:Description>

</rdf:RDF>
```

You can express nested certifications by including a reference to another Certification Format document.

Certifications can refer to or be referenced by metadata descriptions in external Catalog Format documents. This makes it possible to represent certification metadata such as prerequisites, equivalents, and authorship. For example, a Catalog Format document describing the prerequisites for the example certification above might be:

```
<rdf:Description
   about=http://www.saba.com/certifications/Microsoft.xml#MSCD">
   <dc:relation rdf:parseType="Resource">
      <dcq:relationType>Prerequisite</dcq:relationType>
      <rdf:value rdf:resource=
        "http://www.saba.com/certifications/Microsoft.xml#MSINTRO"/>
   </dc:relation>
</rdf:Description>
```

Certification DTD Elements

The table below lists and describes the elements defined in the Certification DTD:

Table 2: *Certification DTD Elements*

Element	Description	Attributes
certificationLibrary	The **certificationLibrary** element is the root element in a Certification Format document. It defines one or more certifications The **certificationLibrary** element is always required and occurs exactly once in every Certification Format document. Subelements of **certificationLibrary** include: **certification**	

Table 2: *Certification DTD Elements*

Element	Description	Attributes
certification (subelement of CertificationLibrary)	The **certification** element defines one or more paths to be followed in order to complete the certification. It also includes an optional summary and set of objectives for the certification. Occurs one or more times per **certificationLibrary** instance. Subelements of **certification** include: **summary** **objective** **path**	Date Attributes **id** name description metadataRef acquireWithin expiresIn
summary (subelement of certification)	Defines a text description for the certification. Occurs zero or one times per **certification** instance.	
objective (subelement of certification)	Defines an objective for the certification. Typically, this is a statement of the skills and knowledge acquired via the certification. Occurs zero or more times per **certification** instance.	
path (subelement of certification)	Defines a path for completing the certification. A path consists of a sequence of learning modules, where each module represents one step along the path. A certification can have multiple paths. Each path must specify whether it is for new certifications or re-certifications. Occurs one or more times per **certification** instance. Subelements of **path** include: **module** **option**	Date Attributes name type

Table 2: *Certification DTD Elements*

Element	Description	Attributes
module (subelement of path)	Defines one step in a path for completing a certification. A learning module consists of one or more options (learning interventions) for completing the module. A **module** can be optional or required and can specify a minimum number of required options. Occurs zero or more times per **path** instance.	name description type minimum
option (subelement of module)	Defines an option (learning intervention) for completing the learning module. Typically, this is a URL reference to a course defined in a Catalog Format document or to a specific competency or another certification. It can also be a text-based description of an action to take. Occurs one or more times per **module** instance.	href

Date Attributes

Some Certification Format elements share a common set of date attributes. The table below lists and describes these date attributes:

Table 3: *Date Attributes*

Attribute	Description
startDate	Specifies the start date for optional date range within which a certification or specific certification path is valid and available. Uses ISO8601 format. This attribute is optional.
endDate	Specifies the end date for optional date range within which a certification or specific certification path is valid and available. Uses ISO8601 format. This attribute is optional. **Note** You can represent discontinued certifications by specifying an **endDate** attribute in the past. Replacement certifications for discontinued certifications can be represented using a Catalog Format document containing a **relation** attribute with the value **replaces**.

Other Attributes

The table below lists and describes other attributes defined in the Certification DTD:

Table 4: *Other Certification DTD Attributes*

Attribute	Description
id	Standard XML identifier. Must be unique within the document. This attribute is required.
name (certification, path, module)	Specifies a name for the certification, path, or module. This attribute is optional.
description (certification, module)	Specifies a text description for the certification or module. This attribute is optional.
metadataRef	URL pointer to an external Catalog Format document containing metadata for the certification library. This attribute is optional.
acquireWithin	Specifies the number of days within which the certification must be acquired. This attribute is optional.
expiresIn	Specifies the number of days for which the certification is valid. This attribute is optional.
type (path)	Contains a value that indicates whether the certification path is intended for new certifications or re-certifications. Value can be: • **new** • **refresh** The default value is **new**.
type (module)	Contains a value that indicates whether the learning module is an optional or required step within the certification path. Value can be: • **optional** • **required** The default value is **required**.

Table 4: *Other Certification DTD Attributes*

Attribute	Description
minimum	Specifies the minimum number of options required to complete the module. This attribute is optional. The default value is **1**, which indicates that any one option fulfills the module.
href	Contains a URL pointer to an external resource for fulfilling the option. This attribute is optional.

Certification Results DTD

The Certification Reults DTD is designed to capture the information provided by third-party certification agencies. It supports the update of learner certifications in cases where certifications are managed and granted by an external agency.

Note The Certification Results Format only expresses whether or not a certification was successfully attained. It is not intended to be a full-fledged test result format.

To view the DTD, see “Certification Results Format DTD”.

The following is an example of a simple Certification Result Format document:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE results SYSTEM
   "http:\\www.saba.com\xml\certification_result10.dtd">

<results>
   <result>
      <learner>John Doe</learner>
      <certification>http://www.saba.com/certifications.xml#MSCD
      </certification>
      <date>1999-12-01</date>
      <score value="pass"/>
   </result>
   <result>
      <learner>Pat Rose</learner>
      <certification>http://www.saba.com/certifications.xml#MSCD
      </certification>
      <date>1999-12-01</date>
      <score value="fail"/>
   </result>
</results>
```

Certification Results DTD Elements

The table below lists and describes the elements defined in the Certification Results DTD:

Table 5: *Certification Results DTD Elements*

Element	Description	Attributes
results	The **results** element is the root element in a Certification Results Format document. It captures one or more sets of results reported by an external certification testing agency. The results element is always required and occurs exactly once in every Certification Results Format document. Subelements of **results** include: **result**	
result (subelement of results)	Contains information pertaining to a set of results. Occurs one or more times per **results** instance. Subelements of **result** include: **learner** **certification** **date** **score**	
learner (subelement of result)	Specifies the name of the individual applying for the certification. The format of this value is provider-specific Occurs exactly one time per **result** instance.	
certification (subelement of result)	Specifies the assessment needed to achieve the certification. The format of this value is provider-specific, although a typical format might be a URL pointer to a certification defined in a Certification Format document. Occurs exactly one time per **result** instance.	
date (subelement of result)	Specifies the date on which the certification was assessed. Uses ISO8601 format. Occurs exactly one time per **result** instance.	

Table 5: *Certification Results DTD Elements*

Element	Description	Attributes
score (subelement of result)	Specifies whether the certification was passed or failed. score is an empty element with attribute **value**. **value** is a required attribute that contains one of the following values: • **pass** • **fail** Occurs exactly one time per **result** instance.	value

Profile Format Specification

This document is the technical specification for Profile Format, the component of Universal Learning Format designed for describing learner profiles.

It includes the following sections:

Introduction

Profile Format is an XML-based representation for describing learner profile information. Learner profiles comprise a variety of data about learners, including personal and job information, learning history, goals and plans, and held competencies and certifications. Profile Format captures this information in an XML-based format using RDF to define metadata for describing learners. Profile Format incorporates several existing metadata standards, including the Dublin Core and vCard, which ensures compatibility with existing person/profile descriptions.

Usage

By employing Profile Format to describe the learners in a system, learning providers can extend their learning management architecture to support all of the following:

- Searches of critical learner metadata such as name, title, role, learning results, and held competencies and certifications
- Tracking the learning history of individual learners
- Assignment of competencies (with proficiency levels) and certifications to learners
- Assignment of learning goals to learners and tracking of progress towards fulfillment of those goals
- Creation of distributed profiles, where portions of a learner's profile are provided by different sources
- Compatibility with standard web search engines

Structure

Profile Format is based on open standards and is designed to reflect the following principles:

- Compatibility with emerging industry standards for learning profiles, including ongoing work in IMS and IEEE.
- Extensibility to easily accommodate future growth and change.

Profile Format employs an XML standard known as RDF (Resource Description Framework), the standard for defining metadata to describe web-based resources. The use of RDF makes it possible to define a set of unique RDF properties and merge these properties with properties defined in existing standards, such as vCard and Dublin Core. RDF also provides a unified mechanism for manipulating and querying this merged metadata. Furthermore, the use of RDF allows Profile Format to support distributed profiles, where portions of a learner's profile are provided by different sources.

Profile Format Schemas

Profile Format is supported by the following XML-based representations:

Schema	Description	Where To Find
Profile Format RDF Schema	Defines the format for describing learner profiles.	The Profile Format RDF schema has the following URL: http://www.saba.com/RDF/profile10.rdf To view the LCF DTD, see "Learning Content Format DTD".
Profile Format XML Schema	Defines the format for describing learner profiles.	To view the LCF XML Schema, see "LCF XML Schema".

Note The remainder of this document defines the specifications for the Profile Format RDF Schema.

Profile Format RDF Schema

The data model used to describe resources in Profile Format is defined in a set of external RDF schemas. Profile Format documents must reference these schemas in an XML namespace in order to validate the properties they describe.

For example, in order to use properties from the Dublin Core schema, a Profile Format document must define an XML namespace that references the URL of the Dublin Core schema, as follows:

```
xmlns:dc="http://purl.org/dc/elements/1.1/"
```

The following table lists and describes the RDF schemas used by Profile Format:

Table 1: *Profile Format Schemas*

Schema	Description
RDF	RDF is a standard framework for describing and interchanging metadata. The **RDF** schema represents a set of properties for describing web resources. It has the following URL: http://www.w3.org/1999/02/22-rdf-syntax-ns# The **RDFS** schema represents the meaning, characteristics, and relationships of the RDF properties. It has the following URL: http://www.w3.org/TR/1999/PR-rdf-schema-19990303# For more information on RDF and RDFS, see: http://www.w3.org/TR/REC-rdf-syntax/
Profile	**Profile Format** is an RDF schema designed to represent information about learner profiles. The Profile Format schema has the following URL: http://www.saba.com/RDF/profile10.rdf To view the Profile Format RDF schema, see Profile Format RDF Schema.
vCard	**vCard** is a standard RDF schema comprising elements that represent information about people and organizations such as that which is profiled on a business card. Some of the IMS metadata elements are vCard properties, which are represented using an RDF mapping. The vCard schema has the following URL: http://www.imc.org/vCard/3.0# Fore more information on support for vCard within RDF, see: http://www.dstc.edu.au/Research/Projects/rdf/draft-iannella-vcard-rdf-00.txt

Table 1: *Profile Format Schemas*

Schema	Description
Dublin Core	**Dublin Core** is a standard RDF schema comprising fifteen “core” elements that represent essential aspects related to the description of resources. A subset of the IMS metadata elements maps directly to the Dublin Core. The **Dublin Core Qualifiers** schema extends the descriptions of the fifteen “core” elements through the use of qualification and substructure. The Dublin Core schema has the following URL: http://purl.org/dc/elements/1.1/ The Dublin Core Qualifiers schema has the following URL: http://purl.org/dc/qualifiers/1.0/ For more information on support for Dublin Core within RDF, see: http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/
Competency	**Competency Format** is an RDF schema designed to represent information about held competency levels. The Competency schema has the following URL: http://www.saba.com/RDF/competency10.rdf To view the Competency Format RDF schema, see Competency Schema. **Note** For detailed information about the Competency Format RDF Schema, see the Competency Format Specification document.
Certification	**Certification Format** is an RDF schema designed to represent information about held certifications. The Certification schema has the following URL: http://www.saba.com/RDF/certification10.rdf To view the Certification Format RDF schema, see Certification Schema. **Note** For detailed information about the Certification Format RDF Schema, see the Certification Format Specification document.

Description Element

A Profile Format document is an RDF document that contains one or more **Description** elements, where each element describes a learner in the system. Each **Description** element contains a unique identifier and a set of property/value pairs that fully describe the learner. These properties can draw from any of the Profile Format RDF schemas.

Each **Description** element has an attribute that unambiguously identifies the learner being described. This attribute can be either of the following:

- **about**
- **id**

See below for more detailed information on using these identifier attributes.

The **Description** element can also include the **xml:lang** attribute for specifying the language in which the metadata description is authored. The **xml:lang** attribute contains the ISO 639/RFC 1766 language code with an optional geographic identifier, such as **en** for English, or **fr** for French.

Using the “id” Attribute

The **id** attribute specifies a unique ID for the learner, for example:

```
<rdf:Description id=”sally_brown” xml:lang=”en-US”>
...
</rdf:Description>
```

Using the “about” Attribute

The **about** attribute specifies the URL of a resource, for example:

```
<rdf:Description about=”http://www.saba.com/people/sally_brown”
xml:lang=”en-US”>
...
</rdf:Description>
```

The **about** attribute is useful when the metadata description applies to learner information defined in some other location, such as a home page.

Note Profile Format documents generated for data defined in Saba, either for export or for use in internal metadata searches, can use the **about** attribute to indicate the internal ID of the described offering.

Summary of Profile Format Properties

The table below lists and describes the properties defined by Profile Format:

Note For more detailed information about these properties, see the section entitled "Detailed Description of Profile Format Properties" on page 10.

Table 2: *Profile Format Properties (page 1 of 3)*

Schema	Property	Description	Applies To
RDF (rdf)	type	Learner	Resource
vCard (vCard)	N	Name	Resource
	FN	Formatted Name	Resource
	BDAY	Birthdate	Resource
	ADR	Address	Resource
	TEL	Telephone Number	Resource
	EMAIL	Email address	Resource
	TZ	Timezone	Resource
	TITLE	Title	Resource
	ROLE	Role in organization	Resource
	ORG	Membership in organization	Resource
	KEY	Public key value	Resource
Profile (profile)	learning	Completed or current learning offering	Resource
	status	Status of a learning offering or goal	profile:learning profile:goal
	result	Result of taking an offering	profile:learning

Table 2: *Profile Format Properties (page 2 of 3)*

Schema	Property	Description	Applies To
	date	Date of a result Date of an observation	profile:result profile:observation
	score	Score of a result	profile:result
	goal	A goal	Resource profile:observation
	description	Detailed goal description	profile:goal
	startDate	Start date for a goal	profile:goal
	endDate	Completion date for a goal	profile:goal
	priority	Priority of a goal	profile:goal
	parent	Parent goal	profile:goal
	observation	Observation on progress towards a goal	Resource
	comment	Observation comment	profile:observation
	observer	Observation made by	profile:observation
	completion	Percentage of goal completed	profile:observation
	language	Preferred language	Resource
	country	Preferred country	Resource
	identifier	A unique identifier	Resource
	source	Organization or source of the identifier	profile:identifier
Competency (comp)	competency	A held competency	Resource
	date	Date a competency was attained	comp:competency

Table 2: *Profile Format Properties (page 3 of 3)*

Schema	Property	Description	Applies To
	provider	Provider of a competency	comp:competency
	details	How a competency was attained	comp:competency
Certification (cert)	certification	A held certification	Resource
	date	Date a certification was attained	cert:certification
	startDate	Date a certification was started	cert:certification
	status	Current status of certification	cert:certification
Dublin Core (dc)	language	Language of resource	Resource
	description	Description	Resource
	publisher	Publisher	Resource
	date	Date	Resource
	coverage	Date range	Resource

Note Since Profile Format is expressed as an RDF document, you can associate any valid RDF property with a catalog entry, as long as you reference the RDF schema from which it derives. This openness provides a straightforward path for future extensibility.

Detailed Description of Profile Format Properties

Profile Format subdivides learner information into the following categories:

Table 3: *Profile Format Property Categories*

Category	Description
Personal Information	Includes information such as name, address, title, role(s), and organization membership. All personal information is represented using RDF mappings of the vCard specification. For more information about these properties, see "Personal Information (from vCard Schema)" on page 14.
Learning Information	Includes information on held learning and planned learning. This category contains a pointer to a learning resource described in a Catalog Format document, such as a class or WBT, with optional properties capturing the status and results of the learning. For more information about these properties, see " Learning Information" on page 19.
Goal Information	Includes information on personal and professional objectives. This category provides a description of the goal, the plan of action for achieving the goal, and the status of the goal. For more information about these properties, see " Goal Information" on page 21.
Observation Information	Includes information on a learner's progress towards specific goals. This category captures specific, measurable milestones on the path towards achieving a goal. For more information about these properties, see " Observation Information" on page 21.
Competency Information	Includes information on held competencies. This category contains a pointer to a competency defined in a Competency Format document, with optional properties describing how the competency was attained. For more information about these properties, see " Competency Information" on page 23.

Table 3: *Profile Format Property Categories*

Category	Description
Certification Information	Includes information on held certifications. This category contains a pointer to a certification track defined in a Certification Format document, with optional properties describing how the certification was attained. For more information about these properties, see “ Certification Information” on page 25.
Preference Information	Includes information on learner preferences, such as home language and country. For more information about these properties, see “ Preference information” on page 26.
Profile Information	Includes information about the profile itself, such as the date it was generated and the language it is in. All profile information is represented using the RDF mappings of Dublin Core. For more information about these properties, see “ Profile Information (from Dublin Core Schema)” on page 27.

Note The sections below provide detailed information about the properties belonging to each of these categories.

Personal Information (from vCard Schema)

Profile Format uses the RDF mappings for the **vCard** specification to represent personal information about learners, including data such as name, address, title, roles, and organization.

vCard is a metadata standard for representing information about people and organizations such as that which is profiled on a business card.

For more information about the vCard standard,see:
http://www.imc.org/pdi/

Fore more information on support for vCard within RDF, see:
http://www.dstc.edu.au/Research/Projects/rdf/draft-iannella-vcard-rdf-00.txt

Profile Format supports the following properties as defined by the vCard specification:

Table 4: *Personal Information Properties (from vCard specification)*

Property	Representation	Description
Name	vCard:N	Contains a series of substructure properties that define pieces of the learner's name. The substructure properties include: • **Family** — family name • **Given** — given name • **Other** — additional name • **Prefix** — honorific prefix • **Suffix** — honorific suffix
Formatted Name	vCard:FN	Specifies the learner's full name.
Birthdate	vCard:BDAY	Specifies the learner's date of birth.
Delivery Address	vCard:ADR	Specifies the learner's mailing address.
Telephone Number	vCard:TEL	Specifies the learner's telephone number.
Electronic Mail	vCard:EMAIL	Specifies the learner's Internet email address. **Note** Profile Format supports a single value that represents a learner's Internet email address.
Timezone	vCard:TZ	Specifies the learner's time zone.
Title	vCard:TITLE	Specifies the learner's title.
Business Category	vCard:ROLE	Specifies the learner's role in the organization.
Organization Name and Unit	vCard:ORG	Contains a series of substructure properties that define pieces of the organization's identification. The substructure properties include: • **Orgname** — name of the organization • **Orgunit** — name of the organizational unit

Table 4: *Personal Information Properties (from vCard specification)*

Property	Representation	Description
Public Key	vCard:KEY	Specifies the public encryption key associated with the learner. Valid values for this property are a public key that conforms to a bilaterally agreed to representation. If the representation is a binary format, then the public key must be further encoded. The default format is clear-text. If a binary format is used, then it is specified by the property parameter.
Identifier	profile:identifier	Specifies a unique identifier for the learner. This property is qualified with a **profile:source** property that identifies the source of the identifier. Currently, the only supported value for **profile:source** is **Social Security**.

Profile Format Example with Personal Information

The following Profile Format example provides basic contact information about a learner:

```
<?xml version="1.0" encoding="UTF-8"?>
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:vCard="http://imc.org/vCard/3.0#" >

   <rdf:Description about="http://www.saba.com/people/sally_brown" >
      <rdf:type rdf:resource=
      "http://www.saba.com/RDF/profile
      10.rdf#learner"/>
      <vCard:FN>Sally Brown</vCard:FN>
      <vCard:N rdf:parseType="Resource">
         <vCard:Family>Brown</vCard:Family>
         <vCard:Given>Sally</vCard:Given>
         <vCard:Prefix>Ms</vCard:Prefix>
      </vCard:N>
      <vCard:TITLE> Senior Technical Staff </vCard:TITLE>
      <vCard:ROLE> Programmer </vCard:ROLE>
      <vCard:TEL rdf:parseType="Resource">
         <rdf:value> 650-581-4444 </rdf:value>
         <vCard:TYPE rdf:resource ="http://imc.org/vCard/3.0#work" />
      </vCard:TEL>
      <vCard:EMAIL> sbrown@saba.com </vCard:EMAIL>
      <vCard:ADR rdf:parseType="Resource">
         <vCard:Street>2400 Bridge Parkway</vCard:Street>
```

```
            <vCard:Locality>Redwood Shores</vCard:Locality>
            <vCard:Region>CA</vCard:Region>
            <vCard:Pcode>94065</vCard:Pcode>
            <vCard:TYPE rdf:resource ="http://imc.org/vCard/3.0#work" />
         </vCard:ADR>
         <profile:identifier rdf:parseType="Resource">
            <rdf:value>110-00-0000</rdf:value>
            <profile:source>Social Security</profile:source>
         </profile:identifier>
      </rdf:Description>
</RDF>
```

Learning Information

Profile Format defines a set of RDF properties that capture information on a learner's current learning (current enrollments) and learning history (transcript).

The **learning** property specifies a URL to a learning resource described in a Catalog Format document. The specified resource is a held offering in the learner's transcript.

In its simplest form, the **learning** property contains only the URL of the held learning offering, for example:

```
<profile:learning rdf:resource="http://www.saba.com/learning/
catalog.rdf#JAVA101"/>
```

The **learning** property can also be a structured property, with substructure properties that qualify the status of the learning offering and the conditions under which it was attained. For qualified instances of the **learning** property, the URL of the learning resource is captured using the **rdf:value** property.

The table below lists and describes the properties that can be used to qualify the **learning** property:

Table 5: *Qualifiers for learning Property*

Property	Representation	Description
status	profile:status	Specifies the status of the learning offering. Learning can be either completed (part of a learner's transcript) or current (part of a learner's current enrollments). Valid status values for completed learning are: • Complete • Incomplete • Dropped • No Show • Cancelled Valid status values for current learning are: • Enrolled • In Progress • Waitlisted • Pending Approval
result	profile:result	Specifies the result of taking the learning. Typically, the **result** property holds a value such as **Pass** or **Fail** or a grade such as **A** or **B**. To represent multiple results (such as multiple tests taken during the course of a class), the **result** property can be structure as an RDF Bag, and each instance can be qualified with the **date** and **score** properties.
date	profile:date	Specifies the date the result was attained. Uses ISO 8601 format.
score	profile:score	Specifies the score based on which the result was attained

Profile Format Example with Learning Information

The following Profile Format example captures a learner's current learning profile and learning transcript:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:profile="http://www.saba.com/RDF/profile10.rdf">

<rdf:Description about = "http://www.saba.com/people/sally_brown" >
<rdf:type rdf:resource=
"http://www.saba.com/RDF/profile10.rdf#learner"/>

<!-- current learning -->

   <profile:learning rdf:parseType="Resource">
      <rdf:value resource="http://www.saba.com/learning/
      catalog.rdf#Banking101.2"/>
      <profile:status>Enrolled</profile:status>
   </profile:learning>

   <profile:learning rdf:parseType="Resource">
      <rdf:value resource="http://www.saba.com/learning/
      catalog.rdf#AlgebraIntro.1"/>
      <profile:status>In Progress</profile:status>
   </profile:learning>

<!-- learning history -->

<!-- a class with a simple result -->
   <profile:learning rdf:parseType="Resource">
      <rdf:value resource="http://www.saba.com/learning/
      catalog.rdf#Essays.1"/>
      <profile:status>Complete</profile:status>
      <profile:result>Pass</profile:result>
   </profile:learning>

<!-- a class for which a single test was taken -->
   <profile:learning rdf:parseType="Resource">
      <rdf:value resource="http://www.saba.com/learning/
      catalog.rdf#Java101.1"/>
      <profile:status>Incomplete</profile:status>
      <profile:result rdf:parseType="Resource">
         <profile:score>95</profile:score>
         <rdf:value>Pass</rdf:value>
      </profile:result>
   </profile:learning>
```

```
<!-- a class for which multiple tests were taken -->
   <profile:learning rdf:parseType="Resource">
      <rdf:value resource=
      "http://www.saba.com/learning/catalog.rdf#MathIntro.1"/>
      <profile:status>Incomplete</profile:status>
      <profile:result>
         <rdf:Bag>
            <rdf:li parseType="Resource">
               <profile:date>2000-03-15</profile:date>
               <profile:score>95</profile:score>
               <rdf:value>A</rdf:value>
            </rdf:li>
            <rdf:li parseType="Resource">
               <profile:date>2000-03-30</profile:date>
               <profile:score>85</profile:score>
               <rdf:value>B</rdf:value>
            </rdf:li>
         </rdf:Bag>
      </profile:result>
   </profile:learning>

</rdf:Description>
</rdf:RDF>
```

Catalog Format Example with Profile Information

The following example shows a fragment of a Catalog Format document that contains some of the learning resources referred to in the example above. It demonstrates how additional class information, such as class start date and class location, can be obtained by following the offering references specified in a Profile Format document.

```
<!-- generic offering templates -->
<rdf:Description about="Banking101">
   <dc:identifier>Banking101</dc:identifier>
   <dc:description>Introducting to banking concepts</dc:description>
   <rdf:type resource=
   "http://www.saba.com/RDF/offering10.rdf#offering"/>
</rdf:Description>

<rdf:Description about="AlgebraIntro">
   <dc:identifier>AlgebraIntro</dc:identifier>
   <dc:description>Algebra for Dummies</dc:description>
   <rdf:type resource="http://www.saba.com/RDF/
   offering10.rdf#offering"/>
</rdf:Description>
```

```
<!-- instances of classes based on the above templates -->
<rdf:Description id="Banking101.1">
   <rdf:type resource="http://www.saba.com/RDF/offering10.rdf#event"/>
   <dc:relation rdf:parseType="Resource">
      <dcq:relationType>InstanceOf</dcq:relationType>
       <rdf:value rdf:resource="#Banking101"/>
   </dc:relation>
   <schedule:startDate>2000-10-09</schedule:startDate>
   <schedule:endDate>2000-10-13</schedule:endDate>
   <schedule:location>Seattle</schedule:location>
</rdf:Description>

<rdf:Description id="Banking101.2">
   <rdf:type resource="http://www.saba.com/RDF/offering10.rdf#event"/>
   <dc:relation rdf:parseType="Resource">
      <dcq:relationType>InstanceOf</dcq:relationType>
      <rdf:value rdf:resource="#Banking101"/>
   </dc:relation>
   <schedule:startDate>2000-11-06</schedule:startDate>
   <schedule:endDate>2000-11-10</schedule:endDate>
   <schedule:location>Seattle</schedule:location>
</rdf:Description>

<rdf:Description id="AlgebraIntro.1">
   <rdf:type resource="http://www.saba.com/RDF/offering10.rdf#event"/>
   <dc:relation rdf:parseType="Resource">
      <dcq:relationType>InstanceOf</dcq:relationType>
      <rdf:value rdf:resource="#AlgebraIntro"/>
   </dc:relation>
   <schedule:startDate>2000-08-09</schedule:startDate>
   <schedule:endDate>2000-08-16</schedule:endDate>
   <schedule:location>San Francisco</schedule:location>
</rdf:Description>
```

Note While catalog references will most typically point to pre-existing Catalog Format documents, there may also be cases where profile information needs to be exported as a single, self-contained document. In this case the catalog descriptions may be placed into the same file as the learner descriptions.

Goal Information

Profile Format defines a set of RDF properties that capture information about a learner's goals. Goals can encompass both business and professional objectives for a learner and include the following additional information:

- planned actions for achieving the goal
- learning interventions
- accomplishments

The **goal** property defines a learner's goal.

In its simplest form, the **goal** property contains the name of an in-progress goal and has an optional **id** attribute, for example:

```
<profile:goal rdf:id="JAVA1">Become a Java expert</profile:goal>
```

Alternatively, the goal can be represented as a reference to a particular competency level, certification, or learning intervention. In these cases the **resource** attribute is used to reference the relevant URL:

```
<profile:goal rdf:resource=
   "http://www.saba.com/competencies/programming#Java.Expert"/>
<profile:goal rdf:resource="http://www.saba.com/certifications/
certifications.xml#MSCD"/>
<profile:goal rdf:id="VJ++"
   rdf:resource="http://www.saba.com/courses/ms.rdf#VJ60">
   Learn Visual J++
</profile:goal>
```

The **goal** property can also be a structured property, with substructure properties that provide details about the goal and its status. For qualified instances of the **goal** property, the URL of a qualified goal is captured using the **rdf:value** property.

The table below lists and describes the properties that can be used to qualify the **goal** property:

Table 6: *Qualifiers for goal Property*

Property	Representation	Description
description	profile:description	Specifies a text description of the goal.
startDate	profiile:startDate	Specifies the target start date for beginning work on the goal. Uses ISO 8601 format.
endDate	profile:endDate	Specifies the target completion date for a goal. Uses ISO 8601 format.

Table 6: *Qualifiers for goal Property*

Property	Representation	Description
priority	profile:priority	Specifies a numeric value for the relative priority of this goal, where
status	profile:status	Specifies the current status of the goal, for example **Accomplished** or **In Progress**.
parent	profile:parent	Contains a reference to the parent goal from which the goal is descended. The reference is a pointer to a goal defined in a Profile Format document. **Note** Goals do not have to reference a parent goal.

Note You can use RDF Bags or Seqs to group goals together. Use an RDF Seq to indicate a required order for a set of goals.

Profile Format Example with Goal Information

The following Profile Format example defines a set of related goals for a learner:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:profile="http://www.saba.com/RDF/profile10.rdf">

<rdf:Description about="http://www.saba.com/people/sally_brown">
<rdf:type rdf:resource=
"http://www.saba.com/RDF/profile10.rdf#learner"/>

   <profile:goal rdf:id="JAVA1">Become a Java expert</profile:goal>
      <profile:goal>
         <rdf:Seq>
            <rdf:li parseType="Resource">
               <rdf:value>Read Java in a Nutshell</rdf:value>
               <profile:startDate>2000-04-24</profile:startDate>
               <profile:endDate>2000-04-28</profile:endDate>
               <profile:parent rdf:resource="#JAVA1"/>
            </rdf:li>
            <rdf:li parseType="Resource">
               <rdf:value rdf:resource=
               "http://www.saba.com/courses/ms.rdf#JavaPatterns">
               Take a class in Java design concepts</rdf:value>
               <profile:startDate>2000-05-01</profile:startDate>
```

```
                <profile:endDate>2000-05-31</profile:endDate>
                <profile:parent rdf:resource="#JAVA1"/>
             </rdf:li>

             <rdf:li id="JAVAPORT" parseType="Resource">
                <rdf:value>Port current C++ code to Java</rdf:value>
                <profile:startDate>2000-05-01</profile:startDate>
                <profile:endDate>2000-06-31</profile:endDate>
                <profile:description>A major, hands-on undertaking to
                provide practical experience in Java design and
                coding</profile:description>
                <profile:priority>2</profile:priority>
                <profile:parent rdf:resource="#JAVA1"/>
             </rdf:li>
          </rdf:Seq>
       </profile:goal>

   </rdf:Description>
</rdf:RDF>
```

Observation Information

Profile Format defines a set of RDF properties that capture information reflecting a learner's progress towards specific goals. These observations track specific, measurable milestones on the path towards achieving a goal.

The **observation** property defines an observation on a goal. It is a structured property that can contain the following substructure properties:

Table 7: *Qualifiers for observation Property*

Property	Representation	Description
value	profile:value	Specifies the value of an observation. The value of an observation can be one of a predefined set of possible values for tracking progress towards a goal. It can either be plain text or a numeric value.
date	profile:date	Specifies the date on which an observation is made. Uses ISO 8601 format.
comment	profile:comment	Contains a text description providing further details on the observation.
observer	profile:observer	Specifies the name of the person or entity performing the observation.

Table 7: *Qualifiers for observation Property*

Property	Representation	Description
completion	profile:completion	Specifies a percentage value (1-100) indicating progress towards completion of the goal.
goal	profile:goal	Contains a reference to the goal for which the observation is made. The reference is a pointer to a goal defined in a Profile Format document.

Profile Format Example with Observation Information

For example, the following properties define a set of observations toward the "JAVAPORT" goal defined above:

```
<profile:observation>
   <rdf:Bag>
      <rdf:li parseType="Resource">
         <profile:date>2000-05-06</profile:date>
         <profile:completion>20</profile:completion>
         <profile:goal rdf:resource="#JAVAPORT"/>
         <profile:observer rdf:resource=
         "http://www.saba.com/people/charles_brown"/>
      </rdf:li>
      <rdf:li parseType="Resource">
         <profile:date>2000-05-13</profile:date>
         <profile:completion>40</profile:completion>
         <profile:comment>Slightly ahead of schedule</profile:comment>
         <profile:goal rdf:resource="#JAVAPORT"/>
      </rdf:li>
   </rdf:Bag>
</profile:observation>
```

Competency Information

Profile Format includes the RDF properties defined by Competency Format for tracking held competencies.

The table below lists and describes the properties defined in the Competency Format RDF Schema:

Table 8: *Competency Format RDF Schema Properties*

Property	Representation	Description
competency	ewp:competency	Specifies a set of one or more competencies and proficiency levels associated with a learning offering. The **competency** property is a literal or RDF Bag. Typically, its value is a URL pointer to a competency ID in a competency library defined using Competency Format and published as a web document. **Note** For convenience, it is also possible for the **competency** property to contain just the ID value of the proficiency level. This is most useful in a scenario where you are generating competency metadata for the purpose of querying. Assuming a fixed set of pre-imported competencies, specifying just the ID value allows subsequent queries to be simpler.
attained	ewp:attained	Specifies the date a held competency was attained. Uses ISO 8601 format.
provider	ewp:provider	Specifies the competency provider that provided a held proficiency level.
details	ewp:details	Specifies any further information about how a held proficiency level was attained.

Profile Format Example with Competency Information

The following Profile Format fragment demonstrates the use of simple and complex competency properties.

```
<Description about="http://www.saba.com/people/sally_brown">
<rdf:type rdf:resource=
"http://www.saba.com/RDF/profile10.rdf#learner"/>
   <ewp:competency>
      <rdf:Bag>
         <rdf:li resource=
         "http://www.saba.com/competencies/programming#XML.Novice"/>
         <rdf:li parseType="resource">
            <rdf:value resource=
            "http://www.saba.com/competencies/programming#Java.Expert"/>
            <ewp:date>1999-02-25</ewp:date>
            <ewp:provider resource="http://www.sabanet/AllAboutJava/"/>
            <ewp:details>Passed with flying colors</ewp:details>
         </rdf:li>
      </rdf:Bag>
   </ewp:competency>
</Description>
```

Certification Information

Profile Format includes the RDF properties defined by Certification Format for tracking held certifications.

The table below lists and describes the attributes defined in the Certification Format RDF Schema:

Table 9: *Certification Format RDF Schema Attributes*

Property	Representation	Description
certification	cer:certification	Specifies a held certification. The **certification** property is a literal or RDF Bag containing one or more values for certifications. Typically, its value is a URL pointer to a certification defined in a Certification Format document and published as a web document. **Note** For convenience, it is also possible for the certification attribute to contain just the ID value of Saba's internal certification object. This is most useful in a scenario where you are generating certification metadata for the purpose of querying. Assuming a fixed set of pre-imported certifications, specifying just the ID value allows subsequent queries to be simpler.
attained	cer:attained	Specifies the date a held certification was attained. Uses ISO 8601 format.
started	cer:started	Specifies the date a held certification was started. Uses ISO 8601 format.
status	cer:status	Specifies the current status of the certification. Status accepts the following values: **Planned** **In Progress** **Acquired** **Expired** **Discontinued** The default value is **Acquired**.

Profile Format Example with Certification Information

The following Profile Format fragment demonstrates the use of simple and complex certification properties

```
<Description about="http://www.saba.com/people/sally_brown">
   <cer:certification>
      <rdf:Bag>
         <rdf:li resource=
         "http://www.saba.com/certifications/certifications.xml#MSCD"/>
         <rdf:li parseType="Resource">
            <rdf:value resource="http://www.saba.com/certifications/
            certifications.xml#MSCD"/>
            <cer:date>1999-02-25</cer:date>
            <cer:status>Acquired</cer:status>
         </rdf:li>
      </rdf:Bag>
   </cer:certification>
</Description>
```

Preference information

Profile Format defines a set of RDF properties that capture information reflecting the learning preferences of a learner.

The table below lists and describes these learner preference properties:

Table 10: *Profile Format Learner Preference Properties*

Property	Representation	Description
language	profile:language	Specifies the preferred language of the learner. Uses ISO 639 format.
country	profile:country	Specifies the home country of the learner. Uses ISO 3166 format.

Profile Information (from Dublin Core Schema)

Profile Format includes a subset of the RDF properties defined by Dublin Core for specifying information about a specific profile.

Dublin Core is a standard metadata schema comprising fifteen "core" elements that represent essential data related to the description of resources on the web.

For more information on the Dublin Core, see the Dublin Core home page at:
http://purl.org/dc/

For more information on support for Dublin Core within RDF, see:
http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/

Profile Format supports the following properties as defined by the Dublin Core specification:

Table 11: *Profile Information Properties (from Dublin Core specification)*

Property	Representation	Description
description	dc:description	Specifies a description of the profile. Different Profile Format documents can capture different aspects of a learner's profile, such as a transcript or a goals summary. The **description** property can be used to describe the captured profile.
publisher	dc:publisher	Specifies the source of the profile, e.g. the system in which the learner's profile is stored.
date	dc:date	Specifies the date on which the profile was generated. Uses ISO 8601 format.
language	dc:language	Specifies the language in which the profile is authored. Uses ISO 639/RFC 1766 format.
coverage	dc:coverage	Specifies the date range for the time period that the profile covers. Uses ISO 8601 format.

Profile Format Examples

The following Profile Format example incorporates properties from several of the namespaces supported by Profile Format.

```
<?xml version="1.0" encoding="UTF-8"?>

<RDF xmlns = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:vCard = "http://imc.org/vCard/3.0#"
     xmlns:profile="http://www.saba.com/RDF/profile10.rdf"
     xmlns:cert="http://www.saba.com/RDF/certification10.rdf">

<Description about = "http://www.saba.com/people/sally_brown" >
<rdf:type rdf:resource=
"http://www.saba.com/RDF/profile10.rdf#learner"/>

<!-- personal information -->
   <vCard:FN>Sally Brown</vCard:FN>
   <vCard:ROLE> Programmer </vCard:ROLE>
   <vCard:ORG rdf:parseType="Resource">
      <vCard:Orgname>Saba Software</vCard:Orgname>
      <vCard:Orgunit>
         <Seq>
            <li>Research and Development</li>
            <li>Platform Group</li>
         </Seq>
      </vCard:Orgunit>
   </vCard:ORG>

<!-- learning information -->
   <profile:learning>
      <Bag>

<!-- current learning -->
         <li parseType="Resource">
            <value resource=
            "http://www.saba.com/learning/catalog.rdf#Banking101.2"/>
            <profile:status>Enrolled</profile:status>
         </li>

<!-- learning history -->
         <li resource=
         "http://www.saba.com/learning/catalog.rdf#MathIntro.1"/>
         <li resource=
         "http://www.saba.com/learning/catalog.rdf#Philosophy.5"/>
```

```
      </Bag>
   </profile:learning>

<!-- goal information -->
   <profile:goal>
      <Bag>
         <li>Become a Java expert</li>
         <li parseType="Resource">
            <value>Reduce bug count by 50%</value>
            <profile:status>In Progress</profile:status>
         </li>
         <li parseType="Resource">
            <value>Learn CVS</value>
            <profile:status>Completed</profile:status>
         </li>
      </Bag>
   </profile:goal>

<!-- held certifications -->
   <cert:certification>
      <Bag>
         <li resource=
         "http://www.saba.com/certifications/certifications.xml#MSCD"/>
         <li resource=
         "http://www.saba.com/certifications/certifications.xml#NOVL"/>
      </Bag>
   </cert:certification>

<!-- preferences -->
   <profile:language>en</profile:language>
   <profile:country>US</profile:country>

<!-- profile information -->
   <profile:publisher>http://www.saba.com</profile:publisher>
   <profile:date>2000-04-25</profile:date>

</Description>
</RDF>
```

Learning Content Format DTD

```
<!-- Learning Content Format (LCF) DTD
Daniel Lipkin
v1.0 -->

<!-- Learning Content Format (LCF) describes the assembly of on-line educational
material -->

<!-- Common attributes shared by objectives and content elements. -->
<!-- name is a human-readable identifier for the content element.
description is a text description of the content element.
metadataRef is an optional URI to a Catalog Format document containing metadata
for the specified element.
lang indicates the language used by the content element using RFC1766.  -->
<!ENTITY % common-attr "id ID #REQUIRED
                        name CDATA #IMPLIED
                        description CDATA #IMPLIED
                        metadataRef CDATA #IMPLIED
                        xml:lang NMTOKEN #IMPLIED">

<!-- Attributes shared by all content elements.  Notice that this includes all
the common attributes defined above. -->
<!-- objectiveRef references objectives associated with this content element.
timeAllowed indicates the number of minutes a specific learner is allowed to view
the content.  It is mostly intended for questions on tests, but may also be
applied to any content element.
accessCount indicates the number of distinct time a single learner is allowed to
view the content. If left unspecified, there are no restrictions on the number of
times the content can be viewed.
isVisible indicates whether or not the content is displayed.
startDate and endDate define a range of time in which a content element may be
viewed.
startDate and endDate are represented using ISO8601 format. -->
<!ENTITY % content-attr "%common-attr;
                         objectiveRef IDREFS #IMPLIED
                         timeAllowed CDATA #IMPLIED
                         accessCount CDATA #IMPLIED
                         isVisible (true | false) "true"
```

```
                        startDate CDATA #IMPLIED
                        endDate CDATA #IMPLIED">

<!-- Attributes shared by all question elements. Notice that this includes all
the content attributes defined above. -->
<!-- points defines the number of points a given question is worth.
questionnaire refers to the questionnaires with which this question is
associated.
title associates a title with this question. -->
<!ENTITY % question-attr "%content-attr;
                          points CDATA #IMPLIED
                          questionnaire IDREFS #IMPLIED
                          title CDATA #IMPLIED">

<!-- The different types of tests defined by LCF -->
<!ENTITY %questionnaire-elements "multipleChoice | trueFalse | essay">

<!-- The different types of content defined by LCF.   -->
<!ENTITY %content-elements "text | learningObject | questionnaire">

<!-- An LCF file contains a Table of Contents, followed by zero or more
objectives, followed by one or more content elements. -->

<!--An LCF file places all content elements at the top-most level, and uses a
Table of Contents to indicate the structure and nesting of the content elements.
This separation of content from structure makes it easier to vary them
independently, and ensures that deeply nested content does not require recursive
extraction. -->
<!ELEMENT lcf (toc,(objective)*,(%content-elements; | %questionnaire-elements; )+)>
<!ATTLIST lcf %content-attr;>

<!-- The type attribute provides a hint as to how the contents of the LCF are to
be interpreted. A type of "class" means the file defines a single, launchable
class. This is directly analogous to an IMS Content Packaging file, where each
"learningObject" is an external reference to a physical resource. A type of
"course" means the file defines a course structure.  This is directly analogous
to an ADL Structure Course Format file, where each "learningObject" element is an
au. A type of "questionnaire" means that the file contains an on-line
questionnaire only. -->
<!ATTLIST lcf type (class | course | questionnaire) "class">

<!-- A text element displays the specified text. -->
<!ELEMENT text  (#PCDATA )>
<!ATTLIST text %content-attr;>
```

```
<!-- A learningObject element is a reference to some form of external learning
content, identified by the URL value of the href attribute. The formats supported
for learningObject references are implementation-specific, although support for
common web formats such as HTML documents and JPEG images is recommended. -->
<!ELEMENT learningObject EMPTY>
<!ATTLIST learningObject %content-attr;
                  href CDATA #REQUIRED>

<!-- A questionnaire consists of optional introductory text, followed by one or
more questions. Questions can be multipleChoice, trueFalse, or essay.
A questionnaire has a type attribute to specify the context in which the
questionnaire is used.
A questionnaire contains one or more scorebands that specify score ranges.
There are two ways to specify the contents of a questionnaire:
1.  Use the TOC structure to indicate the exact order in which questionnaire
questions are presented.
2.  Use the "questionnaire" attribute to indicate the questionnaire to which
question belongs, and set a numeric value for the "randomize" attribute of the
questionnaire. The specified number of questions will be displayed in random
order. -->
<!ELEMENT questionnaire  (scoreband)*>
<!ATTLIST questionnaire %content-attr;
                  type (test | survey | assessment) "test">

<!-- passingScore is a simple mechanism to indicate a passing score for a test.
Tests may also employ scorebands to capture more sophisticated scoring ranges.
-->
<!ATTLIST questionnaire passingScore CDATA #IMPLIED>

<!-- resumable specifies whether or not the questionnaire state can be saved so
it may be resumed at a later time. -->
<!ATTLIST questionnaireresumable (true | false) #IMPLIED>

<!-- displayType specifies whether questions are presented on separate web pages
or on a single page. -->
<!ATTLIST questionnairedisplayType (together | separate) "separate">

<!-- randomize specifies whether questions should be presented in random order.
It contains the number of questions to present. If a value for randomize is not
specified, all questions are presented in order. -->
<!ATTLIST questionnairerandomize CDATA #IMPLIED>

<!-- warningTime specifies whether a warning should be presented prior to the end
of the questionnaire. It contains the number of minutes before the end of the
test at which the warning appears. -->
<!ATTLIST questionnairewarningTime CDATA #IMPLIED>
```

```
<!-- A scoreband has a required numeric upper bound and lower bound. It also has
optional feedback text content. -->
<!ELEMENT scoreband (text)?>
<!ATTLIST scoreband id ID #REQUIRED
                  lowerBound CDATA #REQUIRED
                  upperBound CDATA #REQUIRED>

<!-- A multiple choice question consists of a question followed by one or more
choices, followed by optional hints. In practice, there will be at least three
choice. -->
<!ELEMENT multipleChoice (question, choice+, hints?)>
<!ATTLIST multipleChoice %content-attr;
                  %question-attr;>

<!-- A question is either text or a reference to an external resource.  -->
<!ELEMENT question (text | learningObject)+>
```

```
<!-- The hints element contains one or more hints. The "type" attribute specifies
how hints are presented.  "Incremental" means that all hints up and to including
the current request are presented; "multilevel" means that only the hint element
corresponding to the number of time a hint has been requested is displayed. -->
<!ELEMENT hints (hint+)>
<!ATTLIST hints type (incremental | multilevel) "multilevel">

<!-- The text of a hint. -->
<!ELEMENT hint (#PCDATA)>
<!ATTLIST hint xml:lang NMTOKEN #IMPLIED>

<!-- One of the possible answers. At least one of the choices must have a value
of "true" for the "answer" attribute. -->
<!ELEMENT choice (#PCDATA)>
<!ATTLIST choice xml:lang NMTOKEN #IMPLIED
               answer (true | false) "false">

<!-- A True or False question. The "answer" attribute specifies the correct
answer for this question. -->
<!ELEMENT trueFalse (question, hints?)>
<!ATTLIST trueFalse %content-attr;
               %question-attr;
               answer (true | false) #REQUIRED>

<!-- An essay question. -->
<!ELEMENT essay (question, hints?)>
<!ATTLIST essay %content-attr;
               %question-attr;>

<!-- Objectives are statements of skills, knowledge, and attitudes to be acquired
by the student.  Each objective has an ID, and optional name and description.
Objectives may be nested. -->
<!ELEMENT objective (objective*)>
<!ATTLIST objective %common-attr;>

<!-- The toc (table of contents) defines the structure of the learning content. -
->
<!ELEMENT toc (item+)>

<!-- The item elements can be nested within each other to convey organizational
structure.  The title attribute defines the title of this entry.  The ref
attribute refers to a content element within the file. -->
<!ELEMENT item (item)*>
<!ATTLIST item title CDATA #IMPLIED>
<!ATTLIST item ref IDREF #IMPLIED>
```

LCF XML Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema">

   <xsd:annotation>
      <xsd:documentation>Learning Content Format Schema Daniel Lipkin v1.0
      </xsd:documentation>
   </xsd:annotation>

   <xsd:element name="lcf">
      <xsd:complexType content="elementOnly">
         <xsd:element name="toc" type="tocType" />
         <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="objective" type="objectiveType" />
         </xsd:choice>
         <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:group ref="content" />
            <xsd:group ref="questionnaire" />
         </xsd:choice>
         <xsd:attributeGroup ref="contentAttributes" />
         <xsd:attribute name="type" use="default" value="class">
            <xsd:simpleType base="xsd:string">
               <xsd:enumeration value="class" />
               <xsd:enumeration value="course" />
               <xsd:enumeration value="questionnaire" />
            </xsd:simpleType>
         </xsd:attribute>
         <xsd:attribute name="xmlns:xsi" type="xsd:uriReference" use="default"
         value="http://www.w3.org/1999/XMLSchema-instance" />
         <xsd:attribute name="xsi:schemaLocation" type="xsd:string" />
      </xsd:complexType>
      <xsd:key name="content">
         <xsd:selector>*</xsd:selector>
         <xsd:field>@id</xsd:field>
      </xsd:key>
```

```
      <xsd:key name="objective">
         <xsd:selector>//objective</xsd:selector>
         <xsd:field>@id</xsd:field>
      </xsd:key>
      <xsd:key name="questionnaire">
         <xsd:selector>questionnaire</xsd:selector>
         <xsd:field>@id</xsd:field>
      </xsd:key>
      <xsd:keyref name="item" refer="content">
         <xsd:selector>toc//item</xsd:selector>
         <xsd:field>@ref</xsd:field>
      </xsd:keyref>
      <xsd:keyref name="objectiveRef" refer="objective">
         <xsd:selector>*</xsd:selector>
         <xsd:field>@objectiveRef</xsd:field>
      </xsd:keyref>
      <xsd:keyref name="questionnaireRef" refer="questionnaire">
         <xsd:selector>multipleChoice | trueFalse | essay</xsd:selector>
         <xsd:field>@questionnaire</xsd:field>
      </xsd:keyref>
   </xsd:element>

   <xsd:group name="content">
      <xsd:element name="text" type="textType" />
      <xsd:element name="learningObject" type="learningObjectType" />
      <xsd:element name="questionnaire" type="questionnaireType" />
   </xsd:group>

   <xsd:group name="questionnaire">
      <xsd:element name="multipleChoice" type="multipleChoiceType" />
      <xsd:element name="trueFalse" type="trueFalseType" />
      <xsd:element name="essay" type="essayType" />
   </xsd:group>

   <xsd:complexType name="tocType" content="elementOnly">
      <xsd:element name="item" type="itemType" minOccurs="1"
      maxOccurs="unbounded" />
   </xsd:complexType>

   <xsd:complexType name="itemType" content="elementOnly">
      <xsd:choice minOccurs="0" maxOccurs="unbounded">
         <xsd:element name="item" type="itemType" />
      </xsd:choice>
      <xsd:attribute name="title" type="xsd:string" />
      <xsd:attribute name="ref" type="xsd:IDREF" />
   </xsd:complexType>
```

```
<xsd:complexType name="objectiveType" content="elementOnly">
   <xsd:element name="objective" type="objectiveType" minOccurs="0"
   maxOccurs="unbounded" />
   <xsd:attributeGroup ref="commonAttributes" />
</xsd:complexType>

<xsd:complexType name="textType" base="xsd:string">
   <xsd:attributeGroup ref="contentAttributes" />
</xsd:complexType>

<xsd:complexType name="learningObjectType" content="empty">
   <xsd:attributeGroup ref="contentAttributes" />
   <xsd:attribute name="href" type="xsd:uriReference" use="required" />
</xsd:complexType>

<xsd:complexType name="questionnaireType" content="elementOnly">
   <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="scoreband" type="scorebandType" />
   </xsd:choice>
   <xsd:attributeGroup ref="contentAttributes" />
   <xsd:attribute name="type" use="default" value="test">
      <xsd:simpleType base="xsd:string">
         <xsd:enumeration value="test" />
         <xsd:enumeration value="survey" />
         <xsd:enumeration value="assessment" />
      </xsd:simpleType>
   </xsd:attribute>
   <xsd:attribute name="passingScore" type="xsd:decimal" />
   <xsd:attribute name="resumable" type="xsd:boolean" />
   <xsd:attribute name="displayType" use="default" value="separate">
      <xsd:simpleType base="xsd:string">
         <xsd:enumeration value="together" />
         <xsd:enumeration value="separate" />
      </xsd:simpleType>
   </xsd:attribute>
   <xsd:attribute name="randomize" type="xsd:positiveInteger" />
   <xsd:attribute name="warningTime" type="xsd:positiveInteger" />
</xsd:complexType>
```

```
<xsd:complexType name="scorebandType" content="elementOnly">
   <xsd:choice minOccurs="0" maxOccurs="1">
      <xsd:element name="text" type="textType" />
   </xsd:choice>
   <xsd:attribute name="id" type="xsd:ID" use="required" />
   <xsd:attribute name="lowerBound" type="xsd:nonNegativeInteger"
   use="required" />
   <xsd:attribute name="upperBound" type="xsd:positiveInteger"
   use="required" />
</xsd:complexType>

<xsd:complexType name="multipleChoiceType" content="elementOnly">
   <xsd:sequence>
      <xsd:element name="question" type="questionType" />
      <xsd:element name="choice" type="choiceType" minOccurs="1"
      maxOccurs="unbounded" />
     <xsd:element name="hints" type="hintsType" minOccurs="0" maxOccurs="1"/>
   </xsd:sequence>
   <xsd:attributeGroup ref="questionAttributes" />
</xsd:complexType>

<xsd:complexType name="questionType" content="elementOnly">
   <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="text" type="textType" />
      <xsd:element name="learningObject" type="learningObjectType" />
   </xsd:choice>
</xsd:complexType>

<xsd:complexType name="choiceType" base="xsd:string">
   <xsd:attribute name="xml:lang" type="xsd:language" />
   <xsd:attribute name="answer" type="xsd:boolean" use="default"
   value="false"/>
</xsd:complexType>

<xsd:complexType name="hintsType" content="elementOnly">
   <xsd:sequence>
      <xsd:element name="hint" type="hintType" minOccurs="1"
      maxOccurs="unbounded" />
   </xsd:sequence>
   <xsd:attribute name="type" use="default" value="multilevel">
      <xsd:simpleType base="xsd:string">
         <xsd:enumeration value="incremental" />
         <xsd:enumeration value="multilevel" />
      </xsd:simpleType>
   </xsd:attribute>
</xsd:complexType>
```

```
   <xsd:complexType name="hintType" base="xsd:string">
      <xsd:attribute name="xml:lang" type="xsd:language" />
   </xsd:complexType>
   <xsd:complexType name="trueFalseType" content="elementOnly">
      <xsd:element name="question" type="questionType" />
      <xsd:element name="hints" type="hintsType" minOccurs="0" maxOccurs="1" />
      <xsd:attributeGroup ref="questionAttributes" />
      <xsd:attribute name="answer" type="xsd:boolean" use="required" />
   </xsd:complexType>

   <xsd:complexType name="essayType" content="elementOnly">
      <xsd:element name="question" type="questionType" />
      <xsd:element name="hints" type="hintsType" minOccurs="0" maxOccurs="1" />
      <xsd:attributeGroup ref="questionAttributes" />
   </xsd:complexType>

   <xsd:attributeGroup name="commonAttributes">
      <xsd:attribute name="id" type="xsd:ID" use="required" />
      <xsd:attribute name="name" type="xsd:string" />
      <xsd:attribute name="description" type="xsd:string" />
      <xsd:attribute name="metadataRef" type="xsd:uriReference" />
      <xsd:attribute name="xml:lang" type="xsd:language" />
   </xsd:attributeGroup>

   <xsd:attributeGroup name="contentAttributes">
      <xsd:attributeGroup ref="commonAttributes" />
      <xsd:attribute name="objectiveRef" type="xsd:IDREFS" />
      <xsd:attribute name="timeAllowed" type="xsd:positiveInteger" />
      <xsd:attribute name="accessCount" type="xsd:positiveInteger" />
      <xsd:attribute name="isVisible" type="xsd:boolean" />
      <xsd:attribute name="startDate" type="xsd:date" />
      <xsd:attribute name="endDate" type="xsd:date" />
   </xsd:attributeGroup>

   <xsd:attributeGroup name="questionAttributes">
      <xsd:attributeGroup ref="contentAttributes" />
      <xsd:attribute name="points" type="xsd:positiveInteger" />
      <xsd:attribute name="questionnaire" type="xsd:IDREFS" />
      <xsd:attribute name="title" type="xsd:string" />
   </xsd:attributeGroup>

</xsd:schema>
```

Competency Format DTD

```
<!-- Competency Format DTD
Daniel Lipkin
v1.0-->

<!-- A competencyLibrary defines one or more competencies.
A competencyLibrary contains an optional set of group definition, an optional
shared set of competency levels, and one or more competency definitions.
metadataRef is an optional URI to a Catalog Format document containing metadata
for this competency library. -->
<!ELEMENT competencyLibrary (group*, sharedLevels?, competency+)>
<!ATTLIST competencyLibrary metadataRef CDATA #IMPLIED
                  xml:lang NMTOKEN #IMPLIED>

<!-- Competencies can be grouped, and groups can be nested. -->
<!ELEMENT group (name, description?, group*)>
<!ATTLIST group id ID #REQUIRED>

<!-- SharedLevels defines one or more competency levels available for all
competencies.  -->
<!ELEMENT sharedLevels (level+)>

<!-- A competency has a name and an optional description.
By default, it inherits the competencylevels defined by the shared levels.
It may also choose to override these defaults by defining one or more levels
specific to this competency.
Note that if there is no sharedLevel definition, each competency must explicitly
define at least one level.
A competency can also be associated with a group. -->
<!ELEMENT competency (name, description?, level*)>
<!ATTLIST competency id ID #REQUIRED
                  groupRef IDREF #IMPLIED>
```

```
<!-- A competency level has an optional name, an optional description, and one or
more optional evidence descriptions. It has an optional numeric value as a
property. -->
<!ELEMENT level (name?, description?, evidence*)>
<!ATTLIST level id ID #REQUIRED>
<!ATTLIST level value CDATA #IMPLIED>

<!ELEMENT name (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT evidence (#PCDATA)>
```

Competency Format XML Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema">

   <xsd:annotation>
      <xsd:documentation>Competency Format Schema Daniel Lipkin v1.0
      </xsd:documentation>
   </xsd:annotation>

   <xsd:element name="competencyLibrary" type="competencyLibraryType">
      <xsd:key name="group">
         <xsd:selector>group</xsd:selector>
         <xsd:field>@id</xsd:field>
      </xsd:key>
      <xsd:key name="competency">
         <xsd:selector>competency</xsd:selector>
         <xsd:field>@id</xsd:field>
      </xsd:key>
      <xsd:key name="level">
         <xsd:selector>//level</xsd:selector>
         <xsd:field>@id</xsd:field>
      </xsd:key>
      <xsd:keyref name="groupRef" refer="group">
         <xsd:selector>competency</xsd:selector>
         <xsd:field>@groupRef</xsd:field>
      </xsd:keyref>
   </xsd:element>

   <xsd:complexType name="competencyLibraryType">
      <xsd:element ref="group" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="sharedLevels" minOccurs="0" maxOccurs="1">
         <xsd:complexType>
            <xsd:element ref="level" minOccurs="0" maxOccurs="1" />
         </xsd:complexType>
      </xsd:element>
      <xsd:element ref="competency" minOccurs="1" />
      <xsd:attribute name="metadataRef" type="uriReference" />
      <xsd:attribute name="xml:lang" type="xsd:language" />
```

```
      <xsd:attribute name="xmlns:xsi" type="xsd:uriReference" use="default"
      value="http://www.w3.org/1999/XMLSchema-instance" />
      <xsd:attribute name="xsi:schemaLocation" type="xsd:string" />
   </xsd:complexType>

   <xsd:complexType name="group">
      <xsd:element name="name" type="xsd:string" />
      <xsd:element name="description" type="xsd:string" minOccurs="0"
      maxOccurs="1" />
      <xsd:element name="group" type="group" minOccurs="0"
      maxOccurs="unbounded"/>
      <xsd:attribute name="id" type="xsd:ID" />
   </xsd:complexType>

   <xsd:complexType name="competency">
      <xsd:element name="name" type="xsd:string" />
      <xsd:element name="description" type="xsd:string" minOccurs="0"
      maxOccurs="1" />
      <xsd:element ref="level" minOccurs="0" maxOccurs="unbounded" />
      <xsd:attribute name="id" type="xsd:ID" />
      <xsd:attribute name="groupRef" type="xsd:IDREF" />
   </xsd:complexType>

   <xsd:complexType name="level">
      <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1" />
      <xsd:element name="description" type="xsd:string" minOccurs="0"
      maxOccurs="1" />
      <xsd:element name="evidence" type="xsd:string" minOccurs="0"
      maxOccurs="unbounded" />
      <xsd:attribute name="id" type="xsd:ID" />
      <xsd:attribute name="value" type="xsd:positiveInteger" />
   </xsd:complexType>
</xsd:schema>
```

Competency RDF Schema

```
<?xml version="1.0" ?>
   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
      xmlns:dc="http://purl.org/dc/elements/1.0">

   <rdf:Description about="">
      <dc:Title>Competency Metadata</dc:Title>
      </rdf:Description>

   <rdf:Description ID="competency">
      <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Competency</rdfs:label>
      <rdfs:comment>A competency state possessed by a competency holder,
      competency provider, or competency model.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
   </rdf:Description>

   <rdf:Description ID="attained">
      <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Attained</rdfs:label>
      <rdfs:comment>Date a competency was attained. Used ISO 8601 format.
      </rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
   </rdf:Description>

   <rdf:Description ID="provider">
      <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Provider</rdfs:label>
      <rdfs:comment>The entity providing a help competency level.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
   </rdf:Description>
```

```
   <rdf:Description ID="details">
      <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Details</rdfs:label>
      <rdfs:comment>Additional information about how the competency was attained.
      </rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
   </rdf:Description>
</rdf:RDF>
```

Certification Format DTD

```
<!-- Certification Format DTD
Daniel Lipkin
v1.0-->

<!-- Common date attributes. StartDate and stopDate defines the period of time in
which a specific track or the overall certification are available.
They are represented using ISO8601 format.  -->
<!ENTITY % date-attr "startDate CDATA #IMPLIED
                  endDate CDATA #IMPLIED">

<!-- A certificationLibrary defines one or more certifications. -->
<!ELEMENT certificationLibrary (certification+)>

<!-- A certification defines one or more paths to be followed to complete the
certification. It contains an optional summary and set of objectives.  -->
<!ELEMENT certification (summary?, objective*, path+)>

<!ATTLIST certification id ID #REQUIRED
                        name CDATA #IMPLIED
                        description CDATA #IMPLIED>

<!ATTLIST certification %date-attr;>

<!-- MetadataRef is a optional URI to a Catalog Format document containing
metadata for this certification library. -->
<!ATTLIST certification metadataRef CDATA #IMPLIED>

<!-- The time limit in days for acquiring this certification -->
<!ATTLIST certification acquireWithin CDATA #IMPLIED>

<!-- The time period in days for which a certification is valid -->
<!ATTLIST certification expiresIn CDATA #IMPLIED>

<!-- An in-depth description of the certification. -->
<!ELEMENT summary (#PCDATA)>
<!-- An objective for this certification:  statement of skills, knowledge, and
attitudes to be acquired. -->
```

```
<!ELEMENT objective (#PCDATA)>

<!-- A path is a disjointed sequence of modules. Each module represents one step
along the specific path. -->
<!ELEMENT path (module)+>
<!ATTLIST path name CDATA #IMPLIED>

<!-- Paths can either be for new certification or re-certification. -->
<!ATTLIST path type (new | refresh) "new">

<!-- Each path has a startDate and endDate during which that track is available.
-->
<!ATTLIST path %date-attr;>

<!-- Each module contains one or more options for fulfilling the requirement.  -
->
<!ELEMENT module (option)+>
<!ATTLIST module name CDATA #IMPLIED
                 description CDATA #IMPLIED>

<!-- A module may be optional or required. -->
<!ATTLIST module type (optional | required) "required">

<!-- A module may specify a minimum number, in which case that number of options
must be taken to fulfill the requirement. The default value is 1, that is, any
one option fulfills the requirement. -->
<!ATTLIST module minimum CDATA #IMPLIED>

<!-- Each option can be a URI to the resource fulfilling the certification
requirement. This will typically be a course specified within a Catalog Format
document, although it may also refer to a specific competency or even another
certification. An option may also be a text-based description of an action to
take.  -->
<!ELEMENT option (#PCDATA)>
<!ATTLIST option href CDATA #IMPLIED>
```

Certification Format XML Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema">

   <xsd:annotation>
      <xsd:documentation>Certification Format Schema Daniel Lipkin v1.0
      </xsd:documentation>
   </xsd:annotation>

   <xsd:element name="certificationLibrary">
      <xsd:complexType content="elementOnly">
        <xsd:element name="certification" type="certificationType" minOccurs="1"
         maxOccurs="unbounded" />
         <xsd:attribute name="xmlns:xsi" type="xsd:uriReference" use="default"
         value="http://www.w3.org/1999/XMLSchema-instance" />
         <xsd:attribute name="xsi:schemaLocation" type="xsd:string" />
      </xsd:complexType>
   </xsd:element>
      <xsd:complexType name="certificationType" content="elementOnly">
         <xsd:element name="summary" type="xsd:string" minOccurs="0"
         maxOccurs="1" />
         <xsd:element name="objective" type="xsd:string" minOccurs="0"
         maxOccurs="unbounded" />
         <xsd:element name="path" type="pathType" minOccurs="1"
         maxOccurs="unbounded" />
         <xsd:attribute name="id" type="xsd:ID" use="required" />
         <xsd:attribute name="name" type="xsd:string" />
         <xsd:attribute name="description" type="xsd:string" />
         <xsd:attribute name="startDate" type="xsd:date" />
         <xsd:attribute name="endDate" type="xsd:date" />
         <xsd:attribute name="metadataRef" type="xsd:uriReference" />
         <xsd:attribute name="acquireWithin" type="xsd:integer" />
         <xsd:attribute name="expiresIn" type="xsd:integer" />
      </xsd:complexType>
      <xsd:complexType name="pathType" content="elementOnly">
         <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element name="module" type="moduleType" />
         </xsd:choice>
```

```
      <xsd:attribute name="name" type="xsd:string" />
      <xsd:attribute name="type" use="default" value="new">
         <xsd:simpleType base="xsd:string">
            <xsd:enumeration value="new" />
            <xsd:enumeration value="refresh" />
         </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="startDate" type="xsd:date" />
      <xsd:attribute name="endDate" type="xsd:date" />
   </xsd:complexType>

   <xsd:complexType name="moduleType" content="elementOnly">
      <xsd:choice minOccurs="1" maxOccurs="unbounded">
         <xsd:element name="option" type="optionType" />
      </xsd:choice>
      <xsd:attribute name="name" type="xsd:string" />
      <xsd:attribute name="description" type="xsd:string" />
      <xsd:attribute name="type" use="default" value="required">
         <xsd:simpleType base="xsd:string">
            <xsd:enumeration value="optional" />
            <xsd:enumeration value="required" />
         </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name="minimum" type="xsd:positiveInteger" />
   </xsd:complexType>

   <xsd:complexType name="optionType" base="xsd:string">
      <xsd:attribute name="href" type="xsd:uriReference" />
   </xsd:complexType>
</xsd:schema>
```

Certification Results Format DTD

```
<!-- Certification Results DTD
Daniel Lipkin
v1.0 -->

<!-- This DTD captures the results of a certification agency. Notice that it is
extremely simple and limited to capturing the data returned by certification
agencies such as Sylvan.  -->

<!ELEMENT results (result)+>

<!ELEMENT result (learner, certification, date, score)>

<!-- learner captures the name of the individual applying for the certification.
The exact format of this value is provider-specific. -->
<!ELEMENT learner (#PCDATA)>

<!-- test identifies the test taken to achieve the certification. The exact
format of this value is provider-specific. One typical format might be a URI to a
certification defined in a Certification Format document. -->
<!ELEMENT certification (#PCDATA)>

<!-- date specifies the data in ISO8601 format the certification was assessed. -
->
<!ELEMENT date (#PCDATA)>

<!-- result specifies whether the certification was passed or failed. -->
<!ELEMENT score EMPTY>
<!ATTLIST score value (pass | fail) #REQUIRED>
```

Offerings RDF Schema

```
<?xml version="1.0" ?>
   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
      xmlns:dc="http://purl.org/dc/elements/1.1/">

   <rdf:Description about="">
      <dc:title>Catalog Format - Classes</dc:title>
      <dc:description>Class hierarchy used by Catalog Format.
      </dc:description>
   </rdf:Description>

<!-- Resource class hierarchy defined by Catalog Format -->

   <rdf:Description ID="offering">
      <rdf:type resource=
         "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
      <rdfs:subClassOf rdf:resource=
         "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource"/>
      <rdfs:label>Offering</rdfs:label>
      <rdfs:comment>A resource available for purchase.</rdfs:comment>
   </rdf:Description>

   <rdf:Description ID="competency">
      <rdf:type resource=
         "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
      <rdfs:subClassOf rdf:resource=
         "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource"/>
      <rdfs:label>Competency</rdfs:label>
      <rdfs:comment>A competency library.</rdfs:comment>
   </rdf:Description>
```

```
<rdf:Description ID="certification">
   <rdf:type resource=
      "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
   <rdfs:subClassOf rdf:resource=
      "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource"/>
   <rdfs:label>Certification</rdfs:label>
   <rdfs:comment>A certification track.</rdfs:comment>
</rdf:Description>

<rdf:Description ID="event">
   <rdf:type resource=
      "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
   <rdfs:subClassOf rdf:resource="#offering"/>
   <rdfs:label>Event</rdfs:label>
   <rdfs:comment>An offering that is time and location based.</rdfs:comment>
</rdf:Description>

<rdf:Description ID="inventory">
   <rdf:type resource=
      "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
   <rdfs:subClassOf rdf:resource="#offering"/>
   <rdfs:label>Inventory</rdfs:label>
   <rdfs:comment>
      An offering that is delivered out of inventory.</rdfs:comment>
</rdf:Description>

<rdf:Description ID="online">
   <rdf:type resource=
      "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
   <rdfs:subClassOf rdf:resource="#offering"/>
   <rdfs:label>Online</rdfs:label>
   <rdfs:comment>An on-line offering.</rdfs:comment>
</rdf:Description>

<rdf:Description ID="virtualclass">
   <rdf:type resource=
      "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
   <rdfs:subClassOf rdf:resource="#online"/>
   <rdfs:label>Virtual Class</rdfs:label>
   <rdfs:comment>An on-line offering that is time-based.</rdfs:comment>
</rdf:Description>
```

```
   <rdf:Description ID="package">
      <rdf:type resource=
         "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Offering"/>
      <rdfs:subClassOf rdf:resource="#offering"/>
      <rdfs:label>Package</rdfs:label>
      <rdfs:comment>A group of offerings.</rdfs:comment>
   </rdf:Description>

<!-- Properties defined by Catalog Format -->

   <rdf:Description ID="acronym">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Acronym</rdfs:label>
      <rdfs:comment>Acronym associated with a learning offering.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
      <rdfs:domain rdf:resource="#offering"/>
   </rdf:Description>

   <rdf:Description ID="contents">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Contents</rdfs:label>
      <rdfs:comment>Offerings contained in a package.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
      <rdfs:domain rdf:resource="#package"/>
   </rdf:Description>
</rdf:RDF>
```

IMS Metadata RDF Schema

```
<?xml version="1.0" ?>
   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
      xmlns:dc="http://purl.org/dc/elements/1.1/">

   <rdf:Description about="">
      <dc:Title>Catalog Format</dc:Title>
      <dc:Description>RDF mapping of IMS Core LOM.</dc:Description>
   </rdf:Description>

   <rdf:Description ID="catalogue">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Catalogue</rdfs:label>
      <rdfs:comment>Source of the CatalogEntry.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
   </rdf:Description>

   <rdf:Description ID="version">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Version</rdfs:label>
      <rdfs:comment>Edition of this resource.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
   </rdf:Description>

   <rdf:Description ID="contribute">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Contribute</rdfs:label>
      <rdfs:comment>Suppliments Dublin Core with Role and Date of contributions.
      Contains an IDREF to a DC Contribute element.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
   </rdf:Description>
```

```
<rdf:Description ID="role">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Role</rdfs:label>
   <rdfs:comment>Kind of contribution.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
   </rdf:Description>

<rdf:Description ID="date">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Date</rdfs:label>
   <rdfs:comment>Date of contribution in ISO 8601 format.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="metadataScheme">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
<rdfs:label>MetadataScheme</rdfs:label>
<rdfs:comment>Scheme used for the metadata</rdfs:comment>
<rdf:value>RDF/LOM-1.0</rdf:value>
<rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="location">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Location</rdfs:label>
   <rdfs:comment>Location of the resource, in order of preference.
   </rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="cost">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Cost</rdfs:label>
   <rdfs:comment>Boolean specifying whether the use of the resource requires
   payment.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>
```

```
<rdf:Description ID="copyright">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Copyright</rdfs:label>
   <rdfs:comment>Boolean specifying whether the resource has copyrights or
   other restrictions in its use.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="classification">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Classification</rdfs:label>
   <rdfs:comment>Describes various characteristics of the resource.
   </rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
   <rdfs:domain rdf:resource="#offering"/>
</rdf:Description>

<rdf:Description ID="purpose">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Purpose</rdfs:label>
   <rdfs:comment>Identifies a particular characteristic.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="description">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Description</rdfs:label>
   <rdfs:comment>Describes a particular characteristic.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="keywords">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Keywords</rdfs:label>
   <rdfs:comment>Descriptions of the resource within a particular
   classification.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>
```

```
<rdf:Description ID="status">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Status</rdfs:label>
   <rdfs:comment>Condition of the resource. One of: Draft, Final, Revised,
   Unavailable</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="interactivity">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Interactivity Type</rdfs:label>
   <rdfs:comment>Type of interactivity supported by this resource. One of:
   Active, Expositive, Mixed, Undefined</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
   <rdfs:domain rdf:resource="#offering"/>
</rdf:Description>

<rdf:Description ID="target">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Target</rdfs:label>
   <rdfs:comment>Learning Context; the typical kind of learners.
   </rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
   <rdfs:domain rdf:resource="#offering"/>
</rdf:Description>

<rdf:Description ID="difficulty">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Difficulty</rdfs:label>
   <rdfs:comment>Difficulty of the resource, ranging from 0-4.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
   <rdfs:domain rdf:resource="#offering"/>
</rdf:Description>

<rdf:Description ID="source">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Source</rdfs:label>
   <rdfs:comment>Source of a taxonomy classification.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>
```

```
   <rdf:Description ID="taxon">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Taxon</rdfs:label>
      <rdfs:comment>Entry in a taxonomic path.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
   </rdf:Description>

   <rdf:Description ID="learningTime">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Typical Learning Time</rdfs:label>
      <rdfs:comment>Approximate or typical time it takes to work with the
      resource, in ISO8601 format</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
      <rdfs:domain rdf:resource="#offering"/>
   </rdf:Description>

</rdf:RDF>
```

Scheduling RDF Schema

```
<?xml version="1.0" ?>
   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
      xmlns:dc="http://purl.org/dc/elements/1.1/">

   <rdf:Description about="">
      <dc:title>Catalog Format- Scheduling elements</dc:title>
     <dc:description>Properties for describing scheduling attributes of catalog
offerings.
      </dc:description>
   </rdf:Description>

   <rdf:Description ID="location">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Location</rdfs:label>
      <rdfs:comment>Physical location of a training event.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
      <rdfs:domain rdf:resource="http://www.saba.com/RDF/offering10.rdf#event"/>
   </rdf:Description>

   <rdf:Description ID="startDate">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Start Date</rdfs:label>
      <rdfs:comment>Start date of a training event.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
     <rdfs:domain rdf:resource="http://www.saba.com/RDF/offering10.rdf#event"/>
      <rdfs:domain rdf:resource=
         "http://www.saba.com/RDF/offering/1.0#virtualclass"/>
   </rdf:Description>
```

```
   <rdf:Description ID="template">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Template</rdfs:label>
      <rdfs:comment>Template for days and hours of a training event.
         </rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
     <rdfs:domain rdf:resource="http://www.saba.com/RDF/offering10.rdf#event"/>
      <rdfs:domain rdf:resource=
         "http://www.saba.com/RDF/offering/1.0#virtualclass"/>
   </rdf:Description>

   <rdf:Description ID="minimumLearners">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Minimum Number of Learners</rdfs:label>
      <rdfs:comment>The minimum number of learners that can be enrolled in a
         class.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
      <rdfs:domain rdf:resource=
         "http://www.saba.com/RDF/offering10.rdf#offering"/>
      <rdfs:range rdf:resource=
         "http://www.w3.org/1999/XMLSchema-datatypes/nonNegativeInteger"/>
   </rdf:Description>

   <rdf:Description ID="maximumLearners">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Maximum Number of Learners</rdfs:label>
      <rdfs:comment>The maximum number of learners that can be enrolled in a
         class.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
      <rdfs:domain rdf:resource=
         "http://www.saba.com/RDF/offering10.rdf#offering"/>
      <rdfs:range rdf:resource=
         "http://www.w3.org/1999/XMLSchema-datatypes/nonNegativeInteger"/>
   </rdf:Description>
</rdf:RDF>
```

Price List RDF Schema

```
<?xml version="1.0" ?>
   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
      xmlns:dc="http://purl.org/dc/elements/1.1/">
   <rdf:Description about="">
      <dc:title>Price List format for use with Catalog Format
      </dc:title>
      <dc:description>Properties for describing e-commerce attributes of catalog
         offerings.</dc:description>
   </rdf:Description>

   <rdf:Description ID="pricelist">
      <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
      <rdfs:subClassOf rdf:resource=
         "http://www.w3.org/2000/01/rdf-schema#Resource"/>
      <rdfs:label>Price List</rdfs:label>
      <rdfs:comment>A price list structure.</rdfs:comment>
   </rdf:Description>

   <rdf:Description ID="price">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Price</rdfs:label>
      <rdfs:comment>Unit price for this resource.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
      <rdfs:domain rdf:resource=
         "http://www.saba.com/RDF/offering10.rdf#offering"/>
      <rdfs:domain rdf:resource="#pricelist"/>
   </rdf:Description>

   <rdf:Description ID="startDate">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Start Date</rdfs:label>
      <rdfs:comment>First date a price applies.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
   </rdf:Description>
```

```
<rdf:Description ID="endDate">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>End Date</rdfs:label>
   <rdfs:comment>Last date a price applies.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="currency">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Currency</rdfs:label>
   <rdfs:comment>Currency used by this price.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="quantityRange">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Quantity Range</rdfs:label>
   <rdfs:comment>Quantities to which this price applies.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="min">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Minimum</rdfs:label>
   <rdfs:comment>Minimum quantity to which this price applies.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="max">
   <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Maximum</rdfs:label>
   <rdfs:comment>Maximum quantity to which this price applies.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>
```

```
   <rdf:Description ID="method">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Method</rdfs:label>
      <rdfs:comment>Selling method. One of: register, ship, rent, checkout,
         download</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
   </rdf:Description>

   <rdf:Description ID="target">
      <rdf:type rdf:resource=
         "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Target</rdfs:label>
      <rdfs:comment>Offering to which this price applies.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
      <rdfs:range rdf:resource=
      "http://www.saba.com/RDF/offering10.rdf#offering"/>
   </rdf:Description>
</rdf:RDF>
```

Profile Format RDF Schema

```
<?xml version="1.0" ?>
   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:dc="http://purl.org/dc/elements/1.1/">

   <rdf:Description about="">
      <dc:title>Profile Format</dc:title>
      <dc:description>Properties for describing a learner's
      profile.<dc:description>
   </rdf:Description>

   <rdf:Description ID="learner">
      <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
      <rdfs:subClassOf rdf:resource=
      "http://www.w3.org/2000/01/rdf-schema#Resource"/>
      <rdfs:label>Learner</rdfs:label>
      <rdfs:comment>A learner being profiled.</rdfs:comment>
   </rdf:Description>

   <rdf:Description ID="learning">
      <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Learning</rdfs:label>
      <rdfs:comment>Current and past learning.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
      <rdfs:domain rdf:resource=
      "http://www.saba.com/RDF/profile10.rdf#learner"/>
   </rdf:Description>
```

```
<rdf:Description ID="status">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Status</rdfs:label>
   <rdfs:comment>Status of a learning offering. One of: Complete, Incomplete,
   Dropped, No Show, Cancelled, Enrolled, In Progress, Waitlisted, Pending
   Approval. Also, status of a goal. Open vocabulary.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="result">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Result</rdfs:label>
   <rdfs:comment>Result of taking a learning offering.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="date">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Date</rdfs:label>
   <rdfs:comment>Date for a result or observation.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
   <rdfs:range rdf:resource=
   "http://www.w3.org/1999/XMLSchema-datatypes/date"/>
</rdf:Description>

<rdf:Description ID="score">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Score</rdfs:label>
   <rdfs:comment>Score within a result.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="goal">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Goal</rdfs:label>
   <rdfs:comment>A learner's goal.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
   <rdfs:domain rdf:resource=
   "http://www.saba.com/RDF/profile10.rdf#learner"/>
</rdf:Description>
```

```
<rdf:Description ID="description">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Description</rdfs:label>
   <rdfs:comment>A goal description.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="startDate">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Start Date</rdfs:label>
   <rdfs:comment>Start date for a goal.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
   <rdfs:range rdf:resource=
   "http://www.w3.org/1999/XMLSchema-datatypes/date"/>
</rdf:Description>

<rdf:Description ID="endDate">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>End Date</rdfs:label>
   <rdfs:comment>Target end date for a goal.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
   <rdfs:range rdf:resource=
   "http://www.w3.org/1999/XMLSchema-datatypes/date"/>
</rdf:Description>

<rdf:Description ID="priority">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Priority</rdfs:label>
   <rdfs:comment>Priority of a goal.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
   <rdfs:range rdf:resource=
   "http://www.w3.org/1999/XMLSchema-datatypes/integer"/>
</rdf:Description>
```

```
<rdf:Description ID="parent">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Parent</rdfs:label>
   <rdfs:comment>Parent goal.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
   <rdfs:range rdf:resource=
   "http://www.w3.org/1999/XMLSchema-datatypes/IDREF"/>
</rdf:Description>

<rdf:Description ID="observation">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Observation</rdfs:label>
   <rdfs:comment>Observation on progress towards a goal.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
   <rdfs:domain rdf:resource=
   "http://www.saba.com/RDF/profile10.rdf#learner"/>
</rdf:Description>

<rdf:Description ID="comment">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Comment</rdfs:label>
   <rdfs:comment>An observation comment.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="observer">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Observer</rdfs:label>
   <rdfs:comment>Entity making an observation.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
</rdf:Description>

<rdf:Description ID="completion">
   <rdf:type rdf:resource=
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:label>Completion</rdfs:label>
   <rdfs:comment>Percentage of goal completed.</rdfs:comment>
   <rdfs:isDefinedBy rdf:resource=""/>
   <rdfs:range rdf:resource=
   "http://www.w3.org/1999/XMLSchema-datatypes/nonNegativeInteger"/>
</rdf:Description>
```

```
   <rdf:Description ID="language">
      <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Language</rdfs:label>
      <rdfs:comment>Learner's preferred language.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
      <rdfs:domain rdf:resource=
      "http://www.saba.com/RDF/profile10.rdf#learner"/>
   </rdf:Description>

   <rdf:Description ID="country">
      <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Country</rdfs:label>
      <rdfs:comment>Learner's home country.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
      <rdfs:domain rdf:resource=
      "http://www.saba.com/RDF/profile10.rdf#learner"/>
   </rdf:Description>

   <rdf:Description ID="identifier">
      <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Identfier</rdfs:label>
      <rdfs:comment>A unique identifier.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
      <rdfs:domain rdf:resource=
      "http://www.saba.com/RDF/profile10.rdf#learner"/>
   </rdf:Description>

   <rdf:Description ID="source">
      <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
      <rdfs:label>Source</rdfs:label>
      <rdfs:comment>Source of an identifier.</rdfs:comment>
      <rdfs:isDefinedBy rdf:resource=""/>
   </rdf:Description>
</rdf:RDF>
```

Profile Format XML Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:profile="http://www.saba.com/RDF/profile10.rdf"
   xmlns:competency="http://www.saba.com/RDF/competency10.rdf"
   xmlns:certification="http://www.saba.com/RDF/certification10.rdf"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:vCard="http://imc.org/vCard/3.0#"
   targetNamespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   elementFormDefault="qualified" attributeFormDefault="unqualified">

<!--  import all the dependent schemas -->
   <xsd:import namespace="http://imc.org/vCard/3.0#"
      schemaLocation="profile10_vCard.xsd" />
   <xsd:import namespace="http://purl.org/dc/elements/1.1/"
      schemaLocation="profile10_dc.xsd" />
   <xsd:import namespace="http://www.saba.com/RDF/profile10.rdf"
      schemaLocation="profile10_profile.xsd" />
   <xsd:import namespace="http://www.saba.com/RDF/competency10.rdf"
      schemaLocation="profile10_competency.xsd" />
   <xsd:import namespace="http://www.saba.com/RDF/certification10.rdf"
      schemaLocation="profile10_certification.xsd"/>
   <xsd:element name="RDF">
      <xsd:complexType>
        <xsd:element name="Description" type="rdf:DescriptionType" minOccurs="0"
         maxOccurs="unbounded" />
      </xsd:complexType>
   </xsd:element>

   <xsd:complexType name="DescriptionType">
<!--  each description has a "type" element with a fixed value -->
      <xsd:element name="type">
         <xsd:complexType content="empty">
            <xsd:attribute name="resource" type="xsd:uriReference" use="fixed"
            value="http://www.saba.com/RDF/profile10.rdf#learner" />
         </xsd:complexType>
      </xsd:element>
```

```
<!--  include elements defined in various dependent schemas -->
      <xsd:group ref="vCard:personal" />
      <xsd:group ref="profile:personal" />
      <xsd:group ref="profile:learning" />
      <xsd:group ref="profile:goals" />
      <xsd:group ref="profile:observations" />
      <xsd:group ref="competency:competencies" />
      <xsd:group ref="certification:certifications" />
      <xsd:group ref="profile:preferences" />
      <xsd:group ref="dc:dublinCore" />
      <xsd:attribute name="about" type="xsd:uriReference" use="required" />
   </xsd:complexType>
</xsd:schema>
```

vCard XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema" xmlns:rdf="http://
www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:vCard="http://imc.org/vCard/3.0#"
targetNamespace="http://imc.org/vCard/3.0#" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xsd:import namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
schemaLocation="profile10_rdf.xsd"/>
   <!-- PERSONAL properties  -->
   <!-- group to represent all RDF properties for personal information (from the
vCard namespace).  This definition is based on the RDF mappings of vCard found at
http://www.dstc.edu.au/Research/Projects/rdf/draft-iannella-vcard-rdf-00.txt -->
   <xsd:group name="personal">
       <xsd:all>
           <xsd:element name="N">
               <xsd:complexType>
                   <xsd:all>
                       <xsd:element name="Family" type="xsd:string"/>
                       <xsd:element name="Given" type="xsd:string"/>
                       <xsd:element name="Other" type="xsd:string"/>
                       <xsd:element name="Prefix" type="xsd:string"/>
                       <xsd:element name="Suffix" type="xsd:string"/>
                   </xsd:all>
                   <xsd:attribute ref="rdf:parseType"/>
               </xsd:complexType>
           </xsd:element>
           <xsd:element name="FN" type="xsd:string"/>
           <xsd:element name="BDAY" type="xsd:date"/>
           <xsd:element name="ADR">
               <!-- represent the vCard:ADR property, which takes the form:
                   <vCard:ADR rdf:parseType="Resource">
                     <vCard:Street>2400 Bridge Parkway</vCard:Street>
                     <vCard:Locality>Redwood Shores</vCard:Locality>
                     <vCard:Region>CA</vCard:Region>
                     <vCard:Pcode>94065</vCard:Pcode>
                    <vCard:TYPE rdf:resource ="http://imc.org/vCard/3.0#work" />
                  </vCard:ADR> -->
               <xsd:complexType>
                   <xsd:all>
                       <xsd:element name="Pobox" type="xsd:string"/>
                       <xsd:element name="Extadd" type="xsd:string"/>
                       <xsd:element name="Street" type="xsd:string"/>
                       <xsd:element name="Locality" type="xsd:string"/>
                       <xsd:element name="Region" type="xsd:string"/>
                       <xsd:element name="Pcode" type="xsd:string"/>
```

```
                        <xsd:element name="Country" type="xsd:string"/>
                        <xsd:element name="TYPE" ref="rdf:ADRType"/>
                    </xsd:all>
                    <xsd:attribute ref="rdf:parseType"/>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="TEL">
                <!-- represent the vCard:TEL property, which takes the form:
                    <vCard:TEL rdf:parseType="Resource">
                        <rdf:value> 650-581-4444 </rdf:value>
                        <vCard:TYPE rdf:resource="http://imc.org/vCard/
3.0#work"/>
                    </vCard:TEL> -->
                <xsd:complexType>
                    <xsd:element ref="rdf:value"/>
                    <xsd:element name="TYPE" ref="rdf:TELType"/>
                    <xsd:attribute ref="rdf:parseType"/>
                </xsd:complexType>
            </xsd:element>
            <!-- support a single email property with a type of "internet" -->
            <xsd:element name="EMAIL" type="xsd:string"/>
            <xsd:element name="TZ" type="xsd:string"/>
            <xsd:element name="TITLE" type="xsd:string"/>
            <xsd:group ref="vCard:roleGroup"/>
            <xsd:element name="ORG">
                <!-- represent the vCard:ORG property, which takes the form:
                        <vCard:ORG parseType="Resource">
                         <vCard:Orgname> QQQ.COM Pty Ltd </vCard:Orgname>
                          <vCard:Orgunit>
                            <seq>
                              <li> Commercialisation Directorate </li>
                              <li> Engineering Office </li>
                              <li> Java Unit </li>
                            </seq>
                         </vCard:Orgunit>
                        </vCard:ORG> -->
                <xsd:complexType>
                    <xsd:element name="Orgname" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
                    <xsd:group ref="vCard:orgunitGroup"/>
                    <xsd:attribute ref="rdf:parseType"/>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="KEY">
                <xsd:complexType content="empty">
                    <xsd:attribute ref="rdf:resource"/>
```

```
                </xsd:complexType>
            </xsd:element>
        </xsd:all>
    </xsd:group>
    <xsd:group name="roleGroup">
        <xsd:choice>
            <xsd:element name="ROLE" type="xsd:string"/>
            <xsd:element name="ROLE" ref="rdf:Bag"/>
        </xsd:choice>
    </xsd:group>
    <!--a vCard Orgunit property can either be a single value or an RDF Seq -->
    <xsd:group name="orgunitGroup">
        <xsd:choice>
            <xsd:element name="Orgunit" type="xsd:string"/>
            <xsd:element name="Orgunit" ref="rdf:Seq"/>
        </xsd:choice>
    </xsd:group>
</xsd:schema>
```

Dublin Core XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema" xmlns:dc="http://
purl.org/dc/elements/1.1/" targetNamespace="http://purl.org/dc/elements/1.1/"
elementFormDefault="qualified" attributeFormDefault="unqualified">
   <!-- DUBLIN CORE properties; Profile Format supports a subset of the full DC
-->
   <xsd:group name="dublinCore">
       <xsd:all>
           <xsd:element name="description" type="xsd:string"/>
           <xsd:element name="publisher" type="xsd:string"/>
           <xsd:element name="date" type="xsd:date"/>
           <xsd:element name="language" type="xsd:language"/>
           <xsd:element name="coverage" type="xsd:string"/>
       </xsd:all>
   </xsd:group>
</xsd:schema>
```

Profile XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema" xmlns:profile="http://
www.saba.com/RDF/profile10.rdf" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#" targetNamespace="http://www.saba.com/RDF/profile10.rdf"
elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xsd:include schemaLocation="profile10_profile2.xsd"/>
   <xsd:import namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
schemaLocation="profile10_rdf.xsd"/>
   <!-- PERSONAL properties that are not captured by vCard.  The only property
currently defined is the identifier property, which is intended to represent
values such as SSN.  -->
   <xsd:group name="personal">
       <xsd:all>
           <xsd:element name="identifier">
               <xsd:complexType>
                   <xsd:element ref="rdf:value"/>
                   <xsd:element name="source" type="xsd:string"/>
                   <xsd:attribute ref="rdf:parseType"/>
               </xsd:complexType>
           </xsd:element>
       </xsd:all>
   </xsd:group>

   <!-- LEARNING properties  -->
   <!-- group to represent learning information, including current learning and
learning history -->
   <xsd:group name="learning">
       <xsd:sequence maxOccurs="unbounded">
           <xsd:group ref="profile:learningGroup"/>
       </xsd:sequence>
   </xsd:group>
   <!-- a learning property is either a simple URI reference, a complex
structure, or a Bag -->
   <xsd:group name="learningGroup">
       <xsd:choice>
           <xsd:element name="learning" ref="rdf:value"/>
           <xsd:element name="learning" type="rdf:complexLearning"/>
           <xsd:element name="learning" type="rdf:learningContainer"/>
       </xsd:choice>
   </xsd:group>
   <!-- GOAL properties -->
   <xsd:group name="goals">
       <xsd:sequence maxOccurs="unbounded">
           <xsd:group ref="profile:goalGroup"/>
```

```
        </xsd:sequence>
    </xsd:group>
    <!-- a goal property is either a goal name with an optional ID,  a structured
goal, or a Bag or Seq -->
    <xsd:group name="goalGroup">
        <xsd:choice>
            <xsd:element name="goal" type="profile:simpleGoal"/>
            <xsd:element name="goal" type="rdf:complexGoal"/>
            <xsd:element name="goal" type="rdf:goalContainer"/>
        </xsd:choice>
    </xsd:group>
    <xsd:complexType name="simpleGoal" base="xsd:string" derivedBy="extension">
        <xsd:attribute ref="rdf:id"/>
        <xsd:attribute ref="rdf:resource"/>
    </xsd:complexType>
    <!-- OBSERVATION properties  -->
    <!-- observation can either be a single property or a Bag of observations -->
    <xsd:group name="observations">
        <xsd:choice>
            <xsd:element name="observation" ref="rdf:observationType"/>
            <xsd:element name="observation" ref="rdf:observationBag"/>
        </xsd:choice>
    </xsd:group>
    <!-- USER PREFERENCE properties  -->
    <xsd:group name="preferences">
        <xsd:all>
            <xsd:element name="language" type="xsd:language"/>
            <xsd:element name="country" type="xsd:string"/>
        </xsd:all>
    </xsd:group>
</xsd:schema>
```

Competency XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema"
xmlns:competency="http://www.saba.com/RDF/competency10.rdf" xmlns:rdf="http://
www.w3.org/1999/02/22-rdf-syntax-ns#" targetNamespace="http://www.saba.com/RDF/
competency10.rdf" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xsd:include schemaLocation="profile10_competency2.xsd"/>
    <xsd:import namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
schemaLocation="profile10_rdf.xsd"/>
    <!-- COMPETENCY properties  -->
    <xsd:group name="competencies">
        <xsd:sequence maxOccurs="unbounded">
            <xsd:group ref="competency:competencyGroup"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:group name="competencyGroup">
        <xsd:choice>
            <xsd:element name="competency" type="rdf:value"/>
            <xsd:element name="competency" type="rdf:competencyType"/>
            <xsd:element name="competency" type="rdf:competencyBag"/>
        </xsd:choice>
    </xsd:group>
</xsd:schema>
```

Certification XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema"
xmlns:certification="http://www.saba.com/RDF/certification10.rdf"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" targetNamespace="http://
www.saba.com/RDF/certification10.rdf" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xsd:include schemaLocation="profile10_certification2.xsd"/>
   <xsd:import namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
schemaLocation="profile10_rdf.xsd"/>
   <!-- CERTIFICATION properties begin here -->
   <xsd:group name="certifications">
       <xsd:sequence maxOccurs="unbounded">
           <xsd:group ref="certification:certificationGroup"/>
       </xsd:sequence>
   </xsd:group>
   <xsd:group name="certificationGroup">
       <xsd:choice>
           <xsd:element name="certification" type="rdf:value"/>
           <xsd:element name="certification" type="rdf:certificationType"/>
           <xsd:element name="certification" type="rdf:certificationBag"/>
       </xsd:choice>
   </xsd:group>
</xsd:schema>
```

RDF XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema" xmlns:rdf="http://
www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:profile="http://www.saba.com/RDF/
profile10.rdf" xmlns:competency="http://www.saba.com/RDF/competency10.rdf"
xmlns:certification="http://www.saba.com/RDF/certification10.rdf"
targetNamespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xsd:include schemaLocation="profile10_rdf_common.xsd"/>
    <xsd:import namespace="http://www.saba.com/RDF/profile10.rdf"
schemaLocation="profile10_profile2.xsd"/>
    <xsd:import namespace="http://www.saba.com/RDF/competency10.rdf"
schemaLocation="profile10_competency2.xsd"/>
    <xsd:import namespace="http://www.saba.com/RDF/certification10.rdf"
schemaLocation="profile10_certification2.xsd"/>
    <!-- simple rdf Bag element containing string elements -->
    <xsd:complexType name="Bag">
        <xsd:element name="Bag">
            <xsd:complexType>
                <xsd:sequence maxOccurs="unbounded">
                    <xsd:element name="li" type="xsd:string"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:complexType>
    <!-- simple rdf Seq element containing string elements -->
    <xsd:complexType name="Seq">
        <xsd:element name="Seq">
            <xsd:complexType>
                <xsd:sequence maxOccurs="unbounded">
                    <xsd:element name="li" type="xsd:string"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:complexType>
    <!-- the definition of the vCard:TYPE property for use in the vCard:ADR
property.  Defined here so that the resource attribute is in the rdf namespace.
-->
    <xsd:complexType name="ADRType" content="empty">
        <xsd:attribute name="resource">
            <xsd:simpleType base="xsd:uriReference">
                <xsd:enumeration value="http://imc.org/vCard/3.0#dom"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#intl"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#postal"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#parcel"/>
```

```
                <xsd:enumeration value="http://imc.org/vCard/3.0#home"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#work"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#pref"/>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:complexType>
    <!-- the definition of the vCard:TYPE property for use in the vCard:TEL
property.  Defined here so that the resource attribute is in the rdf namespace.
-->
    <xsd:complexType name="TELType" content="empty">
        <xsd:attribute name="resource">
            <xsd:simpleType base="xsd:uriReference">
                <xsd:enumeration value="http://imc.org/vCard/3.0#home"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#msg"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#work"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#pref"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#voice"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#fax"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#cell"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#video"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#pager"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#bbs"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#car"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#isdn"/>
                <xsd:enumeration value="http://imc.org/vCard/3.0#pcs"/>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:complexType>
    <xsd:complexType name="resultBagType">
        <xsd:element name="Bag">
            <xsd:complexType>
                <xsd:sequence minOccurs="1" maxOccurs="unbounded">
                    <xsd:choice>
                        <xsd:element name="li" type="xsd:string"/>
                        <xsd:element name="li" type="profile:resultType"/>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:complexType>
    <xsd:complexType name="complexLearning">
        <xsd:group ref="profile:complexLearning"/>
        <xsd:attribute ref="rdf:parseType"/>
    </xsd:complexType>
    <xsd:complexType name="learningContainer">
        <xsd:element name="Bag">
```

```
            <xsd:complexType>
                <xsd:sequence minOccurs="1" maxOccurs="unbounded">
                    <xsd:choice>
                        <xsd:element name="li" ref="rdf:value"/>
                        <xsd:element name="li" type="rdf:complexLearning"/>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:complexType>

    <xsd:complexType name="complexGoal">
        <xsd:group ref="profile:complexGoal"/>
        <xsd:attribute ref="rdf:parseType"/>
        <!-- TODO:  add a key for this relationship -->
        <xsd:attribute ref="rdf:id"/>
    </xsd:complexType>

    <xsd:complexType name="goalContainer">
        <xsd:choice>
            <!-- note:  cannot use an abstract type here because we do not want to
require an xsi:type attribute -->
            <xsd:element name="Bag" type="rdf:goalElements"/>
            <xsd:element name="Seq" type="rdf:goalElements"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="goalElements">
        <xsd:sequence minOccurs="1" maxOccurs="unbounded">
            <xsd:choice>
                <xsd:element name="li" type="profile:simpleGoal"/>
                <xsd:element name="li" type="rdf:complexGoal"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="observationBag">
        <xsd:element name="Bag">
            <xsd:complexType>
                <xsd:sequence minOccurs="1" maxOccurs="unbounded">
                    <xsd:element name="li" type="rdf:observationType"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:complexType>
    <xsd:complexType name="observationType">
        <xsd:group ref="profile:observationGroup"/>
        <xsd:attribute ref="rdf:parseType"/>
```

```
    </xsd:complexType>
    <xsd:complexType name="competencyBag">
        <xsd:element name="Bag">
            <xsd:complexType>
                <xsd:sequence minOccurs="1" maxOccurs="unbounded">
                    <xsd:choice>
                        <xsd:element name="li" type="rdf:value"/>
                        <xsd:element name="li" type="rdf:competencyType"/>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:complexType>
    <xsd:complexType name="competencyType">
        <xsd:group ref="competency:competencyElementGroup"/>
        <xsd:attribute ref="rdf:parseType"/>
    </xsd:complexType>
    <xsd:complexType name="certificationBag">
        <xsd:element name="Bag">
            <xsd:complexType>
                <xsd:sequence minOccurs="1" maxOccurs="unbounded">
                    <xsd:choice>
                        <xsd:element name="li" type="rdf:value"/>
                        <xsd:element name="li" type="rdf:certificationType"/>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:complexType>
    <xsd:complexType name="certificationType" >
        <xsd:group ref="certification:certificationElementGroup"/>
        <xsd:attribute ref="rdf:parseType"/>
    </xsd:complexType>
</xsd:schema>
```