A WEB-SERVICE-BASED E-LEARNING SERVICE INFRASTRUCTURE FOR ACHIEVING DYNAMIC AND COLLABORATIVE E-LEARNING

By

GILLIEAN LEE

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2005

To my wife, Rang-wook, and my parents.

ACKNOWLEDGMENTS

The work with this dissertation has been extensive, requiring much work and fun. Without help and support from many persons, I would never have been able to finish this work.

First of all, I would like to thank my advisor Stanley Su for his inspiring and encouraging way to guide me, and his invaluable comments during the whole work with this dissertation. He showed me how to approach a research problem and how to devise and provide a systematic solution. He also provided me with many opportunities to publish papers and meet people in the e-learning area.

I would like to thank Howard Beck for his guide, challenging questions and support. I will also give a special thanks to Sem Lampotang for his strong support and decision to make use of this work for his work without hesitation. I am grateful to Herman Lam for his encouragement and very helpful comments throughout this work. A special thanks goes to Abdelsalam Helal for his guidance. I would like thank Markus Schneider for his support and serving as an advisory committee.

Thanks to all my colleagues at the Database Systems R&D Center who gave me lot of help. Especially I would like to thank Seema Degwekar and Xu Zhang for helping me with implementations and Brian Weinrich for proofreading this dissertation. A special thanks to Sharon Grant for reviewing many research papers even with very short notices.

I cannot thank enough to my parents for giving me life in the first place and their endless support. I also would like to thank my parents-in-law for their support. Last, but

not least, I would like to give a very special thanks to my wife, Rang-wook Oh, who always has been there for me and to my lovely daughter Narae for their patience and understanding all the hard works I went through.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

A WEB-SERVICE-BASED E-LEARNING SERVICE INFRASTRUCTURE FOR ACHIEVING DYNAMIC AND COLLABORATIVE E-LEARNING

By

Gilliean Lee

August 2005

Chair: Stanley Y. W. Su
Major Department: Computer and Information Science and Engineering

Multimedia data and application systems that are accessible over the Web are valuable assets for developing instructional materials for teaching, training, problem solving and decision support. These assets can be used for constructing learning objects, each of which is a reusable granule of instruction designed to meet a specific instructional objective. In order to search, access, and deliver the learning objects, an infrastructure for the registration, discovery, and invocation of these objects is required. Also, there is a need to develop infrastructural technologies to enable general Internet users, not just students and educators, to engage in life-long instruction and learning. These technologies should allow the establishment of many virtual e-learning communities, each of which is formed by people with a common interest in exchanging knowledge useful for problem-solving and decision support in a specific problem domain.

In this work, we model distributed and sharable learning resources by two types of Learning Objects (LOs): Atomic Learning Object and Composite Learning Object. Both

types of LOs are uniformly published as Web-services in a constraints-based Web service broker. XML-based languages for modeling these two types of LOs are defined. The languages also serve as the interchange formats for transferring and sharing LOs.

We have developed an e-learning service infrastructure that facilitates the authoring, registration, discovery, storing, and invocation of LOs. The infrastructure is built on top of an extended Web-service framework, which leverages the constraints-based Web service broker and distributed LO repositories to make LOs searchable, sharable and reusable. A set of distributed software components has been developed to provide e-learning services. It includes authoring tools for constructing LOs, and a Learning Process Management System for the registration, management and delivery of LOs and the assessment of learners' comprehension. The techniques used to achieve active, adaptive, flexible and customizable execution of learning processes such as dynamic binding to LOs, rule-based execution of learning processes that makes use of the profiles and runtime status models of learners, and model-based assessment are presented. The roles and functions of virtual e-learning community members are also discussed.

The e-learning technologies we developed have been applied to encapsulate the instructional materials of the Web-based Virtual Anesthesia Machine and the simulator as LOs, deliver instructional materials to learners, use the simulation system for practice, and perform assessment. Several techniques, such as retrying or disabling activities depending on a learner's profile, performance and needs, dynamic binding to suitable learning objects, optional assessment presentation, and the support of different learning approaches by customization of sequencing control modes, have been applied to achieve personalized learning.

CHAPTER 1
INTRODUCTION

The advent of Internet and Web technologies has enabled organizations and individuals all over the world to place enormous quantity of multimedia data (texts, Web pages, graphs, images, Web simulation, video and audio) on the Web. Increasingly, Web-based application systems are also made accessible to Web users through browsers. These data and application systems are valuable resources for constructing instructional materials for teaching, training, problem solving and decision support. One approach to make use of these distributed, heterogeneous data and application systems is to apply the object-oriented technology and "wrap" them as distributed objects. These distributed objects can then be used as learning assets to compose learning objects (LOs) for instruction and training purposes.

In recent years, there have been a number of initiatives in developing and standardizing Web-based learning technologies. The Advanced Distributed Learning Initiative [ADL, 2003], the IMS Global Learning Consortium [IMS, 2004], and the Open Knowledge Initiative [Eduworks & OKI Leadership, 2002], are a few examples.

The Sharable Content Object Reference Model (SCORM) [ADL, 2004a] is a reference model initiated by the Advanced Distributed Learning (ADL) program of the Department of Defense (DoD) and the White House Office of Science and Technology Policy (OSTP). According to the SCORM's specification, it is envisaged that Internet users and heterogeneous LMSs would use the Web as a universal platform for accessing and launching sharable content objects and for establishing close communication,

interaction and coordination among content object developers, course authors, users, and administrators. To realize this vision, sharable content objects must be durable, interoperable, accessible and reusable. In order to meet these requirements, it is necessary to have a uniform way of modeling, not only learning resources, but also heterogeneous learning tools and LMSs, as well as an information infrastructure to enable the interoperation and sharing of their contents and functionalities. Also, the aggregation model that defines the learning sequence or process has to be flexible, adaptable and customizable to meet different learners' needs and learning contexts. Our work to meet the above needs is consistent with the vision and goals of the ADL Program.

There are systems such as MERLOT [MERLOT, 2004], MIT OpenCourseWare [OCW, 2003], CLOE [CLOE, 2004] and EOE [The EOE Foundation, 2004], which offer repositories of learning assets to virtual learning communities. Each learning asset is described by metadata and a Universal Resource Locator (URL), which is searchable either by categories or by keywords using a search engine. These systems function more like directories provided by digital libraries. Users of these systems have to manually search the directories and access the learning assets that are "pointed to" by the URLs stored in the directories. The above systems do not provide externally accessible programming interfaces for accessing the contents of the directories and the associated learning assets. Nor do they provide facilities for assembling instructional materials out of the learning assets available in the Web, delivering content to users based on their needs and profiles, and assessing users' understanding of the delivered content.

On the other hand, there are commercial instructional management systems (IMSs) such as WebCT [WebCT, 2005], Blackboard [Blackboard, 2005], and eCollege [eCollege,

2005], and research systems such as LON CAPA [Laboratory for Instructional Technology in Education, 2004] and Sakai [Sakai Project, 2005], which are designed to support course development, deliver contents to students, perform assessment, and provide tools for course management. These systems are designed and implemented for creating, teaching and managing courses and their targeted users are instructors, students and course administrators of educational institutions.

The existing systems that provide directory accesses to learning assets and the existing instructional management systems represent two ends of a spectrum. Useful systems can be built in between these two ends. For example, a system may provide tools for Internet users, who are not necessarily course developers or instructors, to design, develop and share LOs. LOs do not have to be instructional materials for formal courses. They can be small granules of information content that are useful to some learners and contain practice and assessment items to exercise and assess the learners' comprehension of the content. The learners do not have to be students who take courses in schools or receive some formal training in business and government organizations. They can be ordinary Internet users who need help in various decision-making and problem-solving situations. Since such a system is not a full-fledged IMS, it will be simpler and easier to develop and maintain than IMSs.

Our vision is to build an e-learning service infrastructure, which provides tools for authoring learning objects, facilities for aggregating and sequencing of small granules of learning objects to form reusable composite learning objects, and a Learning Process Management System for learning object registration, management, delivery and assessment. The e-learning service infrastructure will enable the global participation of

Internet users in life-long instruction and learning. Internet users with similar interest in a specific subject or domain will be able to form a virtual e-learning community to share their knowledge. This is consistent with the vision of "building the technological infrastructure to support dynamic, ad-hoc communities of lifelong learners who interact within an environment of learning objects through a creative blend of advanced computing technologies, high performance networks, authoring and collaboration tools" stated in York et al. [2002].

Internet users have much more diverse backgrounds than students in schools. A learning process management system that delivers identical contents to all learners cannot accommodate diverse Web users' needs. Web-based learning should accommodate learners' different profiles, which include users' backgrounds, competencies and interests. A Learning Process Management System must be flexible, adaptive and customizable in order to satisfy heterogeneous learners' needs.

Learning is largely a social activity; and even the most well developed multimedia interactive materials lack the flexibility of human interaction [Lewis, 2002]. Therefore, a Learning Process Management System that facilitates interactions and collaborations among the people involved in learning will enhance the learning experience.

This work focuses on the creation of an Web-services-based learning, problem-solving and decision-support environment, in which Internet users (not limited to educators) who are knowledgeable in any subject domain (e.g., how to purify water, how to care for a patient with a certain disease, etc.) can participate in a global virtual community to create, compose, and evaluate distributed, reusable and assessable multimedia LOs for instructional purposes; and Internet users who engage in lifelong

learning, problem-solving and decision-making can request, access and study the contents of these LOs, be assessed of the knowledge gained, and engage in collaborative learning with other learners and content producers.

The e-learning virtual communities envisioned in this work are different from the existing virtual communities on the Internet. Learning in the virtual e-learning communities is based on reusable, assessable, and programmatically accessible e-learning objects. LOs are stored in distributed repositories of content providers to avoid the bottleneck problem of a centralized repository and to give content providers more authority and freedom to manage their LOs. These objects are designed and evaluated by content producers and content evaluators with the support of user interface tools. They are delivered and managed by the Learning Process Management System, which consists of a Learning Process Execution Engine (LPEE) for processing learning objects, an Event-Trigger-Rule (ETR) Server [Lee, Su & Lam, 2001] and other supporting components for achieving collaborative, adaptive, customizable and flexible e-learning.

To summarize, our vision is the creation of a sustainable e-learning infrastructure and the development of advanced e-learning technologies to enable the global participation of Internet users in life-long instruction and learning. Internet users with similar interest in a specific subject or domain will be able to form and operate their virtual e-learning community to share their knowledge captured in reusable, compose-able and assessable learning objects and use them to aid decision-making and problem solving. The contribution of this work is in the development of an extended Web-services infrastructure and dynamic and collaborative e-learning technologies essential for the realization of our vision. The developed infrastructure and technologies are in forms of

learning object models, learning object authoring tools, learning object definition languages, user interface tools, information and runtime status models of learners, distributed e-learning service architecture, event-trigger-rule-based execution of learning processes, and model-based assessment.

This dissertation is organized as follows: Related works are discussed first. Then, two learning object models for defining what we call Atomic Learning Object (ALO) and Composite Learning Object (CLO) are presented in Chapter 3. The languages used to define ALOs and CLOs are then described in Chapter 4. Next, Chapter 5 describes an e-learning service infrastructure for virtual e-learning communities. The implementation of the infrastructure including tools and software system components are also discussed. Chapter 6 discusses an application of our e-learning technologies to encapsulate the instructional materials of the Web-based Virtual Anesthesia Machine and the simulator as LOs, deliver instructional materials to learners, use the simulation system for practice, and perform assessment. A summary and the follow-up R&D efforts are given in the last section.

CHAPTER 2
RELATED WORK

Effective delivery of learning contents to learners of varied backgrounds is one of the goals of e-learning. Organizing learning contents is an importance task for achieving this goal. There are several research and standardization efforts in developing learning process models. SCORM's Content Aggregation Model [ADL, 2004b] and Cisco's Reusable Learning Object [Termaat et. al., 2003] are two popular models that organize learning objects for effective delivery. The structure of learning content specifies the delivery sequence of the content. A tree structure, which is commonly used to present the 'table of contents' of a book or training manual, is generally used to model a learning process. A tree node represents a granule of learning content to be presented to learners. It can represent a course, module or lesson. The tree structure is used in SCORM, Cisco's reusable learning object (RLO), CLI Virtuoso [Cisco Learning Institute, 2004], KnowledgeTree [Brusilovsky, 2002], and KOD [Sampson, 2002].

2.1 Cisco's Reusable Learning Object (RLO) Strategy

Cisco's Reusable Learning Object (RLO) Strategy is a scheme for configuring LOs that can be used in a learning process or learning management. It recognizes two levels of LOs: Reusable Information Objects (RIOs) at the bottom level and RLOs at the top level. Each RIO contains content items, practice items and assessment items related to a learning objective. These items are stored in a RIO so that the consistency among content delivery, practice and assessment can be maintained when the RIO is used in different contexts. A larger structure called RLO can be formed out of RIOs. It can have an

overview, a summary, an assessment, and five to nine (7±2) RIOs. The assessment in an RLO is simply a collection of the assessment items taken from the RIOs that form the RLO. When an RLO is delivered, the content items in the RIOs are presented, and then the assessment items randomly chosen from its RIOs are given. Figure 2-1 shows a RIO, an RLO and their relationship.

Although an RLO can be meta-tagged, it lacks a meta-data specification that describes RLOs and RIOs. A concrete specification of meta-data is required especially for automated discovery and dynamic binding purposes. In our work, we use an Object-Oriented Constraint Specification Language (OOCSL) introduced in Su et. al. [2001] to specify the meta-data of a learning object.

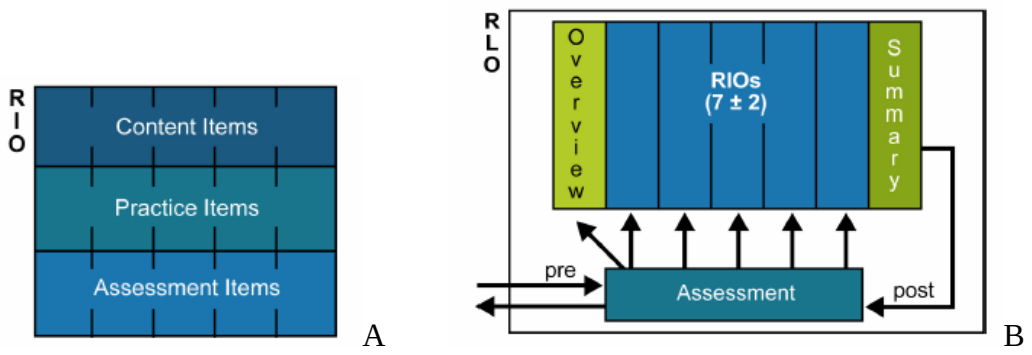


Figure 2-1. Cisco's RLO and RIO. A) Reusable Information Object. B) Reusable Learning Object

2.2. ADL's Sharable Content Object Reference Model (SCORM)

SCORM is a reference model, which is intended to provide a standard framework for building learning systems to enable the reuse and sharing of learning contents developed by different authoring tools. Since our learning process model makes use of some ideas presented in SCORM's sequencing and navigation model, we describe SCORM's model in more details below.

In SCORM, there are two types of specialized instances of resources: *Asset*, and *Sharable Content Object (SCO)*. Asset is a learning content in its most basic form. Assets are electronic representations of media, texts, images, sound tracks, web pages, chat sessions, assessment objects or other pieces of data that can be rendered by a Web client and presented to a learner [ADL, 2004b]. SCO represents a collection of one or more Assets packaged as a single launchable resource that can *communicate* with a Learning Management System (LMS) using its runtime data environment.

SCORM's Sequencing Definition Model [ADL, 2004a] is a rule-based sequencing model, which supports adaptive execution of an activity tree using sequencing rules, rollup rules, and the status of learning objectives. An activity tree is a hierarchical structure of learning activities and their corresponding learning contents. A *cluster*, composed of a parent and its child activities, is the basic building block for sequencing. An activity tree is composed of a set of clusters. The parent activity specifies information about the sequencing strategy for navigating through its child activities. Figure 2-2 shows the structure of an activity tree and its clusters in SCORM's Sequencing Definition Model. An activity can have an associated resource such as SCO resource or Assets resource, or can be made up of additional sub-activities consisting of other clusters.

In SCORM, learning objectives represent a set of locally and globally scoped data items, each with a satisfaction status and a satisfaction measure. Activities may have more than one associated local objective and may reference multiple globally shared objectives. The *Tracking Status Model* records the status of progress associated with objectives, activities, and attempts of activities.

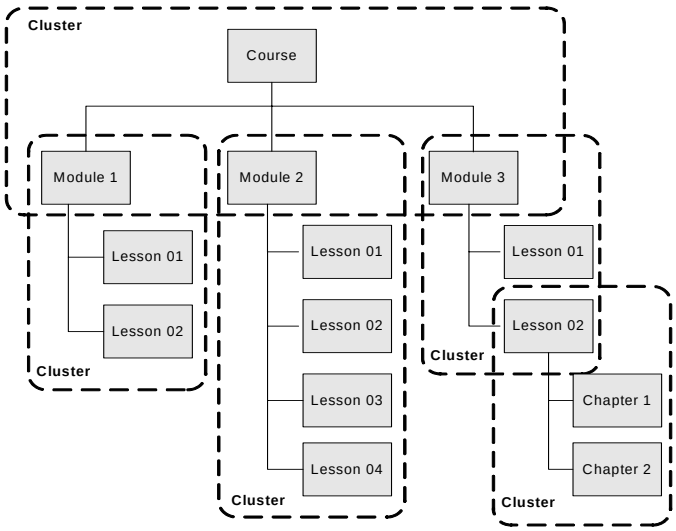


Figure 2-2. A Representation of an Activity Tree Composed of Clusters in SCORM

SCORM's Sequencing Definition Model makes use of a rule-based sequencing model, which adopted the IMS Simple Sequencing Specification [IMS, 2003]. A set of rules is defined for an activity, and is evaluated at specific times during the delivery of an activity. A *sequencing rule* is in the form of ***if [condition] then [action/behavior]***. It consists of a set of conditions and an action/behavior. The action is executed when the conditions of the rules are evaluated to True. The conditions of a sequencing rule are based on the Tracking Status Data. Figure 2-3 shows the possible conditions and actions/behaviors in sequencing rules.

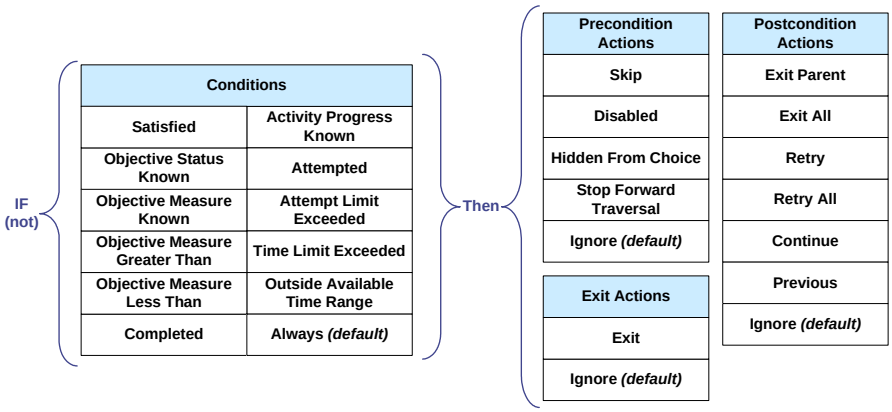


Figure 2-3. Conditions and Action/Behavior in SCORM's Sequencing Definition Model

SCORM's Sequencing Definition Model provides the desirable set of features described above; however, it has a few limitations. First, the definition of learning objects in SCORM does not explicitly distinguish content, practice, and assessment items as in Cisco Systems' RLO, which is adopted in our work. Additionally, we make use of meta-data and their constraints specified in each LO to achieve more accurate, dynamic binding of requests to LOs. Second, non-leaf activities in an Activity Tree neither present contents nor perform assessment in SCORM. We believe that it is useful to allow a non-leaf node to present an introduction and/or a summary of the contents covered by its child activities. Assessment items for testing the integrated knowledge of the contents presented by the child activities should also be allowed. Third, there is no model or specification for conducting assessments in the SCORM sequencing definition model, hence the coverage of an assessment is decided entirely by a SCO bound to the activity instead of the information specified in its sequencing model. As a result, a change of assessment coverage in an activity tree requires a structural change of the learning process and the authoring or modification of an SCO to conduct the new assessment. This may require a substantial amount of time and effort. In this work, we introduce a model-based assessment technique to facilitate flexible assessments.

2.3 Flex-EL

A workflow process model represented by a directed acyclic graph (DAG) is employed in Flex-EL [Lin, 2002]. The model can specify the interaction, coordination and collaboration among the individuals involved in learning and also the sequencing of contents to be presented to learners. The system uses conditional transitions expressed in a workflow model to achieve the flexibility in selecting learning pathways that match each learner's individual pace and learning style. Different from Flex-EL, we model a

learning process by an activity tree, and use sequencing rules and rollup rules as in SCORM. The action part of the rules can change the sequencing paths by allowing learners to retry, hide, and skip activities when the conditions of the rules are met. In addition, we allow the structure of a learning process or sequencing control modes to be customized at runtime by invoking some rules to check a learner's profile and progress.

2.4 KnowledgeTree

The adaptive Web-based educational system, KnowledgeTree, introduced by Brusilovsky [2002], separates learning materials into primary materials for average learners and additional materials for learners with different learning styles and knowledge. The system uses learning goals, preferences and knowledge of individual learners to select the most appropriate learning materials. Our dynamic binding of requests to LOs is similar to their idea of binding learning materials at runtime. However, we leverage on the standardized Web-services technology for the registration and discovery of LOs and add meta-data and constraints in LO specifications.

2.5 L3

L3 [Altenhofen, 2002] separates learning processes from strategies, that is, navigation rules, and introduces meta-tags for describing the knowledge type of learning activities and the inter-relationships among the activities. A strategy chosen at runtime and the meta-tags of learning activities enable adaptive navigation of a learning process. Different from their work, our learning process model allows sequencing rules and rollup rules to be attached to each activity at both design time and runtime in order to achieve dynamic properties in a learning process execution.

2.6 NETg

NETg [Thomson, 2005] is a commercial learning management system. With pre-assessment technique, NETg can selectively deliver contents that a learner needs to take according to the assessment result. However, different from this work, they do not make use of post-assessment results and learner profile for adaptive delivery. In NETg, the assessments can be taken anytime a learner wants, and the deliverable learning contents are written in a proprietary format instead of the general Web document format.

2.7 MERLOT

MERLOT (Multimedia Educational Resource for Learning and Online Teaching) [MERLOT, 2004] is a free and open resource repository designed primarily for faculty and students of higher education. MERLOT is also a community of people who are involved in education. Community members may contribute materials, assignments and comments. Meta-data-based search and category-based search methods through a Web browser are provided, and both require human intervention. Learning resources can only be accessed through the URLs provided by the system. It does not provide an access mechanism for an external software component to access the resources.

2.8 CORDRA

CORDRA (Content Object Repository Discovery and Registration/Resolution Architecture) [LSAL, 2004] is a *reference model* for content repositories, brokering of contents, and APIs for content registration, discovery and a standard mechanism to access repositories. Its key capabilities include the following:

- Content can be made widely available.
- Content can persist outside of the realm of a single course.
- Content can be discovered.
- Once discovered, there will be standard mechanisms to access the content

- Content can be managed and deploying organizations will be able to tailor management to their needs.
- Implementation of the model will be open and flexible.

Our design requirements and functionality of the implementation of content repository and constraints-based broker are consistent with CORDRA's functional requirements. We leverage standard Web-service technology and Web-service registry and make some extensions to them for the registration, discovery, and access of learning objects.

CHAPTER 3
LEARNING OBJECT MODELS

On the Web, there is an enormous amount of multimedia data and application systems that can potentially be used for constructing instructional materials for teaching, training, problem-solving and decision support. The *object* concept in Object-Oriented Programming can be applied to encapsulate these materials as objects for their use and reuse.

3.1 What Is a Learning Object?

A *Learning Object (LO)* is a granule of instruction designed to meet a specific instructional objective. We model an LO in terms of content items, practice items, assessment items needed to cover a subject of learning, provide practice items and conduct an assessment. Additionally, we specify its meta-data as introduced in Termaat et al. [2003] and Heins & Himes [2002]. By including these three types of learning items in an LO specification, the author of the LO can ensure that practice and assessment items presented to learners are consistent with the presented content items. After an LO is authored, we publish the LO as a Web-Service in a Constraints-based Web service Broker for interoperability and ease of access.

3.1.1 Atomic Learning Object

An Atomic Learning Object (ALO) is a small granule of instruction, which contains some *content items* for teaching learners a problem-solving skill or providing them the knowledge needed in a decision-making situation. An ALO can optionally include *practice* and *assessment items* for exercising and assessing learners' acquired

skill or knowledge. In addition to these learning items, *meta-data* that describes properties and constraints of the LO, and a *delivery sequence* of content and practice items compose an ALO.

3.1.2 Composite Learning Object

ALOs by themselves will have limited use unless they can be used to compose larger granules of instruction. A Composite Learning Object (CLO) is an aggregation model defined as a tree structure of activities with sequencing information and sequencing rules. The following subsections discuss the specification of an activity tree including sequencing control modes, learning objectives and activity data.

3.1.2.1 Activity tree

When a reader selects a book, he/she usually takes a look at the *table of contents* of the book to view the contents and the overall structure of the book. An activity tree in a CLO follows the tree structure of the table of contents to aggregate deliverable contents and defines the delivery sequence of the contents. An activity tree is designed with its modeling constructs such as *activities*, *connectors*, and *edges*.

- Activity: A unit of processing in the delivery of a CLO instance. An activity can represent a granule of instruction such as a lesson, module, chapter, part, and so on. Depending on its location in an activity tree, an activity is either a leaf or a non-leaf activity. A *non-leaf activity* can include content, practice and assessment items in its specification. A content item in a non-leaf activity can provide an abstract and/or an introduction to learners before the sub-tree rooted at the non-leaf activity is delivered. A summary of the contents of a sub-tree can also be presented by a content item. Assessment and practice items in a non-leaf activity can be designed to assess learners' abilities to integrate the concepts and contents presented in its

child activities. A *leaf activity* can bind to a specific ALO or CLO statically, or it can contain constraints for dynamic binding to an ALO or CLO that is suitable to a learner. Additionally, a set of rules can be included in a CLO specification and be used to alter the navigation path and the structure of an activity tree based on each individual learner's profile (i.e., background, age, preference and competency) and progress. In an activity, learning objectives and activity data can be declared and specified. The sequencing rules and the rule-based execution model will be discussed later in this chapter.

- Connector: A modeling construct that bridges a parent activity and its child activities. It includes Sequencing Control Modes of the child activities, which will be discussed in the next section.
- Edge: A directed edge that connects a parent activity to a connector, and a connector to a child activity.

Figure 3-1 shows an example CLO, 'Relational Model', composed of the modeling constructs discussed above.

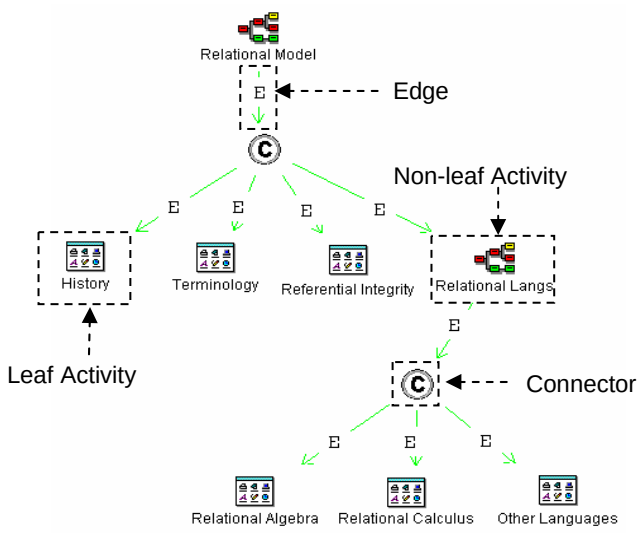


Figure 3-1. Modeling Constructs in a Composite Learning Object

3.1.2.2 Sequencing control modes

Sequencing control modes (SCMs) adopted from SCORM's Sequencing Definition Model defines the sequencing behavior among child activities. SCMs include *Choice*, *Forward-Only* and *Flow* modes. Each mode can be set to True or False, and multiple modes can be enabled to specify combinations of these modes. *Choice* SCM enables the user to choose child activities in any order. *Forward-Only* SCM permits traversal through its child activities only in the forward direction. *Flow* SCM allows sequencing through the child activities in forward and backward directions. The sequence of child activities in Flow and Forward Only SCM is defined in a connector. Figure 3-2 illustrates the three sequencing control modes.

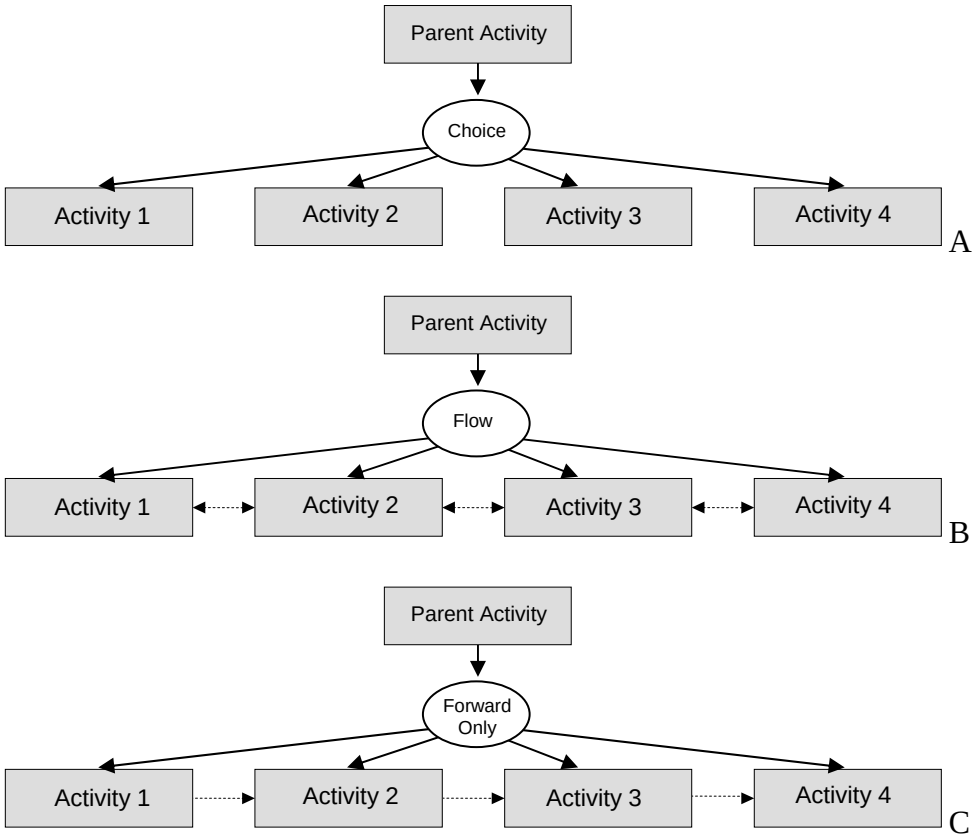


Figure 3-2. Sequencing Control Modes. A) Choice. B) Flow. C) Forward Only.

Unlike SCORM, which defines SCMs in a parent activity of a cluster, we specify them in a connector. By separating the control information (i.e. SCMs) from the activity specification, runtime customization of a learning process instance can be achieved more easily.

3.1.2.3 Learning objective

A learning objective, adopted from SCORM's Sequencing and Navigation model [ADL, 2004a], is a data structure whose data conditions can either be marked as '*Satisfied*' or '*Unsatisfied*' to indicate if a learning objective has been or has not been met.

Learning objectives are declared in an activity. They can be prefixed by the ID of the activity (i.e., '*activityID.objectiveID'*), thus can be unambiguously referenced from any activity of a CLO. To reference a locally declared objective, activityId does not be prefixed. A learning objective includes the data fields such as 'Objective ID (String)', 'Satisfied by Measure (Boolean)', 'Minimum Satisfied Measure (decimal value between 0.0 and 1.0)', and 'Contributes to Rollup (Boolean)'. An objective whose 'Contributes to Rollup' is *True* is referred to as a *primary objective*. Each activity is required to have a primary objective, and the satisfaction status of the primary objective represents the satisfaction status of the activity. At runtime, a learning objective will have a Boolean value for its satisfaction status and a numerical value for its normalized score, that is. measure.

There are two ways to *satisfy* or *unsatisfy* an objective: First, when an objective's *satisfied by measure* field is *True*, a measure (score of an assessment) will be recorded as the result of an assessment. If the measure is equal or above its *minimum satisfied measure*, then the objective will be set to 'Satisfied'. Otherwise, it will be set to 'Unsatisfied'. Second, when an objective's *satisfied by measure* field is *False*, the

objective has to be set to ‘Satisfied’ or ‘Unsatisfied’ by a condition-action rule. For example, in order to satisfy an activity when a learner has taken all the content and practice items, an after-post-assessment rule “*if [true], then [Satisfy this activity]*” can be specified. Details of sequencing rules and the stages of activity processing will be discussed later in this chapter.

3.1.2.4 Activity action

Activity action is a set of simple Java instructions executed at specific execution stages during activity processing. These instructions may check a learner profile and the runtime data model, and assign meaningful values to activity data. Details about their use will be discussed in Chapter 4.

3.1.2.5 Activity data

Activity data are variables declared in an activity. Their values represent the status of the activity, which are used by rules. The data type of each datum can be any one of the simple data types of the Java language such as integer, float, boolean, string, and so forth.

3.1.3 Meta-data

Meta-data of an LO is the data that describes the LO. The meta-data is expressed in terms of a set of attributes and the constraints associated with these attributes. The attributes include Title, Subject, Author, Keywords, Target age, Media format, Language, Cost, Quality Rating, and so on, as in MERLOT, CLOE, and IEEE LOM standard [IEEE LTSC, 2005]. These attributes can be categorized into two types. *Static attributes* such as Title, Subject, Keywords, Language, Target age and Media type are attributes whose constraints are specified by authors. *Dynamic attributes* such as Usage number, Average score and Quality rating are attributes whose constraints are recorded and updated by

evaluators (a role played by qualified individuals in a virtual community) and the Learning Process Management System. Constraints can be categorized into attribute constraints (i.e., the legitimate values of these attributes), and inter-attribute constraints (i.e., their value relationships). An *attribute constraint* has the following form:

- *attribute-Id keyword value | range | enumeration*

  , where *attribute-Id* is the identifier of an attribute, *keyword* can be either a comparison operator '=', '!=', '<', '<=', '>' or '>=', 'RANGE' or 'ENUMERATION'. In case the keyword is a comparison operator, *value* would specify the data value of the attribute in a simple predicate expression. If the keyword is 'RANGE', a *range* specification denoted by *[value..value]* would specify the value range that the attribute can have. In case of 'ENUMERATION', an *enumeration* specification denoted by *{value, value,.., value}* would specify a list of possible values of the attribute.

An inter-attribute constraint is defined as follows:

- *If attribute-constraints then attribute-constraints*

  , where *attribute-constraints* is a Boolean expression in which each term is an attribute constraint.

For example, an attribute constraint of an LO specified by '*Target_age = [3..8]*' specifies that the LO is to be used by children between 3 and 8 years old. An LO that is free of charge to people from a developing country can be specified by an inter-attribute constraint '*If learner.country* $\in$ *developing_country, then cost = 0*'. The meta-data constraints provided by LO producers are used for the registration and discovery of LOs [Lee, Zhang & Su, 2004].

LOs developed for different application domains may require different meta-data attributes. In order to accommodate different applications, meta-data attributes must be customized and/or expanded to meet different application needs. We specify meta-data attributes in an extended WSDL document, which is used by the LO authoring tools and LO search tool, and allow the user to specify meta-data constraints. A part of the extended WSDL document that describes the names and data types of the meta-data attributes used in our current system are shown in Figure 3-3.

```
<service_attributes>
   <service_attribute>
       <name>LO_Type</name>
       <type>string</type>
   </service_attribute>
   <service_attribute>
       <name>Title</name>
       <type>string</type>
   </service_attribute>
   <service_attribute>
       <name>Subject</name>
       <type>string</type>
   </service_attribute>
   <service_attribute>
       <name>Author</name>
       <type>string</type>
   </service_attribute>
   <service_attribute>
       <name>Description</name>
       <type>string</type>
   </service_attribute>
   <service_attribute>
       <name>Keywords</name>
       <type>string</type>
   </service_attribute>
   <service_attribute>
       <name>Material_Type</name>
       <type>string</type>
   </service_attribute>
   <service_attribute>
       <name>Media_Format</name>
       <type>string</type>
   </service_attribute>
   <service_attribute>
       <name>Primary_audience</name>
       <type>string</type>
   </service_attribute>
   <service_attribute>
       <name>Language</name>
       <type>string</type>
   </service_attribute>
   <service_attribute>
       <name>Difficulty</name>
       <type>string</type>
   </service_attribute>
   <service_attribute>
       <name>Workload</name>
       <type>string</type>
   </service_attribute>
   <service_attribute>
       <name>Target_age</name>
       <type>integer</type>
   </service_attribute>
   <service_attribute>
       <name>Level_of_detail</name>
       <type>string</type>
   </service_attribute>
   <service_attribute>
       <name>Study_period</name>
       <type>duration</type>
   </service_attribute>
   <service_attribute>
       <name>Cost</name>
       <type>float</type>
   </service_attribute>
</service_attributes>
```

Figure 3-3. Meta-data Attributes in Extended WSDL Document

3.2 Learning Resources Taxonomy

We distinguish three types of learning resources in a taxonomy of resources. The lowest level of resources is called *learning asset*, which represents the most basic resources available on the Web and can be presented to a learner through a Web browser.

A learning asset can be a simple file, such as a text, image, audio/video clip or plug-in files such as Flash, or a complicated web page that references multiple learning assets. The next level of resources is the *Atomic Learning Object* (ALO) in which each of the learning items includes a learning asset. The highest in the taxonomy is the *Composite Learning Object* (CLO), which is an aggregation model defined as a tree structure of activities with sequencing information. As discussed, a leaf activity can bind to either an ALO or a CLO for the delivery of learning contents, practice and assessment. Figure 3-4 illustrates the taxonomy and reference/binding relationships among learning resources.

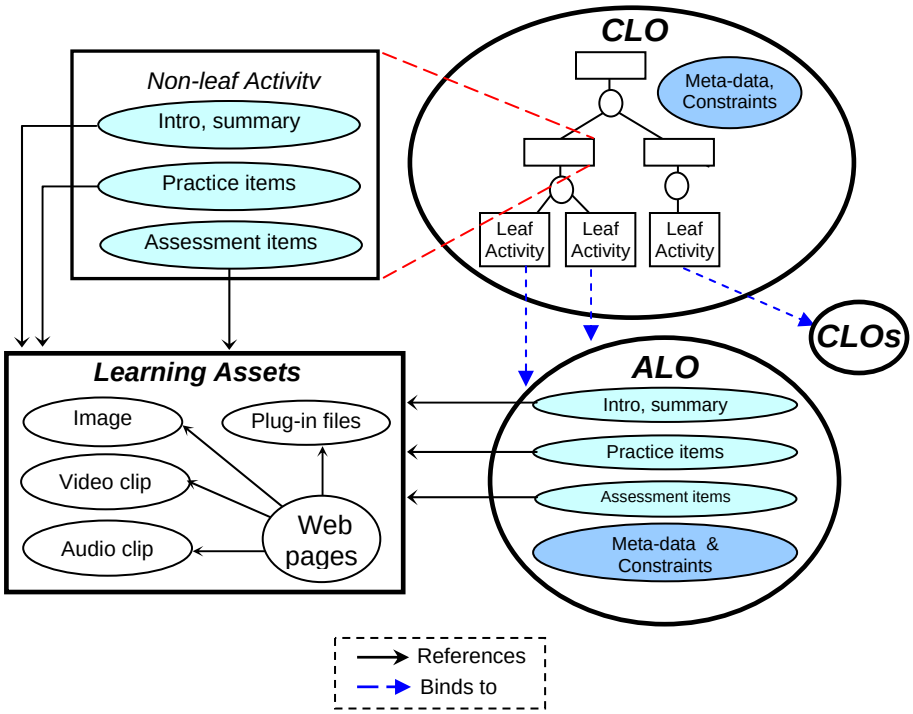


Figure 3-4. Taxonomy of Learning Resources and their Relationships

3.3 Dynamic E-Learning Service – Requirements and Techniques

As discussed, Internet users have much more diverse backgrounds and profiles than students in schools. Therefore, Web-based learning has to be dynamic to accommodate learners' different backgrounds, competencies and interests. To meet this requirement,

learning object services must have the following dynamic properties: *active*, *flexible*, *adaptive*, and *customizable*. They should be "*active*" in that operations on them can automatically trigger rules to enforce policies and constraints, coordinate learning activities, and notify peers and content producers to achieve collaborative e-learning. They should be "*flexible*" in that small granules of LOs and multimedia assets can be used to flexibly compose larger objects, and a request for an object can be flexibly bound to a suitable LO dynamically. They should be "*adaptive*" in that activities can be retried, skipped, exited, enabled or disabled, and sequencing control modes can be changed at runtime to adapt to the *learner's performance* (i.e., assessment results) and interaction with the system. They should be "*customizable*" in that the structures, enabled/disabled status of activities and sequencing control modes can be customized to suit individual learners before the processing of a CLO by using the information provided in *learners' profiles.* A learner's profile contains a learner's identification, demographics, accessibility, background and preference information such as date of birth, name, address, preferred language and learning style, the type of internet connection, disability, interests, learning history, learning objective, and others. Details of the learner profile model will be discussed in Chapter 5. The following sections describe the techniques and mechanisms we have developed to satisfy these requirements.

3.3.1 Rule-Based Execution Model

In this work, we use a rule-based execution model to process a CLO in order to achieve dynamic, flexible, adaptive and customizable e-learning services. The execution model can guide learners through different learning paths that lead to effective learning experience by invoking sequencing rules during the learning process execution. A *CLO instance* refers to a runtime instance of a CLO being enacted for the delivery purpose. A

CLO instance is processed by the Learning Process Management System, which contains a Learning Process Execution Engine (LPEE), an Event-Trigger-Rule (ETR) Server [Lee, Su & Lam, 2001] and a few other server components.

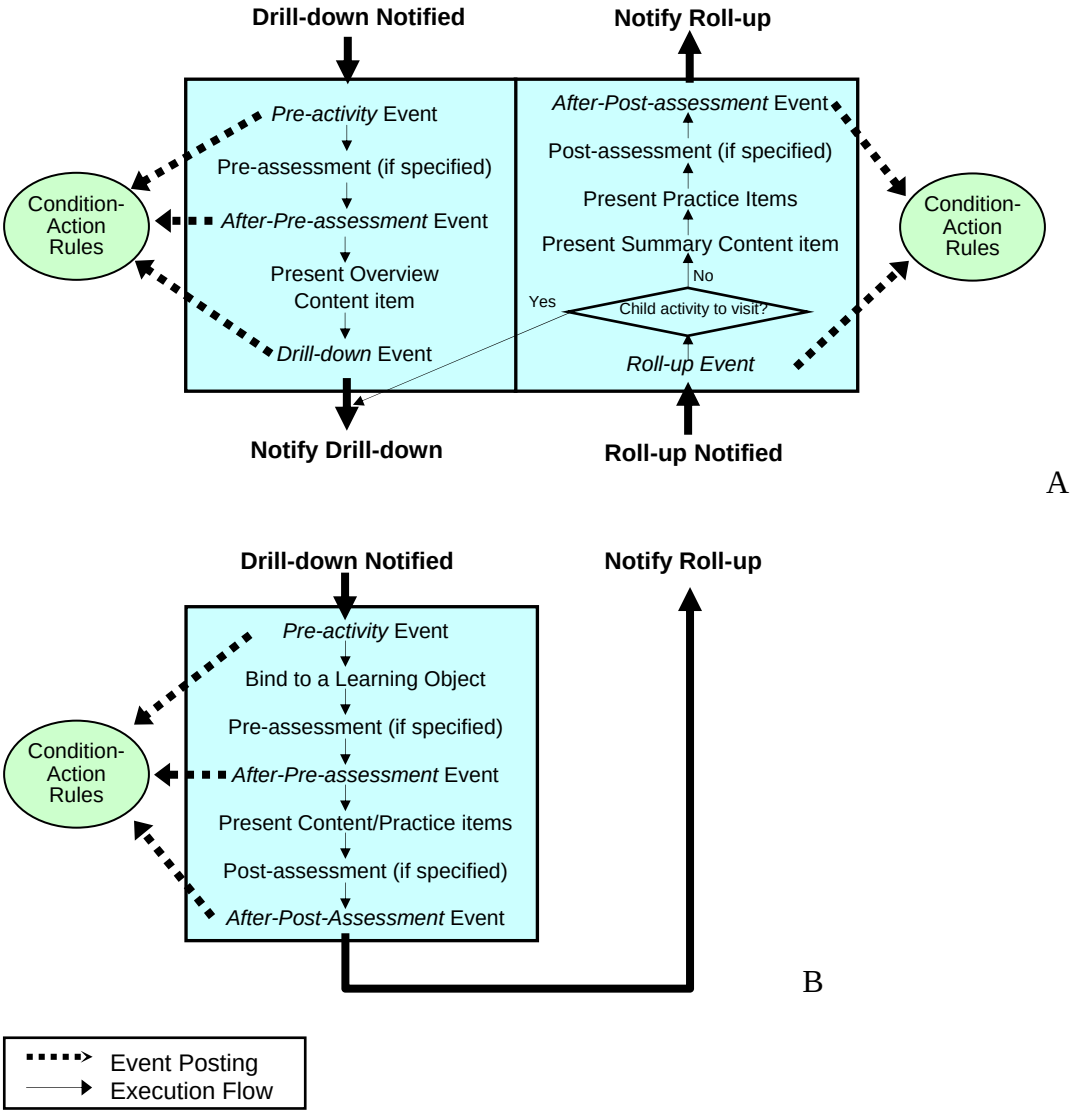


Figure 3-5. Tasks Performed in an Activity Node at Runtime. A) Non-leaf activity. B) Leaf Activity.

Each enactment of a CLO by the LPEE would process the activities of an activity tree by following a sequencing order determined by the followings:

- A pre-order traversal of the activity tree (drill-down and roll-up process).

- Sequencing control modes (Choice, Flow and Forward Only).
- Activation of sequencing rules to retry, skip, exit, enable or disable activities.
- Learner's navigation requests at runtime, for example, exit, next activity, previous activity, choose an activity, continue, and so on.

As shown in Figure 3-5 (A), sequencing rules are executed during the processing of activities. A sequencing rule is a condition-action rule having the structure, **"if [condition] then [action] else [alt-action]"**. The major differences between the processing of leaf and non-leaf activities are that drill-down and roll-up rules do not exist in a leaf activity, and an LO is bound to a leaf activity after the pre-activity event is posted, as shown in Figure 3-5 (B).

During the execution of an activity, events are posted to notify the ETR Server that certain stages of execution have been reached. The ETR Server would invoke sequencing rules associated with the events. As shown in Figure 3-5, a sequencing rule is invoked by an event with the same name. For example, a Pre-activity event activates a Pre-activity sequencing rule. The detailed description of the ETR Server will be given in Chapter 5. An activity may contain the following sequencing rules:

1. **Pre-activity rule** – Processed before the enactment of an activity. Learner's profile or the status of another activity can be referenced in the rule for possible adaptation or customization.
2. **After-pre-assessment rule** – Processed after the pre-assessment task. The result of the pre-assessment can be evaluated for any possible adaptive sequencing.
3. **Drill-down rule** – Processed before leaving the activity to go down the activity tree to process its child activities. Sequencing control modes of the connector between

the activity and its children can be set adaptively at this stage by checking the learner's profile and runtime status. Drill-down rules are applicable only to non-leaf activities.

4. **Roll-up rule** – Processed when an activity is revisited after each of its child activities is finished. A roll-up rule is in the form of ***if [Child_activity_set (Minimum_set) Condition_set] then [action] else [alt-action]***. The *Child_activity_set* specifies a qualifier (All, Any, None, At Least Count, or At Least Percent), which defines the set of child activities referenced during the evaluation of the condition part of a rollup rule. If "At Least Count" or "At Least Percent" is used, it is required that *Minimum_set* is specified. *Minimum_set* represents the minimum count or the percentage of child activities that must be evaluated. The condition set of the rollup rule contains "Satisfied", "Attempted", "Completed", "Objective Status Known", "Objective Measure Known", "Activity Progress Known", "Attempt Limit Exceeded", and "Never" as in SCORM's Sequencing and Navigation [ADL, 2004a]. For example, a rollup rule *"if [At Least Count (2) Satisfied] then [Satisfy this activity]"* states that if any two of the child activities are satisfied, then the activity (parent) is satisfied. Rollup rules are specified in a parent activity.

If there are one or more child activities yet to be processed, sequencing control modes will determine a set of available child activities for the learner's choice. Roll-up rules are applicable only to non-leaf activities.

5. **After-post-assessment rule** – Processed after a post assessment task to check the result of the post-assessment. The result can be used for any possible adaptive sequencing.

The condition part of each sequencing rule may check a learner's profile and the progress status at each stage of activity processing. Thus, an appropriate action can be taken to accommodate a specific learner or a category of learners in different learning contexts. For example, the execution of a sequencing rule can change the sequencing control mode and/or the control structure of the current instance of an activity tree to dynamically adapt to the learning context and the learner's profile and progress. The action part of a rule may specify "Skip Activity", "Exit Activity", "Retry Activity", "Enable/Disable Activity", "Incomplete Activity", "Satisfy/Unsatisfy Activity", "Satisfy/Unsatisfy objective", "Hide Activity from Choice", "Pre/Post-Assess", "Remove/Add Sequencing Control Mode of a Connector", and so forth, so that the delivery sequence of activities can become adaptive, flexible and customizable.

A pre-activity rule specified in a root activity of a CLO can be activated for customization of the CLO instance. For example, a pre-activity rule *"If [learner's goals do not include 'database administrator' or 'DBA'], then [Disable activity 'installing_DBMS']"* would check the learner profile to see if his/her learning goal is to become a database administrator. If not, *'installing_DBMS'* activity that delivers learning items related to the details of a DBMS installation will not be presented to the learner. Before an after-pre-assessment rule is invoked, the result of the pre-assessment will be available. An after-pre-assessment rule *"If [pre-assessment objective is 'Satisfied'], then [Skip this activity]"* would allow the current activity to be skipped when the pre-

assessment result is satisfactory. A drill-down rule of a non-leaf activity specified as *"If [learner prefers 'Receptive' learning style], then [Set Sequencing Control Mode to 'Forward_Only'], else [Set Sequencing Control Mode to 'Choice']"* would not allow a learner who prefers receptive learning style to freely choose the next child activity. Rules as illustrated above can enable the Learning Process Management System to deliver contents to different learners in different order depending on the learners' profiles and the progress they make.

3.3.2 Dynamic Binding

LO repositories established at the sites of content authors who created LOs may be updated frequently, that is, new LOs are created and existing LOs are enhanced with up-to-date contents or infrequently used LOs can be removed from the repositories. One way to leverage LOs stored in the distributed repositories is to register the meta-data and the access methods of the LOs in a Constraint-based LO Broker that is externally accessible. Then learners and the Learning Process Execution Engine can query the LO Broker to find and execute the desired LOs. Another benefit of making LOs programmatically accessible through the LO Broker is to enhance the effectiveness of learning by selecting LOs that suit learners' different competencies and backgrounds. The LO Broker works as a meta-data repository that provides the filtering service.

Dynamic binding refers to the runtime binding of an activity to an LO. It is a more active approach than altering learning paths in that it can choose different learning materials dynamically at runtime. It is achieved by allowing a leaf-activity to specify binding constraints at build-time, and generating a query using the binding constraints to the LO Broker at runtime. The binding constraints describe the requirements of an LO that can satisfy the learning objective of a leaf-activity. They are specified in terms of

attribute and inter-attribute constraints which are in the same format as the meta-data of LOs. The Broker would match the query requirements against the meta-data of the registered LOs to find the suitable LO(s). If more than one LO match with the query, the meta-data of these LOs will be presented to the learner so that he/she can choose an LO of his/her preference. Details about LO registration and binding process will be discussed in Chapter 5.

In order to make dynamic binding adaptive, binding constraints may contain a variable in its attribute constraint specification. This variable will be assigned a value based on a learner's profile, status of objectives, and activity data. In the binding constraints of a leaf activity, the value on the right-hand side of an attribute constraint can be prefixed by 'OBJ:' or 'VAR:', which represents the objective, or the activity data, respectively. At runtime, an objective id prefixed with 'OBJ:' will be replaced by the status of the objective, and a data id with the prefix 'VAR:' will be substituted by the value of the activity data at the time of dynamic binding to generate the query. For example, a binding constraint 'Target_age = VAR:iAge' will be transformed into 'Target_age = 15' when a 15 year old learner is taking the CLO with the activity data *iAge* assigned to the age of the learner. As another example, a learner with a disability can be presented with appropriate content by an attribute constraint '*Media_type ENUMERATION {VAR:content_type}*', where the activity data '*content_type*' can be replaced by an enumeration of strings *{"Movie with caption", "HTML", "Flash"}* for a hearing-impaired learner, or *{"Audio", "Movie with audio"}* for a visually impaired learner. Binding constraints allow the discovery of LOs that have appropriate media types.

3.3.3 Model-based Assessment

Assessment is an important aspect of learning in that it measures a learner's knowledge and problem-solving skills. Assessment results provide important information that can be used to select learning paths, bind activities to suitable LOs, and identify areas that a learner should pay special attention to. SCORM's sequencing definition model does not specify assessment related information, such as when and how an assessment will be conducted. The content of the presented assessment is solely dependent on an assessment LO referenced in a leaf activity. Thus, a change of assessment coverage and location in an activity tree requires a structural change of a learning process and the authoring of an LO that conducts the new assessment. This would require a substantial amount of time and effort.

In this work, we allow the specifications of a pre-assessment and a post-assessment when an activity is defined. We also allow the specification of sequencing rules that can make use of the results of the assessments to select a learning path, and customize a learning process structure as discussed before.

Any activity node in a learning process model can carry out a pre- and/or post-assessment task on the learning contents covered by the tree that is rooted at the node. The assessment items used in the assessment task can be randomly and proportionally selected from the assessment items of the non-leaf nodes and those learning objects bound to the leaf nodes of the tree. The above provision addresses the SCORM's limitation that only leaf nodes of an activity can have assessments, and gives a content producer greater flexibility in specifying when and where assessments are to be carried out.

3.3.3.1 Assessment specification

Assessment specification, which is a part of activity specification, includes information related to the assessment(s) to be conducted in an activity. *Assessment Marker* describes the type of the assessment (pre- and/or post-assessment) to be conducted. *Assessment Parameters* specify the selection percentage (0 ~ 100%) of assessment items and a scoring weight (0 ~ 1.0) for each activity that belongs to the coverage of the assessment. The selection percentage and the scoring weight represent the contribution of the activity for selecting problems and grading the assessment, respectively. The number of assessment items that should be extracted from the items in the activity can be calculated by considering the percentage and the number of assessment items in the activity. For example, if four pre-assessment items exist in an activity and the selection percentage for the activity is set to 50%, then two items are selected randomly from either the activity (in case of non-leaf activity) or the LO bound to the activity (in case of leaf activity) for the assessment. The scoring weight is applied to calculate how many points are contributed to the total score of the assessment. The maximum score, gained score and normalized result score of an assessment are calculated as follows:

$$Max\ Score = \sum_{A_i \in activities\ in\ subtree.} \sum_{P_{ij} \in selected\ assessment\ items\ in\ A_i} Pts(P_{ij}) * weight(A_i)$$

, where $Pts(P_{ij})$ is points allocated to P_{ij}, and $weight(A_i)$ is scoring weight of $A_{i.}$

$$Gained\ Score = \sum_{A_i \in activities\ in\ subtree.} \sum_{P_{ij} \in selected\ assessment\ items\ in\ A_i} PtsGained(P_{ij}) * weight(A_i)$$

, where $PtsGained(P_{ij})$ is points gained from P_i, and $weight(A_i)$ is scoring weight of $A_{i.}$

$$Normalized\ result\ score = Gained\ Score\ /\ Max\ Score$$

The normalized result score is set to the measure of the objective associated with the assessment that has been carried out.

3.3.3.2 Model-based objective measure assignment

As discussed, the satisfaction status of an activity, which is evaluated for the processing of a roll-up rule of its parent activity, is represented by the status of the primary objective of the activity. Additionally, non-primary objective(s) can be declared in an activity. A possible usage of non-primary objectives is for the analysis of an assessment result. In case there is an assessment in an activity, the primary objective's measure may represent the overall gained score, and the measures of the non-primary objectives can represent scores gained from some parts of the assessment, that is, problems chosen from a subset of activities in the assessment coverage. As a result, objectives can be used in an analysis of the assessment, the result of which can be useful for giving advice to learners for future study, and making the execution of the learning process adaptive.

As a part of assessment specification, we provide a mechanism to set multiple objective measures according to the result of an assessment conducted in an activity. An objective is paired with a list of activities that belong to the subtree rooted at the activity. As an example, Figure 3-6 shows a subtree that has an assessment at its root A1. The activity A1 has three objectives: *Overall*, *Left* and *Right*. The objective *Overall* covers the whole subtree, *Left* covers the left subtree, and *Right* covers the right subtree. A list of activities (A2, A4, A5, and A6) is assigned to *Left*, and (A3, A7, A8) to *Right*. With this

specification, the normalized measures of the objectives *Overall*, *Left* and *Right* will be set by calculating the maximum score, gained score and normalized score as discussed.

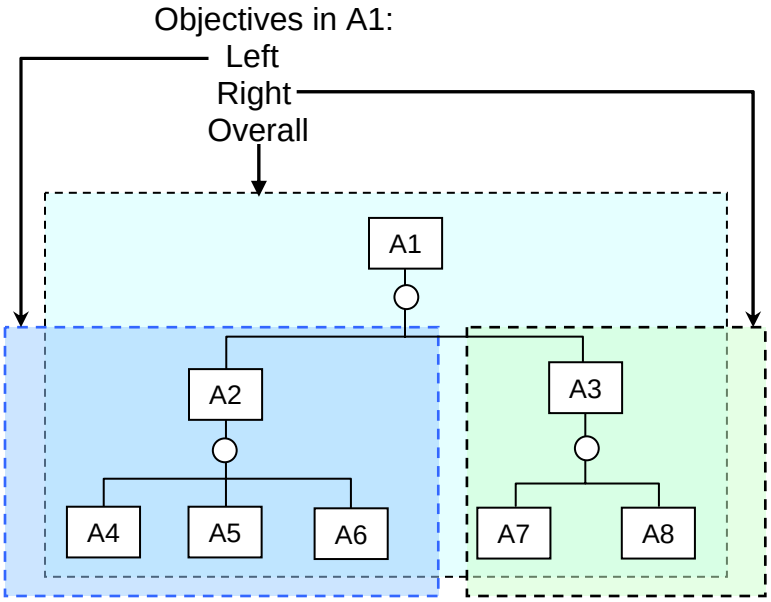


Figure 3-6. Assignment of Measures to Multiple Objectives for an Assessment at A1

Just like there can be multiple quizzes and exams in a course, it is possible to schedule multiple assessments in a CLO. In addition, it is possible that an assessment is conducted optionally by a sequencing rule. As an example, Figure 3-7 shows an activity tree with several assessments. A set of assessment questions will be given at the end of each of the activities at level 2 under the activity A2, that is, A4, A5, and A6. Only those learners, who pass less than two of the three assessments, need to take the assessment on the activity A2. To make the assessment on A2 optional, a roll-up rule of the activity A2 can be specified as "*if [At Least Count (2) are Satisfied], then [Satisfy this activity], else [Unsatisfy this activity]*", and a after-post-assessment rule "*if [this activity is not Satisfied], then [perform a Post-Assessment]*". Assessment parameters need to be specified in A2 for the optional assessment.

In the child activities of A3, there will be no assessment, and there will be a mandatory comprehensive assessment for the entire tree rooted at A1. The roll-up rule of the non-leaf activity A3 is *"If [All child activities are Satisfied], then [Satisfy this activity], else [Unsatisfy this activity]"*. Activities without any assessment will be *satisfied* after they have been navigated by an after-post-assessment rule *"If [true], then [Satisfy this activity]"*. The root activity A1 has a primary objective *primary_objective* and a non-primary objective *assessment_objective* for the post-assessment conducted at the root activity. The entire CLO will be satisfied only if activity A2 and the objective *assessment_objective* are satisfied, which can be specified by an after-post-assessment rule of A1: *"if [(activity A2 is Satisfied) AND (objective(assessment_objective) is Satisfied)], then [Satisfy this activity], else [Unsatisfy this activity]"*.

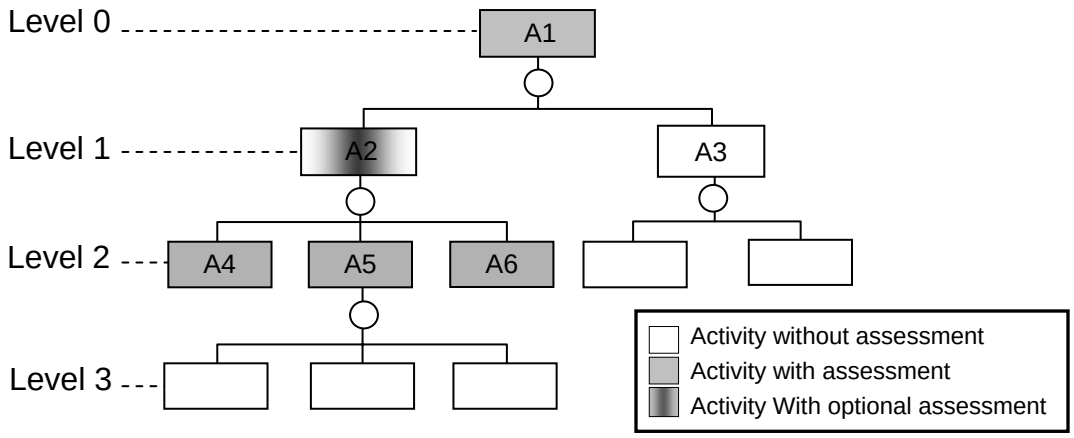


Figure 3-7. Activity Tree with Multiple Assessments

CHAPTER 4
LEARNING CONTENT MODELING LANGUAGE AND LEARNING PROCESS MODELING LANGUAGE

The learning object models, introduced in Chapter 3, are used for defining ALOs and CLOs. They are used as the underlying models for the development of authoring tools to be used by authors of LOs to create ALOs and CLOs. In addition to the underlying models and authoring tools, we need concrete languages that describe these two types of LOs without ambiguity. For the purpose of sharing and exchanging the specifications of ALOs and CLOs, textual languages for defining these two types of LOs are useful. In this work, a *Learning Content Definition Language* (LCDL) and a *Learning Process Definition Language* (LPDL) were designed using the XML technology for defining ALOs and CLOs, respectively. Content authors would utilize authoring tools (to be described in Chapter 5) to define ALOs and CLOs. The tools generate the language descriptions of ALOs and CLOs using these two languages.

4.1 Why XML?

XML (eXtended Markup Language) has become a de facto standard for representing structured and semi-structured information in a textual form [Graham, et. al., 2002]. There are several advantages of the XML technology, which are summarized in [Software AG, 2004] as follows: “

- XML is a meta language. It defines markup languages for specific purposes and schemas for all kinds of data models.
- XML is text-based and easy to read.
- XML is ideal for structured data.

- XML is presentation neutral.
- XML is multi-lingual.
- XML is open technology."

Due to the above advantages and the matured XML technology supported by numerous vendors, XML documents are used across industry and research institutions for describing new technologies and data exchange.

Storing, parsing, and validating data in an XML document requires the definition of an XML data schema. Currently there are two ways to define a XML data schema: XML Schema [W3C, 2001a] and Document Type Definition (DTD). DTD, an older technology, defines a valid XML document format with its proprietary syntax to define XML elements and attributes. XML Schema, developed afterwards, supports new features not supported in DTD, such as definition of new data types, object-oriented concepts including extension/restriction of base data types and type reuse, and namespace that enables incorporation of existing XML Schema definitions to create a new one. In addition, an XML Schema definition is written in a standard XML document to define an XML data format, which is easier to read than a DTD. Due to these advantages, the languages and protocols that describe Web-Service technology, such as Web Service Definition Language (WSDL) [W3C, 2001b], Universal Description, Discovery and Integration (UDDI) [OASIS, 2004], and Simple Object Access Protocol (SOAP) [W3C, 2003], are defined using XML Schemas. We define the learning object languages, that is, LCDL and LPDL, using the XML Schema technology. The XML Schema definitions of LCDL and the LPDL are listed in Appendix A and Appendix B, respectively.

4.2 Meta-model

A meta-model of a modeling language reveals modeling entities, their attributes, and cardinality and containment relationships among the entities. Figure 4-1 shows the meta-model diagram of LPDL and LCDL using the class diagram of the Unified Modeling Language (UML). Each entity, represented by a rectangle in the diagram, is either an XML element or attribute in the XML-based languages.

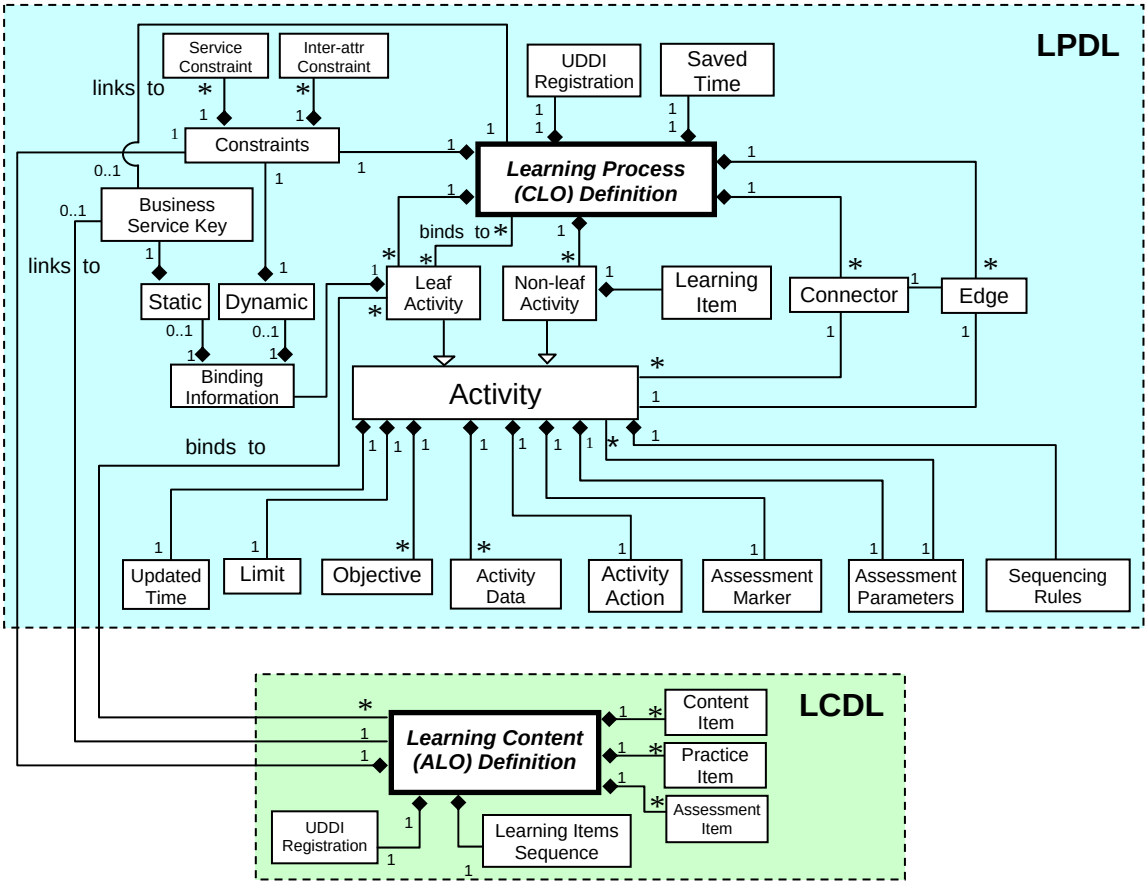


Figure 4-1. Meta-model of Learning Process Definition Language and Learning Content Definition Language and their Relationships

The diagram shows the high-level entities of LPDL and LCDL, and relationships between the entities, such as cardinality, composition, link and binding. To simplify the diagram, a super-class ‘Activity’ is introduced to show that ‘Leaf Activity’ and ‘Non-leaf

Activity' share entities associated with 'Activity'. Details about the meta-model are explained in the following sections.

4.3 Learning Content Definition Language

LCDL is an XML-based language for defining ALOs. A Learning Content Definition, that is, an ALO, consists of *Content Item*, *Practice Item*, *Assessment Item*, *Constraints*, *UDDI Registration*, and *Learning Items Sequence*. A content item contains a learning asset in the form of text, URL, or HTML that presents the content item. A practice item and an assessment item include a content item for its problem content, and an answer. In addition, an assessment item involves a numeric value, that is, a point allocated to the problem, and a Boolean value that indicates whether the item can be automatically graded by simple text matching. The following XML code segment declares learning items consists of four content items and two practice items whose content type is URL, and two assessment items that have text for their problem presentation:

```
.......
<ns1:ContentItems>
   <ns1:ContentItem Name="Q1" Number="1">
      <ns1:URL>http://128.227.176.41:8888/VAM/Low_pressure_system_Q1.
      htm</ns1:URL>
   </ns1:ContentItem>
   <ns1:ContentItem Name="Low Pressure System Learning Objective"
   Number="2">
      <ns1:URL>http://128.227.176.41:8888/VAM/Low_pressure_system_lea
      rning_objectives.htm</ns1:URL>
   </ns1:ContentItem>
   <ns1:ContentItem Name="Q2" Number="3">
      <ns1:URL>http://128.227.176.41:8888/VAM/Low_pressure_system_Q2.
      htm</ns1:URL>
   </ns1:ContentItem>
   <ns1:ContentItem Name="Q1 Learning Objectives" Number="4">
      <ns1:URL>http://128.227.176.41:8888/VAM/Low_pressure_system_Q1_
      Objectives.htm</ns1:URL>
   </ns1:ContentItem>
</ns1:ContentItems>
<ns1:PracticeItems>
   <ns1:PracticeItem Number="1" Name="Q1 VAM Simulation">
      <ns1:Problem Number="0">
         <ns1:URL>http://128.227.176.41:8888/VAM/Low_pressure_system_
         Q1_VAM_simulation.htm</ns1:URL>
      </ns1:Problem>
```

```
        <ns1:Answer>N/A</ns1:Answer>
    </ns1:PracticeItem>
    <ns1:PracticeItem Number="2" Name="Q2 VAM Simulation">
        <ns1:Problem Number="0">
            <ns1:URL>http://128.227.176.41:8888/VAM/Low_pressure_system_
            Q2_VAM_simulation.htm</ns1:URL>
        </ns1:Problem>
        <ns1:Answer>N/A</ns1:Answer>
    </ns1:PracticeItem>
</ns1:PracticeItems>
<ns1:Assessments>
    <ns1:PreAssessmentItems />
    <ns1:PostAssessmentItems>
        <ns1:AssessmentItem Points="3" AutoGrade="true" Number="1">
            <ns1:Problem>
                <ns1:text><p> Q. If a vaporizer concentration dial is
                adjusted to a non-zero setting, will O2 flow generated
                by an O2 flush flow through the vaporizer? </p> <p> A.
                No </p> <p> B. Yes </p> <p> C. Depends on the setting
                of the O2 flowmeter knob </p> <p> D. Depends on the FGF
                setting </p></ns1:text>
            </ns1:Problem>
            <ns1:Answer>A</ns1:Answer>
        </ns1:AssessmentItem>
        <ns1:AssessmentItem Points="3" AutoGrade="true" Number="2">
            <ns1:Problem>
                <ns1:text><p> Q. With the N2O flow meter at its maximum
                setting (> 5 L/min), will a hypoxic gas mixture result
                at the common gas outlet if the O2 flow meter is
                subsequently decreased to its minimum setting (about
                200 mL/min)? </p> <p> A. Yes </p> <p> B. No </p> <p> C.
                Maybe </p></ns1:text>
            </ns1:Problem>
            <ns1:Answer>B</ns1:Answer>
        </ns1:AssessmentItem>
    </ns1:PostAssessmentItems>
</ns1:Assessments>.......
```

Learning items sequence describes a presentation sequence of content and practice items when an ALO is bound to a CLO instance and delivered to learners. *Type* attribute of *LearningItemSequence* element represents the type of the learning item to be presented, and *ItemId* attribute matches with *Number* attribute of the corresponding learning items. *UDDI Registration* records a business interface key and a business service key returned from a UDDI registry when registering the ALO with the Constraint-based Broker. The following defines a learning items sequence and the UDDI registration of an ALO:

```
.......
<ns1:LearningItemSequences>
    <ns1:LearningItemSequence Type="Content" ItemId="2" />
    <ns1:LearningItemSequence Type="Content" ItemId="1" />
    <ns1:LearningItemSequence Type="Practice" ItemId="1" />
    <ns1:LearningItemSequence Type="Content" ItemId="4" />
    <ns1:LearningItemSequence Type="Content" ItemId="3" />
```

```
   <ns1:LearningItemSequence Type="Practice" ItemId="2" />
</ns1:LearningItemSequences>
<ns1:UDDIRegistration BusinessInterfaceKey="UUID:8ACC6F10-115B-11D9-
A2B3-000629DC0A53" ServiceImplementationKey="5B863DF0-7D6A-11D9-9561-
000629DC0A53" />.......
```

Constraints entity is the meta-data that describes an ALO, and is used for the registration and discovery of the ALO through the Constraint-based Broker. The following is an example attribute constraint specification describing that 'Title' is "Low Pressure System – VAM", 'Media_Format' is "HTML" and "Shockwave", 'Language' is "English", 'cost' is 0, and 'Author' is "Gilliean Lee":

```
.......
<ns1:constraints>
   <ns1:service_constraints>
      <ns1:service_constraint>
         <ns1:name>Title</ns1:name>
         <ns1:type>string</ns1:type>
         <ns1:keyword>eq</ns1:keyword>
         <ns1:valueList>"Low Pressure System - VAM"</ns1:valueList>
         <ns1:negotiable>no</ns1:negotiable>
         <ns1:priority>0</ns1:priority>
      </ns1:service_constraint>
      <ns1:service_constraint>
         <ns1:name>Media_Format</ns1:name>
         <ns1:type>string</ns1:type>
         <ns1:keyword>enu</ns1:keyword>
         <ns1:valueList>{"HTML", "Shockwave"}</ns1:valueList>
         <ns1:negotiable>no</ns1:negotiable>
         <ns1:priority>0</ns1:priority>
      </ns1:service_constraint>
      <ns1:service_constraint>
         <ns1:name>Language</ns1:name>
         <ns1:type>string</ns1:type>
         <ns1:keyword>eq</ns1:keyword>
         <ns1:valueList>"English"</ns1:valueList>
         <ns1:negotiable>no</ns1:negotiable>
         <ns1:priority>0</ns1:priority>
         </ns1:service_constraint>
      <ns1:service_constraint>
         <ns1:name>Cost</ns1:name>
         <ns1:type>float</ns1:type>
         <ns1:keyword>eq</ns1:keyword>
         <ns1:valueList>0.0</ns1:valueList>
         <ns1:negotiable>no</ns1:negotiable>
         <ns1:priority>0</ns1:priority>
      </ns1:service_constraint>
      <ns1:service_constraint>
         <ns1:name>Author</ns1:name>
         <ns1:type>string</ns1:type>
         <ns1:keyword>eq</ns1:keyword>
         <ns1:valueList>"Gilliean Lee"</ns1:valueList>
         <ns1:negotiable>no</ns1:negotiable>
         <ns1:priority>0</ns1:priority>
      </ns1:service_constraint>
   </ns1:service_constraints>.......
```

The complete definition of the ALO shown in the above example can be found in Appendix C. Figure 4-2 shows the detailed meta-model of LCDL, consisting of entities and relationships among the entities.

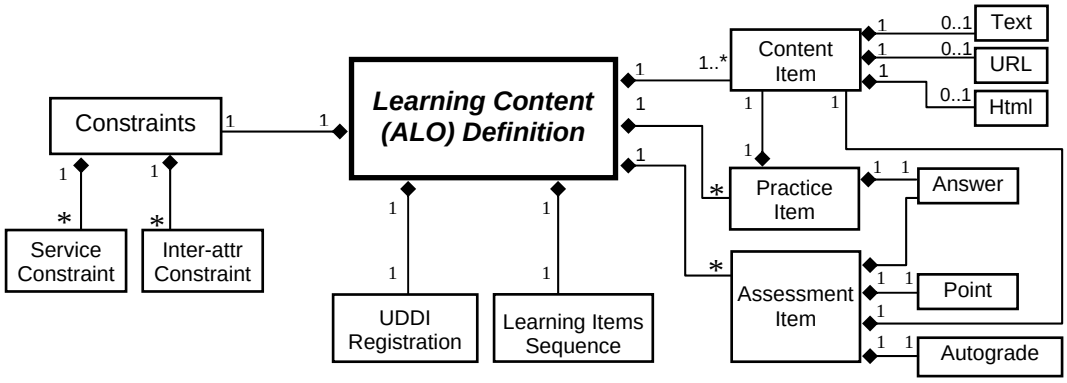


Figure 4-2. Detailed Meta-model for LCDL

4.4 Learning Process Definition Language

LPDL is an XML-based language used for defining a CLO. LPDL specifies a learning process in the form of an activity tree. As discussed in Chapter 3, there are several modeling constructs of the activity tree: leaf activity, non-leaf activity, connector, and edge, each of which is expressed as an XML element in LPDL. The most fundamental modeling entities are *Leaf Activity* and *Non-leaf Activity*, with which the other entities such as *Limit*, *Objective*, *Activity Data*, *Activity Action*, *Binding Info*, *Assessment Marker*, *Assessment Parameters*, *ETR Rules*, *Content items*, *Practice items*, and *Assessment items* are associated. The related entities will be explained when the Activity entities are introduced.

In addition to the modeling constructs of the activity tree, *Constraints* and *UDDI Registration* entities are also specified. *Saved Time* entity stores the most recent time a CLO is created and saved. The information is compared with a deployment time recorded in the LPEE to verify whether a re-deployment of the CLO is necessary or not.

We will use a CLO, which models the instructional process of a Virtual Anesthesia Machine, as an example of LPDL code in this section. The full specification of the CLO can be found in Appendix C.

4.4.1 Non-leaf Activity

In an activity tree, a non-leaf activity is a node that has at least one child node. It can contain contextual contents, that is, introduction and summary, and practices and assessments for exercising and assessing a learner's ability to integrate the contents presented in its child activities. The following XML code segment specifies that an introduction can be accessed through the given URL and a summary is given in the text.

```
.......
<ns1:NonleafContentItems>
   <ns1:Introduction>
      <ns1:URL>http://www.vam.org/VAM/VAM_Intro.htm</ns1:URL>
   </ns1:Introduction>
   <ns1:Summary>
       <ns1:text><p> Original Virtual Anesthesia Machine Web page is
       <a href="http://vam.anest.ufl.edu">http://vam.anest.ufl.edu
       </a>.</p> <p> Special Thanks to Dr. Sem Lampotang </p> <p>
       Copyright: University of Florida. </p>
      </ns1:text>
  </ns1:Summary>
  </ns1:NonleafContentItems>.......
```

The *Limit* entity specifies a delivery limit of the activity in terms of the time experienced or the number of attempts. The *Objective* entity represents a learning objective of the activity. An activity is required to have a primary objective, which represents the satisfaction status of the activity, as discussed in Chapter 3. The *Activity Data* entity contains attributes or variables of simple data types supported by the Java programming language. Their values specify a learner's profile information, the result of some simple calculations, or the current status of an activity. They can be used in the processing of a CLO instance. At runtime, all the values of Activity Data are passed to sequencing condition-action rules, such as Pre-Activity, After-Pre-Assessment, and

After-Post-Assessment rules, for communication between the CLO runtime instance and the rule code. The following XML code segment declares the objectives and activity data of an activity named ‘Part_3_Safety_Exercises’:

```
.......
<ns1:Objectives>
   <ns1:Objective ContributesToRollup="true" Id="oOverall"
       SatisfiedByMeasure="true"
       MinimumSatisfiedNormalizedMeasure="0.5" />
   <ns1:Objective ContributesToRollup="false" Id="oPressureSystem"
       SatisfiedByMeasure="true"
       MinimumSatisfiedNormalizedMeasure="0.5" />
   <ns1:Objective ContributesToRollup="false" Id="oVentilation"
       SatisfiedByMeasure="true"
       MinimumSatisfiedNormalizedMeasure="0.5" />
   <ns1:Objective ContributesToRollup="false"
       Id="oBreathingCircuit" SatisfiedByMeasure="true"
       MinimumSatisfiedNormalizedMeasure="0.5" />
   <ns1:Objective ContributesToRollup="false"
       Id="oScavengingSystem" SatisfiedByMeasure="true"
       MinimumSatisfiedNormalizedMeasure="0.5" />
</ns1:Objectives>
<ns1:ActivityData>
   <ns1:Data DataType="boolean" DataId="bAssessmentReqd" />
   <ns1:Data DataType="int" DataId="iTimes" />
   <ns1:Data DataType="boolean" DataId="bDirective" />
</ns1:ActivityData>.......
```

The *Activity action*, a piece of Java code executed at a specific stage of activity processing, can be defined to perform some simple operations such as checking learner profile and status of objectives, assigning values to activity data, carrying out a data type conversion, and so on. In a non-leaf activity, there are ‘Before-Pre-Activity Actions’, ‘Before-Pre-Assessment Actions’, ‘Before-Introduction Actions’, ‘After-Introduction Actions’, ‘Before-Rollup Actions’, ‘Before-Post-Assessment Actions’, and ‘After-Post-Assessment Actions’. Each action is translated into Java code.

Internally, an activity code class is created for each activity. In the class, activity data are declared as member variables, and a method implements each of the activity actions. LPEE invokes these methods at specific stages of activity processing. Table 4-1 summarizes activity actions in a non-leaf activity and their descriptions.

Table 4-1 Execution Stages and Description of Activity Actions in Non-leaf Activity

Activity Action Name	**Execution Stage**	**Description**
Before-Pre-Activity	Before processing *Pre-Activity* rules	Assign values to activity data to be used for the execution of *Pre-Activity* rules.
Before-Pre-Assessment	Before conducting a pre-assessment	Executed to take any necessary action for the preparation of a pre-assessment.
Before-Introduction	Before processing *After-Pre-Assessment* rules, i.e. before presenting a content item as the introduction	Assign values to activity data to be used for the execution of *After-Pre-Assessment* rules.
After- Introduction	Before drill-down to a child activity, i.e. before processing *Drill-down* rules	Assign values to activity data to be used for the execution of *Drill-down* rules.
Before-Rollup	Before processing *Roll-up* rules, i.e., before roll-up to itself after finishing a child activity	Assign values to activity data to be used for the execution of *Roll-up* rules.
Before-Post-Assessment	Before conducting a post-assessment	Executed to take any necessary action for the preparation of post-assessment.
After-Post-Assessment	Before processing *After-Post-Assessment* rules	Assign values to activity data to be used for the execution of *After-Post-Assessment* rules.

Assessment information for the model-based assessment discussed in Chapter 3 is specified as a part of an activity. The *Assessment Marker* specifies whether a pre- and/or post-assessment is going to be conducted in the activity. For each assessment, there exist the *Assessment Parameters*, which include the *Assessment Selection Parameters* and the *Objective Setting*s. The *Assessment Selection Parameters* specify the activities participating in the assessment. It is composed of a set of tuples *(activity ID, selection percentage, scoring weight)*, where *activity ID* is an identification of a descendent activity from which assessment items will be collected. *Selection percentage* specifies what percentage of assessment items will be chosen from the activity, and *scoring weight* is the weight of selected assessment items to be used for grading the assessment.

Objective Setting entity, which is a subset of descendent activities, indicates the activities that contribute to the calculation of the objective's measure.

The following XML code segment shows an example of assessment parameters specified in a parent activity 'Part_3_Safety_Exercises', which has the following child activities: 'High_Pressure_System', 'Low_Pressure_System', 'Breathing_Circuit', 'Manual_Ventilation', 'Mechanical_Ventilation', and 'Scavenging_System'. In the parent activity, there are objectives 'oOverall', 'oPressureSystem', 'oBreathingCircuit', 'oVentilation', and 'oScavengingSystem' whose measures will be assigned as the result of an assessment conducted in the activity:

```
.......
<ns1:AssessmentParameters>
   <ns1:PostAssessmentParameters>
      <ns1:AssessmentSelectionParameter ActivityID="Part_3_Safety_Exercises"
            ScoringWeight="1" SelectionPercentage="0.5" />
      <ns1:AssessmentSelectionParameter ActivityID="High_Pressure_System"
            ScoringWeight="1" SelectionPercentage="0.5" />
      <ns1:AssessmentSelectionParameter ActivityID="Low_Pressure_System"
            ScoringWeight="1" SelectionPercentage="0.5" />
      <ns1:AssessmentSelectionParameter ActivityID="Breathing_Circuit"
            ScoringWeight="1" SelectionPercentage="0.5" />
      <ns1:AssessmentSelectionParameter ActivityID="Manual_Ventilation"
            ScoringWeight="1" SelectionPercentage="0.5" />
      <ns1:AssessmentSelectionParameter ActivityID="Mechanical_Ventilation"
            ScoringWeight="1" SelectionPercentage="0.5" />
      <ns1:AssessmentSelectionParameter ActivityID="Scavenging_System"
            ScoringWeight="1" SelectionPercentage="0.5" />
   </ns1:PostAssessmentParameters>
   <ns1:PostAssessmentObjectiveSettings>
       <ns1:ObjectiveSetting ObjectiveID="oOverall">
          <ns1:ActivityID>Part_3_Safety_Exercises</ns1:ActivityID>
          <ns1:ActivityID>High_Pressure_System</ns1:ActivityID>
          <ns1:ActivityID>Low_Pressure_System</ns1:ActivityID>
          <ns1:ActivityID>Breathing_Circuit</ns1:ActivityID>
          <ns1:ActivityID>Manual_Ventilation</ns1:ActivityID>
          <ns1:ActivityID>Mechanical_Ventilation</ns1:ActivityID>
          <ns1:ActivityID>Scavenging_System</ns1:ActivityID>
       </ns1:ObjectiveSetting>
       <ns1:ObjectiveSetting ObjectiveID="oPressureSystem">
          <ns1:ActivityID>High_Pressure_System</ns1:ActivityID>
          <ns1:ActivityID>Low_Pressure_System</ns1:ActivityID>
       </ns1:ObjectiveSetting>
       <ns1:ObjectiveSetting ObjectiveID="oBreathingCircuit">
          <ns1:ActivityID>Breathing_Circuit</ns1:ActivityID>
       </ns1:ObjectiveSetting>
       <ns1:ObjectiveSetting ObjectiveID="oVentilation">
          <ns1:ActivityID>Manual_Ventilation</ns1:ActivityID>
          <ns1:ActivityID>Mechanical_Ventilation</ns1:ActivityID>
       </ns1:ObjectiveSetting>
       <ns1:ObjectiveSetting ObjectiveID="oScavengingSystem">
          <ns1:ActivityID>Scavenging_System</ns1:ActivityID>
       </ns1:ObjectiveSetting>
    </ns1:PostAssessmentObjectiveSettings>
 </ns1:AssessmentParameters>.......
```

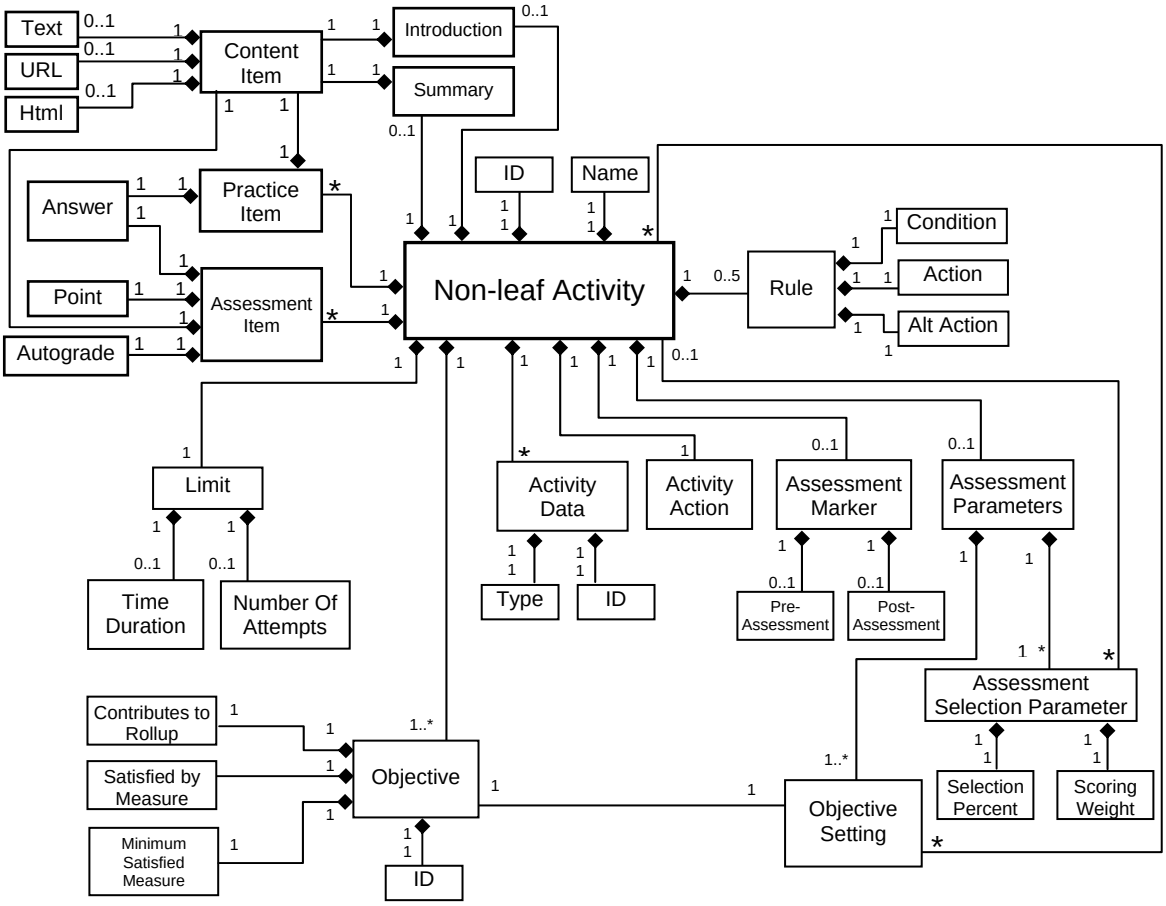


Figure 4-3. Entities in the Meta-model of a Non-leaf Activity in Learning Process Definition Language

As discussed, a condition-action rule has the structure "*If [Condition], then [Action], else [Alt-Action]*". Thus, the *Rule* entity consists of a *Condition*, an *Action*, and an *Alternative Action*. *Condition* is in the form of a Boolean expression, and *Action* and *Alternative Action* are Java code segments to be executed when the *Condition* is satisfied or unsatisfied, respectively. Sequencing rules for a non-leaf activity include 'Pre-activity', 'After-Pre-Assessment', 'Drill-down', 'Roll-up', and 'After-Post-Assessment' rules. Figure 4-3 depicts a detailed meta-model of a Non-leaf Activity entity with its related entities.

4.4.2 Leaf Activity

A leaf activity is a leaf node in an activity tree. Unlike a non-leaf activity, which includes learning items in it, a leaf activity binds to an LO at runtime either statically or dynamically. *Binding Info* specifies the method of LO binding and the required information for each method. In the case of static binding, a business service key and a business interface key is specified. Dynamic binding involves the requirements of LO that the activity needs to bind, which are specified in the same format as the meta-data of LOs. Implementation details on the registration and dynamic binding mechanism will be given in Chapter 5. In the following example of *Binding Info*, it specifies the information necessary for dynamic binding, where the attribute *Keywords* should be “input output of anesthesia machine” and the attribute *Language* should be set to the value of activity data ‘sLanguage’ at the time of binding:

```
.......
<ns1:BindingInfo Mode="Dynamic">
   <ns1:DynamicBindingInfo>
      <ns1:constraints>
         <ns1:service_constraints>
            <ns1:service_constraint>
               <ns1:name>Keywords</ns1:name>
               <ns1:type>string</ns1:type>
               <ns1:keyword>eq</ns1:keyword>
               <ns1:valueList>"input output of anesthesia machine"
               </ns1:valueList>
               <ns1:negotiable>No</ns1:negotiable>
               <ns1:priority>0</ns1:priority>
            </ns1:service_constraint>
            <ns1:service_constraint>
               <ns1:name>Language</ns1:name>
               <ns1:type>string</ns1:type>
               <ns1:keyword>eq</ns1:keyword>
               <ns1:valueList>"VAR:sLanguage"</ns1:valueList>
               <ns1:negotiable>No</ns1:negotiable>
               <ns1:priority>0</ns1:priority>
            </ns1:service_constraint>
         </ns1:service_constraints>
      </ns1:constraints>
   </ns1:DynamicBindingInfo>
</ns1:BindingInfo>.......
```

Table 4-2 Execution Stages and Description of Activity Actions in Leaf Activity

Activity Action Name	**Execution Stage**	**Description**
Before-Pre-Activity	Before processing *Pre-Activity* rules	Assign values to activity data to be used for the execution of *Pre-Activity* rules.
Before-Binding	Before binding to a learning object	Assign values to activity data that can be referenced for updating dynamic binding constraints.
Before-Pre-Assessment	Before conducting a pre-assessment	Executed to take any necessary action for the preparation of pre-assessment.
Before-Content	Before processing *After-Pre-Assessment* rules, i.e. before presenting Introduction content item	Assign values to activity data to be used for the execution of *After-Pre-Assessment* rules.
Before-Post-Assessment	Before conducting a post-assessment	Executed to take any necessary action for the preparation of post-assessment.
After-Post-Assessment	Before processing *After-Post-Assessment* rules	Assign values to activity data to be used for the execution of *After-Post-Assessment* rules.

In a leaf activity, there are the following activity actions: ‘Before-Pre-Activity Actions’, ‘Before-Binding Actions’, ‘Before-Pre-Assessment Actions’, ‘Before-Content Actions’, ‘Before-Post-Assessment Actions’, and ‘After-Post-Assessment Actions’. Table 4-2 describes the execution stages and the purposes of the activity actions. A leaf activity specifies Pre-activity, After-Pre-Assessment, and After-Post-Assessment sequencing rules.

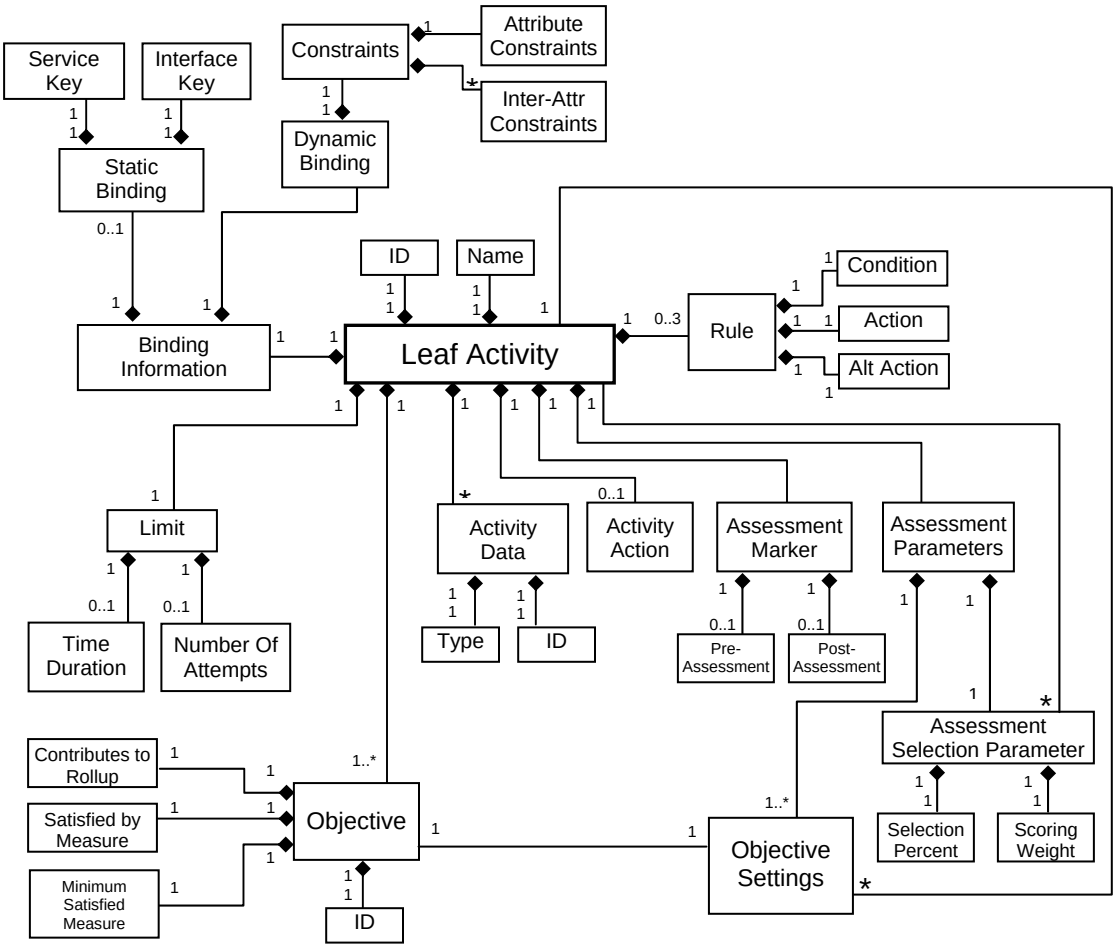


Figure 4-4. Meta-model of Leaf Activity and Related Entities in the Learning Process Definition Language

The definitions of the *Limit*, *Activity Data*, *Objective*, *Assessment Marker*, *Assessment Parameters* entities in Leaf Activities are identical to those in Non-leaf Activities. Figure 4-4 shows the detailed meta-model of Leaf Activity and its related entities.

4.4.3 Connector

A *connector* relates a parent activity and its child activities in a cluster. *Sequencing Control Modes* of the cluster are specified in a connector. Additionally, it can include an ordered list of child activities, which gives the processing sequence of these child

activities in the *Forward Only* and *Flow* sequencing control modes. Figure 4-5 shows the detailed meta-model for *connector*, *edge*, and their related entities.

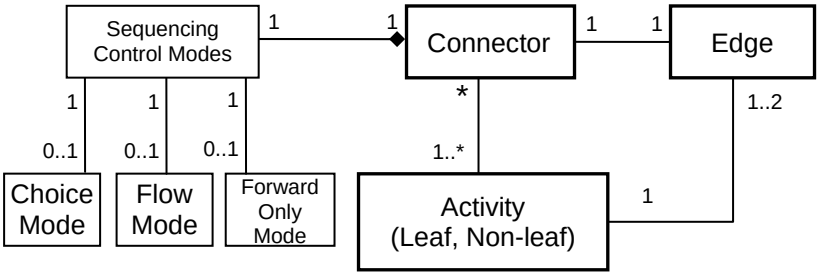


Figure 4-5 Meta-model of Connector and Edge in the Learning Process Definition Language

The following XML code defines two connectors having the 'ForwardOnly' and 'Flow' sequencing control modes, respectively:

```
.......
<ns1:Connector Id="connector1" SequencingControlMode="ForwardOnly">
   <ns1:SiblingActivityID>Part_1_Basic_Concepts</ns1:SiblingActivityID>
   <ns1:SiblingActivityID>Part_2_IO_of_VAM</ns1:SiblingActivityID>
   <ns1:SiblingActivityID>Part_3_Safety_Exercises</ns1:SiblingActivityID>
</ns1:Connector>
<ns1:Connector Id="connector2" SequencingControlMode="Flow">
   <ns1:SiblingActivityID>High_Pressure_System</ns1:SiblingActivityID>
   <ns1:SiblingActivityID>Low_Pressure_System</ns1:SiblingActivityID>
   <ns1:SiblingActivityID>Breathing_Circuit</ns1:SiblingActivityID>
   <ns1:SiblingActivityID>Manual_Ventilation</ns1:SiblingActivityID>
   <ns1:SiblingActivityID>Mechanical_Ventilation</ns1:SiblingActivityID>
   <ns1:SiblingActivityID>Scavenging_System</ns1:SiblingActivityID>
 </ns1:Connector>.......
```

4.4.4 Edge

An *Edge* is a directed edge, which connects an activity to a connector or a connector to an activity. It has a 'from' attribute and a 'to' attribute. The value of each attribute can be either an activity ID or a connector ID. The meta-model of Edge is shown in Figure 4-5. The next XML code defines edges that connect an activity to a connector, and vice versa:

```
 .......
 <ns1:Edge To="connector1" Id="VAM::connector1" From="VAM" />
 <ns1:Edge To="Part_1_Basic_Concepts"
   Id="connector1::Part_1_Basic_Concepts" From="connector1" />
 <ns1:Edge To="Part_2_IO_of_VAM" Id="connector1::Part_2_IO_of_VAM"
   From="connector1" />
 <ns1:Edge To="Part_3_Safety_Exercises"
   Id="connector1::Part_3_Safety_Exercises" From="connector1" />
```

```
<ns1:Edge To="connector2" Id="Part_3_Safety_Exercises::connector2"
  From="Part_3_Safety_Exercises" />
<ns1:Edge To="High_Pressure_System"
  Id="connector2::High_Pressure_System" From="connector2" />
<ns1:Edge To="Low_Pressure_System" Id="connector2::Low_Pressure_System"
  From="connector2" />
<ns1:Edge To="Breathing_Circuit" Id="connector2::Breathing_Circuit"
  From="connector2" />
<ns1:Edge To="Manual_Ventilation" Id="connector2::Manual_Ventilation"
  From="connector2" />
<ns1:Edge To="Mechanical_Ventilation"
  Id="connector2::Mechanical_Ventilation" From="connector2" />
<ns1:Edge To="Scavenging_System" Id="connector2::Scavenging_System"
  From="connector2" /> .......
```

CHAPTER 5
E-LEARNING SERVICE INFRASTRUCTURE

We envision a new type of virtual e-learning community that can be established based on reusable, assessable, and programmatically accessible e-learning objects that are created and stored in a distributed fashion over the Internet.

5.1 E-Learning Infrastructure for Virtual E-Learning Communities

The e-learning infrastructure provides a dynamic and collaborative e-learning environment in which a large number of virtual collaborative e-learning communities can be dynamically established and evolved. Each collaborative community is dynamically formed by a community of Internet users who have a common interest in a specific subject of learning, and who can provide and/or use the knowledge for solving problems in that subject domain.

5.1.1 Roles in a Virtual E-learning Community

Members of a virtual e-learning community can play the following roles: *content provider*, *content composer*, *content evaluator*, *content learner*, and *community host administrator*. A member can play one or more of these roles.

Content providers create and register ALOs. Since an ALO is an aggregation of learning items (i.e., content, practice and assessment items), which make references to learning assets, a content provider can either take advantage of the learning assets available in the Internet, or produce learning assets by using appropriate authoring tools to build ALOs. HTML authoring tools, audio and movie editing tools, Macromedia Director, and so on, can be used to create learning assets. The constructed ALOs are stored in the provider's

local repository, maintained by the Apache Xindice XML DBMS. Each ALO, together with its meta-data and business service key, is registered as a Web-service in the Constraints-based LO Broker deployed at a *community host*'s site for later discovery and delivery. Community members, and the Learning Process Management System can browse and query the LO Broker for LOs. The LO Broker performs constraints matching to discover the best LO that matches with a learner's request and profile.

Content composers' responsibility is to compose and register CLOs by making use of the CLO authoring tool. Because non-leaf activities can include learning items, they also need to be able to access learning assets as content providers. They can work with content providers, or take the role of a content provider. Constructed CLOs are stored in the content composer's local repository. The meta-data and access information of the CLO is again registered with the LO Broker and made accessible and reusable to other community members.

Content evaluators, elected by a virtual e-learning community, would review, evaluate and rate the ALOs and CLOs that are registered with the LO Broker based on learners' feedback and LOs' use statistics, and, together with the administrator, approve or disapprove their inclusion for the community's use, and their modification and removal. It is necessary that content providers, content composers, and content evaluators are domain experts in the subject area of LOs that they produce and evaluate.

Administrators of the community host are responsible for the operation and maintenance of the Learning Process Management System installed at the host site. Additionally, they take care of the registration and enforcement of the community's policy rules, which govern LO registrations with the LO Broker. Policy rules can

recommend to members the LOs that have high evaluation scores, or drop LOs from the registry that are rarely used or have very low rating.

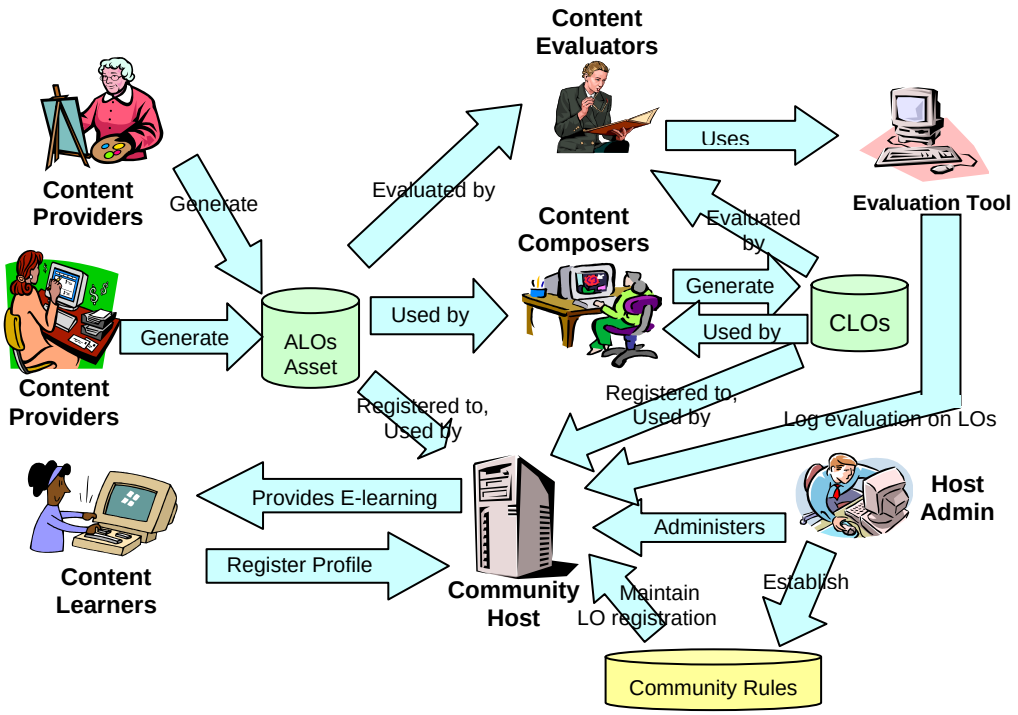


Figure 5-1. Roles and Relationships among Members of a Virtual E-Learning Community

Content learners are members of a virtual e-learning community who use the LOs created by the virtual community. During the learner registration, they are required to provide information that constitute learner profiles, which are used by the Learning Process Management System to achieve adaptive and customizable delivery of learning experience. In addition to learning from LOs, they are authorized to evaluate the LOs that have been delivered to them. They make use of the Web User Interface provided by the community host. The roles and relationships among members of a virtual e-learning community are depicted in Figure 5-1.

5.1.2 Web-Services-Based E-Learning Service Infrastructure

The infrastructure is composed of distributed software components including the LO authoring tools, the distributed LO repositories, the Learning Process Management System, the LO Broker, and Web Servers. Each network site of a virtual community has a Web server to provide HTTP and SOAP Web-services. LO repositories are stored at the sites where they are created to avoid the communication/storage bottleneck of a centralized repository and to give more authority to content providers and content composers in their creation, modification and removal of LOs.

Different software components are installed at different sites depending on the roles that the members of a virtual community play. The *Content Provider site* is equipped with the ALO authoring tool and other learning asset authoring tools. It also has an XML DBMS to store and maintain the local LO repository. The *Content Composer site* has a CLO authoring tool in addition to the facilities available to a content provider. Using the LO authoring tools, content composers and providers store LOs in their local LO repositories, and register them with the Constraints-based LO Broker installed at the host site. The *Community Host* site has the Learning Process Management System, the Constraints-based LO Broker and a local LO repository. The local LO repository maintains all the data necessary for the execution and delivery of LOs to learners, that is, the learner profile models, the runtime status models and LOs that are fetched from the distributed repositories at the sites of content providers and composers for processing by the Learning Process Management System. The *Content Learner site* needs only a Web browser. A learner can browse or query the LO Broker for registered LOs by specifying his/her learning requirements in terms of the meta-data and constraints associated with a desired LO. The learner can then store the LOs of his/her choice in a learner space, called

MyCLOs, for later execution. He/she can execute the LOs in his/her learner space. The facilities at different sites of the web-service-based e-learning service infrastructure are shown in Figure 5-2. The implementation details of the infrastructure are discussed in the next section.

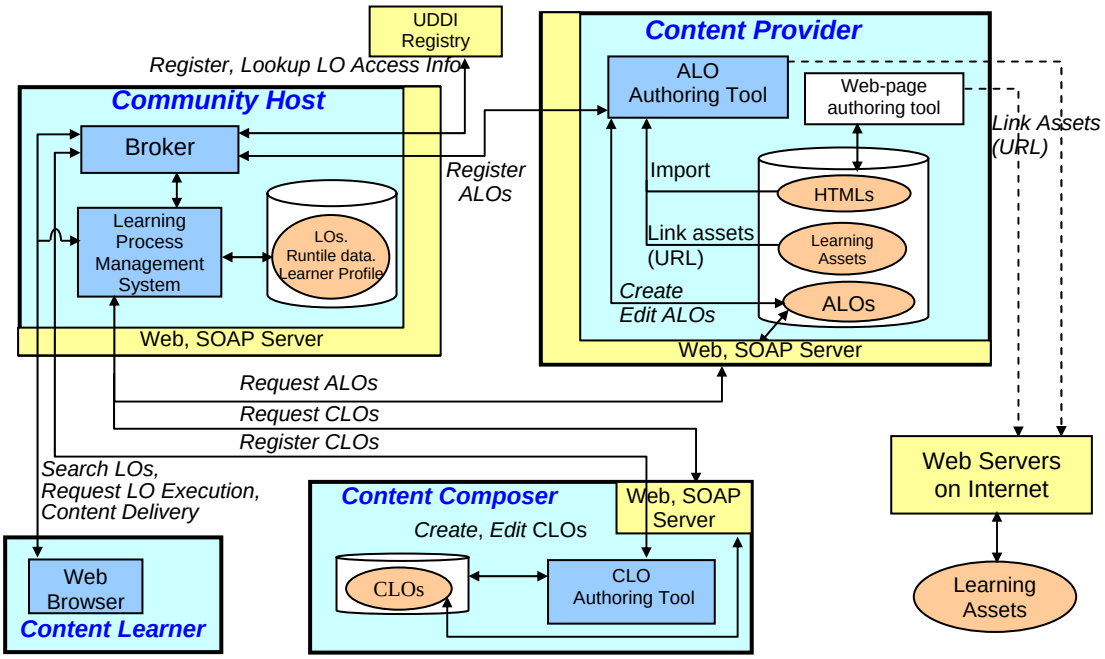


Figure 5-2. Web-service-based E-Learning Service Infrastructure

5.2 Implementation

E-learning services enable the authoring, registration, delivery and management of LOs. They are needed for the creation of virtual e-learning communities. In order to provide these e-learning services, a service framework that facilitates the authoring, registration, discovery, binding, and invocation of LOs is necessary. In addition, a uniform way of modeling multimedia learning assets and Web-based service components in the form of objects is needed. The Web-services technology developed by the IT industry can be adapted and extended to serve this purpose.

5.2.1 Constraints-enabled Web-service Model

The Universal Description, Discovery and Integration (UDDI) [OASIS, 2004] provides a general framework to allow objects to be defined as Web-services using the Web Service Description Language (WSDL). The major elements of WSDL used for Web service description are as follows [W3C, 2001]. The relationships among these elements are illustrated in Figure 5-3:

- **types** – a container for data type definitions using some type system (such as XSD).
- **message** – an abstract, typed definition of the data being communicated. It consists of the data types defined in 'types' elements.
- **operation** – an abstract description of an action supported by the service. Input / output parameters are defined using the messages defined in 'message' elements.
- **portType** – an abstract set of operations supported by one or more endpoints.
- **binding** – a concrete protocol and data format specification for a particular port type.
- **port** – a single endpoint defined as a combination of a binding and a network address.
- **service** – a collection of related endpoints.

A WSDL document can be posted on the Internet, and its access point (URL) together with a textual description and the category information can be registered with a Web-services registry such as the UDDI registry. The contents of the registry can then be navigated or searched manually or programmatically to discover and obtain the access information of suitable Web-services. As an alternative to downloading identified objects for local use, the Web-services model allows the runtime binding of a service request to a remote Web-service, and provides the Simple Object Access Protocol (SOAP) [W3C, 2003] that allows the activation of a remote service by exchanging SOAP XML messages. By applying the Web-service technology, LOs and other instructional resources, tools,

and software systems can be uniformly specified as Web-services. However, the existing specification and implementation of UDDI does not provide constraints specification and filtering capabilities to match service constraints with a service requestor's requirements to identify suitable services. Without these constraints specification and filtering capabilities, an LO can be selected by the registry, but the selected LO may not satisfy the requestor's requirements.

In this work, we have extended the WSDL specification and the UDDI registry's capabilities by allowing constraints to be specified in WSDL, and adding constraint processing capability to the UDDI registry. In the extended WSDL, a *constraints* element that describes the meta-data of a Web-service is written using the LO meta-data format, and appended as a part of the Web-service description. The extended WSDL information model is shown in Figure 5-3.

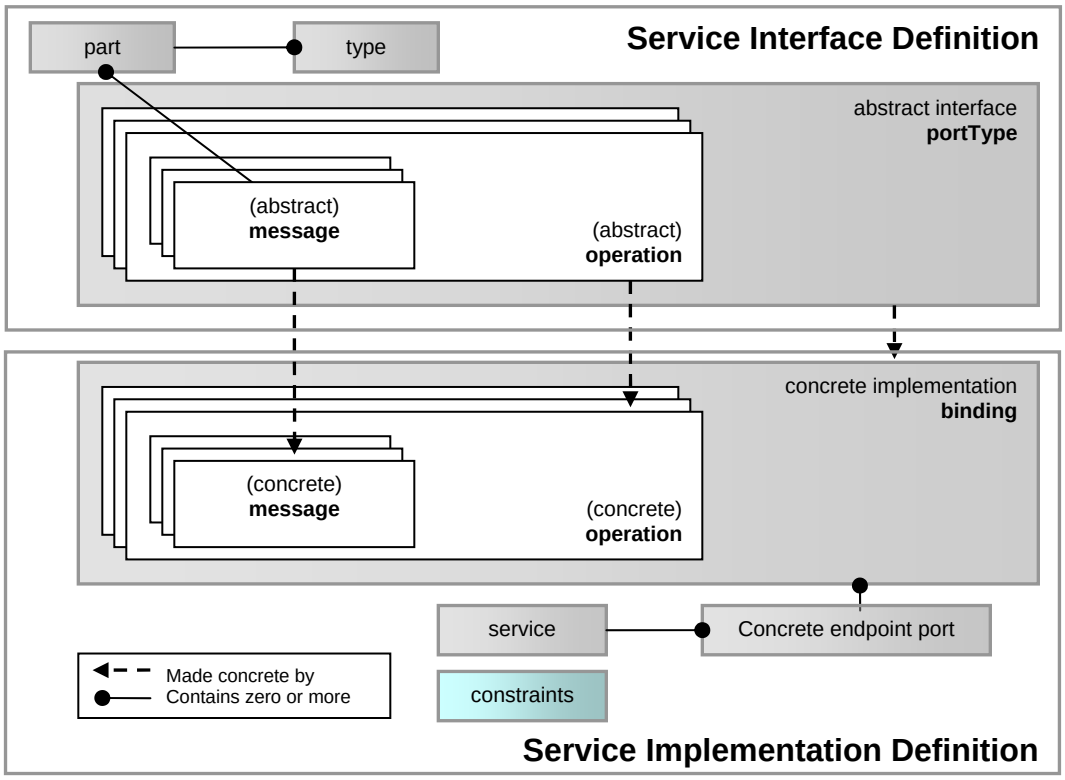


Figure 5-3. Extended WSDL Information Model (In part from [Graham, et. al. 2002])

Figure 5-4 shows a part of an extended WSDL of an LO that includes an operation *getLO*, the access point of which is http://128.227.176.41:8888/axis/LODBService.jws. The meta-data attribute '*Keywords*' of the LO has the enumerated values: "Anesthesia Machine", "Anesthesia", "Virtual Anesthesia Machine", "Simulation", and "Low Pressure System", the attribute '*Media_Format*' has the enumerated values "HTML" and "Shockwave", and the attributed '*Cost*' has 0 as its value.

```
<?xml version="1.0" encoding="UTF-8" ?>
<wsdl:definitions targetNamespace="urn:LODBService"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="urn:LODBService"
xmlns:intf="urn:LODBService"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <wsdl:message name="getLORequest">
        <wsdl:part name="sLOId" type="xsd:string" />
        <wsdl:part name="sLOType" type="xsd:string" />
        <wsdl:part name="sLOPart" type="xsd:string" />
    </wsdl:message>
        <wsdl:message name="getLOResponse">
        <wsdl:part name="getLOReturn" type="xsd:string" />
    </wsdl:message>
    <wsdl:portType name="LODBService">
        <wsdl:operation name="getLO" parameterOrder="sLOId sLOType sLOPart">
            <wsdl:input message="intf:getLORequest" name="getLORequest" />
            <wsdl:output message="intf:getLOResponse" name="getLOResponse" />
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="LODBServiceSoapBinding" type="intf:LODBService">
        <wsdlsoap:binding style="rpc"
        transport="http://schemas.xmlsoap.org/soap/http" />
        <wsdl:operation name="getLO">
            <wsdlsoap:operation soapAction="" />
            <wsdl:input name="getLORequest">
                <wsdlsoap:body
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:LODBService" use="encoded" />
            </wsdl:input>
            <wsdl:output name="getLOResponse">
                <wsdlsoap:body
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:LODBService" use="encoded" />
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="LODBServiceService">
        <wsdl:port binding="intf:LODBServiceSoapBinding" name="LODBService">
            <wsdlsoap:address
            location="http://128.227.176.41:8888/axis/LODBService.jws" />
```

```
        </wsdl:port>
    </wsdl:service>
    <constraints>
        <service_constraints>
            <service_constraint>
                <name>Keywords</name>
                <type>string</type>
                <keyword>enu</keyword>
                <valueList>{"Anesthesia Machine", "Anesthesia", "Virtual Anesthesia
                Machine", "Simulation", "Low Pressure System", "High Pressure
                System", "Breathing Circuit", "Manual Ventilation", "Mechanical
                Ventilation", "Scavenging System"}</valueList>
                <negotiable>No</negotiable>
                <priority>0</priority>
            </service_constraint>
            <service_constraint>
                <name>Media_Format</name>
                <type>string</type>
                <keyword>enu</keyword>
                <valueList>{"HTML", "Shockwave"}</valueList>
                <negotiable>No</negotiable>
                <priority>0</priority>
            </service_constraint>
            <service_constraint>
                <name>Cost</name>
                <type>float</type>
                <keyword>eq</keyword>
                <valueList>0.0</valueList>
                <negotiable>No</negotiable>
                <priority>0</priority>
            </service_constraint>
        </service_constraints>
        <interattribute_constraints />
    </constraints>
</wsdl:definitions>
```

Figure 5-4. Extended WSDL Document Example

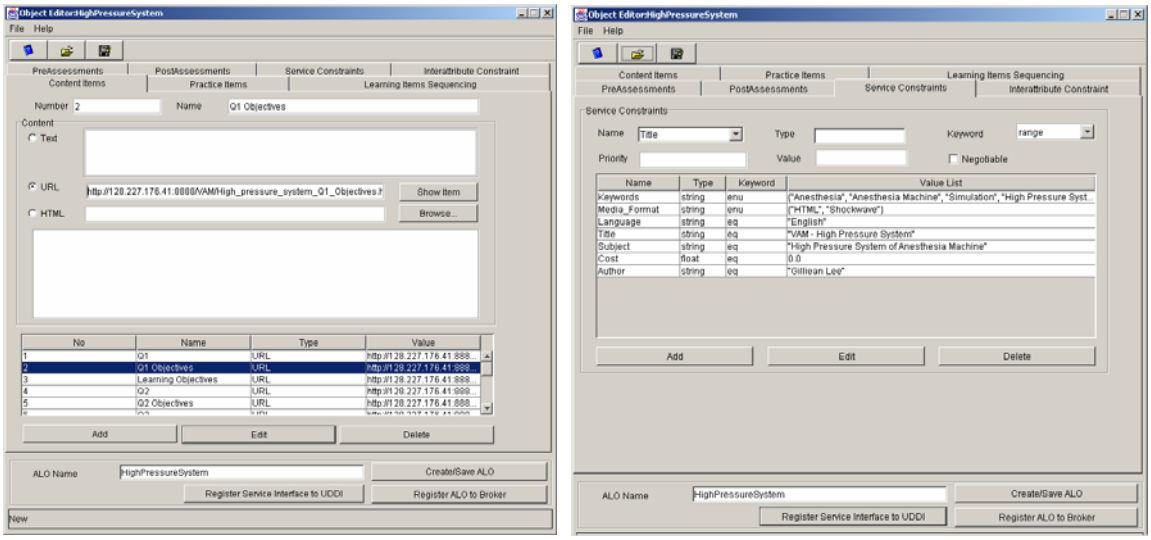
A Constraints-based Web service Registry [Degwekar, Su & Lam, 2004], which stores constraints and performs constraints matching to select suitable Web-services, is utilized as a constraints-enabled Web-services registry. The registry leverages Apache jUDDI registry (UDDI v.2.0) [The Apache Software Foundation, 2003] as a back-end to acquire globally unique registration keys and store access points of LO repositories where LOs are located. Since we make use of the registry as an LO registry that matches LOs with browse requests, we shall refer it as the ‘Constraints-based LO Broker’.

LOs registered as Web services are accessible by calling Web-service operations of the LO repositories (*getLO*, *addLO*, etc) deployed at the access points of the Web-

services. The registry serves as a *meta-data repository* in contrast to LO repositories. More about the Constraints-based Registry will be discussed in a later section.

5.2.2 Authoring Tools & LO Repository

Authoring tools with graphical user interfaces have been implemented to help authors compose ALOs and CLOs efficiently and effectively. We make use of the Java Architecture for XML Binding (JAXB) [Sun Microsystems, 2003] technology to implement XML parsing and updating in our authoring tools. The JAXB package generates a set of Java class files using an XML Schema definition as an input. The use of JAXB package and the generated Java class files greatly simplifies the parsing, updating, writing, and validation of XML files, which conform to the XML Schema definition provided as an input.



A B

Figure 5-5. Graphical User Interface Frame of the ALO Authoring Tool. A) Content Items Authoring Tab. B) Service Constraints (Meta-data) Tab.

5.2.2.1 ALO authoring tool

We have developed an ALO authoring tool that facilitates authoring and registration of ALOs. It is developed using Java Swing components. Its user interface is

similar to a tabbed dialog box in a windows-based operating system. The tool is used to add, delete, and edit learning items, and to create, edit and register ALOs. The user interface of the ALO authoring tool is shown in Figure 5-5.

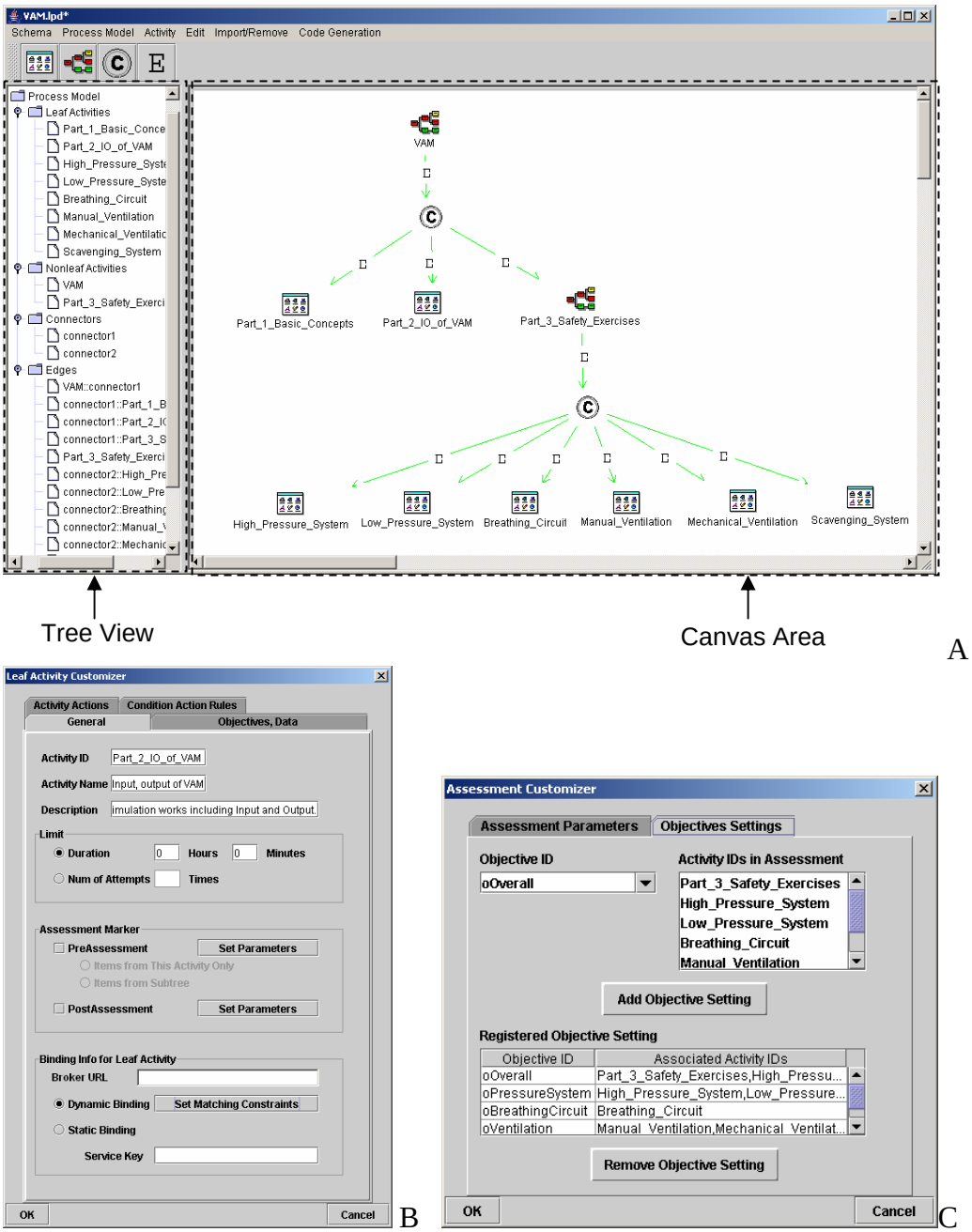


Figure 5-6. Graphical User Interface Frame of the CLO Authoring Tool. A) User Interface Frame with Tree View and Canvas Area. B) Leaf Activity Dialog Box. C) Assessment Dialog Box.

5.2.2.2 CLO authoring tool

The CLO authoring tool supports the authoring and registration of CLOs. As shown in Figure 5-6 (A), the tool's frame provides a *canvas area* on the right side for designing an activity tree graphically, and a *tree view* on the left side of the frame where modeling constructs are categorized and presented. Using the CLO authoring tool, a content composer can create, edit and register CLOs. Dialog boxes for editing the properties of the modeling constructs such as activities, connectors and edges are also provided. Functions such as 'Add', 'Delete' and 'Edit' learning items are provided for defining learning items in non-leaf activities. Activity and Assessment dialog boxes are shown in Figure 5-6 (B) and (C).

5.2.2.3 LO repository

The ALO and CLO authoring tools store and retrieve LOs from an LO repository stored in an XML database managed by the Web-based native XML DBMS, Xindice [The Apache Software Foundation, 2004]. The Learning Process Management System also accesses LOs from distributed LO repositories for processing. For interoperability and ease of access, each LO repository is implemented as a Web-service. The Java Web Service (JWS) technology supported by the Apache Axis SOAP engine is used for the deployment of the Web-service of the LO repository. Any part of an LO, such as meta-data, contents items, assessment items and practice items, can be accessed from the LO repository. The WSDL document that describes the LO repository's Web-service is given in Appendix D. Details about the LO registration process with the Constraints-based LO Broker and the accessing of LOs from LO repositories will be discussed later.

5.2.3 Learning Process Management System

LOs are registered, discovered and processed by the Learning Process Management System, which consists of the following main components: a Learning Process Execution Engine (LPEE), an Event-Trigger-Rule (ETR) Server, an Assessment Component, an LO Broker, a local LO Repository, and a User Interface Component. The architecture of Learning Process Management System is illustrated in Figure 5-7. The components of the system are discussed in the following sections.

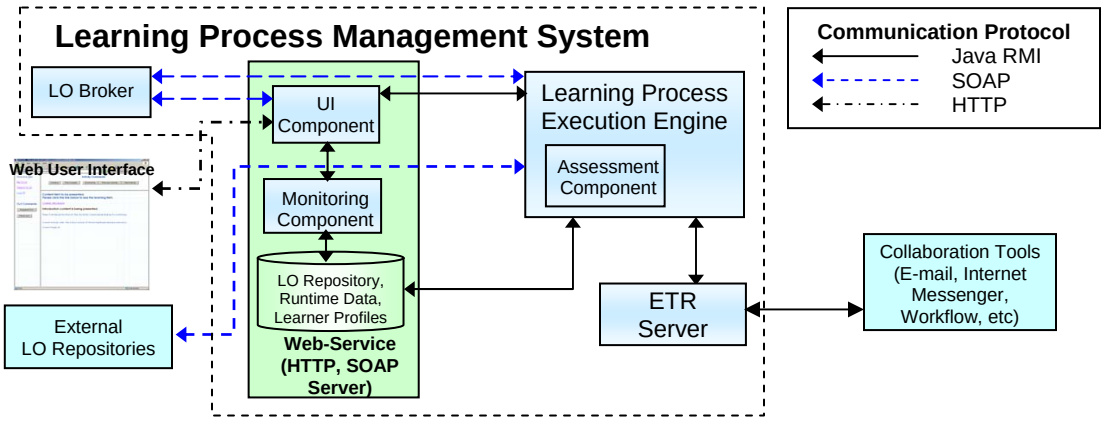


Figure 5-7. Architecture of Learning Process Management System

5.2.3.1 Learning process execution engine

The LPEE processes LOs using the rule-based execution model to facilitate active, adaptive, flexible and customizable e-learning service. When a learner requests the execution of an ALO, a predefined CLO having a single leaf activity is instantiated and the leaf activity is bound to the ALO. When the execution of a CLO is requested, LPEE fetches the requested CLO from a repository that stores the CLO by making a Web-service call, and then deploys the CLO by generating activity codes and installing events, triggers and rules specified in the CLO in the Event-Trigger-Rule Server. Next, an instance of CLO is created for execution. A runtime status model, described in an XML document, is created and maintained in a local LO repository stored in an XML database.

- Runtime Status Model

The model keeps the following information necessary for the execution of a CLO:

  - Status of the instance: started, suspended, and finished.
  - Current activity id.
  - Status of activities: unattempted, attempted, and completed.
  - Status of learning objectives: unknown, satisfied, unsatisfied, and current measure.
  - Status of activity attempts.

Upon the creation of a CLO instance, a runtime status model described in an XML document, and a temporary copy of the CLO are created and maintained in the local LO repository. The local LO repository is built on top of Xindice XML database, which provides the persistent storage. LPEE maintains the runtime status model document for the CLO instance. Upon receiving sequencing commands, the states of activities and the instance are recorded in the runtime status model according to the state transition diagrams shown in Figure 5-8. The runtime status model is defined in an XML Schema shown in Appendix E.

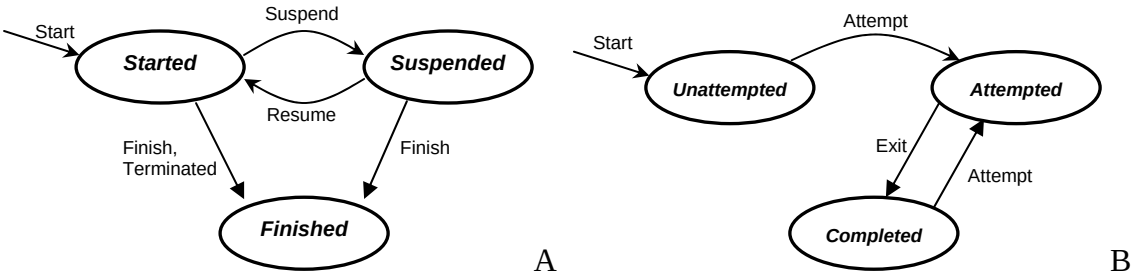


Figure 5-8. State Transition Diagrams for Learning Process and Activity. A) Process Instance Status Transition Diagram B) Activity Status Transition Diagram

A *monitoring component* makes use of the runtime status model to provide the status information about the CLO instance to a learner. A *UI component* invokes the

monitoring component to create a Web page that presents the status information along with the activity tree of the CLO instance.

- Architecture of LPEE

A learning process instance handler object (*LPInstance Handler*) takes care of the execution of a learning process (CLO) instance by managing the execution of its activities and maintaining its runtime status model through an Activity Handler and the LO Repository Access Module. An *Activity Handler* object is responsible for the execution of an activity. It instantiates the activity code object of an activity and invokes its appropriate methods in accordance with the stage of execution. Some of the methods post events to the ETR Server to invoke sequencing rules that are installed when the CLO is deployed. The parsed structure of the CLO definition is maintained so that the LO specification can be retrieved rapidly. The runtime objects of a CLO instance and their relationships with a learning object definition are illustrated in Figure 5-9.

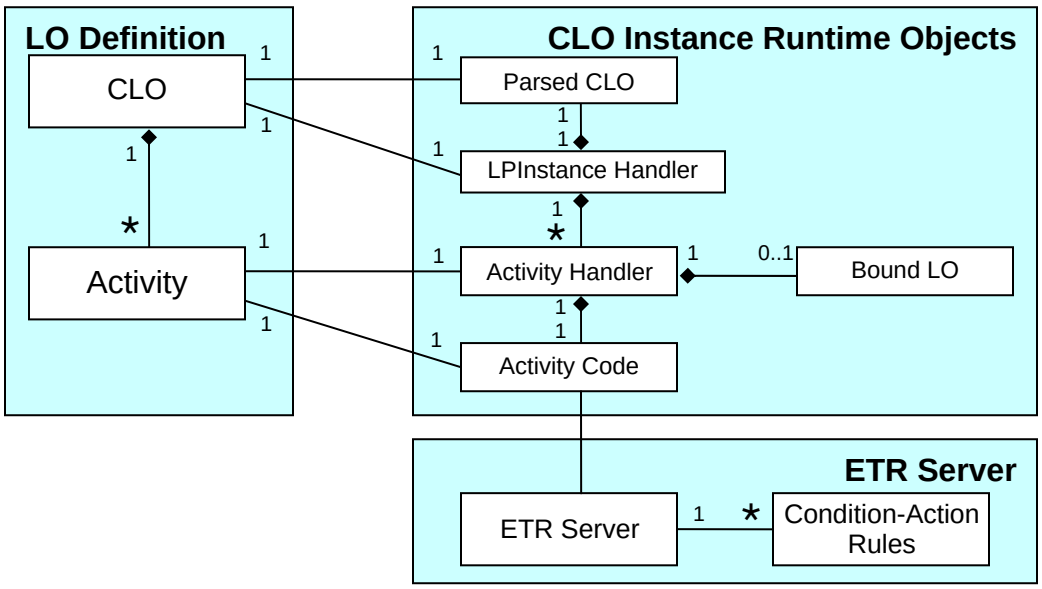


Figure 5-9. Relationships between Learning Object Definition and Runtime Objects of a CLO Instance

The *LMS Engine Core* object implements the core functionality of LPEE. The runtime objects, including *activity codes*, *parsed CLOs*, *LPInstance Handlers*, and *Activity Handlers*, are maintained in four hash tables. Using the CLO instance ID or a combination the instance ID and an activity ID, the runtime objects of a CLO instance can be retrieved from the hash tables. LPEE is encapsulated as a Remote Method Invocation (RMI) server accessible to sequencing rules and the UI component.

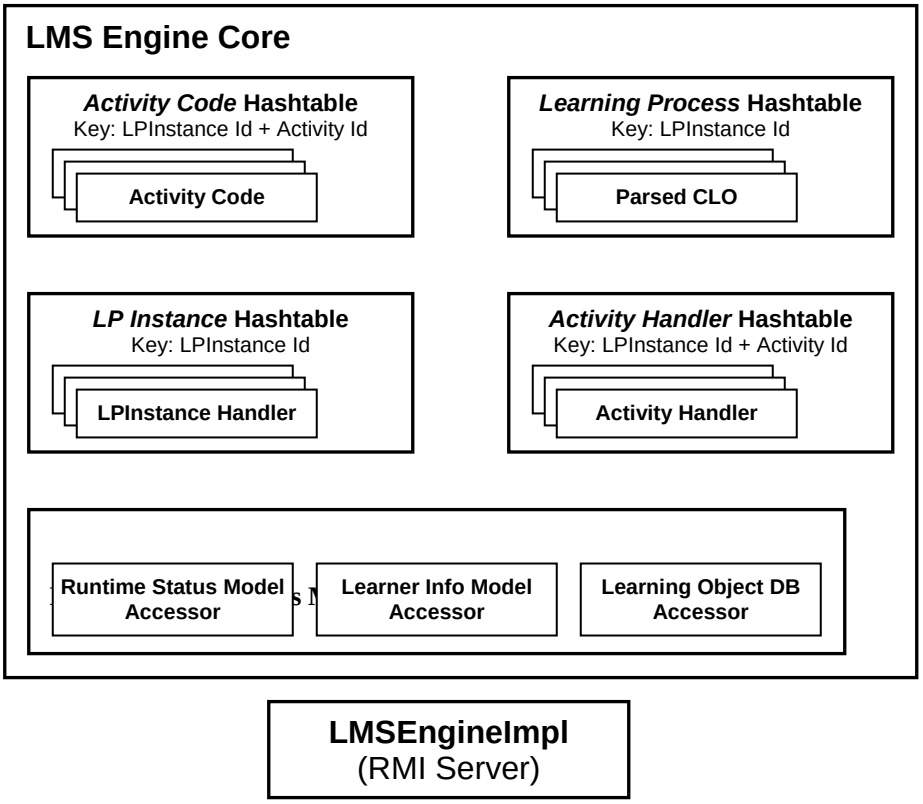


Figure 5-10. Architecture of the Learning Process Execution Engine

The *LO Repository Access Module* consists of three classes that access the Xindice XML database: *Runtime Status Model Accessor*, *Learner Information Model Accessor*, and *Learning Object DB Accessor*. These classes make use of XPath and XUpdate technologies to query and update the XML database. The *Runtime Status Model Accessor* class is responsible for the management of runtime status models. The *Learner*

Information Model Accessor class queries and updates the learner profile models. The *Learning Object DB Accessor* class stores temporary LOs and accesses LO specifications. The architecture of LPEE is shown in Figure 5-10.

- LO Deployment Process

A CLO has to be deployed before an instance of the CLO can be instantiated. The deployment process involves the generation of activity codes for all the activities of the CLO, and the installation of sequencing rules, which involves code generation for events, triggers, and rules, and installing the code in the ETR Server. During the deployment process, the *installProcessInstance* method of the LMS Engine Core object invokes the *deployCLO* method of CLO RuntimeDB Access object. The *deployCLO* method checks whether the CLO has to be deployed by comparing the CLO's last deployment time and the CLO's update time. If the CLO is deployed for the first time, the *deployCLO* method will generate code for events, triggers, rules, and activities that are necessary for the execution of the CLO instance. If the CLO has been deployed before and the update time is later than the deployment time, then some of the activities of the CLO have to be deployed to reflect the update. The deployment method deploys only the activities whose update time is later than the deployment time by generating and installing codes for the events, triggers and rules associated with the updated activities and the codes for the updated activities. An object diagram of the objects related to the deployment process is shown in Figure 5-11.

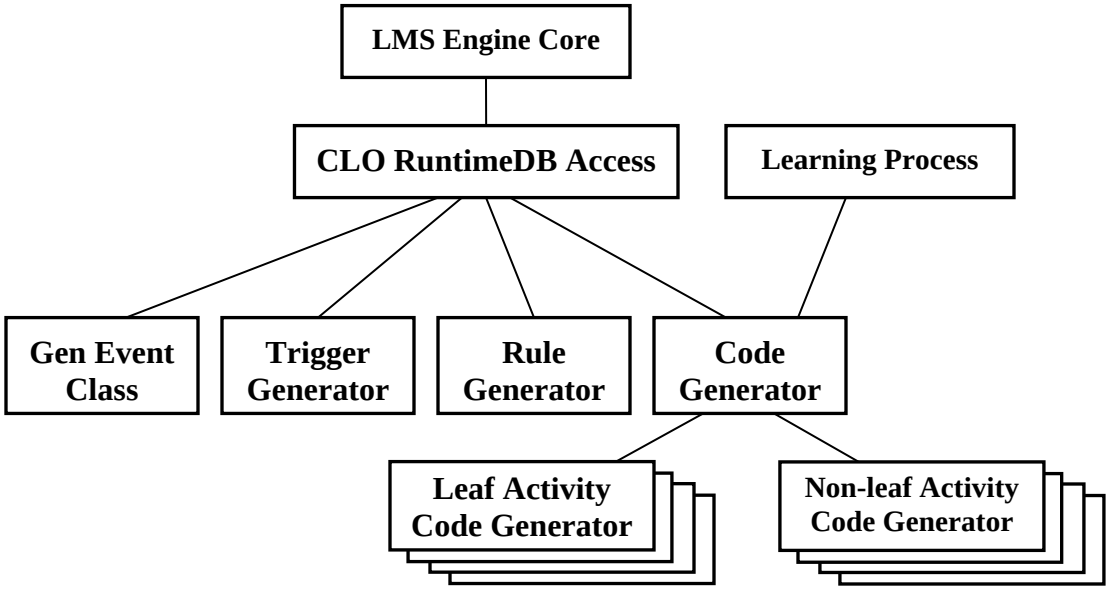


Figure 5-11. Object Diagram of CLO Deployment Process

- Learner Profile Model

Learner profile information provides the basis for the adaptation and customization of learning objects. An LO can give different learning experiences to different learners with diverse competencies and backgrounds. The IMS Learner Information Package Specification [IMS, 2005] provides a specification of learner information model. Our Learner Profile Model adopts most of the core information in the IMS specification and adds additional features such as the specification of preferred learning style as shown below:

- Identification – Learner ID, Password, Address, Name, Contact Information (phone, e-mail, fax, etc.), Date of Birth, Gender, Place of Birth.
- Goal – Description, Setup data, Target date, Completion Date, Status (active, inactive, completed), Sub-goals.
- Qualification, Certificate, License
- Accessibility
  - Language Proficiency – Oral Speak, Oral Comprehension, Read, Write

- Preference
  - Network Connection – Dial up, DSL, T1, T2, etc.
  - Preferred Learning Style – Receptive, Directive, Guided Discovery, Explorative.
  - Preferred Media Type – Movie, Audio, Document, Flash, Presentation, etc.
  - Disability – Hearing, Visual, Mobility.
- Learning History – Start data, Finish date, Evaluation Result, Status (Active, Suspended, Finished, Failed).
- Affiliation – Affiliated organizations and related information.
- Interest – Description.

The XML Schema Definition of the learner profile model is shown in Appendix E.

5.2.3.2 Event-Trigger-Rule server

The ETR Server is the rule engine that implements and executes sequencing rules and community policies and regulations. These rules, policies and regulations are defined in terms of events, triggers and condition-action rules. In the next few paragraphs, we explain how a sequencing rule of a CLO is implemented as an event, a trigger and a rule, and how the sequencing rule is executed.

An *event* is a data structure that can be posted to represent an occurrence of an event type, and is implemented as a Java class. The attributes of an event class include CLO instance related information, such as CLO ID, process instance ID, activity ID and activity data. A constructor of the class is the only class method specified in the event class.

A *rule*, which is also implemented as a Java class, consists of a condition specification, an action specification, and alternative action specification. Each of the

condition, action and alternative action of the rule is implemented as a method of the rule class. The data attributes of the rule class also include the CLO instance related information as in the event class (i.e., CLO ID, process instance ID, activity ID and activity data) so that the CLO instance information of an event occurrence can be passed to an invoked rule. In addition, the rule class has an attribute, the value of which is a reference to the LMSEngine. Through this reference, the action and alternative action parts of a rule can invoke the methods of the LMSEngine.

A *trigger* is a Java object that links an event to a rule in order to allow the ETR server to execute the rule when the associated event is posted. A trigger object specifies the event that will activate the trigger, a rule that will be invoked by the trigger, and the mapping between the parameters of the event and the parameters of the rule. The parameter mapping information allows values of the data attributes of the event to be transferred to the data attributes of the rule. During the CLO deployment process, the code of the event and rule classes are generated, compiled and installed in the ETR Server. A trigger is not code-generated, but a trigger object is instantiated with the information specified above. It is registered with the ETR Server after instantiation. When the Learning Process Execution Engine posts an event to the ETR Server during the execution of a CLO instance, the ETR Server identifies the trigger object associated with the event. Using the specification in the trigger object, the ETR Server identifies the associated rule, passes the values of the data attributes of the event to the attributes of the associated rule, and invokes the condition method of the rule class. The condition method invokes either the action method or the alternative action method of the rule class depending on the result of the condition evaluation. The action and alternative action

methods can identify the activity, the CLO instance, and the status of the activity by referring to the data attributes of the rule class. These methods can in turn invoke the methods of LMSEngine for adaptive, flexible and customizable delivery of e-learning services.

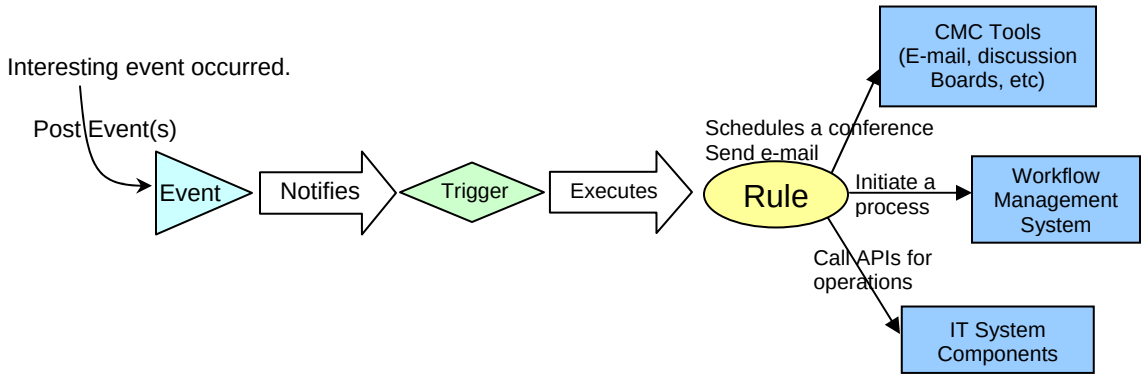


Figure 5-11. Event and Rule Mechanisms for Coordination and Collaboration

In addition to sequential rule processing, the general ETR mechanism allows any party interested in an event to subscribe to the event and receive a notification when an occurrence of the event is posted. When an event is posted at a computing node, all subscribers' nodes are notified, and triggers and rules associated with the event are executed in a distributed fashion by the replicas of the ETR Server installed at the subscribers' sites. We exploit the event notification mechanism to support collaboration in e-learning by integrating it with Computer Mediated Communication (CMC) tools. An event is posted when a situation that requires collaboration has occurred. Rules triggered by each instance of the event can perform collaborative operations such as sending e-mails to collaborators, posting message to a discussion board, initiating a workflow process, or arranging a conference call with fellow learners, the content author, and/or a mentor. The event and rule mechanisms enable timely collaboration among people involved in e-learning. Figure 5-11 shows the event notification mechanism that

facilitates collaborative e-learning through the activation of CMC tools, workflow processes, and other IT systems.

5.2.3.3 Assessment component

The Assessment Component performs the model-based assessment. It carries out the assessment process, which involves selecting assessment items, collecting selected assessment items from the activities participating in an assessment, grading the answers submitted by learners, and recording the assessment result in the runtime status model of the CLO instance that conducts the assessment.

An assessment in a non-leaf activity is carried out by making use of the assessment items specified in the non-leaf activity as well as all or part of the assessment items of all its child activities. The assessment component makes use of the *Parsed CLO* (internal tree structure representation of a CLO) to identify the child activities of a non-leaf activity. Additionally, assessment items in non-leaf activities can be retrieved from the *Parsed CLO*.

A class named *AssessmentInfo* is responsible for collecting and providing assessment items, grading the answers provided by a learner and assigning measures of the assessment to learning objectives. The information of each assessment item is maintained in an object of the class named *AssessmentItemInfo*. The collected assessment items are stored in a vector named *vAssessmentItemInfo*. The assessment items in the vector are delivered in their indexed order. A hash table named *htAssessmentItems* is used to store the vectors associated with learning objectives. The table uses learning objective id as the key. Since assessment items of each objective are stored in a separate vector, the measure of an objective can be easily calculated.

In case a leaf activity is bound to a CLO, an *AssessmentInfo* object of the CLO is instantiated and stored in a vector in a hash table, *htAssessmentInfo*, which uses learning objective id as the hash key. The *AssessmentInfo* object is also stored in the *vAssessmentInfoOfBoundCLO* vector for presentation. Assigning measures to objectives, obtaining the bound LOs for collecting assessment items, and so on, are accomplished by communicating with the *LMSEngineCore* object. When a learner submits an answer to an assessment item, the user interface component would communicate with the RMI server of LMSEngine (*LMSEngineImpl*) to record the answer. The runtime structure of the assessment component is illustrated in Figure 5-12.

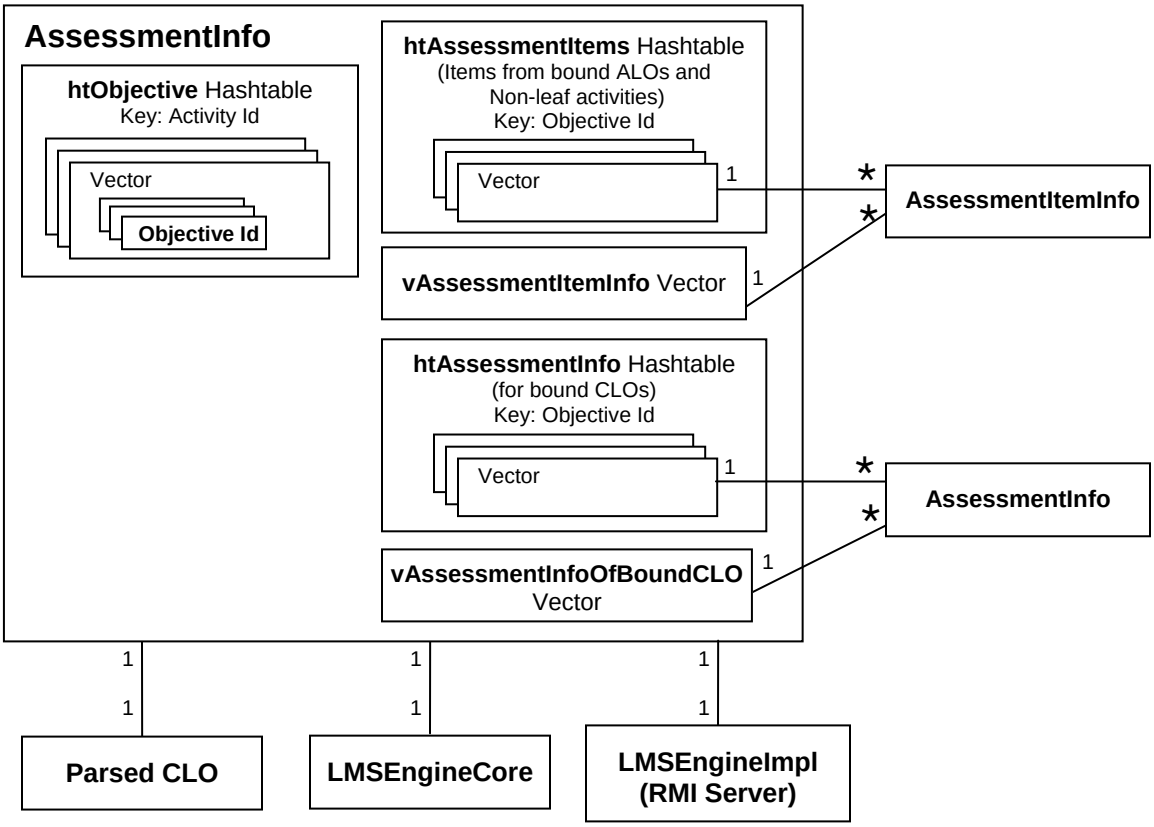


Figure 5-12. Runtime Structure of Assessment Component

5.2.3.4 Constraints-based LO broker

The Constraints-based LO Broker, which is an extension of UDDI v2.0 registry, supports registration, browsing and dynamic binding of LOs. To register an LO with the Broker, the *RegBroker* (*QueryBroker*) Web service operation should be invoked. The operation requires the URL of an extended WSDL document describing its registration (query) constraints as a parameter. The extended WSDL document includes the meta-data description as well as the general WSDL elements, as discussed.

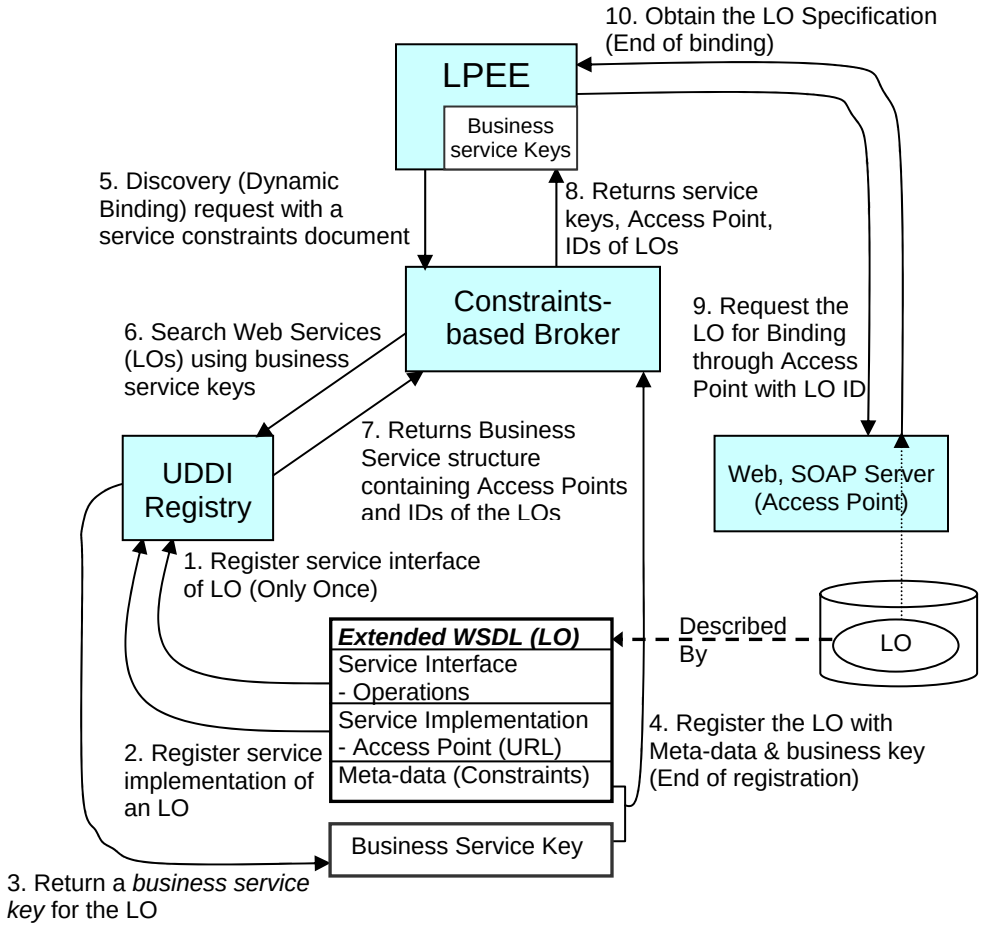


Figure 5-13. Registration and Discovery Process of LO

LO authoring tools register LOs with the Broker, and take the following steps to accomplish the task. First, an LO Web-Service interface is registered as a *service*

interface with the UDDI registry. The registration operation returns an interface key (*tModel* key). The interface can be considered as a *business type* that is common to all LOs of that type. So, the interface registration is a one-time process. Then an LO, which is an implementation of the service interface, is registered with the UDDI registry as a *business service*, and a *business service key* is returned to the authoring tool when the registration of the LO is completed. Together with the interface key and the business key, the meta-data and constraints of the LO are registered with the Constraints-based Broker, which serves as a meta-data repository having a constraint-matching facility.

The LPEE can discover an LO for dynamic binding using the constraint-matching facility provided by the LO Broker. During the processing of a CLO instance, the LPEE would issue a discovery request to the LO Broker by submitting an extended WSDL document that includes a set of requirements. The constraint-matching facility would match the specified requirements against the meta-data and constraints of LOs that have been registered as Web services. The LO Broker finds a set of service keys of the LOs that match with the requirements. Using the service keys, the LO Broker accesses the UDDI registry to obtain LO access information, which includes the IDs and access points of the LOs. The Web user interface retrieves the meta-data of each LO from its LO repository to display the information to a learner to help him/her choose the right LO. Finally, the LPEE accepts the learner's choice, and then invokes the *getLO* operation at the access point of the chosen LO (i.e. the LO repository) with the ID of the selected LO as an input parameter. The *getLO* operation fetches the LO and returns it to LPEE for delivery. In Figure 5-13, steps 1 through 4 describe the registration procedure, and steps 5 through 10 describe the dynamic discovery and binding process. In case of a static

binding request, the LO Broker is given the business service key of a pre-selected LO instead of service requirements in step 5. The LO Broker directly passes the business service key to the UDDI registry without constraint-matching to obtain the service access information. The WSDL document that describes the LO Broker Web service is given in Appendix D.

5.2.3.5 User interface component

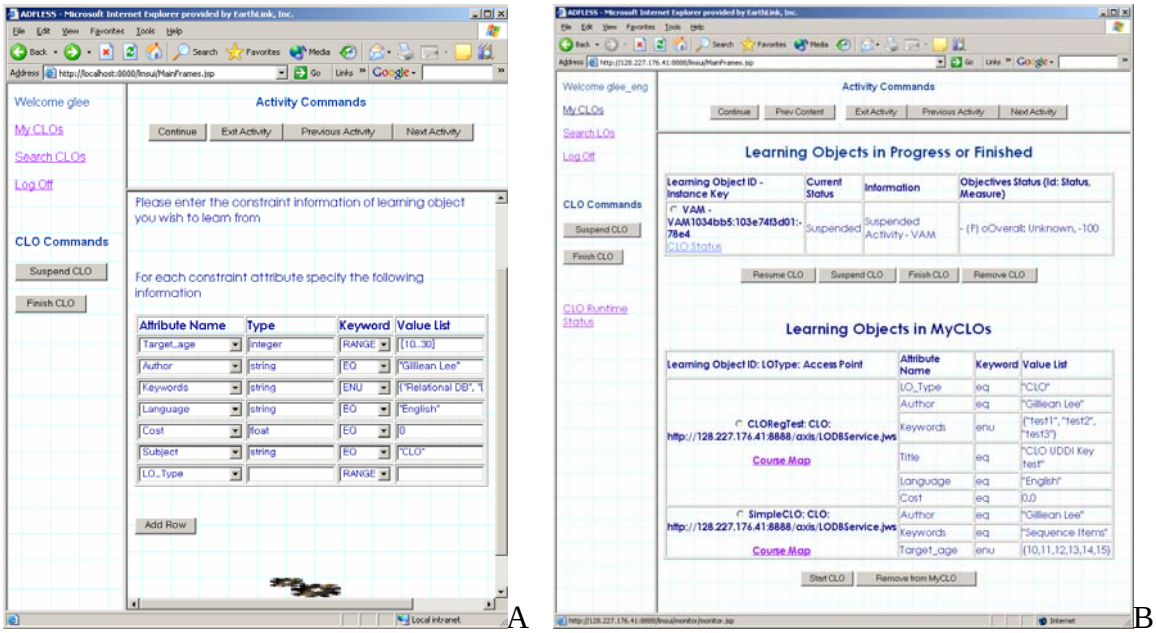
The User Interface Component is a set of Java Server Pages (JSPs), which generate Web pages for communication between users and server components, such as LPEE, Assessment Component, Monitoring Component and Constraints-based Broker. Through the Web user interface, a learner can start, suspend, finish, navigate, and monitor a CLO instance, and search LOs using meta-data constraints.

MainFrames JSP carries out user authentication and loading of a main Web page. CLO instance commands, such as instantiation, completion, suspension, and resumption of a CLO instance, are handled by *cloCommands* JSP. *ActivityCommands* JSP processes the activity navigation commands, such as *Continue*, *Previous Contents*, *Next Activity* and *Previous Activity*. Communication with the Assessment Component for assigning answers to assessment items is accommodated through *assessmentCommand* JSP. *Monitoring* JSP provides the status information of a CLO instance. Web user interface that facilitates LO discovery through the Constraints-based LO Broker is provided by *searchLOs* JSP. A user space called ‘MyCLOs’ can be accessed through a Web page generated by *myclos* JSP.

5.2.3.6 Web user interface

A learner interacts with the Learning Process Management System through Web pages that the User Interface Component generates. After a learner logs in the system,

he/she is able to search and execute LOs, access his/her learner space ‘MyCLOs’, or log off.



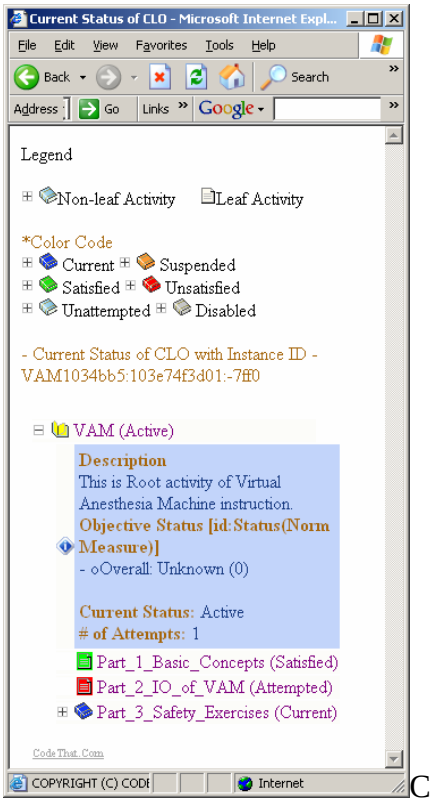


Figure 5-14. Web User Interface. A) Search Constraint Specification. B) ‘MyCLOs’ Learner Space. C) CLO Status Monitoring Information.

When a learner chooses to search LOs, a page that accepts search requirements is provided as shown in Figure 5-14 (A). A list of LOs that meets the requirements and their meta-data is displayed as the result of the search request. In the search result page, LOs can be added to ‘MyCLOs’ for later execution, or an LO can be instantiated for execution. In ‘MyCLOs’ Web page, a learner is able to start an LO registered in ‘MyCLOs’, gain learning experience by executing the LO, suspend, finish, or resume existing LO instances. Figure 5-14 (B) shows the ‘MyCLOs’ page. During the execution of a CLO instance, a learner can monitor the status of a CLO instance through a Web browser as shown in Figure 5-14 (C).

CHAPTER 6
APPLYING LEARNING OBJECT AND DYNAMIC E-LEARNING TECHNOLOGIES TO SIMULATION-BASED MEDICAL INSTRUCTION

There are many areas of education where our learning object models and e-learning service infrastructure can be applied. As a part of this work, we aim to apply the e-learning technologies to a practical area where there can be a direct impact.

6.1 Instruction on Anesthesia Machines

Complex medical equipment is regularly used in hospitals and clinics where patient safety depends on the proper interaction between skilled practitioners and equipment [Dalley, et. al., 2004] [Reason, 2000]. Even though anesthesia is increasingly safe, critical accidents in which patients are seriously injured still occur [Dalley, et. al., 2004] [Weinger, 1999]. Misuse of equipment is far more common than pure equipment errors/failures in the medical environment. Human error is a dominating factor in up to 90% of the problems caused by equipment [Weinger, 1999] [Williamson, et. al., 1993].

There are several Web contents that provide learning materials on anesthesia machines. For example, the resource reported in [Dosch, 2004] provides extensive Web contents about a variety of anesthesia machines. The material by the College of Veterinary Medicine [Washington State University, 2004] is a comprehensive resource that covers circuits, vaporizers, gas cylinders, pressure regulators, flow meters, scavenging, ventilators, and endotracheal tubes. However, they do not provide simulation software for practice, nor a way of assessment to measure learners' understanding of the materials. [Gas Man, 2004] is a commercial product for teaching, simulating and

experimenting with anesthesia uptake and distribution. It provides a text tutorial and simulation software. The text tutorial is not accessible on the Web; it is published as a book. The simulation software is not implemented as a learning object. To the authors' knowledge, no anesthesia simulation system has been developed based on the learning object technology at the time of writing.

6.2 Introduction to the Virtual Anesthesia Machine

There are several ways for practitioners and medical students to learn how to use an anesthesia machine. First, a person who is proficient in the use of the machine can provide a training session by giving a demonstration and using textbooks and reference materials supplied by the manufacturer of the machine. Second, a practitioner can learn by using the anesthesia machine with a Human Patient Simulator (HPS) [Dalley, et. al., 2004]. Third, a Virtual Anesthesia Machine (VAM) [Lampotang, et. al.] that simulates a real counterpart can be used for educational purposes. VAM is a Web-based simulation system developed at the University of Florida. It, together with a Web-accessible workbook, can be accessed by registered users free of charge through Web-enabled personal computers. VAM is an interactive simulation of a generic anesthesia machine with model-driven, real-time animation of color-coded molecules and dynamic representations of flows, pressures, volumes and concentrations of gases. It is international in scope and features legends in 23 languages and is used in over 336 institutions and programs worldwide. The VAM web site at http://vam.anest.ufl.edu receives more than 2,000,000 hits per year.

A major advantage of VAM is that users can focus on learning about anesthesia without being distracted by the concern of placing an actual patient at risk [White, 2002]. Also, it is a cost-efficient way to learn because it is free to use and does not require any

actual anesthesia equipment or patient. In addition, practitioners and medical students can learn the inner workings of the anesthesia machine because VAM is a transparent reality simulator; that is, it is a model-driven, display-based simulation that represents internal, abstract and invisible functions with explicitly visible and manipulatable symbols to assist learners in exploring, developing and confirming mental models. The focus of the simulation is on high fidelity behavior rather than life-like appearance [Lampotang, et. al.].

In spite of VAM's success and advantages, there are a number of limitations in the existing system. First, it is a monolithic system in that the components and their accompanying instructional materials, questions and answers for practice and questions for assessment are not modularized to make them reusable for constructing other simulation systems. Second, users of VAM and its workbook can freely operate on any part of the system and access any content of the workbook. Although, the workbook provides some guide for structured exercises, the learners can choose not to follow the guide. Thus, the last of the four learning approaches (*receptive*, *directive*, *guided discovery* and *explorative*) proposed by Dr. Ruth Clark [Clark, 2000] is emphasized. Some learners may desire or need more structured learning approaches than the exploratory approach. Lastly, the ability of the existing VAM system to adapt and customize its instruction delivery to suit the learners' profiles and needs is currently limited to language and medical gas color code. Searching for contents on the VAM Web site is accomplished via a search box, driven by Google, which is a part of the header of every Web page on the VAM Web site. This chapter discusses the application of the

learning object and e-learning service technologies we have developed to address these limitations.

6.3 Application to Medical Instruction using VAM

The Virtual Anesthesia Machine (VAM) simulates the inner workings of an anesthesia machine and ventilator – the complex equipment in which oxygen, nitrous oxide and anesthetics mix together to provide patients insensitivity to pain during surgery or procedures. The objective of VAM is to increase the safety of patients by providing practitioners clear understanding of the flow of gas in the machine, user actions and the consequences of equipment malfunctions [White, 2002].

VAM is composed of a transparent reality simulator developed using Macromedia Director, and a Web-based workbook (http://vam.anest.ufl.edu/members/workbook/apsf-workbook-english.html) delivered in the Macromedia FlashPaper format. The simulator currently supports 22 languages, and the workbook is in 11 different languages. In the visual display of the simulator, the language of the user interface as well as the color code for medical gases are changed according to a learner's selection or data retrieved from a user database.

The workbook consists of three parts. Part 1 covers the basic concepts of the anesthesia machine. Part 2 teaches input/output controls of VAM. Users can adjust gas pipelines, valves, gas flow meter knobs, buttons, and so on, and visually observe the effects of the adjustment on the flow of molecules of gases, the readings of pressure gauges, valves, and so forth. Part 3 covers safety exercises and is composed of six sections for covering the high pressure system, the low pressure system, the breathing circuit, manual ventilation, mechanical ventilation, and the scavenging system, respectively. Each section is comprised of a set of learning objectives, a set of

instructional materials, and a set of post-assessment questions. Each instructional material has a question, an answer, a VAM demonstration instruction, an explanation of the demonstration, and some learning objectives. The post-assessment questions (quiz) can be answered online at a Website specified in the workbook.

The workbook, the simulator and the quiz Web site need to be used together for effective delivery of learning experience. However, the current VAM system is not structured to provide a smooth transition between the workbook contents, the simulator, and the post-assessment questions. Even though directions are provided in the workbook, it is the responsibility of a learner to open the workbook online, to start the simulator, to go to the quiz Website at appropriate times. The learner is free to do anything with the contents and the simulator (i.e., explorative learning). The learning process is generally not guided by the system.

6.3.1 Encapsulating VAM as Learning Objects

There are two main advantages to create LOs that encapsulate the workbook contents, the VAM simulator, and the quiz problems. First, each LO can be reused for instruction of other types of anesthesia machines, which may use the same or similar component(s). Second, it is possible to deliver learning experiences provided by the workbook's contents, the use of the simulator for practice, and the use of quiz problems for assessment in a seamless fashion and under the control of the Learning Process Management System.

In this application scenario, we created an ALO for Part 1 and another ALO for Part 2 of the workbook. Part 3 is composed of several ALOs each of which contains instructional materials to cover a component of the anesthesia machine. For these ALOs, the workbook contents are extracted and regenerated as separate HTML pages. The quiz

problems available at the quiz Website are extracted and used as post-assessment items. We constructed a CLO for modeling the entire VAM instruction as an activity tree, in which each of the leaf activities is bound to an ALO. Figure 6-1 shows the activity tree of the CLO, and the relationship among the ALOs and VAM resources.

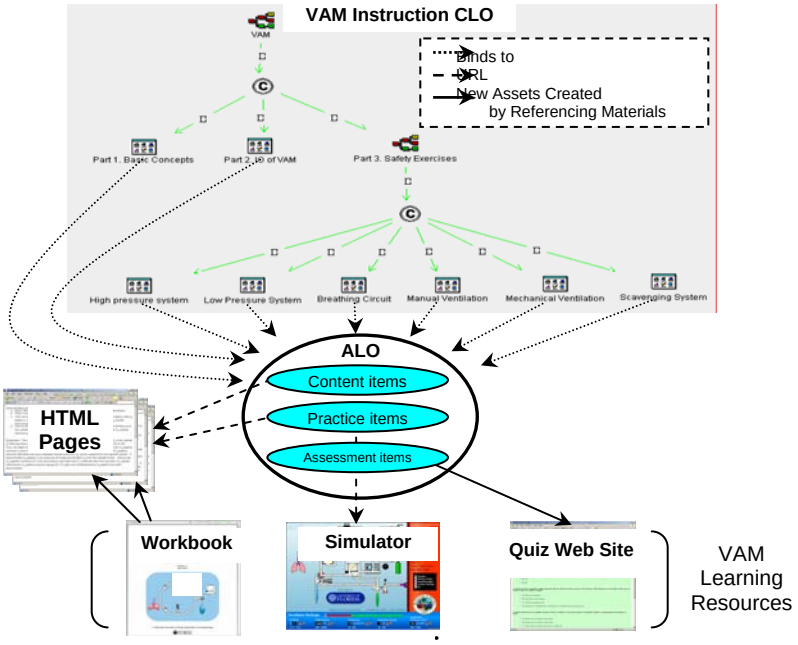
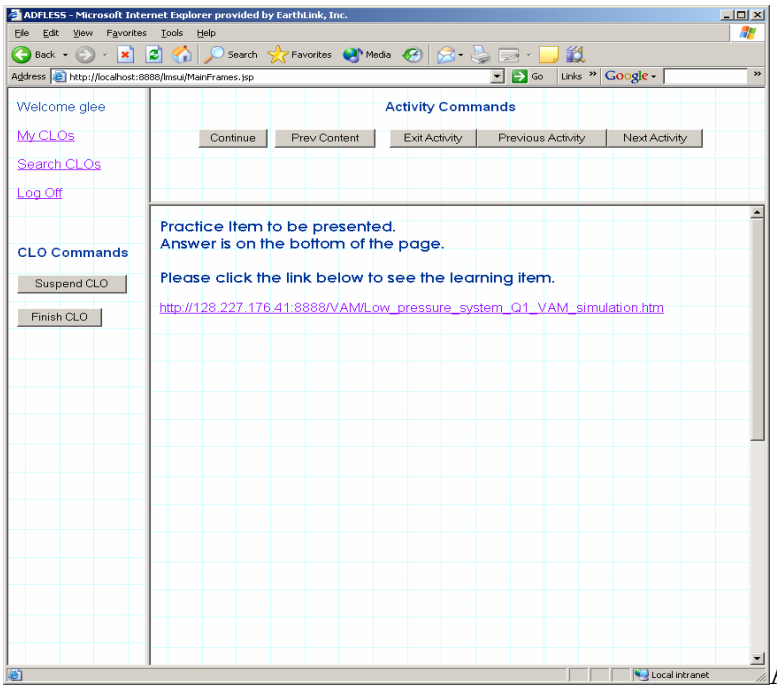


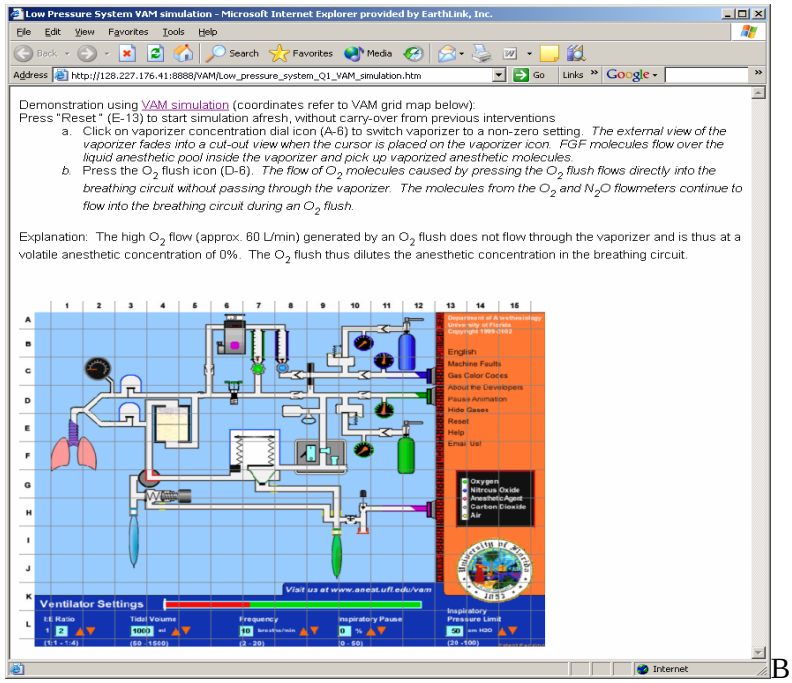
Figure 6-1. Construction of LOs using VAM Learning Resources

An ALO bound to a leaf activity of Part 3 begins its presentation with a textual description of learning objectives of the ALO. Then a sequence of several content and practice items of the workbook is delivered. A content item is presented to the learner in the form of Q&A as in the VAM workbook. For example, a content item in the High Pressure System ALO is as follows: "***Question***: *During the morning pre-use check of an anesthesia machine, you open the O2 cylinder to check the O2 cylinder pressure. The O2 cylinder reads full (2,000 psig). You subsequently close the O2 cylinder. With the anesthesia machine turned on with a minimal flow of 200mL/min of O2, what does the O2 cylinder pressure gauge now read with the O2 pipeline supply connected and the O2 cylinder closed?* ***Answer***:

Even though the O2 cylinder has been closed, the O2 cylinder pressure gauge will continue to read full.”



A



B

Figure 6-2. Web User Interface of VAM Instruction. A) E-Learning Service System Delivering Practice Item. B) A Practice Item with Demonstration Scenario.

Practice items include a text-based demonstration scenario related to the Q&A, and provides a Web link for invoking the VAM simulator. The delivery sequence of the items follows the order in the workbook. After finishing the content and practice items, post-assessment items in the multiple-choice format are provided and the result is graded. All these content, practice and assessment items are delivered in sequence through a Web user interface of our Learning Processor Management System. Figure 6-2 (A) shows the Web user interface that presents a practice item of the URL type. By clicking the link to the practice item, a new Web browser that presents the practice item will popup, as shown in 6-2 (B). The user can then invoke the VAM simulator by clicking the link in the new Web browser, and follow the demonstration scenario of the practice item.

6.3.2 Dynamic Execution of VAM Instruction

A number of dynamic techniques, including sequencing rules and dynamic binding, are incorporated in the design of the VAM CLO to achieve adaptive, customizable, and flexible delivery of instruction. They are described below:

(1) A learner is *required* to *take the content, practice, and assessment items* to satisfy an activity. In the activities 'Part 1 Basic Concepts' and 'Part 2 IO of VAM', all the learning items have to be delivered to a learner in order to make each of the activities 'Satisfied'. This is specified by an after-post-assessment rule "*if [true], then [set the activity 'Satisfied']*" in each activity.

(2) A learner is required to *retry* an activity if the result of post-assessment is not satisfactory. A leaf activity, 'High Pressure System', requires a learner to retry when he/she does not obtain a satisfactory score in the post-assessment of the activity.

(3) Dissimilar contents are delivered to different learners depending on the *learning goals* specified in their *learner profiles*. The pre-activity rule of the root activity 'VAM'

checks if a learner's goal is related to anesthesia. If it is, then a non-leaf activity 'Part 3 – Safety Exercises', which contains the details of system components and exercises, will be delivered to the learner. Otherwise, the non-leaf activity is *disabled* so that the contents in the activity and its child activities are not presented to the learner. In the CLO, we consider that the activity 'Part 3 – Safety Exercises' covers too many details that are not necessary for a person without a learning goal in anesthesia.

(4) ALOs written in the *language* of a learner's preference will be chosen and delivered. The language requirement is specified as a binding constraint of a leaf activity. For example, in the CLO, activities 'Part 1-Basic Concepts' and 'Part 2-IO of VAM' can be bound to ALOs that are written in either English or Korean, depending on the language specified in the learner's profile model.

(5) Various *learning approaches* that suit different learners' backgrounds and preferences are supported. A learner with the background knowledge of anesthesia or a preference to the learning approach, *guided discovery,* is given the freedom to choose the next activity when the non-leaf activity 'Part 3 – Safety Exercises' is delivered. That is, the *Choice* sequencing control mode is set in the connector of the activity 'Part 3 – Safety Exercises'. On the other hand, a learner who prefers the *receptive* or *directive* learning approach will be restricted to choose the next activity; that is, the Flow sequencing control mode will be set in the connector. A *customized* LO is delivered by a pre-activity rule of the root activity.

(6) *Depending on the performance* of a learner, a *post-assessment* can be conducted in the 'Part 3 – Safety Exercises' activity. There is an after-post-assessment rule in the activity stating "*If [4 or less children are satisfied], then [conduct Post-assessment], else*

[set the activity 'Satisfied']", and each child activity of 'Part 3 – Safety Exercises' is designed to conduct a post assessment. If the assessment results are satisfactory in four or less child activities, a post-assessment is presented to the learner. Otherwise, the parent activity will be set to 'Satisfied'.

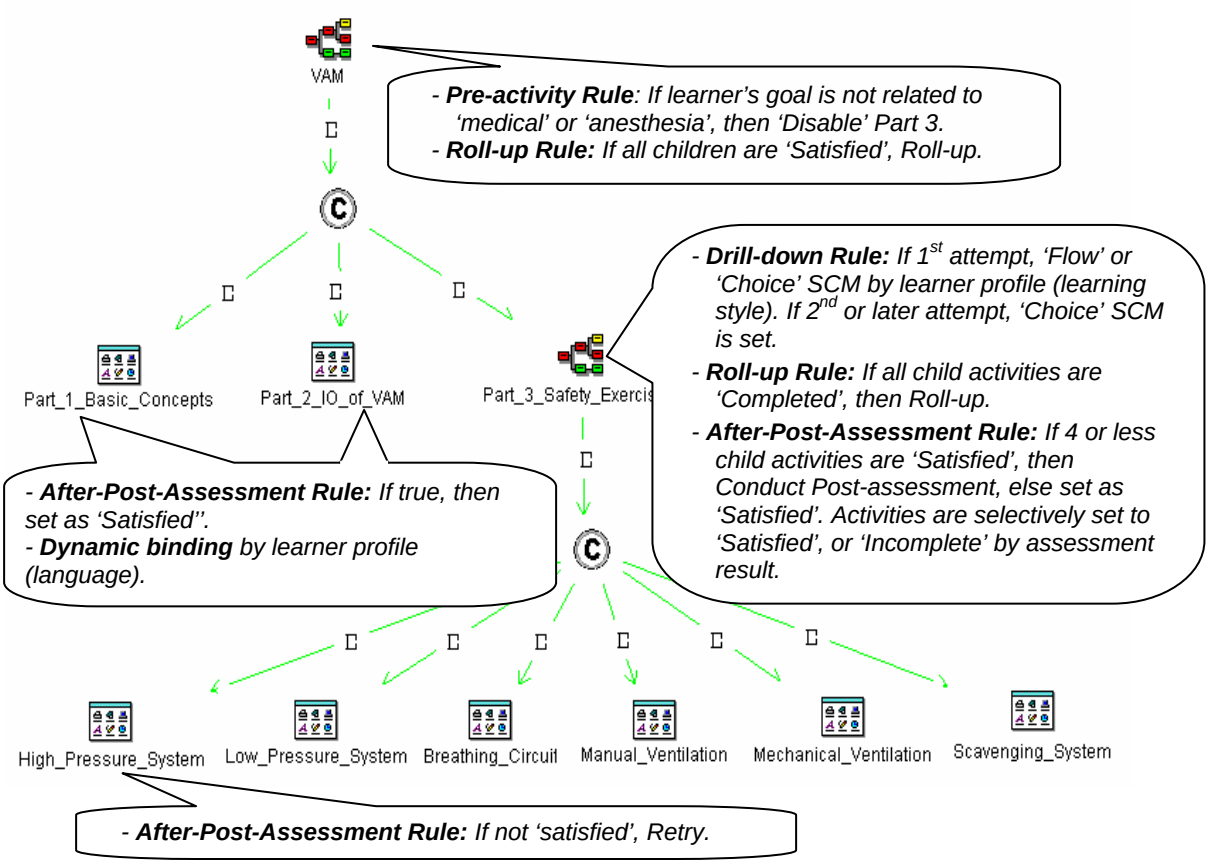


Figure 6-3. Condition/Action Rules and Dynamic Binding Defined in 'VAM' Instruction CLO.

(7) A learner is required to *re-take* the instruction of components (activities) that he/she has *failed*. If a learner takes the optional assessment in 'Part 3 – Safety Exercises' activity discussed in (6), the assessment result is analyzed to identify the parts (components) of VAM the learner has to focus on. By referring to the status of learning objectives that are set by the assessment result, the system can figure out the components where the learner has difficulties. An after-post-assessment rule in 'Part 3 – Safety

Exercises' sets the status of its child activities as 'Incomplete' for the unsatisfactory parts, and as 'Satisfied' otherwise. When the learner re-takes the 'Part 3 – Safety Exercises', the statuses of the activities (Attempted or Satisfied) are shown to the learner so that he/she would know which activities to take in order to satisfy and complete the CLO.

(8) After the entire child activities of the root activity 'VAM' have been 'Satisfied', the instruction of the CLO is satisfied. This is specified by a roll-up rule "*If [all child 'Satisfied]', then [Roll-up]*" in the root activity. The rules that facilitate the dynamic execution of 'VAM' CLO are shown in Figure 6-3.

6.3.3 Importance of Learning-Object-Technology-Based Medical Instruction

Educators in medical education are beginning to propose that assessment should be based on performance end-points rather than an arbitrarily-defined time end-point that is so prevalent in paper and pencil tests. In other words, in this new approach to instruction, it does not matter how long a learner has to practice to learn a given learning objective or skill, as long as the learner can prove at the end of the learning session that the performance end-point has been reached. An LO approach to delivering instruction reduces instructional manpower needs and is thus compatible with assessment based on performance end-points rather than temporal ones, because the LO intrinsically accepts and accounts for the fact that all learners do not learn at the same rate.

Accreditation bodies in medicine are also beginning to focus on performance or outcome itself instead of infrastructure. The Accreditation Council for the Graduate Medical Education (ACGME) oversees medical instruction in the United States as its name implies. In July 2001, the ACGME embarked on a 10-year project, named the Outcome project [ACGME, 2001]. The ACGME Web site states: "The Outcome project is a long-term initiative by which the ACGME is increasing emphasis on educational outcomes in

the accreditation of residency education programs". The same Web site also adds: "The current model of accreditation captures the potential of a GME (Graduate Medical Education) program to educate residents by focusing on structure and process components." In other words, accreditation was based on the availability of classrooms, instructional tools and materials, and the curriculum, not on measuring whether the residents (learners) had learned. Medicine is often hands-on and medical simulation is well-suited for practice and assessment of both cognitive and psychomotor skills without placing patients at risk. The encapsulation of assessment within medical simulation LOs fits well within the goals of ACGME as the individualized assessment scores could be used as one of the tools to evaluate learning outcome.

6.4 Chapter Summary

The e-learning technologies presented in this dissertation (i.e., learning object models, authoring tools, Web-service infrastructure, learning process management system, and event-driven, rule-based execution of learning processes) have been applied to encapsulate the instructional materials of the Web-based Virtual Anesthesia Machine and the simulator as LOs, deliver instructional materials to learners, use the simulation system for practice, and perform assessment. The instructional materials are integrated and delivered seamlessly providing more guide in our e-learning service infrastructure.

We have demonstrated that the event-driven, rule-based execution of a learning process, which models the VAM instruction as a CLO, can make the delivery of instruction and assessment active, customizable, flexible and adaptive. Several techniques, such as retrying or disabling activities depending on a learner's profile, performance and needs, dynamic binding to suitable learning objects, optional assessment presentation,

and the support of different learning approaches by customization of sequencing control modes, have been applied to achieve personalized learning.

CHAPTER 7
SUMMARY

7.1 Summary and Conclusion

It is our vision and belief that the citizens of the world can be educated and benefit if all Internet users can actively engage in sharing their knowledge and learning from one another to solve the problems they face in their lives. An e-learning service infrastructure that is based on reusable multi-media learning objects, and advanced e-learning technologies that support collaborative, adaptable, flexible and customizable learning are the keys to realize this vision. In this work, we have developed two learning object models, two learning object definition languages, a learner profile model, a runtime status model, and several user interface tools and software system components to enable the modeling, specification, registration, discovery, invocation and management of distributed e-learning objects. The developed tools and software system components can be replicated and installed at many Websites to enable the establishment of many virtual e-learning communities for knowledge sharing in different problem domains.

To demonstrate the utility of the developed technologies and infrastructure, we have applied them in delivering instructions on the function and operation of a generic anesthesia machine. We modularized and converted the instructional materials of an existing Web-based Virtual Anesthesia Machine into a number of reusable Atomic Learning Objects, each of which contains content, practice and assessment items as well as meta-data with constraint specifications. We also introduced a Composite Learning Object to capture the structure and the sequence of delivering the Atomic Learning

Objects. The Web-based Virtual Anesthesia Machine is used by learners as an online simulator to practice on the instructional materials delivered to them. The developed LOs can be reused for other related medical instructions. The developed tools and the Learning Process Management System provide an integrated computing environment, in which a learner can learn the function and operation of the generic anesthesia machine, use the virtual machine to practice on what he/she has learned, and be assessed on the information and knowledge gained. In this work, we have also demonstrated the use of an Event-Trigger-Rule Server as a component of the Learning Process Management System to achieve active, adaptive, customizable and flexible rule-based instruction delivery. Sequencing rules are defined and invoked in a learning process to, for example, retry or disable activities depending on a learner's profile, performance and need, dynamically discover and bind to suitable atomic learning objects, and modify sequencing control modes based on a learner's runtime status model. The Learning Process Management System supports four general learning strategies (i.e., receptive, directive, guided discovery and explorative) to enable personalized learning.

7.2 Future Work

A learner of a specific learning style can benefit from personalized learning contents. There are research efforts on theories on learning styles [Felder & Brent, 2005] [Jester & Miller, 2000] [Kolb, 1984]. Research on customization/personalization of learning contents to suit different learning styles is a worthwhile undertaking.

Another worthwhile research problem is the use of ontology in e-learning. An ontology is a specification of a conceptualization [Gruber, 1993]. Ontologies are useful for representing terminologies in a machine interpretable way and for reasoning about conceptual structures used to represent the meanings of terms. The new W3C standard for

ontologies, the Web Ontology Language (OWL) [W3C, 2004], provides a mechanism for sharing ontologies on the Web. In a collaborative e-learning environment, in which individuals who play different roles in a virtual community would exchange instructional materials, assessment items and practice items, terminology becomes an important issue. Specific terms used by one person in naming entities and attributes and in metadata descriptions can be quite different from those used by another person. Also, users searching for LOs that are registered as Web services in the Web Service Broker will likewise encounter mismatches between the terms used in their searches and the terms used to register these LOs. The use of ontology technologies to address the terminology issues is an important future research.

Learning Object (LO) evaluation is essential for promoting reusability because the evaluation information can be used by learners in selecting the suitable learning objects. However, the simple quality rating used in the existing learning object repositories does not provide sufficient information to the learners. A learning process can have a complex structure containing many LOs. Evaluation factors such as drop-out rate, average score, number of usage, and learners' reviews need to be considered. Evaluation of CLOs can be a challenging research problem.

APPENDIX A
XML SCHEMA DEFINITION OF LEARNING CONTENT DEFINITION LANGUAGE

```
<?xml version="1.0" encoding="UTF-8" ?>
- <xsd:schema targetNamespace="http://www.cise.ufl.edu/~glee/LCDL1_0"
    xmlns:lcdl="http://www.cise.ufl.edu/~glee/LCDL1_0"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">XML Schema for Learning Content Definition Model Language
        Copyright by Gilliean Lee. 2003.</xsd:documentation>
    </xsd:annotation>
  - <xsd:element name="LearningObject">
    - <xsd:complexType>
      - <xsd:sequence>
          <xsd:element ref="lcdl:ContentItems" />
          <xsd:element ref="lcdl:PracticeItems" minOccurs="0" />
          <xsd:element ref="lcdl:Assessments" minOccurs="0" />
          <xsd:element ref="lcdl:LearningItemSequences" minOccurs="0" />
          <xsd:element ref="lcdl:UDDIRegistration" minOccurs="0" maxOccurs="unbounded" />
          <xsd:element ref="lcdl:constraints" minOccurs="0" />
        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required" />
        <xsd:attribute name="Name" type="xsd:string" />
      </xsd:complexType>
    </xsd:element>
  - <xsd:element name="LearningItemSequences">
    - <xsd:complexType>
      - <xsd:sequence>
          <xsd:element name="LearningItemSequence" type="lcdl:LearningItemSequenceType"
            minOccurs="0" maxOccurs="unbounded" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  - <xsd:complexType name="LearningItemSequenceType">
      <xsd:attribute name="ItemId" type="xsd:NMTOKEN" use="required" />
      <xsd:attribute name="Type" type="lcdl:ItemTypes" use="required" />
    </xsd:complexType>
  - <xsd:simpleType name="ItemTypes">
    - <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Content" />
        <xsd:enumeration value="Practice" />
      </xsd:restriction>
    </xsd:simpleType>
  - <xsd:element name="Assessments">
    - <xsd:complexType>
      - <xsd:sequence>
          <xsd:element name="PreAssessmentItems" type="lcdl:AssessmentItems" minOccurs="0"
            />
          <xsd:element name="PostAssessmentItems" type="lcdl:AssessmentItems" minOccurs="0"
            />
```

```
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
  - <xsd:complexType name="AssessmentItems">
      - <xsd:sequence>
            <xsd:element name="AssessmentItem" type="lcdl:AssessmentItem" minOccurs="0"
                maxOccurs="unbounded" />
            <xsd:element ref="lcdl:Code" minOccurs="0" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="Code" type="xsd:string" />
  - <xsd:complexType name="AssessmentItem">
      - <xsd:complexContent>
          - <xsd:extension base="lcdl:Item">
                <xsd:attribute name="Points" use="required" type="xsd:decimal" />
                <xsd:attribute name="AutoGrade" use="required" type="xsd:boolean" />
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
  - <xsd:complexType name="Attachment">
      - <xsd:sequence>
            <xsd:element name="Binary" type="xsd:string" />
        </xsd:sequence>
        <xsd:attribute name="ContentType" type="lcdl:ContentTypes" />
    </xsd:complexType>
  - <xsd:simpleType name="ContentTypes">
      - <xsd:restriction base="xsd:string">
            <xsd:enumeration value="rm" />
            <xsd:enumeration value="mov" />
            <xsd:enumeration value="wmf" />
            <xsd:enumeration value="doc" />
            <xsd:enumeration value="pdf" />
            <xsd:enumeration value="mp3" />
            <xsd:enumeration value="mid" />
        </xsd:restriction>
    </xsd:simpleType>
  - <xsd:element name="PracticeItems">
      - <xsd:complexType>
          - <xsd:sequence>
                <xsd:element name="PracticeItem" type="lcdl:Item" minOccurs="0"
                    maxOccurs="unbounded" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
  - <xsd:complexType name="Item">
      - <xsd:sequence>
            <xsd:element name="Problem" type="lcdl:ContentItem" />
            <xsd:element name="Answer" minOccurs="0" type="xsd:string" />
        </xsd:sequence>
        <xsd:attribute name="Number" use="required" type="xsd:unsignedInt" />
        <xsd:attribute name="Name" use="optional" type="xsd:string" />
    </xsd:complexType>
  - <xsd:element name="ContentItems">
      - <xsd:complexType>
          - <xsd:sequence>
                <xsd:element name="ContentItem" type="lcdl:ContentItem" minOccurs="0"
                    maxOccurs="unbounded" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
  - <xsd:complexType name="ContentItem">
      - <xsd:choice>
```

```
            <xsd:element ref="lcdl:HTMLDoc" />
            <xsd:element name="text" type="xsd:string" />
            <xsd:element name="URL" type="xsd:anyURI" />
        </xsd:choice>
        <xsd:attribute name="Number" use="required" type="xsd:unsignedInt" />
        <xsd:attribute name="Name" use="optional" type="xsd:string" />
    </xsd:complexType>
  - <xsd:element name="HTMLDoc">
      - <xsd:complexType>
          - <xsd:sequence>
                <xsd:element ref="lcdl:Code" minOccurs="0" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
  - <xsd:element name="UDDIRegistration">
      - <xsd:complexType>
            <xsd:sequence />
            <xsd:attribute name="BusinessInterfaceKey" type="xsd:string" />
            <xsd:attribute name="ServiceImplementationKey" type="xsd:string" />
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="constraints" type="lcdl:constraintsType" />
  - <xsd:complexType name="constraintsType">
  - <xsd:sequence>
    <xsd:element ref="lcdl:constraint" minOccurs="0" maxOccurs="unbounded" />
    <xsd:element ref="lcdl:service_constraints" minOccurs="0" maxOccurs="unbounded" />
    <xsd:element ref="lcdl:interattribute_constraints" minOccurs="0" maxOccurs="unbounded" />
    <xsd:any namespace="##other" minOccurs="0" />
      </xsd:sequence>
      </xsd:complexType>
    <xsd:element name="constraints" type="lpdl:constraintsType" />
  - <xsd:complexType name="constraintsType">
      - <xsd:sequence>
            <xsd:element ref="lpdl:constraint" minOccurs="0" maxOccurs="unbounded" />
            <xsd:element ref="lpdl:service_constraints" minOccurs="0" maxOccurs="unbounded" />
            <xsd:element ref="lpdl:interattribute_constraints" minOccurs="0" maxOccurs="unbounded" />
            <xsd:any namespace="##other" minOccurs="0" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="constraint" type="lpdl:constraintType" />
  - <xsd:complexType name="constraintType">
      - <xsd:sequence>
            <xsd:element ref="lpdl:input_constraint" minOccurs="0" maxOccurs="unbounded" />
            <xsd:element ref="lpdl:output_constraint" minOccurs="0" maxOccurs="unbounded" />
            <xsd:element ref="lpdl:operation_constraint" minOccurs="0" maxOccurs="unbounded" />
            <xsd:any namespace="##other" minOccurs="0" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="input_constraint" type="lpdl:attributeconstraintType" />
  - <xsd:complexType name="attributeconstraintType">
      - <xsd:sequence>
            <xsd:element ref="lpdl:name" minOccurs="1" maxOccurs="1" />
            <xsd:element ref="lpdl:type" minOccurs="1" maxOccurs="1" />
            <xsd:element ref="lpdl:keyword" minOccurs="1" maxOccurs="1" />
            <xsd:element ref="lpdl:valueList" minOccurs="1" maxOccurs="1" />
            <xsd:element ref="lpdl:negotiable" minOccurs="1" maxOccurs="1" />
            <xsd:element ref="lpdl:priority" minOccurs="1" maxOccurs="1" />
            <xsd:any namespace="##other" minOccurs="0" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="name" type="xsd:string" />
    <xsd:element name="type" type="lpdl:typeType" />
```

```
- <xsd:simpleType name="typeType">
    - <xsd:restriction base="xsd:string">
          <xsd:enumeration value="string" />
          <xsd:enumeration value="float" />
          <xsd:enumeration value="integer" />
          <xsd:enumeration value="boolean" />
          <xsd:enumeration value="datetime" />
          <xsd:enumeration value="date" />
          <xsd:enumeration value="time" />
          <xsd:enumeration value="duration" />
      </xsd:restriction>
  </xsd:simpleType>
  <xsd:element name="keyword" type="lpdl:keywordType" />
- <xsd:simpleType name="keywordType">
    - <xsd:restriction base="xsd:string">
          <xsd:enumeration value="range" />
          <xsd:enumeration value="enu" />
          <xsd:enumeration value="eq" />
          <xsd:enumeration value="!=" />
          <xsd:enumeration value="<" />
          <xsd:enumeration value="<=" />
          <xsd:enumeration value=">" />
          <xsd:enumeration value=">=" />
      </xsd:restriction>
  </xsd:simpleType>
  <xsd:element name="valueList" type="xsd:string" />
  <xsd:element name="negotiable" type="lpdl:negotiableType" />
- <xsd:simpleType name="negotiableType">
    - <xsd:restriction base="xsd:string">
          <xsd:enumeration value="yes" />
          <xsd:enumeration value="no" />
      </xsd:restriction>
  </xsd:simpleType>
  <xsd:element name="priority" type="xsd:positiveInteger" />
  <xsd:element name="output_constraint" type="lpdl:attributeconstraintType" />
  <xsd:element name="operation_constraint" type="lpdl:attributeconstraintType" />
  <xsd:element name="service_constraints" type="lpdl:serviceconstraintsType" />
- <xsd:complexType name="serviceconstraintsType">
    - <xsd:sequence>
          <xsd:element ref="lpdl:service_constraint" minOccurs="0" maxOccurs="unbounded" />
          <xsd:any namespace="##other" minOccurs="0" />
      </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="service_constraint" type="lpdl:attributeconstraintType" />
  <xsd:element name="interattribute_constraints" type="lpdl:interattributeconstraintsType" />
- <xsd:complexType name="interattributeconstraintsType">
    - <xsd:sequence>
          <xsd:element ref="lpdl:interattribute_constraint" minOccurs="0" maxOccurs="unbounded" />
          <xsd:any namespace="##other" minOccurs="0" />
      </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="interattribute_constraint" type="lpdl:interattributeconstraintType" />
- <xsd:complexType name="interattributeconstraintType">
    - <xsd:sequence>
          <xsd:element ref="lpdl:name" />
          <xsd:element name="antecedence" type="xsd:string" />
          <xsd:element name="consequence" type="xsd:string" />
          <xsd:element ref="lpdl:negotiable" />
          <xsd:element ref="lpdl:priority" />
          <xsd:any namespace="##other" minOccurs="0" />
      </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:schema>
```

APPENDIX B
XML SCHEMA DEFINITION OF LEARNING PROCESS DEFINITION LANGUAGE

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <xsd:schema targetNamespace="http://www.cise.ufl.edu/~glee/LPDL1_0"
      xmlns:lpdl="http://www.cise.ufl.edu/~glee/LPDL1_0"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
      attributeFormDefault="unqualified">
    - <xsd:annotation>
          <xsd:documentation xml:lang="en">XML Schema for Learning Process Definition Model Language
              1.0 Copyright by Gilliean Lee. 2003.</xsd:documentation>
      </xsd:annotation>
    - <xsd:element name="Connector">
        - <xsd:complexType>
            - <xsd:sequence>
                  <xsd:element name="SiblingActivityID" type="xsd:NMTOKEN" minOccurs="1"
                      maxOccurs="unbounded" />
              </xsd:sequence>
              <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required" />
              <xsd:attribute name="Name" type="xsd:string" />
              <xsd:attribute name="SequencingControlMode" type="lpdl:SequencingControlModes"
                  use="optional" default="Choice" />
          </xsd:complexType>
      </xsd:element>
    - <xsd:element name="Edge">
        - <xsd:complexType>
              <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required" />
              <xsd:attribute name="From" type="xsd:NMTOKEN" use="required" />
              <xsd:attribute name="To" type="xsd:NMTOKEN" use="required" />
              <xsd:attribute name="Name" type="xsd:string" />
          </xsd:complexType>
      </xsd:element>
    - <xsd:element name="LeafActivity">
        - <xsd:complexType>
            - <xsd:sequence>
                  <xsd:element name="UpdateTime" type="xsd:unsignedLong" minOccurs="0" />
                  <xsd:element ref="lpdl:Description" minOccurs="0" />
                  <xsd:element ref="lpdl:Limit" />
                  <xsd:element ref="lpdl:Objectives" minOccurs="0" />
                  <xsd:element ref="lpdl:ActivityData" minOccurs="0" />
                  <xsd:element ref="lpdl:BindingInfo" />
                  <xsd:element ref="lpdl:LeafActivityActions" minOccurs="0" />
                  <xsd:element ref="lpdl:LeafConditionActionRules" minOccurs="0" />
                  <xsd:element ref="lpdl:LeafAssessmentMarker" minOccurs="0" />
                  <xsd:element ref="lpdl:AssessmentParameters" minOccurs="0" />
              </xsd:sequence>
              <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required" />
              <xsd:attribute name="Name" type="xsd:string" />
          </xsd:complexType>
      </xsd:element>
    - <xsd:element name="NonleafActivity">
        - <xsd:complexType>
            - <xsd:sequence>
                  <xsd:element name="UpdateTime" type="xsd:unsignedLong" minOccurs="0" />
                  <xsd:element ref="lpdl:Description" minOccurs="0" />
                  <xsd:element ref="lpdl:Limit" />
```

```
                <xsd:element ref="lpdl:Objectives" minOccurs="0" />
                <xsd:element ref="lpdl:ActivityData" minOccurs="0" />
                <xsd:element ref="lpdl:NonleafActivityActions" minOccurs="0" />
                <xsd:element ref="lpdl:NonleafConditionActionRules" minOccurs="0" />
                <xsd:element ref="lpdl:NonleafAssessmentMarker" minOccurs="0" />
                <xsd:element ref="lpdl:NonleafContentItems" minOccurs="0" />
                <xsd:element ref="lpdl:NonleafPracticeItems" minOccurs="0" />
                <xsd:element ref="lpdl:NonleafAssessments" minOccurs="0" />
                <xsd:element ref="lpdl:AssessmentParameters" minOccurs="0" />
            </xsd:sequence>
            <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required" />
            <xsd:attribute name="Name" type="xsd:string" />
        </xsd:complexType>
    </xsd:element>
- <xsd:element name="NonleafContentItems">
    - <xsd:complexType>
        - <xsd:sequence>
                <xsd:element name="Introduction" type="lpdl:NonleafContentItem" minOccurs="0" />
                <xsd:element name="Summary" type="lpdl:NonleafContentItem" minOccurs="0" />
            </xsd:sequence>
    </xsd:complexType>
    </xsd:element>
- <xsd:complexType name="NonleafContentItem">
    - <xsd:choice>
            <xsd:element name="html" type="xsd:string" />
            <xsd:element name="text" type="xsd:string" />
            <xsd:element name="URL" type="xsd:anyURI" />
        </xsd:choice>
    </xsd:complexType>
- <xsd:element name="NonleafPracticeItems">
    - <xsd:complexType>
        - <xsd:sequence>
            <xsd:element name="NonleafPracticeItem" type="lpdl:PAItem" minOccurs="0"
                maxOccurs="unbounded" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
- <xsd:complexType name="PAItem">
    - <xsd:sequence>
            <xsd:element name="Problem" type="lpdl:NonleafContentItem" />
            <xsd:element name="Answer" type="xsd:string" />
        </xsd:sequence>
        <xsd:attribute name="Number" use="required" type="xsd:unsignedInt" />
    </xsd:complexType>
- <xsd:element name="NonleafAssessments">
    - <xsd:complexType>
        - <xsd:sequence>
                <xsd:element name="PreAssessmentItems" type="lpdl:NonleafAssessmentItems"
                    minOccurs="0" />
                <xsd:element name="PostAssessmentItems" type="lpdl:NonleafAssessmentItems"
                    minOccurs="0" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
- <xsd:complexType name="NonleafAssessmentItems">
    - <xsd:sequence>
            <xsd:element name="AssessmentItem" type="lpdl:NonleafAssessmentItem" minOccurs="0"
                maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
- <xsd:complexType name="NonleafAssessmentItem">
    - <xsd:complexContent>
        - <xsd:extension base="lpdl:PAItem">
```

```
            - <xsd:sequence>
                  <xsd:element ref="lpdl:Code" minOccurs="0" />
              </xsd:sequence>
              <xsd:attribute name="Points" use="required" type="xsd:decimal" />
              <xsd:attribute name="AutoGrade" use="required" type="xsd:boolean" />
          </xsd:extension>
      </xsd:complexContent>
  </xsd:complexType>
  <xsd:element name="Description" type="xsd:string" />
- <xsd:element name="ActivityData">
    - <xsd:complexType>
        - <xsd:sequence>
              <xsd:element name="Data" type="lpdl:Data" minOccurs="0" maxOccurs="unbounded" />
          </xsd:sequence>
      </xsd:complexType>
  </xsd:element>
- <xsd:element name="LeafAssessmentMarker">
    - <xsd:complexType>
          <xsd:attribute name="PreAssessment" type="xsd:boolean" use="optional" default="false" />
          <xsd:attribute name="PostAssessment" type="xsd:boolean" use="optional" default="false" />
      </xsd:complexType>
  </xsd:element>
- <xsd:element name="NonleafAssessmentMarker">
    - <xsd:complexType>
          <xsd:attribute name="PreAssessment" type="xsd:boolean" use="optional" default="false" />
          <xsd:attribute name="PostAssessment" type="xsd:boolean" use="optional" default="false" />
        - <xsd:attribute name="PreAssessmentMode" use="optional" default="ThisActivityOnly">
            - <xsd:simpleType>
                - <xsd:restriction base="xsd:string">
                      <xsd:enumeration value="IncludeSubtree" />
                      <xsd:enumeration value="ThisActivityOnly" />
                  </xsd:restriction>
              </xsd:simpleType>
          </xsd:attribute>
      </xsd:complexType>
  </xsd:element>
- <xsd:element name="AssessmentParameters">
    - <xsd:complexType>
        - <xsd:sequence>
              <xsd:element name="PreAssessmentParameters"
                  type="lpdl:AssessmentSelectionParameters" minOccurs="0" />
              <xsd:element name="PostAssessmentParameters"
                  type="lpdl:AssessmentSelectionParameters" minOccurs="0" />
              <xsd:element name="PreAssessmentObjectiveSettings" type="lpdl:ObjectiveSettings"
                  minOccurs="0" />
              <xsd:element name="PostAssessmentObjectiveSettings" type="lpdl:ObjectiveSettings"
                  minOccurs="0" />
          </xsd:sequence>
      </xsd:complexType>
  </xsd:element>
- <xsd:complexType name="AssessmentSelectionParameters">
    - <xsd:sequence>
          <xsd:element ref="lpdl:AssessmentSelectionParameter" minOccurs="0"
              maxOccurs="unbounded" />
      </xsd:sequence>
  </xsd:complexType>
- <xsd:element name="AssessmentSelectionParameter">
    - <xsd:complexType>
          <xsd:attribute name="ActivityID" type="xsd:NMTOKEN" use="required" />
          <xsd:attribute name="ScoringWeight" type="lpdl:Percent" use="required" />
          <xsd:attribute name="SelectionPercentage" type="lpdl:Percent" use="required" />
      </xsd:complexType>
```

```
  </xsd:element>
- <xsd:complexType name="ObjectiveSettings">
    - <xsd:sequence>
          <xsd:element ref="lpdl:ObjectiveSetting" minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
  </xsd:complexType>
- <xsd:element name="ObjectiveSetting">
    - <xsd:complexType>
        - <xsd:sequence>
              <xsd:element name="ActivityID" type="xsd:NMTOKEN" minOccurs="0"
                  maxOccurs="unbounded" />
          </xsd:sequence>
          <xsd:attribute name="ObjectiveID" type="xsd:NMTOKEN" use="required" />
      </xsd:complexType>
  </xsd:element>
- <xsd:element name="NonleafActivityActions">
    - <xsd:complexType>
        - <xsd:sequence>
              <xsd:element name="BeforePreActivity_Actions" type="xsd:string" minOccurs="0" />
              <xsd:element name="BeforePreAssessment_Actions" type="xsd:string" minOccurs="0" />
              <xsd:element name="BeforeIntroPresentation_Actions" type="xsd:string" minOccurs="0"
                  />
              <xsd:element name="AfterIntroPresentation_Actions" type="xsd:string" minOccurs="0"
                  />
              <xsd:element name="BeforeRollup_Actions" type="xsd:string" minOccurs="0" />
              <xsd:element name="BeforePostAssessment_Actions" type="xsd:string" minOccurs="0"
                  />
              <xsd:element name="AfterPostAssessment_Actions" type="xsd:string" minOccurs="0" />
          </xsd:sequence>
      </xsd:complexType>
  </xsd:element>
- <xsd:element name="LeafActivityActions">
    - <xsd:complexType>
        - <xsd:sequence>
              <xsd:element name="BeforePreActivity_Actions" type="xsd:string" minOccurs="0" />
              <xsd:element name="BeforeBinding_Actions" type="xsd:string" minOccurs="0" />
              <xsd:element name="BeforePreAssessment_Actions" type="xsd:string" minOccurs="0" />
              <xsd:element name="BeforeContentPresentation_Actions" type="xsd:string"
                  minOccurs="0" />
              <xsd:element name="BeforePostAssessment_Actions" type="xsd:string" minOccurs="0"
                  />
              <xsd:element name="AfterPostAssessment_Actions" type="xsd:string" minOccurs="0" />
          </xsd:sequence>
      </xsd:complexType>
  </xsd:element>
- <xsd:complexType name="Actions">
    - <xsd:sequence>
          <xsd:element ref="lpdl:Action" minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
  </xsd:complexType>
- <xsd:element name="Action">
    - <xsd:complexType>
        - <xsd:choice>
              <xsd:element ref="lpdl:Code" />
              <xsd:element ref="lpdl:LMSAction" />
              <xsd:element ref="lpdl:WebServiceCall" />
          </xsd:choice>
      </xsd:complexType>
  </xsd:element>
- <xsd:element name="LMSAction">
    - <xsd:complexType>
        - <xsd:choice>
```

```
                <xsd:element name="PreconditionAction" type="lpdl:PreconditionAction" />
                <xsd:element name="PostconditionAction" type="lpdl:PostconditionAction" />
                <xsd:element name="RollupAction" type="lpdl:RollupAction" />
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
  - <xsd:simpleType name="PreconditionAction">
      - <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Skip" />
            <xsd:enumeration value="Disabled" />
            <xsd:enumeration value="Hidden from Choice" />
            <xsd:enumeration value="Stop Forward Traversal" />
            <xsd:enumeration value="Ignore" />
        </xsd:restriction>
    </xsd:simpleType>
  - <xsd:simpleType name="PostconditionAction">
      - <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Exit Parent" />
            <xsd:enumeration value="Exit All" />
            <xsd:enumeration value="Retry" />
            <xsd:enumeration value="Retry All" />
            <xsd:enumeration value="Continue" />
            <xsd:enumeration value="Previous" />
            <xsd:enumeration value="Ignore" />
        </xsd:restriction>
    </xsd:simpleType>
  - <xsd:simpleType name="RollupAction">
      - <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Satisfied" />
            <xsd:enumeration value="Not Satisfied" />
            <xsd:enumeration value="Completed" />
            <xsd:enumeration value="Incomplete" />
            <xsd:enumeration value="PostAssess" />
        </xsd:restriction>
    </xsd:simpleType>
  - <xsd:simpleType name="ChildActivitySet">
      - <xsd:restriction base="xsd:string">
            <xsd:enumeration value="All" />
            <xsd:enumeration value="Any" />
            <xsd:enumeration value="None" />
            <xsd:enumeration value="At Least Count" />
            <xsd:enumeration value="At Least Percent" />
        </xsd:restriction>
    </xsd:simpleType>
  - <xsd:simpleType name="Percent">
      - <xsd:restriction base="xsd:decimal">
            <xsd:minInclusive value="0.0" />
            <xsd:maxInclusive value="1.0" />
        </xsd:restriction>
    </xsd:simpleType>
  - <xsd:simpleType name="MinSet">
        <xsd:union memberTypes="lpdl:Percent xsd:positiveInteger" />
    </xsd:simpleType>
  - <xsd:element name="BindingInfo">
      - <xsd:complexType>
          - <xsd:choice>
                <xsd:element ref="lpdl:DynamicBindingInfo" />
                <xsd:element ref="lpdl:StaticBindingInfo" />
            </xsd:choice>
            <xsd:attribute name="Mode" type="lpdl:BindingMode" />
            <xsd:attribute name="BrokerURL" type="xsd:anyURI" />
        </xsd:complexType>
```

```
  </xsd:element>
- <xsd:element name="DynamicBindingInfo">
    - <xsd:complexType>
        - <xsd:sequence>
              <xsd:element ref="lpdl:constraints" />
          </xsd:sequence>
      </xsd:complexType>
  </xsd:element>
- <xsd:element name="StaticBindingInfo">
    - <xsd:complexType>
        - <xsd:sequence>
              <xsd:element name="BusinessServiceKey" type="xsd:string" />
          </xsd:sequence>
      </xsd:complexType>
  </xsd:element>
- <xsd:element name="RollupCARule">
    - <xsd:complexType>
        - <xsd:sequence>
              <xsd:element name="CARule" type="lpdl:Trigger" minOccurs="0" />
          </xsd:sequence>
          <xsd:attribute name="ChildActivitySet" type="lpdl:ChildActivitySet" default="All" />
          <xsd:attribute name="MinimumSet" type="lpdl:MinSet" use="optional" />
      </xsd:complexType>
  </xsd:element>
- <xsd:element name="Objectives">
    - <xsd:complexType>
        - <xsd:sequence>
              <xsd:element ref="lpdl:Objective" minOccurs="0" maxOccurs="unbounded" />
         </xsd:sequence>
     </xsd:complexType>
  </xsd:element>
- <xsd:element name="Objective">
    - <xsd:complexType>
        - <xsd:sequence>
              <xsd:element ref="lpdl:AttainmentLevel" minOccurs="0" maxOccurs="unbounded" />
          </xsd:sequence>
          <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required" />
          <xsd:attribute name="SatisfiedByMeasure" type="xsd:boolean" default="false" />
          <xsd:attribute name="MinimumSatisfiedNormalizedMeasure" type="lpdl:NormalizedValue" />
          <xsd:attribute name="ContributesToRollup" type="xsd:boolean" default="false" />
      </xsd:complexType>
  </xsd:element>
- <xsd:element name="AttainmentLevel">
    - <xsd:complexType>
          <xsd:attribute name="Comments" type="xsd:string" use="required" />
          <xsd:attribute name="MaxNormalizedScore" type="lpdl:NormalizedValue" use="required" />
          <xsd:attribute name="MinNormalizedScore" type="lpdl:NormalizedValue" use="required" />
      </xsd:complexType>
  </xsd:element>
- <xsd:element name="NonleafConditionActionRules">
    - <xsd:complexType>
        - <xsd:sequence>
              <xsd:element name="PreActivityCARule" type="lpdl:Trigger" minOccurs="0" />
              <xsd:element name="PreAssessedCARule" type="lpdl:Trigger" minOccurs="0" />
              <xsd:element name="DrillDownCARule" type="lpdl:Trigger" minOccurs="0" />
              <xsd:element ref="lpdl:RollupCARule" minOccurs="0" />
              <xsd:element name="PostAssessedCARule" type="lpdl:Trigger" minOccurs="0" />
          </xsd:sequence>
      </xsd:complexType>
  </xsd:element>
- <xsd:element name="LeafConditionActionRules">
    - <xsd:complexType>
```

```
        - <xsd:sequence>
              <xsd:element name="PreActivityCARule" type="lpdl:Trigger" minOccurs="0" />
              <xsd:element name="PreAssessedCARule" type="lpdl:Trigger" minOccurs="0" />
              <xsd:element name="PostAssessedCARule" type="lpdl:Trigger" minOccurs="0" />
          </xsd:sequence>
      </xsd:complexType>
  </xsd:element>
- <xsd:simpleType name="NormalizedValue">
    - <xsd:restriction base="xsd:decimal">
          <xsd:minInclusive value="-1.0" />
          <xsd:maxInclusive value="1.0" />
      </xsd:restriction>
  </xsd:simpleType>
- <xsd:complexType name="ParamMappings">
    - <xsd:sequence>
          <xsd:element name="ParamMapping" type="lpdl:ParamMapping" minOccurs="0"
              maxOccurs="unbounded" />
      </xsd:sequence>
  </xsd:complexType>
- <xsd:complexType name="ParamMapping">
      <xsd:attribute name="FromId" type="xsd:NMTOKEN" />
      <xsd:attribute name="ToId" type="xsd:NMTOKEN" />
  </xsd:complexType>
- <xsd:complexType name="Data">
      <xsd:attribute name="DataId" type="xsd:NMTOKEN" use="required" />
      <xsd:attribute name="DataType" type="xsd:NMTOKEN" use="required" />
  </xsd:complexType>
- <xsd:complexType name="Event">
    - <xsd:sequence>
          <xsd:element name="EventParameter" type="lpdl:Data" minOccurs="0"
              maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:attribute name="EventId" type="xsd:NMTOKEN" use="required" />
  </xsd:complexType>
- <xsd:complexType name="ConditionActionRule">
    - <xsd:sequence>
          <xsd:element name="Condition" type="xsd:string" minOccurs="0" />
          <xsd:element name="Actions" type="xsd:string" minOccurs="0" />
          <xsd:element name="AltActions" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
  </xsd:complexType>
- <xsd:complexType name="Trigger">
    - <xsd:sequence>
          <xsd:element name="Event" type="lpdl:Event" minOccurs="0" />
          <xsd:element name="ConditionActionRule" type="lpdl:ConditionActionRule" />
      </xsd:sequence>
  </xsd:complexType>
- <xsd:element name="Limit">
    - <xsd:complexType>
        - <xsd:choice>
              <xsd:element name="Duration" type="xsd:duration" />
              <xsd:element name="NumOfAttempts" type="xsd:positiveInteger" />
          </xsd:choice>
      </xsd:complexType>
  </xsd:element>
  <xsd:element name="Code" type="xsd:string" />
- <xsd:element name="WebServiceCall">
    - <xsd:complexType>
        - <xsd:sequence>
              <xsd:element name="Location" type="xsd:anyURI" minOccurs="0" />
              <xsd:element name="InputParamMappings" type="lpdl:ParamMappings" minOccurs="0"
                  />
```

```
            <xsd:element name="OutputParamMapping" type="lpdl:ParamMapping" minOccurs="0" />
        </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
- <xsd:simpleType name="SequencingControlModes">
    <xsd:list itemType="lpdl:SCModes" />
  </xsd:simpleType>
- <xsd:simpleType name="SCModes">
  - <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="Choice" />
        <xsd:enumeration value="ForwardOnly" />
        <xsd:enumeration value="Flow" />
        <xsd:enumeration value="ChoiceExit" />
    </xsd:restriction>
  </xsd:simpleType>
- <xsd:simpleType name="BindingMode">
  - <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="Static" />
        <xsd:enumeration value="Dynamic" />
    </xsd:restriction>
  </xsd:simpleType>
- <xsd:element name="UDDIRegistration">
  - <xsd:complexType>
        <xsd:sequence />
        <xsd:attribute name="BusinessInterfaceKey" type="xsd:string" />
        <xsd:attribute name="ServiceImplementationKey" type="xsd:string" />
    </xsd:complexType>
  </xsd:element>
- <xsd:element name="LearningProcess">
  - <xsd:complexType>
    - <xsd:sequence>
            <xsd:element name="SavedTime" type="xsd:unsignedLong" minOccurs="0" />
            <xsd:element ref="lpdl:LeafActivity" minOccurs="0" maxOccurs="unbounded" />
            <xsd:element ref="lpdl:NonleafActivity" minOccurs="0" maxOccurs="unbounded" />
            <xsd:element ref="lpdl:Connector" minOccurs="0" maxOccurs="unbounded" />
            <xsd:element ref="lpdl:Edge" minOccurs="0" maxOccurs="unbounded" />
            <xsd:element ref="lpdl:UDDIRegistration" minOccurs="0" maxOccurs="unbounded" />
            <xsd:element ref="lpdl:constraints" minOccurs="0" />
        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required" />
        <xsd:attribute name="Name" type="xsd:string" />
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="constraints" type="lpdl:constraintsType" />
- <xsd:complexType name="constraintsType">
  - <xsd:sequence>
        <xsd:element ref="lpdl:constraint" minOccurs="0" maxOccurs="unbounded" />
        <xsd:element ref="lpdl:service_constraints" minOccurs="0" maxOccurs="unbounded" />
        <xsd:element ref="lpdl:interattribute_constraints" minOccurs="0" maxOccurs="unbounded" />
        <xsd:any namespace="##other" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="constraint" type="lpdl:constraintType" />
- <xsd:complexType name="constraintType">
  - <xsd:sequence>
        <xsd:element ref="lpdl:input_constraint" minOccurs="0" maxOccurs="unbounded" />
        <xsd:element ref="lpdl:output_constraint" minOccurs="0" maxOccurs="unbounded" />
        <xsd:element ref="lpdl:operation_constraint" minOccurs="0" maxOccurs="unbounded" />
        <xsd:any namespace="##other" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="input_constraint" type="lpdl:attributeconstraintType" />
```

```
- <xsd:complexType name="attributeconstraintType">
    - <xsd:sequence>
          <xsd:element ref="lpdl:name" minOccurs="1" maxOccurs="1" />
          <xsd:element ref="lpdl:type" minOccurs="1" maxOccurs="1" />
          <xsd:element ref="lpdl:keyword" minOccurs="1" maxOccurs="1" />
          <xsd:element ref="lpdl:valueList" minOccurs="1" maxOccurs="1" />
          <xsd:element ref="lpdl:negotiable" minOccurs="1" maxOccurs="1" />
          <xsd:element ref="lpdl:priority" minOccurs="1" maxOccurs="1" />
          <xsd:any namespace="##other" minOccurs="0" />
      </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="name" type="xsd:string" />
  <xsd:element name="type" type="lpdl:typeType" />
- <xsd:simpleType name="typeType">
    - <xsd:restriction base="xsd:string">
          <xsd:enumeration value="string" />
          <xsd:enumeration value="float" />
          <xsd:enumeration value="integer" />
          <xsd:enumeration value="boolean" />
          <xsd:enumeration value="datetime" />
          <xsd:enumeration value="date" />
          <xsd:enumeration value="time" />
          <xsd:enumeration value="duration" />
      </xsd:restriction>
  </xsd:simpleType>
  <xsd:element name="keyword" type="lpdl:keywordType" />
- <xsd:simpleType name="keywordType">
    - <xsd:restriction base="xsd:string">
          <xsd:enumeration value="range" />
          <xsd:enumeration value="enu" />
          <xsd:enumeration value="eq" />
          <xsd:enumeration value="!=" />
          <xsd:enumeration value="<" />
          <xsd:enumeration value="<=" />
          <xsd:enumeration value=">" />
          <xsd:enumeration value=">=" />
      </xsd:restriction>
  </xsd:simpleType>
  <xsd:element name="valueList" type="xsd:string" />
  <xsd:element name="negotiable" type="lpdl:negotiableType" />
- <xsd:simpleType name="negotiableType">
    - <xsd:restriction base="xsd:string">
          <xsd:enumeration value="yes" />
          <xsd:enumeration value="no" />
      </xsd:restriction>
  </xsd:simpleType>
  <xsd:element name="priority" type="xsd:positiveInteger" />
  <xsd:element name="output_constraint" type="lpdl:attributeconstraintType" />
  <xsd:element name="operation_constraint" type="lpdl:attributeconstraintType" />
  <xsd:element name="service_constraints" type="lpdl:serviceconstraintsType" />
- <xsd:complexType name="serviceconstraintsType">
    - <xsd:sequence>
          <xsd:element ref="lpdl:service_constraint" minOccurs="0" maxOccurs="unbounded" />
          <xsd:any namespace="##other" minOccurs="0" />
      </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="service_constraint" type="lpdl:attributeconstraintType" />
  <xsd:element name="interattribute_constraints" type="lpdl:interattributeconstraintsType" />
- <xsd:complexType name="interattributeconstraintsType">
    - <xsd:sequence>
          <xsd:element ref="lpdl:interattribute_constraint" minOccurs="0" maxOccurs="unbounded" />
          <xsd:any namespace="##other" minOccurs="0" />
```

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="interattribute_constraint" type="lpdl:interattributeconstraintType" />
  - <xsd:complexType name="interattributeconstraintType">
      - <xsd:sequence>
            <xsd:element ref="lpdl:name" />
            <xsd:element name="antecedence" type="xsd:string" />
            <xsd:element name="consequence" type="xsd:string" />
            <xsd:element ref="lpdl:negotiable" />
            <xsd:element ref="lpdl:priority" />
            <xsd:any namespace="##other" minOccurs="0" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

APPENDIX C
SAMPLE ATOMIC LEARNING OBJECT AND COMPOSITE LEARNING OBJECT

C.1 Sample Atomic Learning Object – Low Pressure System


```
<?xml version="1.0"?>
<ns1:LearningObject xmlns:ns1="http://www.cise.ufl.edu/~glee/LCDL1_0" Name="LowPressureSystem"
Id="LowPressureSystem">
  <ns1:ContentItems>
    <ns1:ContentItem Name="Q1" Number="1">
      <ns1:URL>http://128.227.176.41:8888/ VAM/Low_pressure_system_Q1.htm</ns1:URL>
    </ns1:ContentItem>
    <ns1:ContentItem Name="Low Pressure System Learning Objective" Number="2">
      <ns1:URL>http://128.227.176.41:8888/
                    VAM/Low_pressure_system_learning_objectives.htm</ns1:URL>
    </ns1:ContentItem>
    <ns1:ContentItem Name="Q2" Number="3">
      <ns1:URL>http://128.227.176.41:8888/ VAM/Low_pressure_system_Q2.htm</ns1:URL>
    </ns1:ContentItem>
    <ns1:ContentItem Name="Q1 Learning Objectives" Number="4">
      <ns1:URL>http://128.227.176.41:8888/
                    VAM/Low_pressure_system_Q1_Objectives.htm</ns1:URL>
    </ns1:ContentItem>
    <ns1:ContentItem Name="Q2 Learning Objectives" Number="5">
      <ns1:URL>http://128.227.176.41:8888/
                    VAM/Low_pressure_system_Q2_Objectives.htm</ns1:URL>
    </ns1:ContentItem>
    <ns1:ContentItem Name="Q3" Number="6">
      <ns1:URL>http://128.227.176.41:8888/ VAM/Low_pressure_system_Q3.htm</ns1:URL>
    </ns1:ContentItem>
    <ns1:ContentItem Name="Q3 Learning Objectives" Number="7">
      <ns1:URL>http://128.227.176.41:8888/
                    VAM/Low_pressure_system_Q3_Objectives.htm</ns1:URL>
    </ns1:ContentItem>
    <ns1:ContentItem Name="Q4" Number="8">
      <ns1:URL>http://128.227.176.41:8888/ VAM/Low_pressure_system_Q4.htm</ns1:URL>
    </ns1:ContentItem>
    <ns1:ContentItem Name="Q4 Learning Objectives" Number="9">
      <ns1:URL>http://128.227.176.41:8888/
                    VAM/Low_pressure_system_Q4_Objectives.htm</ns1:URL>
    </ns1:ContentItem>
    <ns1:ContentItem Name="Q5" Number="10">
      <ns1:URL>http://128.227.176.41:8888/ VAM/Low_pressure_system_Q5.htm</ns1:URL>
    </ns1:ContentItem>
    <ns1:ContentItem Name="Q5 Learning Objectives" Number="11">
      <ns1:URL>http://128.227.176.41:8888/
                      VAM/Low_pressure_system_Q5_Objectives.htm</ns1:URL>
    </ns1:ContentItem>
    <ns1:ContentItem Name="Q6" Number="12">
      <ns1:URL>http://128.227.176.41:8888/ VAM/Low_pressure_system_Q6.htm</ns1:URL>
    </ns1:ContentItem>
    <ns1:ContentItem Name="Q6 Learning Objectives" Number="13">
      <ns1:URL>http://128.227.176.41:8888/
                    VAM/Low_pressure_system_Q6_Objectives.htm</ns1:URL>
    </ns1:ContentItem>
  </ns1:ContentItems>
  <ns1:PracticeItems>
    <ns1:PracticeItem Number="1" Name="Q1 VAM Simulation">
```

```
        <ns1:Problem Number="0">
          <ns1:URL>http://128.227.176.41:8888/
                          VAM/Low_pressure_system_Q1_VAM_simulation.htm</ns1:URL>
        </ns1:Problem>
        <ns1:Answer>N/A</ns1:Answer>
      </ns1:PracticeItem>
      <ns1:PracticeItem Number="2" Name="Q2 VAM Simulation">
        <ns1:Problem Number="0">
          <ns1:URL>http://128.227.176.41:8888/
                          VAM/Low_pressure_system_Q2_VAM_simulation.htm</ns1:URL>
        </ns1:Problem>
        <ns1:Answer>N/A</ns1:Answer>
      </ns1:PracticeItem>
      <ns1:PracticeItem Number="3" Name="Q3 VAM Simulation">
        <ns1:Problem Number="0">
          <ns1:URL>http://128.227.176.41:8888/
                          VAM/Low_pressure_system_Q3_VAM_simulation.htm</ns1:URL>
        </ns1:Problem>
        <ns1:Answer>N/A</ns1:Answer>
      </ns1:PracticeItem>
      <ns1:PracticeItem Number="4" Name="Q4 VAM Simulation">
        <ns1:Problem Number="0">
          <ns1:URL>http://128.227.176.41:8888/
                          VAM/Low_pressure_system_Q4_VAM_simulation.htm</ns1:URL>
        </ns1:Problem>
        <ns1:Answer>N/A</ns1:Answer>
      </ns1:PracticeItem>
      <ns1:PracticeItem Number="5" Name="Q5 VAM Simulation">
        <ns1:Problem Number="0">
          <ns1:URL>http://128.227.176.41:8888/
                          VAM/Low_pressure_system_Q5_VAM_simulation.htm</ns1:URL>
        </ns1:Problem>
        <ns1:Answer>N/A</ns1:Answer>
      </ns1:PracticeItem>
      <ns1:PracticeItem Number="6" Name="Q6 VAM Simulation">
        <ns1:Problem Number="0">
          <ns1:URL>http://128.227.176.41:8888/
                          VAM/Low_pressure_system_Q6_VAM_simulation.htm</ns1:URL>
        </ns1:Problem>
        <ns1:Answer>N/A</ns1:Answer>
      </ns1:PracticeItem>
    </ns1:PracticeItems>
    <ns1:Assessments>
      <ns1:PreAssessmentItems />
      <ns1:PostAssessmentItems>
        <ns1:AssessmentItem Points="3" AutoGrade="true" Number="1">
          <ns1:Problem Number="0">
            <ns1:text>&lt;p&gt; Q. If a vaporizer concentration dial is adjusted to a non-zero setting,
will O2 flow generated by an O2 flush flow through the vaporizer?  &lt;/p&gt;

&lt;p&gt; A. No &lt;/p&gt;
&lt;p&gt; B. Yes &lt;/p&gt;
&lt;p&gt; C. Depends on the setting of the O2 flowmeter knob &lt;/p&gt;
&lt;p&gt; D. Depends on the FGF setting &lt;/p&gt;</ns1:text>
          </ns1:Problem>
          <ns1:Answer>A</ns1:Answer>
        </ns1:AssessmentItem>
        <ns1:AssessmentItem Points="3" AutoGrade="true" Number="2">
          <ns1:Problem Number="0">
            <ns1:text>&lt;p&gt; Q. With the N2O flow meter at its maximum setting (&gt; 5 L/min),
will a hypoxic gas mixture result at the common gas outlet if the O2 flow meter is subsequently decreased
to its minimum setting (about 200 mL/min)? &lt;/p&gt;

&lt;p&gt; A. Yes &lt;/p&gt;
&lt;p&gt; B. No &lt;/p&gt;
&lt;p&gt; C. Maybe &lt;/p&gt;</ns1:text>
```

```
            </ns1:Problem>
            <ns1:Answer>B</ns1:Answer>
         </ns1:AssessmentItem>
         <ns1:AssessmentItem Points="3" AutoGrade="true" Number="3">
            <ns1:Problem Number="0">
               <ns1:text>&lt;p&gt; Q. Will liquid anesthetic spill from a loose filler cap if the vaporizer
concentration dial is set to 0? &lt;/p&gt;

&lt;p&gt; A. Yes &lt;/p&gt;
&lt;p&gt; B. No &lt;/p&gt;
&lt;p&gt; C. Only if the peak inspiratory pressure exceeds 20 cm H2O &lt;/p&gt;</ns1:text>
            </ns1:Problem>
            <ns1:Answer>B</ns1:Answer>
         </ns1:AssessmentItem>
         <ns1:AssessmentItem Points="3" AutoGrade="true" Number="4">
            <ns1:Problem Number="0">
               <ns1:text>&lt;p&gt; Q. In an O2/N2O anesthesia machine, the O2 flow meter is always
downstream of the N2O flow meter. &lt;/p&gt;

&lt;p&gt; A. True &lt;/p&gt;
&lt;p&gt; B. False &lt;/p&gt;
&lt;p&gt; C. Depends if there is a third gas like air or helium &lt;/p&gt;</ns1:text>
            </ns1:Problem>
            <ns1:Answer>A</ns1:Answer>
         </ns1:AssessmentItem>
      </ns1:PostAssessmentItems>
   </ns1:Assessments>
   <ns1:LearningItemSequences>
      <ns1:LearningItemSequence Type="Content" ItemId="2" />
      <ns1:LearningItemSequence Type="Content" ItemId="1" />
      <ns1:LearningItemSequence Type="Practice" ItemId="1" />
      <ns1:LearningItemSequence Type="Content" ItemId="4" />
      <ns1:LearningItemSequence Type="Content" ItemId="3" />
      <ns1:LearningItemSequence Type="Practice" ItemId="2" />
      <ns1:LearningItemSequence Type="Content" ItemId="5" />
      <ns1:LearningItemSequence Type="Content" ItemId="6" />
      <ns1:LearningItemSequence Type="Practice" ItemId="3" />
      <ns1:LearningItemSequence Type="Content" ItemId="7" />
      <ns1:LearningItemSequence Type="Content" ItemId="8" />
      <ns1:LearningItemSequence Type="Practice" ItemId="4" />
      <ns1:LearningItemSequence Type="Content" ItemId="9" />
      <ns1:LearningItemSequence Type="Content" ItemId="10" />
      <ns1:LearningItemSequence Type="Practice" ItemId="5" />
      <ns1:LearningItemSequence Type="Content" ItemId="11" />
      <ns1:LearningItemSequence Type="Content" ItemId="12" />
      <ns1:LearningItemSequence Type="Practice" ItemId="6" />
      <ns1:LearningItemSequence Type="Content" ItemId="13" />
   </ns1:LearningItemSequences>
   <ns1:UDDIRegistration BusinessInterfaceKey="UUID:8ACC6F10-115B-11D9-A2B3-000629DC0A53"
            ServiceImplementationKey="5B863DF0-7D6A-11D9-9561-000629DC0A53" />
   <ns1:constraints>
      <ns1:service_constraints>
         <ns1:service_constraint>
            <ns1:name>Title</ns1:name>
            <ns1:type>string</ns1:type>
            <ns1:keyword>eq</ns1:keyword>
            <ns1:valueList>&quot;Low Pressure System - VAM&quot;</ns1:valueList>
            <ns1:negotiable>no</ns1:negotiable>
            <ns1:priority>0</ns1:priority>
         </ns1:service_constraint>
         <ns1:service_constraint>
            <ns1:name>Subject</ns1:name>
            <ns1:type>string</ns1:type>
            <ns1:keyword>eq</ns1:keyword>
            <ns1:valueList>&quot;Low Pressure System of Anesthesia Machine&quot;</ns1:valueList>
            <ns1:negotiable>no</ns1:negotiable>
```

```
            <ns1:priority>0</ns1:priority>
          </ns1:service_constraint>
          <ns1:service_constraint>
            <ns1:name>Keywords</ns1:name>
            <ns1:type>string</ns1:type>
            <ns1:keyword>enu</ns1:keyword>
            <ns1:valueList>{&quot;Low Pressure System&quot;, &quot;Anesthesia&quot;,
    &quot;Anesthesia Machine&quot;}</ns1:valueList>
            <ns1:negotiable>no</ns1:negotiable>
            <ns1:priority>0</ns1:priority>
          </ns1:service_constraint>
          <ns1:service_constraint>
            <ns1:name>Media_Format</ns1:name>
            <ns1:type>string</ns1:type>
            <ns1:keyword>enu</ns1:keyword>
            <ns1:valueList>{&quot;HTML&quot;, &quot;Shockwave&quot;}</ns1:valueList>
            <ns1:negotiable>no</ns1:negotiable>
            <ns1:priority>0</ns1:priority>
          </ns1:service_constraint>
          <ns1:service_constraint>
            <ns1:name>Language</ns1:name>
            <ns1:type>string</ns1:type>
            <ns1:keyword>eq</ns1:keyword>
            <ns1:valueList>&quot;English&quot;</ns1:valueList>
            <ns1:negotiable>no</ns1:negotiable>
            <ns1:priority>0</ns1:priority>
          </ns1:service_constraint>
          <ns1:service_constraint>
            <ns1:name>Cost</ns1:name>
            <ns1:type>float</ns1:type>
            <ns1:keyword>eq</ns1:keyword>
            <ns1:valueList>0.0</ns1:valueList>
            <ns1:negotiable>no</ns1:negotiable>
            <ns1:priority>0</ns1:priority>
          </ns1:service_constraint>
          <ns1:service_constraint>
            <ns1:name>Author</ns1:name>
            <ns1:type>string</ns1:type>
            <ns1:keyword>eq</ns1:keyword>
            <ns1:valueList>&quot;Gilliean Lee&quot;</ns1:valueList>
            <ns1:negotiable>no</ns1:negotiable>
            <ns1:priority>0</ns1:priority>
          </ns1:service_constraint>
        </ns1:service_constraints>
        <ns1:interattribute_constraints />
      </ns1:constraints>
    </ns1:LearningObject>
```


C.2 Sample Composite Learning Object – Virtual Anesthesia Machine


```
<?xml version="1.0"?>
<ns1:LearningProcess Id="VAM" Name="VAM" xmlns:ns1="http://www.cise.ufl.edu/~glee/LPDL1_0">
<ns1:SavedTime>1116435483106</ns1:SavedTime>
    <ns1:LeafActivity Name="" Id="Part_1_Basic_Concepts">
            <ns1:UpdateTime>1116435456966</ns1:UpdateTime>
            <ns1:Description>Basic concepts related to the anesthesia machine.</ns1:Description>
            <ns1:Limit>
                    <ns1:Duration>P0Y0M0DT0H0M0S</ns1:Duration>
            </ns1:Limit>
            <ns1:Objectives>
            <ns1:Objective SatisfiedByMeasure="false" ContributesToRollup="true"
                MinimumSatisfiedNormalizedMeasure="0" Id="oBasicConcepts" />
            </ns1:Objectives>
            <ns1:ActivityData>
                    <ns1:Data DataType="String" DataId="sLanguage" />
            </ns1:ActivityData>
```

```
        <ns1:BindingInfo Mode="Dynamic" BrokerURL="">
                <ns1:DynamicBindingInfo>
                        <ns1:constraints>
                                <ns1:service_constraints>
                                        <ns1:service_constraint>
                                                <ns1:name>Keywords</ns1:name>
                                                <ns1:type>string</ns1:type>
                                                <ns1:keyword>eq</ns1:keyword>
                                                <ns1:valueList>&quot;basic concepts of anesthesia
                                                  machine&quot;</ns1:valueList>
                                                <ns1:negotiable>No</ns1:negotiable>
                                                <ns1:priority>0</ns1:priority>
                                        </ns1:service_constraint>
                                        <ns1:service_constraint>
                                                <ns1:name>Language</ns1:name>
                                                <ns1:type>string</ns1:type>
                                                <ns1:keyword>eq</ns1:keyword>
                                                <ns1:valueList>&quot;VAR:sLanguage&quot;</ns1:
                                                    valueList>
                                                <ns1:negotiable>No</ns1:negotiable>
                                                <ns1:priority>0</ns1:priority>
                                        </ns1:service_constraint>
                                </ns1:service_constraints>
                                <ns1:interattribute_constraints />
                        </ns1:constraints>
                </ns1:DynamicBindingInfo>
        </ns1:BindingInfo>
        <ns1:LeafActivityActions>
                <ns1:BeforePreActivity_Actions>sLanguage =
                    &quot;&quot;;</ns1:BeforePreActivity_Actions>
                <ns1:BeforeBinding_Actions>Vector vLanguage = DataAccessor.getLIMLanguage(this);
                sLanguage = &quot;English&quot;; // default language is English
                for (int i=0;i&lt;vLanguage.size();i++) {
                        String sLang = (String) vLanguage.get(i);
                        if (sLang.equalsIgnoreCase(&quot;English&quot;)) {
                                sLanguage = &quot;English&quot;;
                                break;
                        }
                        else if (sLang.equalsIgnoreCase(&quot;Korean&quot;)) {
                                sLanguage = &quot;Korean&quot;;
                                break;
                        }
                }</ns1:BeforeBinding_Actions>
                <ns1:AfterPostAssessment_Actions>DataAccessor.executeCommand(LMSEngineCore.EX
                    E_ACTIVITY_SATISFIED, this, &quot;&quot;);</ns1:AfterPostAssessment_Actions>
        </ns1:LeafActivityActions>
        <ns1:LeafConditionActionRules>
                <ns1:PreActivityCARule>
                        <ns1:ConditionActionRule />
                </ns1:PreActivityCARule>
                <ns1:PreAssessedCARule>
                        <ns1:ConditionActionRule />
                </ns1:PreAssessedCARule>
                <ns1:PostAssessedCARule>
                        <ns1:ConditionActionRule>
                                <ns1:Condition>true</ns1:Condition>
                                <ns1:Actions>m_lmsEngine.executeCommand(LMSEngineCore.EXE_A
                                    CTIVITY_SATISFIED, m_sProcessInstanceID,  m_sActivityID,
                                    &quot;&quot;);</ns1:Actions>
                                <ns1:AltActions />
                        </ns1:ConditionActionRule>
                </ns1:PostAssessedCARule>
        </ns1:LeafConditionActionRules>
        <ns1:LeafAssessmentMarker PostAssessment="false" PreAssessment="false" />
</ns1:LeafActivity>
<ns1:LeafActivity Name="Input, output of VAM" Id="Part_2_IO_of_VAM">
```

```
<ns1:UpdateTime>1116435459388</ns1:UpdateTime>
<ns1:Description>This module teaches how the VAM simulation works including Input and
    Output.</ns1:Description>
<ns1:Limit>
        <ns1:Duration>P0Y0M0DT0H0M0S</ns1:Duration>
</ns1:Limit>
<ns1:Objectives>
<ns1:Objective SatisfiedByMeasure="false" ContributesToRollup="true"
    MinimumSatisfiedNormalizedMeasure="0" Id="oIOVAM" />
</ns1:Objectives>
<ns1:ActivityData>
        <ns1:Data DataType="String" DataId="sLanguage" />
</ns1:ActivityData>
<ns1:BindingInfo Mode="Dynamic" BrokerURL="">
        <ns1:DynamicBindingInfo>
                <ns1:constraints>
                        <ns1:service_constraints>
                                <ns1:service_constraint>
                                        <ns1:name>Keywords</ns1:name>
                                        <ns1:type>string</ns1:type>
                                        <ns1:keyword>eq</ns1:keyword>
                                        <ns1:valueList>&quot;input output of anesthesia
                                            machine&quot;</ns1:valueList>
                                        <ns1:negotiable>No</ns1:negotiable>
                                        <ns1:priority>0</ns1:priority>
                                </ns1:service_constraint>
                                <ns1:service_constraint>
                                        <ns1:name>Language</ns1:name>
                                        <ns1:type>string</ns1:type>
                                        <ns1:keyword>eq</ns1:keyword>
                                        <ns1:valueList>&quot;VAR:sLanguage&quot;</ns1:
                                            valueList>
                                        <ns1:negotiable>No</ns1:negotiable>
                                        <ns1:priority>0</ns1:priority>
                                </ns1:service_constraint>
                        </ns1:service_constraints>
                        <ns1:interattribute_constraints />
                </ns1:constraints>
        </ns1:DynamicBindingInfo>
</ns1:BindingInfo>
<ns1:LeafActivityActions>
        <ns1:BeforePreActivity_Actions>sLanguage =
            &quot;&quot;;</ns1:BeforePreActivity_Actions>
        <ns1:BeforeBinding_Actions>
        Vector vLanguage = DataAccessor.getLIMLanguage(this);
        sLanguage = &quot;English&quot;; // default language is English
        for (int i=0;i&lt;vLanguage.size();i++) {
                String sLang = (String) vLanguage.get(i);
                if (sLang.equalsIgnoreCase(&quot;English&quot;)) {
                        sLanguage = &quot;English&quot;;
                        break;
                }
                else if (sLang.equalsIgnoreCase(&quot;Korean&quot;)) {
                        sLanguage = &quot;Korean&quot;;
                        break;
                }
        }</ns1:BeforeBinding_Actions>
        <ns1:AfterPostAssessment_Actions>DataAccessor.executeCommand(LMSEngineCore.EX
            E_ACTIVITY_SATISFIED, this, &quot;&quot;);</ns1:AfterPostAssessment_Actions>
</ns1:LeafActivityActions>
<ns1:LeafConditionActionRules>
        <ns1:PostAssessedCARule>
                <ns1:ConditionActionRule>
                        <ns1:Condition>true</ns1:Condition>
```

```
                                <ns1:Actions>m_lmsEngine.executeCommand(LMSEngineCore.EXE_A
                                    CTIVITY_SATISFIED, m_sProcessInstanceID,  m_sActivityID,
                                    &quot;&quot;);</ns1:Actions>
                                <ns1:AltActions />
                        </ns1:ConditionActionRule>
                </ns1:PostAssessedCARule>
        </ns1:LeafConditionActionRules>
        <ns1:LeafAssessmentMarker PostAssessment="false" PreAssessment="false" />
</ns1:LeafActivity>
<ns1:LeafActivity Name="" Id="High_Pressure_System">
        <ns1:UpdateTime>1116435463919</ns1:UpdateTime>
        <ns1:Description>High pressure system Instruction, Exercise and Post
            assessment.</ns1:Description>
        <ns1:Limit>
                <ns1:Duration>P0Y0M0DT0H0M0S</ns1:Duration>
        </ns1:Limit>
        <ns1:Objectives>
            <ns1:Objective SatisfiedByMeasure="true" ContributesToRollup="true"
                MinimumSatisfiedNormalizedMeasure="0.75" Id="oOverall" />
        </ns1:Objectives>
        <ns1:ActivityData>
                <ns1:Data DataType="boolean" DataId="bSatisfied" />
        </ns1:ActivityData>
        <ns1:BindingInfo Mode="Static" BrokerURL="">
                <ns1:StaticBindingInfo>
                <ns1:BusinessServiceKey>EE5F8C20-7A28-11D9-9561-
                    000629DC0A53</ns1:BusinessServiceKey>
                </ns1:StaticBindingInfo>
        </ns1:BindingInfo>
        <ns1:LeafActivityActions>
                <ns1:AfterPostAssessment_Actions>bSatisfied = DataAccessor.isObjectiveSatisfied(this,
                    &quot;oOverall&quot;);</ns1:AfterPostAssessment_Actions>
        </ns1:LeafActivityActions>
        <ns1:LeafConditionActionRules>
                <ns1:PostAssessedCARule>
                        <ns1:ConditionActionRule>
                                <ns1:Condition>bSatisfied</ns1:Condition>
                                <ns1:Actions>System.out.println(&quot;High Pressure System is
                                    SATISFIED.  Retry NOT REQUIRED&quot;);</ns1:Actions>
                                <ns1:AltActions>
// need to be retried if not satisfied (if didn&apos;t pass assessment)
                                System.out.println(&quot;High Pressure System is NOT SATISFIED.
Retry REQUIRED&quot;);
                                m_lmsEngine.executeCommand(LMSEngineCore.EXE_RETRY,
m_sProcessInstanceID, m_sActivityID, &quot;&quot;);</ns1:AltActions>
                        </ns1:ConditionActionRule>
                </ns1:PostAssessedCARule>
        </ns1:LeafConditionActionRules>
        <ns1:LeafAssessmentMarker PostAssessment="true" PreAssessment="false" />
        <ns1:AssessmentParameters>
                <ns1:PostAssessmentParameters>
                        <ns1:AssessmentSelectionParameter ScoringWeight="1"
                            SelectionPercentage="0.75" ActivityID="High_Pressure_System" />
                </ns1:PostAssessmentParameters>
                <ns1:PostAssessmentObjectiveSettings>
                        <ns1:ObjectiveSetting ObjectiveID="oOverall">
                                <ns1:ActivityID>High_Pressure_System</ns1:ActivityID>
                        </ns1:ObjectiveSetting>
                </ns1:PostAssessmentObjectiveSettings>
        </ns1:AssessmentParameters>
</ns1:LeafActivity>
<ns1:LeafActivity Name="" Id="Low_Pressure_System">
        <ns1:UpdateTime>1116435465919</ns1:UpdateTime>
        <ns1:Description>Low Pressure System Instruction, Exercise and Post
            assessment.</ns1:Description>
        <ns1:Limit>
```

```
                <ns1:Duration>P0Y0M0DT0H0M0S</ns1:Duration>
        </ns1:Limit>
        <ns1:Objectives>
                <ns1:Objective SatisfiedByMeasure="true" ContributesToRollup="true"
                    MinimumSatisfiedNormalizedMeasure="0.75" Id="oOverall" />
        </ns1:Objectives>
        <ns1:ActivityData />
        <ns1:BindingInfo Mode="Static" BrokerURL="">
                <ns1:StaticBindingInfo>
                        <ns1:BusinessServiceKey>5B863DF0-7D6A-11D9-9561-
                            000629DC0A53</ns1:BusinessServiceKey>
                </ns1:StaticBindingInfo>
        </ns1:BindingInfo>
        <ns1:LeafActivityActions />
        <ns1:LeafConditionActionRules />
        <ns1:LeafAssessmentMarker PostAssessment="true" PreAssessment="false" />
        <ns1:AssessmentParameters>
                <ns1:PostAssessmentParameters>
                    <ns1:AssessmentSelectionParameter ScoringWeight="1"
                    SelectionPercentage="0.75" ActivityID="Low_Pressure_System" />
                </ns1:PostAssessmentParameters>
                <ns1:PostAssessmentObjectiveSettings>
                        <ns1:ObjectiveSetting ObjectiveID="oOverall">
                                <ns1:ActivityID>Low_Pressure_System</ns1:ActivityID>
                        </ns1:ObjectiveSetting>
                </ns1:PostAssessmentObjectiveSettings>
        </ns1:AssessmentParameters>
</ns1:LeafActivity>
<ns1:LeafActivity Name="" Id="Breathing_Circuit">
        <ns1:UpdateTime>1116435468185</ns1:UpdateTime>
        <ns1:Description>Breathing Circuit Instruction, Exercise and Post
            assessment.</ns1:Description>
        <ns1:Limit>
                <ns1:Duration>P0Y0M0DT0H0M0S</ns1:Duration>
        </ns1:Limit>
        <ns1:Objectives>
                <ns1:Objective SatisfiedByMeasure="true" ContributesToRollup="true"
                    MinimumSatisfiedNormalizedMeasure="0.75" Id="oOverall" />
        </ns1:Objectives>
        <ns1:ActivityData />
        <ns1:BindingInfo Mode="Static" BrokerURL="">
                <ns1:StaticBindingInfo>
                        <ns1:BusinessServiceKey>6C9E7A50-7EC6-11D9-9561-
                            000629DC0A53</ns1:BusinessServiceKey>
                </ns1:StaticBindingInfo>
        </ns1:BindingInfo>
        <ns1:LeafActivityActions />
        <ns1:LeafConditionActionRules />
        <ns1:LeafAssessmentMarker PostAssessment="true" PreAssessment="false" />
        <ns1:AssessmentParameters>
                <ns1:PostAssessmentParameters>
                        <ns1:AssessmentSelectionParameter ScoringWeight="1"
                            SelectionPercentage="0.75" ActivityID="Breathing_Circuit" />
                </ns1:PostAssessmentParameters>
                <ns1:PostAssessmentObjectiveSettings>
                        <ns1:ObjectiveSetting ObjectiveID="oOverall">
                                <ns1:ActivityID>Breathing_Circuit</ns1:ActivityID>
                        </ns1:ObjectiveSetting>
                </ns1:PostAssessmentObjectiveSettings>
        </ns1:AssessmentParameters>
</ns1:LeafActivity>
<ns1:LeafActivity Name="" Id="Manual_Ventilation">
        <ns1:UpdateTime>1116435470278</ns1:UpdateTime>
        <ns1:Description>Manual Ventilation Instruction, Exercise and Post
            assessment.</ns1:Description>
        <ns1:Limit>
```

```
                <ns1:Duration>P0Y0M0DT0H0M0S</ns1:Duration>
        </ns1:Limit>
        <ns1:Objectives>
                <ns1:Objective SatisfiedByMeasure="true" ContributesToRollup="true"
                    MinimumSatisfiedNormalizedMeasure="0.75" Id="oOverall" />
        </ns1:Objectives>
        <ns1:ActivityData />
        <ns1:BindingInfo Mode="Static" BrokerURL="">
                <ns1:StaticBindingInfo>
                        <ns1:BusinessServiceKey>F635C7B0-7ECF-11D9-9561-
                            000629DC0A53</ns1:BusinessServiceKey>
                </ns1:StaticBindingInfo>
        </ns1:BindingInfo>
        <ns1:LeafActivityActions />
        <ns1:LeafConditionActionRules />
        <ns1:LeafAssessmentMarker PostAssessment="true" PreAssessment="false" />
        <ns1:AssessmentParameters>
                <ns1:PostAssessmentParameters>
                        <ns1:AssessmentSelectionParameter ScoringWeight="1"
                            SelectionPercentage="0.75" ActivityID="Manual_Ventilation" />
                </ns1:PostAssessmentParameters>
        <ns1:PostAssessmentObjectiveSettings>
                <ns1:ObjectiveSetting ObjectiveID="oOverall">
                        <ns1:ActivityID>Manual_Ventilation</ns1:ActivityID>
                </ns1:ObjectiveSetting>
        </ns1:PostAssessmentObjectiveSettings>
        </ns1:AssessmentParameters>
</ns1:LeafActivity>
<ns1:LeafActivity Name="" Id="Mechanical_Ventilation">
        <ns1:UpdateTime>1116435474075</ns1:UpdateTime>
        <ns1:Description>Mechanical Ventilation Instruction, Exercise and Post
            assessment.</ns1:Description>
        <ns1:Limit>
                <ns1:Duration>P0Y0M0DT0H0M0S</ns1:Duration>
        </ns1:Limit>
        <ns1:Objectives>
                <ns1:Objective SatisfiedByMeasure="true" ContributesToRollup="true"
                    MinimumSatisfiedNormalizedMeasure="0.75" Id="oOverall" />
        </ns1:Objectives>
        <ns1:ActivityData />
        <ns1:BindingInfo Mode="Static" BrokerURL="">
                <ns1:StaticBindingInfo>
                        <ns1:BusinessServiceKey>54AC9C40-7EE1-11D9-9561-
                            000629DC0A53</ns1:BusinessServiceKey>
                </ns1:StaticBindingInfo>
        </ns1:BindingInfo>
        <ns1:LeafActivityActions />
        <ns1:LeafConditionActionRules />
        <ns1:LeafAssessmentMarker PostAssessment="true" PreAssessment="false" />
        <ns1:AssessmentParameters>
                <ns1:PostAssessmentParameters>
                            <ns1:AssessmentSelectionParameter ScoringWeight="1"
                            SelectionPercentage="0.75" ActivityID="Mechanical_Ventilation" />
                </ns1:PostAssessmentParameters>
        <ns1:PostAssessmentObjectiveSettings>
                <ns1:ObjectiveSetting ObjectiveID="oOverall">
                        <ns1:ActivityID>Mechanical_Ventilation</ns1:ActivityID>
                </ns1:ObjectiveSetting>
        </ns1:PostAssessmentObjectiveSettings>
        </ns1:AssessmentParameters>
</ns1:LeafActivity>
<ns1:LeafActivity Name="" Id="Scavenging_System">
        <ns1:UpdateTime>1116435476481</ns1:UpdateTime>
        <ns1:Description>Scavenging System Instruction, Exercise and Post
            assessment.</ns1:Description>
        <ns1:Limit>
```

```
                <ns1:Duration>P0Y0M0DT0H0M0S</ns1:Duration>
        </ns1:Limit>
        <ns1:Objectives>
                <ns1:Objective SatisfiedByMeasure="true" ContributesToRollup="true"
                    MinimumSatisfiedNormalizedMeasure="0.75" Id="oOverall" />
        </ns1:Objectives>
        <ns1:ActivityData />
        <ns1:BindingInfo Mode="Static" BrokerURL="">
                <ns1:StaticBindingInfo>
                        <ns1:BusinessServiceKey>3E6DF260-7F7E-11D9-9561-
                            000629DC0A53</ns1:BusinessServiceKey>
                </ns1:StaticBindingInfo>
        </ns1:BindingInfo>
        <ns1:LeafActivityActions />
        <ns1:LeafConditionActionRules />
        <ns1:LeafAssessmentMarker PostAssessment="true" PreAssessment="false" />
        <ns1:AssessmentParameters>
                <ns1:PostAssessmentParameters>
                        <ns1:AssessmentSelectionParameter ScoringWeight="1"
                            SelectionPercentage="0.75" ActivityID="Scavenging_System" />
                </ns1:PostAssessmentParameters>
        <ns1:PostAssessmentObjectiveSettings>
                <ns1:ObjectiveSetting ObjectiveID="oOverall">
                        <ns1:ActivityID>Scavenging_System</ns1:ActivityID>
                </ns1:ObjectiveSetting>
        </ns1:PostAssessmentObjectiveSettings>
        </ns1:AssessmentParameters>
</ns1:LeafActivity>
<ns1:NonleafActivity Id="VAM" Name="">
        <ns1:UpdateTime>1116435453982</ns1:UpdateTime>
        <ns1:Description>This is Root activity of Virtual Anesthesia Machine
            instruction.</ns1:Description>
        <ns1:Limit>
                <ns1:Duration>P0Y0M0DT10H0M0S</ns1:Duration>
        </ns1:Limit>
        <ns1:Objectives>
                <ns1:Objective SatisfiedByMeasure="false" ContributesToRollup="true"
                    MinimumSatisfiedNormalizedMeasure="0" Id="oOverall" />
        </ns1:Objectives>
        <ns1:ActivityData>
                <ns1:Data DataType="boolean" DataId="bMedicalGoal" />
                <ns1:Data DataType="boolean" DataId="bAnestGoal" />
        </ns1:ActivityData>
        <ns1:NonleafActivityActions>
                <ns1:BeforePreActivity_Actions>bMedicalGoal = DataAccessor.goalsIncludeWord(this,
&quot;medical&quot;);
                bAnestGoal = DataAccessor.goalsIncludeWord(this,
&quot;anest&quot;);</ns1:BeforePreActivity_Actions>
        </ns1:NonleafActivityActions>
        <ns1:NonleafConditionActionRules>
                <ns1:PreActivityCARule>
                        <ns1:ConditionActionRule>
                                <ns1:Condition>bMedicalGoal || bAnestGoal</ns1:Condition>
                                <ns1:Actions>  System.out.println(&quot;In Pre-Activity Rule of VAM
activity, Part3 is ENABLED&quot;);</ns1:Actions>

        <ns1:AltActions>m_lmsEngine.executeCommand(LMSEngineCore.EXE_DISABLE,
m_sProcessInstanceID,  &quot;Part_3_Safety_Exercises&quot;, &quot;&quot;);
                                System.out.println(&quot;In Pre-Activity Rule of VAM activity, Part3 is
DISABLED&quot;);</ns1:AltActions>
                        </ns1:ConditionActionRule>
                </ns1:PreActivityCARule>
                <ns1:RollupCARule ChildActivitySet="All" MinimumSet="0">
                        <ns1:CARule>
                                <ns1:ConditionActionRule>
                                        <ns1:Condition>Satisfied</ns1:Condition>
```

```
                                        <ns1:Actions>
m_lmsEngine.activityHandlerSetReadyToFihishChildren(m_sProcessInstanceID,
                                          m_sActivityID, true);

m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_SATISFIED, m_sProcessInstanceID,
m_sActivityID, &quot;&quot;);
                                          System.out.println(&quot;In Rollup rule of VAM: All child
SATISFIED&quot;);
                                        </ns1:Actions>
                                        <ns1:AltActions>
m_lmsEngine.activityHandlerSetReadyToFihishChildren(m_sProcessInstanceID,
                                          m_sActivityID, false);

m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_UNSATISFIED, m_sProcessInstanceID,
m_sActivityID, &quot;&quot;);
                                          System.out.println(&quot;In Rollup rule of VAM: All child
UNSATISFIED&quot;);</ns1:AltActions>
                                    </ns1:ConditionActionRule>
                            </ns1:CARule>
                    </ns1:RollupCARule>
            </ns1:NonleafConditionActionRules>
            <ns1:NonleafAssessmentMarker PostAssessment="false" PreAssessmentMode="ThisActivityOnly"
                PreAssessment="false" />
            <ns1:NonleafContentItems>
                    <ns1:Introduction>
                            <ns1:URL>http://128.227.176.41:8888/VAM/VAM_Intro.htm</ns1:URL>
                    </ns1:Introduction>
                    <ns1:Summary>
                            <ns1:text>&lt;p&gt; Original Virtual Anesthesia Machine Web page is &lt;a
href=&quot;http://vam.anest.ufl.edu&quot;&gt;http://vam.anest.ufl.edu&lt;/a&gt;.&lt;/p&gt;
                            &lt;p&gt; Special Thanks to Dr. Sem Lampotang &lt;/p&gt;
                            &lt;p&gt; Copyright: University of Florida. &lt;/p&gt;</ns1:text>
                    </ns1:Summary>
            </ns1:NonleafContentItems>
            <ns1:NonleafPracticeItems>
                    <ns1:NonleafPracticeItem Number="1">
                            <ns1:Problem>
                                    <ns1:text>&lt;H3&gt; This is Practice Item &lt;/H3&gt;</ns1:text>
                            </ns1:Problem>
                            <ns1:Answer>N/A</ns1:Answer>
                    </ns1:NonleafPracticeItem>
            </ns1:NonleafPracticeItems>
            <ns1:NonleafAssessments>
                    <ns1:PreAssessmentItems />
                    <ns1:PostAssessmentItems />
            </ns1:NonleafAssessments>
            <ns1:AssessmentParameters>
                    <ns1:PostAssessmentParameters />
            </ns1:AssessmentParameters>
</ns1:NonleafActivity>
<ns1:NonleafActivity Id="Part_3_Safety_Exercises" Name="">
            <ns1:UpdateTime>1116435461747</ns1:UpdateTime>
            <ns1:Description>Specific safety-related exercises using VAM Simulator.</ns1:Description>
            <ns1:Limit>
                    <ns1:Duration>P0Y0M0DT0H0M0S</ns1:Duration>
            </ns1:Limit>
            <ns1:Objectives>
                    <ns1:Objective SatisfiedByMeasure="true" ContributesToRollup="true"
                        MinimumSatisfiedNormalizedMeasure="0.5" Id="oOverall" />
                    <ns1:Objective SatisfiedByMeasure="true" ContributesToRollup="false"
                        MinimumSatisfiedNormalizedMeasure="0.5" Id="oPressureSystem" />
                    <ns1:Objective SatisfiedByMeasure="true" ContributesToRollup="false"
                        MinimumSatisfiedNormalizedMeasure="0.5" Id="oVentilation" />
                    <ns1:Objective SatisfiedByMeasure="true" ContributesToRollup="false"
                        MinimumSatisfiedNormalizedMeasure="0.5" Id="oBreathingCircuit" />
```

```
                <ns1:Objective SatisfiedByMeasure="true" ContributesToRollup="false"
                    MinimumSatisfiedNormalizedMeasure="0.5" Id="oScavengingSystem" />
        </ns1:Objectives>
        <ns1:ActivityData>
                <ns1:Data DataType="boolean" DataId="bAssessmentReqd" />
                <ns1:Data DataType="int" DataId="iTimes" />
                <ns1:Data DataType="boolean" DataId="bDirective" />
        </ns1:ActivityData>
        <ns1:NonleafActivityActions>
                <ns1:BeforePreActivity_Actions>iTimes = 0;</ns1:BeforePreActivity_Actions>
                <ns1:AfterIntroPresentation_Actions>int iAttemptCount =
DataAccessor.getAttemptCount(this);

                if (iAttemptCount &gt; 1) // second or later attempt
                  bDirective = false;
                else
                  bDirective = (DataAccessor.hasLIMLearningStyle(this, DataAccessor.DIRECTIVE) ||
DataAccessor.hasLIMLearningStyle(this, DataAccessor.RECEPTIVE));
                System.out.println(&quot;LearningStyle: DIRECTIVE or RECEPTIVE- &quot; +
bDirective);</ns1:AfterIntroPresentation_Actions>
                <ns1:BeforeRollup_Actions />
                <ns1:BeforePostAssessment_Actions />
                <ns1:AfterPostAssessment_Actions>iTimes++;
                if (iTimes &lt;= 1) {
                        bAssessmentReqd = !DataAccessor.evaluateChildActivities(this, &quot;At Least
Count&quot;, &quot;4&quot;, &quot;Satisfied&quot;);
                        if (!bAssessmentReqd)
                                DataAccessor.setObjectiveStatus(this, &quot;oOverall&quot;,
&quot;Satisfied&quot;);
                        iTimes = 1;
                }
                else { // exam conducted Maximum only once
                //      bAssessmentReqd = !DataAccessor.isObjectiveSatisfied(this,
&quot;oOverall&quot;);
                        bAssessmentReqd = false;
                //      DataAccessor.setObjectiveStatus(this, &quot;oOverall&quot;,
&quot;Satisfied&quot;);
                }</ns1:AfterPostAssessment_Actions>
        </ns1:NonleafActivityActions>
        <ns1:NonleafConditionActionRules>
                <ns1:DrillDownCARule>
                        <ns1:ConditionActionRule>
                                <ns1:Condition>bDirective</ns1:Condition>
                                <ns1:Actions>System.out.println(&quot;Drilldown Rule Action
executed!&quot;);

                                System.out.println(&quot;addSCM-ForwardOnly requested for &quot;
+ m_sActivityID + &quot;!&quot;);

        m_lmsEngine.removeAllSCMfromChildConnector(m_sProcessInstanceID,  m_sActivityID);
                                m_lmsEngine.addSCMtoChildConnector(m_sProcessInstanceID,
m_sActivityID, &quot;Flow&quot;);</ns1:Actions>
                                <ns1:AltActions>System.out.println(&quot;Drilldown Rule
AlternativeAction executed!&quot;);

                                System.out.println(&quot;addSCM-Choice requested for &quot; +
m_sActivityID + &quot;!&quot;);

        m_lmsEngine.removeAllSCMfromChildConnector(m_sProcessInstanceID,  m_sActivityID);
                                m_lmsEngine.addSCMtoChildConnector(m_sProcessInstanceID,
m_sActivityID, &quot;Choice&quot;);</ns1:AltActions>
                        </ns1:ConditionActionRule>
                </ns1:DrillDownCARule>
                <ns1:RollupCARule ChildActivitySet="All" MinimumSet="0">
                        <ns1:CARule>
                                <ns1:ConditionActionRule>
```

```
                                        <ns1:Condition>Completed</ns1:Condition>
                                        <ns1:Actions>
m_lmsEngine.activityHandlerSetReadyToFihishChildren(m_sProcessInstanceID,
                                          m_sActivityID, true);</ns1:Actions>
                                        <ns1:AltActions>
m_lmsEngine.activityHandlerSetReadyToFihishChildren(m_sProcessInstanceID,
                                          m_sActivityID, false);

m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_UNSATISFIED, m_sProcessInstanceID,
m_sActivityID, &quot;&quot;);
                                        </ns1:AltActions>
                                </ns1:ConditionActionRule>
                        </ns1:CARule>
                </ns1:RollupCARule>
                <ns1:PostAssessedCARule>
                        <ns1:ConditionActionRule>
                                <ns1:Condition>bAssessmentReqd</ns1:Condition>
                                <ns1:Actions>System.out.println(&quot;bAssessmentReqd: &quot; +
bAssessmentReqd);
                                System.out.println(&quot;PostAssessment command
requested.&quot;);
                                m_lmsEngine.executeCommand(LMSEngineCore.EXE_POST_ASSESS,
m_sProcessInstanceID,  m_sActivityID, &quot;&quot;);</ns1:Actions>
                                <ns1:AltActions>System.out.println(&quot;::Value of
&apos;iTimes&apos;: &quot; + iTimes);
                                if (iTimes != 1) {
                                System.out.println(&quot;PostAssessment command NOT
requested.&quot;);
                                // disable all child activities, then selectively enable according to
objective status

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_SATISFIED,
m_sProcessInstanceID, &quot;High_Pressure_System&quot;, &quot;&quot;);

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_SATISFIED,
m_sProcessInstanceID, &quot;Low_Pressure_System&quot;, &quot;&quot;);

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_SATISFIED,
m_sProcessInstanceID, &quot;Breathing_Circuit&quot;, &quot;&quot;);

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_SATISFIED,
m_sProcessInstanceID, &quot;Manual_Ventilation&quot;, &quot;&quot;);

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_SATISFIED,
m_sProcessInstanceID, &quot;Mechanical_Ventilation&quot;, &quot;&quot;);

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_SATISFIED,
m_sProcessInstanceID, &quot;Scavenging_System&quot;, &quot;&quot;);

                                String sPressureSystem =
m_lmsEngine.getObjectiveSatisfiedStatus(m_sProcessInstanceID, m_sActivityID,
&quot;oPressureSystem&quot;);
                                String sVentilation =
m_lmsEngine.getObjectiveSatisfiedStatus(m_sProcessInstanceID, m_sActivityID,
&quot;oVentilation&quot;);
                                String sBreathingCircuit =
m_lmsEngine.getObjectiveSatisfiedStatus(m_sProcessInstanceID, m_sActivityID,
&quot;oBreathingCircuit&quot;);
                                String sScavengingSystem =
m_lmsEngine.getObjectiveSatisfiedStatus(m_sProcessInstanceID, m_sActivityID,
&quot;oScavengingSystem&quot;);
                                String sOverall =
m_lmsEngine.getObjectiveSatisfiedStatus(m_sProcessInstanceID, m_sActivityID, &quot;oOverall&quot;);

                                boolean bSatisfied = true;
                                if (!sPressureSystem.equals(&quot;Satisfied&quot;)) {
```

```
        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_UNSATISFIED,
m_sProcessInstanceID, &quot;High_Pressure_System&quot;, &quot;&quot;);

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_UNSATISFIED,
m_sProcessInstanceID, &quot;Low_Pressure_System&quot;, &quot;&quot;);

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_INCOMPLETE,
m_sProcessInstanceID, &quot;High_Pressure_System&quot;, &quot;&quot;);

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_INCOMPLETE,
m_sProcessInstanceID, &quot;Low_Pressure_System&quot;, &quot;&quot;);
                                }

                                if (!sVentilation.equals(&quot;Satisfied&quot;)) {

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_UNSATISFIED,
m_sProcessInstanceID, &quot;Manual_Ventilation&quot;, &quot;&quot;);

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_UNSATISFIED,
m_sProcessInstanceID, &quot;Mechanical_Ventilation&quot;, &quot;&quot;);

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_INCOMPLETE,
m_sProcessInstanceID, &quot;Manual_Ventilation&quot;, &quot;&quot;);

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_INCOMPLETE,
m_sProcessInstanceID, &quot;Mechanical_Ventilation&quot;, &quot;&quot;);
                                }

                                if (!sBreathingCircuit.equals(&quot;Satisfied&quot;)) {

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_UNSATISFIED,
m_sProcessInstanceID, &quot;Breathing_Circuit&quot;, &quot;&quot;);

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_INCOMPLETE,
m_sProcessInstanceID, &quot;Breathing_Circuit&quot;, &quot;&quot;);
                                }

                                if (!sScavengingSystem.equals(&quot;Satisfied&quot;)) {

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_UNSATISFIED,
m_sProcessInstanceID, &quot;Scavenging_System&quot;, &quot;&quot;);

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_INCOMPLETE,
m_sProcessInstanceID, &quot;Scavenging_System&quot;, &quot;&quot;);
                                }

                                /*if (sOverall.equals(&quot;Satisfied&quot;))
                                        bSatisfied = true;
                                else
                                        bSatisfied = false;

                                if (bSatisfied)

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_SATISFIED,
m_sProcessInstanceID, m_sActivityID, &quot;&quot;);
                                else

        m_lmsEngine.executeCommand(LMSEngineCore.EXE_ACTIVITY_UNSATISFIED,
m_sProcessInstanceID, m_sActivityID, &quot;&quot;);
                                */
                                }</ns1:AltActions>
                        </ns1:ConditionActionRule>
                </ns1:PostAssessedCARule>
        </ns1:NonleafConditionActionRules>
```

```
        <ns1:NonleafAssessmentMarker PostAssessment="false" PreAssessmentMode="ThisActivityOnly"
            PreAssessment="false" />
        <ns1:NonleafContentItems>
                <ns1:Introduction>
                        <ns1:text>&lt;p&gt; Here we will be taking lessons related to &apos;Safety
Exercises&apos; using several components of Anesthesia Machines.&lt;/p&gt;
                        &lt;p&gt; Each of the lessons consists of several Q&amp;As. Each of Q&amp;A
has a Question and Answer, Demonstration with VAM simulator, Explanation, and Learning Objectives of
the Question. &lt;/p&gt;
                        &lt;p&gt; Enjoy! &lt;/p&gt;</ns1:text>
                </ns1:Introduction>
                <ns1:Summary>
                        <ns1:text>&lt;p&gt; Hope you learned a lot about Anesthesia machine!
                            &lt;/p&gt;</ns1:text>
                </ns1:Summary>
        </ns1:NonleafContentItems>
        <ns1:NonleafPracticeItems />
        <ns1:NonleafAssessments>
        <ns1:PreAssessmentItems />
        <ns1:PostAssessmentItems />
        </ns1:NonleafAssessments>
        <ns1:AssessmentParameters>
                <ns1:PostAssessmentParameters>
                        <ns1:AssessmentSelectionParameter ScoringWeight="1"
                            SelectionPercentage="0.5" ActivityID="Part_3_Safety_Exercises" />
                        <ns1:AssessmentSelectionParameter ScoringWeight="1"
                            SelectionPercentage="0.5" ActivityID="High_Pressure_System" />
                        <ns1:AssessmentSelectionParameter ScoringWeight="1"
                            SelectionPercentage="0.5" ActivityID="Low_Pressure_System" />
                        <ns1:AssessmentSelectionParameter ScoringWeight="1"
                            SelectionPercentage="0.5" ActivityID="Breathing_Circuit" />
                        <ns1:AssessmentSelectionParameter ScoringWeight="1"
                            SelectionPercentage="0.5" ActivityID="Manual_Ventilation" />
                        <ns1:AssessmentSelectionParameter ScoringWeight="1"
                            SelectionPercentage="0.5" ActivityID="Mechanical_Ventilation" />
                        <ns1:AssessmentSelectionParameter ScoringWeight="1"
                            SelectionPercentage="0.5" ActivityID="Scavenging_System" />
                </ns1:PostAssessmentParameters>
        <ns1:PostAssessmentObjectiveSettings>
                <ns1:ObjectiveSetting ObjectiveID="oOverall">
                        <ns1:ActivityID>Part_3_Safety_Exercises</ns1:ActivityID>
                        <ns1:ActivityID>High_Pressure_System</ns1:ActivityID>
                        <ns1:ActivityID>Low_Pressure_System</ns1:ActivityID>
                        <ns1:ActivityID>Breathing_Circuit</ns1:ActivityID>
                        <ns1:ActivityID>Manual_Ventilation</ns1:ActivityID>
                        <ns1:ActivityID>Mechanical_Ventilation</ns1:ActivityID>
                        <ns1:ActivityID>Scavenging_System</ns1:ActivityID>
                </ns1:ObjectiveSetting>
                <ns1:ObjectiveSetting ObjectiveID="oPressureSystem">
                        <ns1:ActivityID>High_Pressure_System</ns1:ActivityID>
                        <ns1:ActivityID>Low_Pressure_System</ns1:ActivityID>
                </ns1:ObjectiveSetting>
                <ns1:ObjectiveSetting ObjectiveID="oBreathingCircuit">
                        <ns1:ActivityID>Breathing_Circuit</ns1:ActivityID>
                </ns1:ObjectiveSetting>
                <ns1:ObjectiveSetting ObjectiveID="oVentilation">
                        <ns1:ActivityID>Manual_Ventilation</ns1:ActivityID>
                        <ns1:ActivityID>Mechanical_Ventilation</ns1:ActivityID>
                </ns1:ObjectiveSetting>
                <ns1:ObjectiveSetting ObjectiveID="oScavengingSystem">
                        <ns1:ActivityID>Scavenging_System</ns1:ActivityID>
                </ns1:ObjectiveSetting>
        </ns1:PostAssessmentObjectiveSettings>
        </ns1:AssessmentParameters>
</ns1:NonleafActivity>
<ns1:Connector Name="" SequencingControlMode="Choice" Id="connector1">
```

```
        <ns1:SiblingActivityID>Part_1_Basic_Concepts</ns1:SiblingActivityID>
        <ns1:SiblingActivityID>Part_2_IO_of_VAM</ns1:SiblingActivityID>
        <ns1:SiblingActivityID>Part_3_Safety_Exercises</ns1:SiblingActivityID>
</ns1:Connector>
<ns1:Connector Name="" SequencingControlMode="Choice" Id="connector2">
        <ns1:SiblingActivityID>High_Pressure_System</ns1:SiblingActivityID>
        <ns1:SiblingActivityID>Low_Pressure_System</ns1:SiblingActivityID>
        <ns1:SiblingActivityID>Breathing_Circuit</ns1:SiblingActivityID>
        <ns1:SiblingActivityID>Manual_Ventilation</ns1:SiblingActivityID>
        <ns1:SiblingActivityID>Mechanical_Ventilation</ns1:SiblingActivityID>
        <ns1:SiblingActivityID>Scavenging_System</ns1:SiblingActivityID>
</ns1:Connector>
<ns1:Edge Id="VAM::connector1" To="connector1" From="VAM" />
<ns1:Edge Id="connector1::Part_1_Basic_Concepts" To="Part_1_Basic_Concepts" From="connector1" />
<ns1:Edge Id="connector1::Part_2_IO_of_VAM" To="Part_2_IO_of_VAM" From="connector1" />
<ns1:Edge Id="connector1::Part_3_Safety_Exercises" To="Part_3_Safety_Exercises" From="connector1"
/>
<ns1:Edge Id="Part_3_Safety_Exercises::connector2" To="connector2" From="Part_3_Safety_Exercises"
/>
<ns1:Edge Id="connector2::High_Pressure_System" To="High_Pressure_System" From="connector2" />
<ns1:Edge Id="connector2::Low_Pressure_System" To="Low_Pressure_System" From="connector2" />
<ns1:Edge Id="connector2::Breathing_Circuit" To="Breathing_Circuit" From="connector2" />
<ns1:Edge Id="connector2::Manual_Ventilation" To="Manual_Ventilation" From="connector2" />
<ns1:Edge Id="connector2::Mechanical_Ventilation" To="Mechanical_Ventilation" From="connector2" />
<ns1:Edge Id="connector2::Scavenging_System" To="Scavenging_System" From="connector2" />
<ns1:UDDIRegistration BusinessInterfaceKey="UUID:8ACC6F10-115B-11D9-A2B3-000629DC0A53"
ServiceImplementationKey="311295A0-7F90-11D9-9561-000629DC0A53" />
<ns1:constraints>
        <ns1:service_constraints>
                <ns1:service_constraint>
                        <ns1:name>LO_Type</ns1:name>
                        <ns1:type>string</ns1:type>
                        <ns1:keyword>eq</ns1:keyword>
                        <ns1:valueList>&quot;CLO&quot;</ns1:valueList>
                        <ns1:negotiable>No</ns1:negotiable>
                        <ns1:priority>0</ns1:priority>
                </ns1:service_constraint>
                <ns1:service_constraint>
                        <ns1:name>Title</ns1:name>
                        <ns1:type>string</ns1:type>
                        <ns1:keyword>eq</ns1:keyword>
                        <ns1:valueList>&quot;Virtual Anesthesia Machine&quot;</ns1:valueList>
                        <ns1:negotiable>No</ns1:negotiable>
                        <ns1:priority>0</ns1:priority>
                </ns1:service_constraint>
                <ns1:service_constraint>
                        <ns1:name>Subject</ns1:name>
                        <ns1:type>string</ns1:type>
                        <ns1:keyword>eq</ns1:keyword>
                        <ns1:valueList>&quot; Anesthesia Machine&quot;</ns1:valueList>
                        <ns1:negotiable>No</ns1:negotiable>
                        <ns1:priority>0</ns1:priority>
                </ns1:service_constraint>
                <ns1:service_constraint>
                        <ns1:name>Author</ns1:name>
                        <ns1:type>string</ns1:type>
                        <ns1:keyword>eq</ns1:keyword>
                        <ns1:valueList>&quot;Gilliean Lee&quot;</ns1:valueList>
                        <ns1:negotiable>No</ns1:negotiable>
                        <ns1:priority>0</ns1:priority>
                </ns1:service_constraint>
                <ns1:service_constraint>
                        <ns1:name>Keywords</ns1:name>
                        <ns1:type>string</ns1:type>
                        <ns1:keyword>enu</ns1:keyword>
```

```
                        <ns1:valueList>{&quot;Anesthesia Machine&quot;, &quot;Anesthesia&quot;,
                            &quot;Virtual Anesthesia Machine&quot;, &quot;Simulation&quot;,
                            &quot;Low Pressure System&quot;, &quot;High Pressure System&quot;,
                            &quot;Breathing Circuit&quot;, &quot;Manual Ventilation&quot;,
                            &quot;Mechanical Ventilation&quot;, &quot;Scavenging
                            System&quot;}</ns1:valueList>
                        <ns1:negotiable>No</ns1:negotiable>
                        <ns1:priority>0</ns1:priority>
                </ns1:service_constraint>
                <ns1:service_constraint>
                        <ns1:name>Media_Format</ns1:name>
                        <ns1:type>string</ns1:type>
                        <ns1:keyword>enu</ns1:keyword>
                        <ns1:valueList>{&quot;HTML&quot;,
                            &quot;Shockwave&quot;}</ns1:valueList>
                        <ns1:negotiable>No</ns1:negotiable>
                        <ns1:priority>0</ns1:priority>
                </ns1:service_constraint>
                <ns1:service_constraint>
                        <ns1:name>Cost</ns1:name>
                        <ns1:type>float</ns1:type>
                        <ns1:keyword>eq</ns1:keyword>
                        <ns1:valueList>0.0</ns1:valueList>
                        <ns1:negotiable>No</ns1:negotiable>
                        <ns1:priority>0</ns1:priority>
                </ns1:service_constraint>
        </ns1:service_constraints>
        <ns1:interattribute_constraints />
    </ns1:constraints>
</ns1:LearningProcess>
```

APPENDIX D
LEARNING OBJECT REPOSITORY AND BROKER WEB SERVICE DEFINITION LANGUAGE DOCUMENTS

D.1 Learning Object Repository Web Service Description

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:LODBService"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="urn:LODBService" xmlns:intf="urn:LODBService"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:message name="isLOExistRequest">
    <wsdl:part name="sLOId" type="xsd:string"/>
    <wsdl:part name="sLOType" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="insertLOwithStringRequest">
    <wsdl:part name="sLOId" type="xsd:string"/>
    <wsdl:part name="sLOType" type="xsd:string"/>
    <wsdl:part name="sLO" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="getLOResponse">
    <wsdl:part name="getLOReturn" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="addLORequest">
    <wsdl:part name="sLOId" type="xsd:string"/>
    <wsdl:part name="sLOType" type="xsd:string"/>
    <wsdl:part name="sLOContent" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="insertLOwithStringResponse">
    <wsdl:part name="insertLOwithStringReturn" type="xsd:boolean"/>
  </wsdl:message>
  <wsdl:message name="addLOResponse">
    <wsdl:part name="addLOReturn" type="xsd:boolean"/>
  </wsdl:message>
  <wsdl:message name="getLORequest">
    <wsdl:part name="sLOId" type="xsd:string"/>
    <wsdl:part name="sLOType" type="xsd:string"/>
    <wsdl:part name="sLOPart" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="isLOExistResponse">
    <wsdl:part name="isLOExistReturn" type="xsd:boolean"/>
  </wsdl:message>
  <wsdl:portType name="LODBService">
```

```
  <wsdl:operation name="addLO" parameterOrder="sLOId sLOType sLOContent">
    <wsdl:input message="intf:addLORequest" name="addLORequest"/>
    <wsdl:output message="intf:addLOResponse" name="addLOResponse"/>
  </wsdl:operation>
  <wsdl:operation name="getLO" parameterOrder="sLOId sLOType sLOPart">
    <wsdl:input message="intf:getLORequest" name="getLORequest"/>
    <wsdl:output message="intf:getLOResponse" name="getLOResponse"/>
  </wsdl:operation>
  <wsdl:operation name="insertLOwithString" parameterOrder="sLOId sLOType
          sLO">
    <wsdl:input message="intf:insertLOwithStringRequest"
              name="insertLOwithStringRequest"/>
    <wsdl:output message="intf:insertLOwithStringResponse"
              name="insertLOwithStringResponse"/>
  </wsdl:operation>
  <wsdl:operation name="isLOExist" parameterOrder="sLOId sLOType">
    <wsdl:input message="intf:isLOExistRequest" name="isLOExistRequest"/>
    <wsdl:output message="intf:isLOExistResponse"
              name="isLOExistResponse"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="LODBServiceSoapBinding" type="intf:LODBService">
  <wsdlsoap:binding style="rpc"
          transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="addLO">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="addLORequest">
      <wsdlsoap:body encodingStyle=
                "http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:LODBService" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="addLOResponse">
      <wsdlsoap:body encodingStyle=
                "http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:LODBService" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getLO">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="getLORequest">
      <wsdlsoap:body encodingStyle=
                "http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:LODBService" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="getLOResponse">
      <wsdlsoap:body encodingStyle=
                "http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:LODBService" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="insertLOwithString">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="insertLOwithStringRequest">
```

```
          <wsdlsoap:body encodingStyle=
                    "http://schemas.xmlsoap.org/soap/encoding/"
                    namespace="urn:LODBService" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="insertLOwithStringResponse">
          <wsdlsoap:body encodingStyle=
                    "http://schemas.xmlsoap.org/soap/encoding/"
                    namespace="urn:LODBService" use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="isLOExist">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="isLOExistRequest">
          <wsdlsoap:body encodingStyle=
                    "http://schemas.xmlsoap.org/soap/encoding/"
                    namespace="urn:LODBService" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="isLOExistResponse">
          <wsdlsoap:body encodingStyle=
                    "http://schemas.xmlsoap.org/soap/encoding/"
                    namespace="urn:LODBService" use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="LODBServiceService">
      <wsdl:port binding="intf:LODBServiceSoapBinding" name="LODBService">
        <wsdlsoap:address
                  location="http://128.227.176.41:8888/axis/LODBService.jws"/>
      </wsdl:port>
    </wsdl:service>
  </wsdl:definitions>
```

D.2 Broker Web Service Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:BrokerService"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="urn:BrokerService" xmlns:intf="urn:BrokerService"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <wsdl:types>
  <schema targetNamespace="urn:BrokerService"
        xmlns="http://www.w3.org/2001/XMLSchema">
   <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
   <complexType name="ArrayOf_xsd_string">
    <complexContent>
     <restriction base="soapenc:Array">
      <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string[]"/>
     </restriction>
    </complexContent>
```

```
  </complexType>
 </schema>
</wsdl:types>
  <wsdl:message name="mainRequest">
    <wsdl:part name="args" type="intf:ArrayOf_xsd_string"/>
  </wsdl:message>
  <wsdl:message name="QueryBroker2Response">
    <wsdl:part name="QueryBroker2Return" type="soapenc:Array"/>
  </wsdl:message>
  <wsdl:message name="QueryBroker2Request">
    <wsdl:part name="serviceKey" type="xsd:string"/>
    <wsdl:part name="interfaceKey" type="xsd:string"/>
    <wsdl:part name="xmlURL" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="RegBrokerRequest">
    <wsdl:part name="serviceKey" type="xsd:string"/>
    <wsdl:part name="interfaceKey" type="xsd:string"/>
    <wsdl:part name="xmlURL" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="QueryBrokerResponse">
    <wsdl:part name="QueryBrokerReturn" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="QueryBrokerRequest">
    <wsdl:part name="interfaceKey" type="xsd:string"/>
    <wsdl:part name="xmlURL" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="mainResponse">
  </wsdl:message>
  <wsdl:message name="RegBrokerResponse">
    <wsdl:part name="RegBrokerReturn" type="xsd:string"/>
  </wsdl:message>
  <wsdl:portType name="BrokerService">
    <wsdl:operation name="main" parameterOrder="args">
      <wsdl:input message="intf:mainRequest" name="mainRequest"/>
      <wsdl:output message="intf:mainResponse" name="mainResponse"/>
    </wsdl:operation>
    <wsdl:operation name="QueryBroker" parameterOrder="interfaceKey xmlURL">
      <wsdl:input message="intf:QueryBrokerRequest"
                  name="QueryBrokerRequest"/>
      <wsdl:output message="intf:QueryBrokerResponse"
                  name="QueryBrokerResponse"/>
    </wsdl:operation>
    <wsdl:operation name="QueryBroker2" parameterOrder="serviceKey
            interfaceKey xmlURL">
      <wsdl:input message="intf:QueryBroker2Request"
                  name="QueryBroker2Request"/>
      <wsdl:output message="intf:QueryBroker2Response"
                  name="QueryBroker2Response"/>
    </wsdl:operation>
    <wsdl:operation name="RegBroker" parameterOrder="serviceKey interfaceKey
            xmlURL">
      <wsdl:input message="intf:RegBrokerRequest" name="RegBrokerRequest"/>
```

```
    <wsdl:output message="intf:RegBrokerResponse"
              name="RegBrokerResponse"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="BrokerServiceSoapBinding" type="intf:BrokerService">
  <wsdlsoap:binding style="rpc"
          transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="main">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="mainRequest">
      <wsdlsoap:body
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:BrokerService" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="mainResponse">
      <wsdlsoap:body
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:BrokerService" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="QueryBroker">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="QueryBrokerRequest">
      <wsdlsoap:body
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:BrokerService" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="QueryBrokerResponse">
      <wsdlsoap:body
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:BrokerService" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="QueryBroker2">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="QueryBroker2Request">
      <wsdlsoap:body
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:BrokerService" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="QueryBroker2Response">
      <wsdlsoap:body
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:BrokerService" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="RegBroker">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="RegBrokerRequest">
      <wsdlsoap:body
                encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="urn:BrokerService" use="encoded"/>
    </wsdl:input>
```

```
      <wsdl:output name="RegBrokerResponse">
        <wsdlsoap:body
                    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                    namespace="urn:BrokerService" use="encoded"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="BrokerServiceService">
    <wsdl:port binding="intf:BrokerServiceSoapBinding" name="BrokerService">
      <wsdlsoap:address location=
                "http://128.227.176.41:8888/axis/BrokerService.jws"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

APPENDIX E
XML SCHEMA DEFINITION OF CLO RUNTIME STATUS MODEL AND LEARNER INFORMATION MODEL

E.1 CLO Runtime Status Model Definition

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.cise.ufl.edu/~glee/CLORuntimeDataModel"
xmlns:clordm="http://www.cise.ufl.edu/~glee/CLORuntimeDataModel"
elementFormDefault="qualified">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      XML Schema for CLO Runtime Data Model.
      Copyright by Gilliean Lee. 2004.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="CLORuntimeDataModel">
    <xsd:complexType>
      <xsd:sequence>
            <xsd:element name="CLOStatus" type="clordm:CLOStatus"/>
            <xsd:element name="CurrentActivity" type="xsd:NMTOKEN"/>
            <xsd:element name="SuspendedActivity" type="xsd:NMTOKEN"/>
        <xsd:element ref="clordm:LeafActivity" minOccurs="0"
                maxOccurs="unbounded"/>
        <xsd:element ref="clordm:NonleafActivity" minOccurs="0"
                maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="InstanceId" type="xsd:NMTOKEN" use="required"/>
      <xsd:attribute name="CLOId" type="xsd:string" use="required"/>
      <xsd:attribute name="LearnerId" type="xsd:NMTOKEN" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="LeafActivity">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="ActivityProgressInformation"
                  type="clordm:ActivityProgressInformation"/>
        <xsd:element name="ObjectiveStatusInformation"
                type="clordm:ObjectiveStatusInformation" minOccurs="0"/>
        <xsd:element name="ActivityDataStatusInformation"
                type="clordm:ActivityDataStatusInformation" minOccurs="0"/>
        <xsd:element name="AttemptProgressInformation"
                type="clordm:AttemptProgressInformation" minOccurs="0"
                maxOccurs="unbounded"/>
        <xsd:element name="BoundLOId" type="xsd:NMTOKEN" minOccurs="0"/>
```

```
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Name" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="NonleafActivity">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="ActivityProgressInformation"
                type="clordm:ActivityProgressInformation"/>
      <xsd:element name="ObjectiveStatusInformation"
                type="clordm:ObjectiveStatusInformation" minOccurs="0"/>
      <xsd:element name="ActivityDataStatusInformation"
                type="clordm:ActivityDataStatusInformation" minOccurs="0"/>
      <xsd:element name="AttemptProgressInformation"
                type="clordm:AttemptProgressInformation" minOccurs="0"
                maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Name" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="ObjectiveStatusInformation">
  <xsd:sequence>
    <xsd:element name="ObjectiveSatus" type="clordm:ObjectiveStatus"
              minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ObjectiveStatus">
  <xsd:sequence>
    <xsd:element name="SatisfiedStatus"
              type="clordm:ObjectiveSatisfiedInformation"/>
        <xsd:element name="NormalizedMeasure"
              type="clordm:NormalizedValue"/>
        <xsd:element name="ProgressStatus" type="xsd:boolean"/>
  </xsd:sequence>
     <xsd:attribute name="ObjectiveId" type="xsd:NMTOKEN" use="required"/>
</xsd:complexType>

<xsd:simpleType name="ObjectiveSatisfiedInformation">
     <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="Satisfied"/>
        <xsd:enumeration value="Unsatisfied"/>
   <xsd:enumeration value="Unknown"/>
     </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="ActivityDataStatusInformation">
  <xsd:sequence>
```

```
    <xsd:element name="ActivityDataStatus" type="clordm:ActivityDataStatus"
              minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ActivityDataStatus">
  <xsd:sequence>
            <xsd:element name="DataType" type="xsd:NMTOKEN"/>
            <xsd:element name="Value" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attribute name="DataId" type="xsd:NMTOKEN" use="required"/>
</xsd:complexType>

<xsd:complexType name="ActivityProgressInformation">
  <xsd:sequence>
            <xsd:element name="AbsoluteDuration" type="xsd:unsignedLong"/>
            <xsd:element name="ExperiencedDuration"
                  type="xsd:unsignedLong"/>
            <xsd:element name="AttemptCount" type="xsd:unsignedShort"/>
            <xsd:element name="StartTime" type="xsd:unsignedLong"/>
            <xsd:element name="CurrentStatus"
                  type="clordm:ActivityProgress"/>
            <xsd:element name="ProgressStatus" type="xsd:boolean"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="ActivityProgress">
      <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="Inactive"/>
        <xsd:enumeration value="Active"/>
    <xsd:enumeration value="Suspended"/>
    <xsd:enumeration value="Supressed"/>
    <xsd:enumeration value="Disabled"/>
      </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="AttemptProgressInformation">
  <xsd:sequence>
            <xsd:element name="CompletionAmount"
                  type="clordm:NormalizedValue"/>
            <xsd:element name="AbsoluteDuration" type="xsd:unsignedLong"/>
            <xsd:element name="ExperiencedDuration"
                  type="xsd:unsignedLong"/>
            <xsd:element name="StartTime" type="xsd:unsignedLong"/>
            <xsd:element name="ProgressStatus" type="xsd:boolean"/>
            <xsd:element name="CompletionStatus" type="xsd:boolean"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="CLOStatus">
      <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="Started"/>
    <xsd:enumeration value="Suspended"/>
```

```
      <xsd:enumeration value="Finished"/>
        </xsd:restriction>
  </xsd:simpleType>

   <xsd:simpleType name="NormalizedValue">
     <xsd:restriction base="xsd:decimal">
       <xsd:minInclusive value="0.0"/>
       <xsd:maxInclusive value="1.0"/>
     </xsd:restriction>
   </xsd:simpleType>
</xsd:schema>
```

E.2 Learner Information Model Definition

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.cise.ufl.edu/~glee/LearnerInformationModel"
xmlns:lim="http://www.cise.ufl.edu/~glee/LearnerInformationModel"
elementFormDefault="qualified">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      XML Schema for Learner Information Model.
      Copyright by Gilliean Lee. 2004.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="learnerInformation">
     <xsd:complexType>
       <xsd:sequence>
         <xsd:element name="dateCreated" type="xsd:date"/>
         <xsd:element name="dateUpdated" type="xsd:date"/>
         <xsd:element name="identification" type="lim:identification"/>
         <xsd:element name="goal" type="lim:goal" minOccurs="0"
                     maxOccurs="unbounded"/>
         <xsd:element name="qcl" type="lim:qcl" minOccurs="0"
                     maxOccurs="unbounded"/>
         <xsd:element name="accessibility" type="lim:accessibility"
                     minOccurs="0"/>
         <xsd:element name="learningHistory" type="lim:learningHistory"
                     minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element name="securitykey" type="lim:securitykey" minOccurs="0"
                     maxOccurs="unbounded"/>
         <xsd:element name="affiliation" type="lim:affiliation" minOccurs="0"
                     maxOccurs="unbounded"/>
         <xsd:element name="interest" type="lim:interest" minOccurs="0"/>
         <xsd:element name="CLOSearched" type="lim:CLOSearched"
                     minOccurs="0" maxOccurs="unbounded"/>
       </xsd:sequence>
     </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="CLOSearched">
    <xsd:sequence>
      <xsd:element name="LOID" type="xsd:NMTOKEN"/>
```

```
  </xsd:sequence>
  <xsd:attribute name="UDDIKey" type="xsd:NMTOKEN"/>
</xsd:complexType>

<xsd:complexType name="identification">
  <xsd:sequence>
    <xsd:element name="firstName" type="xsd:string"/>
    <xsd:element name="lastName" type="xsd:NMTOKEN"/>
    <xsd:element name="address" type="lim:address" minOccurs="0"/>
    <xsd:element name="contactInfo" type="lim:contactInformation"
              minOccurs="0"/>
    <xsd:element name="demographics" type="lim:demographics"
              minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="userid" type="xsd:NMTOKEN"/>
</xsd:complexType>

<xsd:complexType name="address">
  <xsd:sequence>
    <xsd:element name="street" type="xsd:string" minOccurs="0"/>
    <xsd:element name="city" type="xsd:string" minOccurs="0"/>
    <xsd:element name="state_province" type="xsd:string" minOccurs="0"/>
    <xsd:element name="postalCode" type="xsd:string" minOccurs="0"/>
    <xsd:element name="country" type="xsd:string" minOccurs="0"/>
    <xsd:element name="timeZone" type="xsd:NMTOKEN" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="contactInformation">
  <xsd:sequence>
    <xsd:element name="telephone" type="xsd:string" minOccurs="0"/>
    <xsd:element name="fax" type="xsd:string" minOccurs="0"/>
    <xsd:element name="mobile" type="xsd:string" minOccurs="0"/>
    <xsd:element name="email" type="xsd:string" minOccurs="0"/>
    <xsd:element name="webHP" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="demographics">
  <xsd:sequence>
    <xsd:element name="placeOfBirth" type="lim:address" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="gender" type="lim:gender"/>
  <xsd:attribute name="dob" type="xsd:date"/>
</xsd:complexType>

<xsd:simpleType name="gender">
     <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="Male"/>
      <xsd:enumeration value="Female"/>
      <xsd:enumeration value="NA"/>
     </xsd:restriction>
</xsd:simpleType>
```

```
<xsd:complexType name="goal">
  <xsd:sequence>
    <xsd:element name="comment" type="xsd:string" minOccurs="0"/>
    <xsd:element name="description" type="xsd:string" minOccurs="0"/>
    <xsd:element name="setupDate"  type="xsd:date" minOccurs="0"/>
    <xsd:element name="targetDate"  type="xsd:date" minOccurs="0"/>
    <xsd:element name="lastUpdated"  type="xsd:date" minOccurs="0"/>
    <xsd:element name="completionDate"  type="xsd:date" minOccurs="0"/>
    <xsd:element name="subgoal" type="lim:goal" minOccurs="0"
              maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:NMTOKEN" use="required"/>
  <xsd:attribute name="priority" type="xsd:integer" use="required"/>
  <xsd:attribute name="status" type="lim:goalStatus" use="required"/>
</xsd:complexType>

<xsd:simpleType name="goalStatus">
     <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="Active"/>
      <xsd:enumeration value="Inactive"/>
      <xsd:enumeration value="Completed"/>
     </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="qcl">
  <xsd:sequence>
    <xsd:element name="description" minOccurs="0" type="xsd:string"/>
    <xsd:element name="achievedDate" minOccurs="0" type="xsd:date"/>
    <xsd:element name="issueOrganization" type="xsd:string" minOccurs="0"/>
    <xsd:element name="registrationNo" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="title" type="xsd:string" use="required"/>
  <xsd:attribute name="level" type="xsd:string"/>
  <xsd:attribute name="type" type="lim:qcltype" use="required"/>
</xsd:complexType>

<xsd:simpleType name="qcltype">
     <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="Qualification"/>
      <xsd:enumeration value="Certification"/>
      <xsd:enumeration value="License"/>
      <xsd:enumeration value="Other"/>
     </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="accessibility">
  <xsd:sequence>
    <xsd:element name="comment" type="xsd:string" minOccurs="0"/>
    <xsd:element name="language" type="lim:language" minOccurs="0"
              maxOccurs="unbounded"/>
    <xsd:element name="preference" type="lim:preference" minOccurs="0"/>
    <xsd:element name="disability" type="lim:disability" minOccurs="0"/>
```

```
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="language">
  <xsd:sequence>
    <xsd:element name="comment" minOccurs="0" type="xsd:string"/>
    <xsd:element name="proficiency" type="lim:proficiency" maxOccurs="4"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="proficiency">
  <xsd:sequence>
  </xsd:sequence>
  <xsd:attribute name="profcode" type="lim:proficiencyCategory"
          use="required"/>
  <xsd:attribute name="proficiencyGrade" type="lim:proficiencyGrade"/>
</xsd:complexType>

<xsd:simpleType name="proficiencyGrade">
     <xsd:restriction base="xsd:NMTOKEN">
       <xsd:enumeration value="Native"/>
         <xsd:enumeration value="Excellent"/>
       <xsd:enumeration value="Good"/>
       <xsd:enumeration value="Fair"/>
       <xsd:enumeration value="Poor"/>
     </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="proficiencyCategory">
     <xsd:restriction base="xsd:NMTOKEN">
       <xsd:enumeration value="OralSpeak"/>
         <xsd:enumeration value="OralComp"/>
       <xsd:enumeration value="Read"/>
       <xsd:enumeration value="Write"/>
     </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="preference">
  <xsd:sequence>
    <xsd:element name="internetConnection" type="lim:networkConnection"
              minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="learningStyle" type="lim:learningStyle" minOccurs="0"
              maxOccurs="unbounded"/>
    <xsd:element name="mediaType" type="lim:mediaType" minOccurs="0"
              maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="networkConnection">
     <xsd:restriction base="xsd:NMTOKEN">
     <xsd:enumeration value="Dialup"/>
        <xsd:enumeration value="DSL"/>
```

```
      <xsd:enumeration value="T1"/>
      <xsd:enumeration value="T2"/>
      </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="learningStyle">
      <xsd:restriction base="xsd:NMTOKEN">
      <xsd:enumeration value="Receptive"/>
         <xsd:enumeration value="Directive"/>
      <xsd:enumeration value="Guided Discovery"/>
      <xsd:enumeration value="Exploratory"/>
      </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="mediaType">
  <xsd:restriction base="xsd:NMTOKEN">
      <xsd:enumeration value="DownloadableMovie"/>
      <xsd:enumeration value="DownloadableMovieCaption"/>
      <xsd:enumeration value="StreamingMovie"/>
      <xsd:enumeration value="StreamingMovieCaption"/>
      <xsd:enumeration value="StreamingAudio"/>
      <xsd:enumeration value="Presentation"/>
      <xsd:enumeration value="Document"/>
      <xsd:enumeration value="Flash"/>
      <xsd:enumeration value="Any"/>
      <xsd:enumeration value="HTML"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="disability">
  <xsd:sequence>
      <xsd:element name="comment" type="xsd:string" minOccurs="0"/>
      <xsd:element name="info" type="lim:disabilityInfo" minOccurs="0"
               maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="disabilityInfo">
  <xsd:sequence>
     <xsd:element name="comment" type="xsd:string" minOccurs="0"/>
     <xsd:element name="degree" type="lim:disabilityDegree"/>
  </xsd:sequence>
  <xsd:attribute name="area" type="lim:disabilityArea" use="required"/>
</xsd:complexType>

<xsd:simpleType name="disabilityArea">
      <xsd:restriction base="xsd:NMTOKEN">
     <xsd:enumeration value="Hearing"/>
     <xsd:enumeration value="Visual"/>
     <xsd:enumeration value="Moving"/>
     <xsd:enumeration value="Mental"/>
      </xsd:restriction>
</xsd:simpleType>
```

```
<xsd:simpleType name="disabilityDegree">
      <xsd:restriction base="xsd:NMTOKEN">
     <xsd:enumeration value="Ignorable"/>
     <xsd:enumeration value="Light"/>
     <xsd:enumeration value="Correctable"/>
     <xsd:enumeration value="NeedAssistance"/>
      </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="securitykey">
  <xsd:sequence>
     <xsd:element name="password" type="xsd:string"/>
     <xsd:element name="description" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="interest">
  <xsd:sequence>
     <xsd:element name="product" type="lim:typeInfo" minOccurs="0"
               maxOccurs="unbounded"/>
     <xsd:element name="description" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="typeInfo">
  <xsd:sequence>
     <xsd:element name="typeName" type="xsd:string"/>
     <xsd:element name="description" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="affiliation">
  <xsd:sequence>
     <xsd:element name="classification" type="lim:classification"/>
     <xsd:element name="organization" type="xsd:string"/>
     <xsd:element name="status" type="lim:status"/>
     <xsd:element name="joinDate" type="xsd:date"/>
     <xsd:element name="role" type="lim:role" minOccurs="0"
               maxOccurs="unbounded"/>
     <xsd:element name="description" type="xsd:string" minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:NMTOKEN"/>
</xsd:complexType>

<xsd:simpleType name="role">
      <xsd:restriction base="xsd:NMTOKEN">
      <xsd:enumeration value="Composer"/>
      <xsd:enumeration value="Provider"/>
      <xsd:enumeration value="Learner"/>
      <xsd:enumeration value="Evaluator"/>
      <xsd:enumeration value="Administrator"/>
      </xsd:restriction>
```

```
</xsd:simpleType>

<xsd:simpleType name="status">
      <xsd:restriction base="xsd:NMTOKEN">
      <xsd:enumeration value="Active"/>
      <xsd:enumeration value="Inactive"/>
      </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="classification">
      <xsd:restriction base="xsd:NMTOKEN">
      <xsd:enumeration value="Recreational"/>
      <xsd:enumeration value="Professional"/>
      <xsd:enumeration value="Academic"/>
      </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="learningHistory">
  <xsd:sequence>
      <xsd:element name="description" type="xsd:string" minOccurs="0"/>
     <xsd:element name="startDate" type="xsd:date"/>
     <xsd:element name="finishDate" type="xsd:date" minOccurs="0"/>
     <xsd:element name="unit" type="lim:unit" minOccurs="0"/>
     <xsd:element name="evaluation" type="lim:evaluation" minOccurs="0"/>
     <xsd:element name="testimonial" type="lim:learnerTestimonial"
               minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="CLOId" type="xsd:NMTOKEN" use="required"/>
  <xsd:attribute name="status" type="lim:learningStatus" use="required"/>
</xsd:complexType>

<xsd:simpleType name="learningStatus">
      <xsd:restriction base="xsd:NMTOKEN">
      <xsd:enumeration value="Active"/>
      <xsd:enumeration value="Suspended"/>
      <xsd:enumeration value="Finished"/>
      <xsd:enumeration value="Exception"/>
      <xsd:enumeration value="Failed"/>
      </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="unit">
  <xsd:sequence>
     <xsd:element name="unitName" type="xsd:string"/>
     <xsd:element name="unitValue" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="evaluation">
  <xsd:sequence>
     <xsd:element name="result" type="lim:evaluationResult"/>
     <xsd:element name="date" type="xsd:date"/>
     <xsd:element name="description" type="xsd:string" minOccurs="0"/>
```

```
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="evaluationResult">
    <xsd:sequence>
      <xsd:element name="noOfAttempt" type="xsd:integer" minOccurs="0"/>
      <xsd:element name="normalizedMeasure" type="lim:normalizedValue"/>
      <xsd:element name="minimumSatisfiedNormalizedMeasure"
                type="lim:normalizedValue" minOccurs="0"/>
      <xsd:element name="comment" type="xsd:string" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="learnerTestimonial">
    <xsd:sequence>
      <xsd:element name="date" type="xsd:date"/>
      <xsd:element name="description" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>

   <xsd:simpleType name="normalizedValue">
     <xsd:restriction base="xsd:decimal">
       <xsd:minInclusive value="0.0"/>
       <xsd:maxInclusive value="1.0"/>
     </xsd:restriction>
   </xsd:simpleType>
</xsd:schema>
```

LIST OF REFERENCES

ACGME. (2001). ACGME Outcome project, Retrieved July 17, 2005, from http://www.acgme.org/outcome/project/proHome.asp

Advanced Distributed Learning (ADL). (2003). "Advanced Distributed Learning – Home", Retrieved June 16, 2004, from http://ADLNet.org

Advanced Distributed Learning (ADL). (2004a). "Sharable Content Object Reference Model (SCORM) 1.3: Sequencing and Navigation", Retrieved January 30, 2004, from http://ADLNet.org

Advanced Distributed Learning (ADL). (2004b). "Sharable Content Object Reference Model (SCORM): The SCORM Content Aggregation Model", Retrieved January 30, 2004, from http://ADLNet.org

Altenhofen, M., & Schaper, J. (2002). "Flexible Instructional Strategies for e-learning", *Proceedings of the 35th Hawaii International Conference on System Sciences*, Big Island, Hawaii, USA.

The Apache Software Foundation. (2003). "Welcome to jUDDI", Retrieved June 11, 2005, from http://ws.apache.org/juddi/

The Apache Software Foundation. (2004). "Apache Xindice", Retrieved June 17, 2004, from http://xml.apache.org/xindice/

Blackboard Inc. (2005). "Blackboard" Retrieved April 14, 2005, from http://www.blackboard.com

Brusilovsky, P., & Nijhavan H. (2002). "A Framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities", *Proceedings of World Conference on E-Learning, E-Learn 2002*, Montreal, Canada, pp.154-161.

Clark, R. (2000). "Four Architectures of Instruction", *Performance Improvement*, 39(10), pp. 31-37.

Cisco Learning Institute. (2004). "CLI Virtuoso E-Learning Platform", Retrieved April 27, 2005, from http://www.ciscolearning.org/Assets/Cisco_Learning/PDF/CLI_Virtuoso_Featues_2.2.pdf

Co-operative Learning Object Exchange (CLOE). (2004). “CLOE - Co-operative Learning Object Exchange”, Retrieved June 16, 2004, from http://cloe.on.ca/

Dalley, P., Robinson, B., Weller, J., & Caldwell, C. (2004). “The Use of High-Fidelity Human Patient Simulation and the Introduction of New Anesthesia Delivery Systems”, *Anesthesia and Analgesia*, vol.99, pp.1737–1741.

Degwekar, S., Su, S. Y. W., & Lam, H. (2004). “Constraint Specification and Processing in Web Services Publication and Discovery”, *Proceedings of the International Conference on Web Services*, San Diego, California, USA.

Dosch, M. (2004). “The Anesthesia Gas Machine.” Retrieved February 17, 2005, from http://www.udmercy.edu/crna/agm/index.htm

The EOE Foundation. (2004). “EOE Foundation Home.” Retrieved June 17, 2004, from http://www.eoe.org/index.htm

eCollege. (2005). “eCollege”, Retrieved April 14, 2005, from http://www.ecollege.com/indexflash.learn

Eduworks, & O.K.I. Leadership. (2002). “OKI White Paper: What is the Open Knowledge Initiative?” Retrieved June 15, 2004, from http://web.mit.edu/oki/product/whtpapers/whatis.html

Felder, R.M., and Brent, R. (2005). "Understanding Student Differences." *J. Engr. Education*, vol.94, no.1, 2005, pp.57-72.

Gas Man. (2004). “Gas Man Home Page”, Retrieved February 17, 2005, from http://www.gasmanweb.com

Graham, S., Simeonov, S., Boubez., T., Davis, D., Daniels, G., Nakamura, Y., & Neyama, R. (2002). *Building Web Services with Java*, p. 34, Sams Publishing.

Gruber, T. R. (1993). “A translation approach to portable ontologies”, *Knowledge Acquisition*, vol.5, no.2, pp.199-220.

Heins, T., & Himes, F. (2002). “Macromedia MX: Strategies and Architectures for eLearningContent”, Retrieved on April 27, 2005, from http://download.macromedia.com/pub/solutions/downloads/elearning/strategy-architecture.pdf

IEEE LTSC (Learning Technology Standards Committee). (2005). “IEEE LTSC - WG12: Learning Object Metadata”, Retrieved on June 12, 2005, from http://ltsc.ieee.org/wg12/

IMS Global Learning Consortium (IMS). (2004). “IMS Global Learning Consortium, Inc.”, Retrieved June 16, 2004, from http://www.imsglobal.org

IMS Global Learning Consortium (IMS). (2005). "IMS Learner Information Package Specification", Retrieved April 7, 2005, from http://www.imsglobal.org/profiles/index.cfm

IMS Global Learning Consortium (IMS). (2003). "IMS Simple Sequencing Specification", Retrieved April 18, 2005, from http://www.imsglobal.org/simplesequencing/index.html

Jester, C., & Miller, S. (2000). "The Four Learning Styles in the DVC Survey", Retrieved on April 22, 2005, from http://www.metamath.com/lsweb/fourls.htm

Kolb, D. (1984). *Experiential Learning: Experience as the Source of Learning and Development*, Prentice Hall, NJ.

Laboratory for Instructional Technology in Education. (2004). "The LearningOnline Network with CAPA", Retrieved June 15, 2004, from http://lon-capa.org/

Lampotang, S, Lizdas, D. E., Gravenstein, N., & Liem, E. B. (In press) "Transparent reality, a simulation based on interactive dynamic graphical models emphasizing visualization", *Educational Technology*.

Lee, G., Zhang, X., & Su, S. Y. W. (2004). "Learning Object Models, Authoring Tools, and Learning Object Registry Service for Achieving Dynamic and Collaborative E-Learning", *Proceedings of the 7th IASTED International Conference on Computers and Advanced Technology in Education 2004*, pp.321-327.

Lee, M., Su, S. Y. W., & Lam, H. (2001). "A Web-based Knowledge Network for Supporting Emerging Internet Applications", *the WWW Journal*, vol.4, no.1/2, pp.121-140.

Lewis, G. (2002). "Introduction to Educational Technologies", Retrieved June 16, 2004, from http://www.warwick.ac.uk/ETS/Publications/Guides/Printer.htm

Lin, J., Ho, C., Sadiq, W., & Orlowska, M.E. (2002). "Using Workflow Technology to Manage Flexible e-Learning Services." *Educational Technology & Society*, vol.5, no.4.

LSAL (Learning Systems Architecture Lab). (2004). "CORDRA: Technical Introduction and Overview", Retrieved April 14, 2005, from http://www.lsal.cmu.edu/lsal/expertise/projects/cordra/intro/intro-v1p00.html

MERLOT. (2004). "MERLOT Home Page", Retrieved June 16, 2004, from http://www.merlot.org/Home.po

OCW. (2003). "MIT OpenCourseWare | OCW Home", Retrieved June 15, 2004, from http://ocw.mit.edu/index.html

OASIS. (2004). “Universal Description, Discovery, and Integration (UDDI)”, Retrieved June 16, 2004, from http://www.uddi.org/

Reason, J. (2000). “Human error: models and management.” *BMJ*, vol.320, pp.768–770.

Sakai Project. (2005). “Sakai Project”, Retrieved April 22, 2005, from http://www.sakaiproject.org/cms/

Sampson, D., Karagiannidis, C., & Cardinali, F. (2002). “An Architecture for Web-based e-Learning Promoting Re-usable Adaptive Educational e-Content”, *Educational Technology & Society*, vol.5, no.4.

Software AG. (2004). “7 good reasons for XML”, Retrieved April 25, 2005, from http://www1.softwareag.com/es/products/tamino/prod_info/7goodreasons.asp

Su, S. Y. W., Huang, C., Hammer, J., Huang, Y., Li, H., Wang, L., Liu, Y., Pluempitiwiriyawej, C., Lee, M., & Lam, H. (2001). “An Internet-based Negotiation Server for E-Commerce”, *VLDB Journal*, vol.10, no.1, pp.72-90.

Sun Microsystems. (2003). “Java Architecture for XML Binding (JAXB)”, Retrieved June 10, 2004, from http://java.sun.com/webservices/jwsdp/index.jsp

Termaat, B., Crowley, R., Agua, B., Tabor, M., McGough, B., Darling, S., Lee, C., Lee, J., Lewis, D., Nichols, B., Dunne, K., Yoshida, C., Epperson, L., Maddocks, P., Donoghue, M., Cisco Networking Academy Program, King, M., Goga, L., Uyeda, S., Birch, S., & Barritt, C. (2003). “Reusable Learning Object Strategy: Designing and Developing Learning Objects for Multiple Learning Approaches”, Retrieved June 29, 2004, from http://business.cisco.com/servletwl3/FileDownloader/iqprd/104108/104108_kbns.pdf

Thomson. (2005). “Thomson NETg – Knowledge Now”, Retrieved April 13, 2005, from http://www.netg.com/

W3C. (2001a). “XML Schema Part 0: Primer”, Retrieved April 10, 2004, from http://www.w3.org/TR/xmlschema-0

W3C. (2001b). “Web Services Description Language (WSDL) 1.1”, Retrieved June 10, 2004, from http://www.w3.org/TR/wsdl

W3C. (2003). “Simple Object Access Protocol”, Retrieved June 15, 2004, from http://www.w3.org/TR/SOAP/

W3C. (2004). “Web Ontology Language (OWL)”, Retrieved on April 21, 2005, from http://www.w3.org/2004/OWL/

Weinger, M. (1999). “Anesthesia Equipment and Human Error”, *Journal of Clinical Monitoring*, vol.15, pp.319–323.

Washington State University. (2004). “Anesthesia Information from the College of Veterinary Medicine”, Retrieved April 14, 2005, from http://www.vetmed.wsu.edu/depts-anesthesia/equipment/equipment.html

WebCT. (2005). “WebCT.com”, Retrieved April 14, 2005 from http://www.webct.com

White, V. (2002). “Virtual Anesthesia Machine: Teaching without Risk”, *UniSci*, Retrieved April 14, 2005 from http://unisci.com/stories/20021/0128025.htm

Williamson, J., Webb, R., Sellen A, et al. (1993). “Human failure: an analysis of 2000 incident reports”, *Anaesthesia and Intensive Care*, vol.21, pp.678–683.

York, B., Dam, A.v., Ullman, J., Soloway, E., Pollack, J., Kay, A., & Kalil, T. (2002). “A Teacher for Every Learner”, Retrieved April 8, 2004, from http://www.cra.org/Activities/grand.challenges/slides/education.pdf

BIOGRAPHICAL SKETCH

Gilliean Lee was born in Seoul, Korea. He received his B.S. and M.S. degrees in computer science from Sogang University, Korea, in 1992 and 1994, respectively. He was then a senior software researcher at Hyundai Electronics Co., Hyundai Information Tech. Co., and Posdata Co. He is a recipient of a University of Florida Alumni Fellowship. He received his M.S. degree in computer and information science and engineering in 2003. His research interests are in e-learning management, Web ontology, e-business, and workflow management.