ADAPTIVE E-LEARNING USING ECpAA RULES, BAYESIAN NETWORKS AND GROUP PROFILE AND PERFORMANCE DATA

By

SANGHYUN SEO JEON

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

2010

To my family, my husband, Okjune, my children, Yerin (Stella) and Hohyun

ACKNOWLEDGMENTS

Thanks to God for making this dissertation possible and providing everything and everyone I needed all the times. At the beginning of every semester, I was wondering whether I could survive the semester. To be a PhD student has been a long struggle with myself, as well as a long journey to discover another side of myself. During this journey, I met many unforgettable people.

First of all, I owe my deepest gratitude to my advisor Dr. Su for his patience and encouragement to persevere as a PhD student, and his many comments and valuable advice during the entire dissertation process. He has taught me much, from development of my original ideas to putting those ideas into words. From the beginning to the end, I have learned more from him than I ever hoped.

I would like to thank Dr. Fishwick for his guidance in approaching problem resolution. I would like to express my gratitude to Dr. Lamportang, who has made the instructional materials of VAM available to us for constructing learning objects and his VAM system available for integration with our system. I am grateful to Dr. Helal for his advice and support of my presentation and work. A special thanks goes to Dr. Lok for his sharp questions, guidance, and his service as a committee member.

I am indebted to many coworkers at the Academic Technology who have supported me all the time. Especially, I would like to thank Michael Stoufer, Leo Wierzbowski, and Melody Kaufmann for reviewing my papers even with a very short notice. A special thank to Adam Bellaire, Douglas Johnson, and Jeff DePree for proofreading this dissertation. I also want to thank Gilliean Lee for the use of some of the system components he developed in our work.

This dissertation would not have been possible without my husband's and my children's support. They always love me regardless of my busy schedule, and continuously encourage me to pursue this journey to the end. I could not finish it without them. My husband is always available when I need him and never fails to support me. My daughter has been extremely valuable in assisting me with nuances of the English language. My son has shown patience and understanding of all the hard work I have gone through. I also offer my regards and blessings to my parents for their endless support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AL	Attempt Limit
ALO	Atomic Learning Object
AGP	Average Grade of Prerequisites
AS	Average Score
BM	Bayesian Model
CA	Condition-Action
CLO	Composite Learning Object
CLS	Competency Level on a Subject
CPA	Current Pre-Assessment score
CPT	Conditional Probability Table
DD	Drill Down
ECA	Event-Condition-Action
ECpAA	Event-Condition_probability-Action-Alternative_action
GELS	Gator E-Learning System
GPA	Grade Point Average
LC	Learning Capability
LG	Learning Goal
LM	Learning Motive
LO	Learning Object
LOM	Learning Object Metadata
LOTR	Learning Object Tools and Repositories
LV	Limit Value
MV	Measure Value
NFS	No Failure Score

PKL	Prior Knowledge Level
PL	Pass Limit
RU	Roll Up

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

ADAPTIVE E-LEARNING USING ECpAA RULES, BAYESIAN NETWORKS AND GROUP PROFILE AND PERFORMANCE DATA

By

Sanghyun Seo Jeon

December 2010

Chair: Stanley Y.W. Su
Major: Computer Engineering

In order to deliver individualized instruction to learners, an adaptive e-Learning system must be able to perform content selection, sequencing and presentation, and to control learners' navigation through content based on their different profiles and performances. However, the profile data provided by learners and the performance data gathered by a system may be incomplete, inaccurate, or contradictory.

This dissertation presents techniques and features, which alleviate the above data problems by evaluating the profile and performance data of each new learner probabilistically based on the profile and performance data of earlier learners. We present the methodology for the construction, search, and selection of learning objects. Our probabilistic rule model allows our system to apply adaptation rules to examine learners' data at various stages of processing a learning activity, and make proper adaptation decisions even though the learners' data may have anomalies. Adaptation rules are processed by a rule engine and a Bayesian Model Processor. Content authors are provided with system facilities to define adaptation rules and Bayesian Models. The prior distribution of a Bayesian model is automatically derived by using the formulas presented in this work together with prior probabilities and weights assigned by the

content author or the instructor. Each new learner’s profile and performance data are used to update the prior distribution, which is then used to evaluate the next new learner. The system thus continues to improve the accuracy of learner evaluation as well as its adaptive capability.

Two applications have been developed to demonstrate several features of adaptation: namely, construction of self-contained and reusable learning objects, dynamic content search and selection, event and rule-based processing of learning objects at six adaptation points, probabilistic specification and evaluation of rule conditions using Bayesian Models, and the support of personalized learning depending on a learner's profile and performance as well as the characteristics of content. Several simulations have been conducted to handle data anomalies, and verify our proposed formulas for deriving the conditional probabilities needed for Bayesian inference. The system architecture and implementation of an adaptive e-learning system are also presented.

CHAPTER 1
INTRODUCTION

1.1 Motivation

The advent of Internet and Web technologies has enabled organizations and individuals all over the world to post and access enormous quantities of multimedia data such as text, images, web simulations, video, and audio. Content authors can effectively construct self-contained, reusable and sharable learning objects from these resources for instructional and training purposes. An adaptive e-learning system's goal is to automatically tailor learning objects to meet the diverse needs of learners. It delivers individualized instruction to each learner by content selection, sequencing, navigation, and presentation of contents based on a learner's profile and performance (Brusilovsky & Maybury, 2002). Such a system can either automatically adapt itself to meet each learner’s needs, or provide the necessary tools for the learner to change the system’s behavior.

Developing an adaptive e-learning system is a very challenging task because the system needs to have specific knowledge about the learner, make the correct adaptation decision, and appropriately apply that decision to change the behavior of the system to suit each learner. These three steps form the process of personalization, each of which poses several challenges. First, acquiring specific knowledge about a learner requires choosing the right data to collect and deciding how to collect them. Second, making an adaptation decision involves determining the proper data variables and values (i.e., the proper data conditions), taking into account the possible correlations among those data conditions, and evaluating them accurately. Finally, the application of the system’s decision entails knowing when adaptation decisions should

be applied and what techniques, tools, and software components are needed. The motivation of our research is to seek answers to these research challenges so that a truly adaptive e-learning system can be developed.

1.2 Research Goal and Contribution

Several adaptive e-learning systems have been developed and tested in various disciplines. The effectiveness of an adaptive system depends heavily on the collection and proper use of profile data provided by learners and performance data collected by the system. Condition-action rules are commonly used to examine these data and guide the system to take proper adaptation actions to meet different learners' needs (De Bra, Stash, & de Lange, 2003; Duitama, Defude, Bouzeghoub, & Lecocq, 2005). The condition part of this type of rules is a Boolean expression which examines some of the learner's profile and performance data that are relevant to an adaptation decision.

There are three basic problems with e-learning systems that use condition-action rules. First, the condition specification of a rule is evaluated deterministically to a true or false value. This means that the content author or instructor (i.e., the expert) must be able to define the precise data conditions under which an adaptation action should be taken. However, in reality, the expert may not have the full knowledge necessary to specify these precise data conditions. Second, some profile data provided by a learner can be missing, incorrect, or contradictory to his/her performance data. These data anomalies can cause serious problems in evaluating the condition specification of a rule; an error made in even a single data condition in a potentially complex specification can cause the entire specification to have a wrong evaluation result, and thus can cause the system to take the wrong action. Third, in traditional rule-based e-learning systems, each data condition given in a rule is evaluated independently. The correlation among

data conditions is not taken into consideration. Since the truth value of one data condition may affect that of another, their correlations are important and should be considered.

To resolve these problems, we take a probabilistic approach to specification and evaluation of the condition part of an adaptation rule. The type of rule we use is called Event-Condition_probability-Action-Alternative_action (ECpAA) rule, which has the following features. First, the condition part of a rule has a probabilistic expression, which specifies that if the probability of the specified data conditions being true is above a given threshold, the adaptation action specified in the action clause is taken. Otherwise the action specified in the alternative action clause is taken to tailor the content to suit a learner. Second, the data conditions given in the condition specification are modeled by a Bayesian Network (Pearl, 1998), which captures the correlations among these data conditions as well as their relative importance (specified by weights) to an adaptation decision. The Bayesian Model is then evaluated by a Bayesian Model Processor to determine the probability of the given data conditions being true. Third, the evaluation is based on not only the profile and performance data of the learner but also the data of other learners who have taken or are currently taking the same learning object. This allows the system to make use of the previous learners' data in making the adaptation decisions for each new learner. Since the condition part of an adaptation rule is specified and evaluated probabilistically and the evaluation takes into consideration the correlations and relative importance of its data conditions as well as the data of previous learners, the evaluation result can be accurate even in the presence of data anomalies and/or inaccurate specification of rule conditions.

Bayesian Networks have been successfully used in some adaptive e-learning systems for various purposes such as assessing a learner's knowledge level (Martin & VanLehn, 1995; Gamboa & Fred, 2001), predicting a learner's goals (Arroyo & Woolf, 2005; Conati, Gertner, & VanLehn, 2002), providing help or feedback (Gertner & VanLehn, 2000; VanLehn & Niu, 2001), or guiding the navigation of content (Butz, Hua, & Maguire, 2008). Using a Bayesian Network requires an informative prior distribution (Kass & Wasserman, 1996) which represents a system's initial assumption on the data of previous learners (Neal, 2001). The prior distribution is composed of prior probabilities and conditional probabilities associated with a set of given data conditions. Choosing an appropriate prior distribution is the key for a successful Bayesian inference (Gelman, 2002). It has been recognized that obtaining an informative prior distribution is the most challenging task in building a probabilistic network (Druzdzel & Gaag, 1995). To facilitate the acquisition of the prior distribution, we develop a user interface tool for the expert (the content author or instructor) to assign prior probabilities and weights to data conditions based on his/her best estimation. We also introduce three formulas for automatically deriving conditional probabilities of correlated data conditions.

All in all, this research aims to improve the accuracy of learner evaluations in an adaptive e-learning system so that learning objects can be delivered in different ways to suit the needs of a diverse learner population. The contributions of this work, which are the key features of our system, are summarized below:

1. The probabilistic approach of the ECpAA rule specification is better suited for handling data anomalies than the deterministic approach of the traditional condition-

action rule specification because the former does not require that the condition specification of a rule be 100% true.

2. The correlations of data conditions given in the Condition_probability part of our rule are important in influencing the system's adaptation decision and are taken into consideration in our system, but not in existing adaptive e-learning systems.

3. The use of Bayesian Networks in our work is an effective way of not only modeling the correlations and relative importance of data conditions given in an adaptation rule, but also evaluating these data conditions probabilistically. They represent the profile and performance data of learners in terms of probability values.

4. Before a Bayesian Model can be used, its prior distribution (i.e., prior probabilities and conditional probabilities of data conditions given in an adaptation rule) needs to be determined. In our work, we have developed a user interface to ease the expert's task of entering prior probabilities and weights associated with the data conditions to be evaluated, and introduced three formulas for the system to automatically derive conditional probabilities. Furthermore, through constantly updating the prior distribution as the data of each new learner become available, our system is able to make use of the profile and performance data of previous learners to continuously improve the accuracy of evaluating the next new learner.

5. In this work, we have implemented an adaptive e-learning system based on the approach and technique described above, and used case studies and simulations to verify that the system does possess desirable adaptive features.

1.3 Paper Organization

This dissertation is organized in the following way. Existing works related to e-learning standards and adaptive e-learning systems are discussed in Chapter 2.

Chapter 3 presents our way of modeling reusable learning objects by using the multimedia learning assets available on the Web and/or provided by content authors, and describes the techniques used for adaptive content search and selection. Chapter 4 discusses the problems associated with existing rule-based adaptive systems in more details, including the issues related to data uncertainty, data correlation, and the use of group data. We also introduce our probabilistic rule model (i.e. ECpAA), and present our approach of using Bayesian Networks to solve the problems associated with existing rule-based adaptive e-learning systems. Chapter 5 proposes three formulas for deriving conditional probability tables that form the prior distribution of a Bayesian Model. Chapter 6 presents the architecture of our system, the Gator E-Learning System. The implementations of its tools and software system components are described. Chapter 7 presents two applications of our adaptive e-learning system in medical instruction and operation of anesthesia machines. Chapter 8 presents our approach of verifying the adaptive features of our system by simulations. A summary and some useful follow-up R&D efforts are given in the last chapter.

CHAPTER 2
RELATED WORK

An adaptive e-learning system must be able to collect data about each learner, evaluate the data accurately, and take the proper adaptation actions for that learner. The development of such a system requires extensive time and cost. Therefore, many research efforts focus on minimizing time and cost by establishing standards, which can provide guided steps for system development and improve the reusability of previously developed materials and components (Paramythis & Loidl-Reisinger, 2004). In this respect, we review existing standards introduced by e-learning communities, and the techniques and features of some developed adaptive e-learning systems and tools. We also discuss what we adopted from these existing works in the development of our system, the Gator E-Learning System (GELS), and how our work differs from them.

2.1 Standards and Specifications of e-Learning Systems

Several standardization bodies such as the Aviation Industry Computer-Based Training Committee (http://www.aicc.org), the IEEE Learning Technology Standards Committee (http://www.ieeeltsc.org), the Advanced Distributed Learning (http://www.adlnet.gov/Technologies/scorm/default.aspx), and the Instructional Management Systems Global Learning Consortium (http://www.imsproject.org/) have issued a number of standards and drafts focused on supporting the development of e-learning systems. However, a standard that comprehensively deals with adaptive e-learning system development does not exist as of yet. Such a standard should be a collection of learner, content, and adaptive learning process standards, because it needs to ensure that the system will select, sequence, navigate, and present learning content to a learner in an adaptive manner. In this section, we survey the existing e-

learning standards for defining metadata of contents, learner data, and learning processes.

2.1.1 Content Metadata Standards

Educational materials on the Web are usually defined, structured and presented using different formats. These resources can be better shared, accessed, and updated if their metadata are specified using a content metadata standard. Standardizing the description of content can especially aid a system in handling learning contents in an organized and efficient manner. This description of content is encoded in the XML metadata.

Metadata are data about data. They describe contents for the purposes of easily sharing the content information and automating the search, retrieval, and use of learning contents in a Web environment. Thus, metadata are very useful in adaptive e-learning because this knowledge about contents can help an adaptive e-learning system quickly choose and present the most suitable learning content to a learner.

Currently, several content metadata standards are available: the Dublin Core Metadata Element Set (Dublin Core Metadata Initiative [DCMI], 2006); the IMS Learning Resource Meta-data Specification Version 1.3 (IMS Global Learning Consortium [IMS], 2005); the Metadata Encoding Transmission Standard (METS, 2010); the Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE, 2006); the ADL Sharable Content Object Reference Model (SCORM) Content Aggregation Model (ADL, 2004); and the IEEE Draft Standard for Learning Object Metadata Specification (IEEE LTSC [LOM], 2002). Among the above, the ‘Dublin Core Schema’ introduced by the Dublin Core Metadata Initiative (http://dublincore.org/) is most popular. Each Dublin Core element is defined by a set of fifteen attributes, which are *contributor,*

coverage, creator, date, description, format, identifier, language, publisher, relation, rights, source, subject, title and *type*. Each element has 7 properties: *the Uniform Resource Identifier (URI), label, definition, comment, type of term, status,* and *date issued*. The Dublin Schema aims to enable intelligent information retrieval of contents.

The LOM (2002) specification was established as an extension of the Dublin Core and became the official standard in June 2002. Metadata in LOM has an 'object cataloging scheme', which uses the XML schema technology. LOM offers nine categories of metadata and attributes for each of these categories, which can be used to tag learning objects. These categories and some selected attributes are shown below:

- General resources: title, language, description, keyword, aggregation level
- Life cycle: version, status, contribute
- Meta-meta-data: metadata scheme
- Technical: format, size, duration, installation remark, other platform requirements
- Educational: interactivity type, learning resource type, typical age range, typical learning time, difficulty, language
- Rights: cost, copyright and other restrictions
- Relation: kind, resource
- Annotation: entity, date, description
- Classification: purpose, taxon path

The goal of LOM is to define a standard set of attributes that can be used to specify the metadata of distributed learning objects and thus facilitate the automatic and dynamic generation of instructions that are composed of these learning objects.

Although LOM is the official standard for specifying content metadata, it is not specifically designed for adaptive systems. Some standardization research efforts have been made to extend LOM's attribute set so that learning objects defined by the enriched attribute set (i.e., the metadata) can be used to match with the data that characterize learners. Thus, the learning object that is most suitable to a learner can be identified. One example is Rumetshofer and Wolfram's work (2003), in which they extend LOM's attribute set by including 'psychological factors' such as cognitive/ learning styles and strategies.

In our work, we also use metadata to define learning objects so that the learning objects can be easily searched and retrieved for purposes of system adaptation. We modify LOM by selecting categories that we deem important to an adaptive system, enriching the attributes of the educational category, and adding a new category called "access point information". The educational category is enhanced by including attributes that also characterize learners. Access point information forms the new category so that distributed learning objects can be more easily searched and retrieved. Details about the role of metadata in the construction, search, and selection of reusable and distributed learning objects will be given in Chapter 3.

2.1.2 Learner Data Standards

An adaptive system must possess not only informative content metadata but also knowledge about a learner in order to correctly match the metadata of learning objects with the data about the learner. There are two types of data about a learner: a learner's profile data and a learner's performance data. These learner information can be better acquired through the usage of standards because learner data standards support the description, collection, maintenance, and retrieval of learner information. In recent

years, there have been several efforts to standardize the specification of a learner. We adopt the specifications of two of the best known efforts, the IMS Learner Information Packaging Specification (IMS [LIP], 2010) and the IEEE LTSC Public and Private Information for Learners (IEEE LTSE [PAPI], 2001), in order to facilitate the collection of learners' profile information and the tracking of their learning progress. Both standards introduce several categories of information for learner specification.

The LIP (2010) uses the following attributes to describe learner profile data: 'identification', 'goal', 'qualifications, certifications and licenses', 'activity', 'interest', 'relationship', 'competency', 'accessibility', 'transcript', 'affiliation' and 'security key'. Our learner profile information adopts all of these attributes except 'activity' and 'relationship' because we believe that 'qualifications, certification and licenses' and 'affiliation' (i.e., records about membership of professional organization) can cover 'activity', which describes any learning related activities. We also collect a learner's learning activities as learner's performance data separately, so that it is not necessary to collect it from a leaner. The 'relationship' is defined as "the container for the definition of the relations between the other core data structure (e.g., 'qcl' and the awarding organization) [in order to construct] complex relationships between the core data structures" (LIP, 2010, p.67). We believe it is not necessary for describing the learner. Instead, we add features such as the specification of preferred 'learning style' and 'context information'. A learner's context information describes the learner's learning environment. Since the development of wireless Internet and the increasingly widespread use of mobile technology, a learner's learning environment has become an important element in an adaptive e-learning design (Ramsden, 2005). In our work, learner profile information is

collected from learners at the time of their registration with our system through a Learner Registration Interface.

Similar to LIP, PAPI (2001) also describes learner profile data. However, PAPI focuses more on describing performance records. It contains six categories: *personal information, relations information, security, preference, performance,* and *portfolio*. Of these, performance information is a record of a learner's performance described by a content identifier, a timestamp, a performance coding scheme, a metric, a valid date, and a certification ID. The performance coding scheme specifies the type of grading, coding, measuring, etc., that the system can use. We adopt the PAPI standard to define the learner performance information model, which is then used to track learners' learning records during the execution of a learning object. The learner performance information is collected by the system based on the interactions between a learner and the system and the results of assessments. Details of GELS' adoption and extension of LIP and PAPI will be discussed in Chapter 6.

2.1.3 Adaptive Learning Process Standards

Internet users have different backgrounds, competencies, preferences, and interests. The aforementioned standards for modeling and collecting profile information, performance information, and content information are important for developing an adaptive e-learning system. However, it is the way that the collected information is used during a learning process that makes an e-learning system adaptive. Since a standard for modeling learning processes will establish a uniform way of designing and sharing them, different e-learning systems will be able to share and reuse these processes. To our knowledge, there is no standard for defining an adaptive learning process, yet.

Therefore, we start by looking at learning process standards for traditional e-learning systems and discuss the requirements of an adaptive learning process.

The commonly accepted learning process standard is the Sequencing and Navigation Standard of SCORM 2004 (ADL, 2004), which is adopted from the IMS Simple Sequencing Specification (IMS, 2003). SCORM's Sequencing and Navigation Standard specifies how to prescribe a content order as well as the conditions that determine if and when content should be delivered to a learner. It consists of a *Sequencing Definition Model*, a *Sequencing Behaviors Model*, and a *Navigation Model*. The Sequencing Definition Model tells how to define sequencing strategies and rules. Since sequencing is controlled by the system, the Sequencing Behaviors Model describes how to handle the sequencing request, such as ‘start’, ‘resume all’, ‘continue’, ‘previous’, and ‘exit’, and what information should be tracked for sequencing purposes. Navigation through contents is controlled by the learner. SCORM's Navigation Model guides an e-learning system on how to make a navigable table of contents and how to provide navigation controls to a learner at run-time, such as allowing the learner to click on a previous button, a next button, and status indicators.

Along with Sequencing and Navigation, SCORM 2004 has other standards such as an Overview, a Content Aggregation Model, a Run-Time Environment, and Compliance Requirements. This collection of standards give the basic framework for e-learning system development and makes SCORM 2004 one of the most widely utilized standards in the e-learning field. However, SCORM 2004 does not provide a standard for developing an adaptive e-learning system. Such a standard would have to provide a facility to specify adaptation rules for a system to carry out different adaptation

strategies. SCORM provides a way of tracking the information obtained from the interaction between a learner and the system in a learning activity. This information can be part of learner data. However, it does not specify how this information should persist from one course to another, so that the information used in one course can be used for its corresponding courses. SCORM also does not take into account a learner's profile such as his/her preferences or learning background.

Much research and many experimental trials have been carried out to add adaptive capabilities and techniques to a system. Rey-Lopez and her colleagues (2006) extended the SCORM Specification by adding a set of adaptation rules which can help a system make suitable adaptation decisions based on a learner's characteristics. Van Rosmalen et al. (2006) presented an adaptation engine, “CopperCore”, which enables a system to achieve and exhibit its adaptation, presentation, and interaction functionalities. A comprehensive summary of current research efforts on adaptive properties and standards can be found in Botsios and Georgiou (2009). However, as of yet, a specific criterion for specifying a set of effective adaptation rules is still under-researched as pointed out in Brusilovsky and Pyelo (2003) due to the complexity of considering numerous factors involved in making adaptation decisions.

Our system, GELS, uses adaptation rules at various stages of processing a learning object. It adopts and extends SCORM's Sequencing and Navigation Standard to achieve adaptive content selection, sequencing, navigation, and delivery based on the metadata of content and the profile and performance data of a learner. The system introduces a new way of adaptation rule specification, and uses an adaptive rule execution engine, a rule editor, and several facilitating tools for defining and processing

adaptation rules. The details on GELS' rules and adaptation strategies will be given in Chapter 4.

In summary, many organizations have made efforts to introduce standards for defining metadata of contents, learners and learning processes. However, as surveyed and examined in this section, the current versions of e-learning standards are not adequate for building an adaptive e-learning system that possesses a comprehensive set of adaptive features. In our work, we do not simply adopt and benefit from some e-learning standards but extend them to enable the implementation of GELS' adaptive features. We use an Extended LOM to specify the metadata of learning objects. For characterizing learners and their knowledge, we use a modified IMS LIP and a modified PAPI, respectively. GELS also adopts SCORM's Sequence and Navigation Standard and uses it in a rule engine to execute a learning process. We supplement SCORM's standard by allowing adaptation rules to be specified and activated during the enactment of a learning process, thus making it adaptive. Since the standards we adopt are widely used in the e-learning community, GELS can easily utilize contents that have been defined by these standards.

2.2 Existing Adaptive Systems and Supporting Tools

Adaptive e-learning systems are often called by different names, most of which use various combinations of terms like Web-based, intelligent, adaptive, and educational/learning. The term "Web" refers to learning content and learning activities that are accessible and executable over the Web. The term "adaptive" indicates that a system can automatically adapt itself to learners with different characteristics and needs. The term "intelligent" signifies that a system uses some sort of artificial intelligence technique to provide support to learners or identify their characteristics,

needs, and situations. Lastly, “educational” or “learning” suggests that the system’s primary application is in education or learning.

There are three categories of existing systems that exhibit adaptive capabilities in different ways. They are *Intelligent Tutoring systems, Adaptive Hypermedia Systems, and Group-based Adaptive Systems*. In the following subsections, we examine the functionalities and techniques of these systems as well as supporting tools used in these systems.

2.2.1 Intelligent Tutoring Systems

An Intelligent Tutoring System (ITS) is an adaptive instructional system that uses an artificial intelligence technique to instruct learners who have different backgrounds, learning capabilities, and goals. Adaptation in ITS is achieved by the incorporation of a “learner knowledge model”. The learner knowledge model is built on information gathered from and about an individual learner in order to assess the learner’s current state. ITS analyzes each learner’s strengths and weaknesses based on the learner model so that the system is able to respond to that learner’s needs and provide individualized instruction and tutoring. For example, Andes, an ITS for instructing classical physics, uses a learner model to provide a learner with individualized coaching in problem solving (Anderson, Corbett, Loedinger, & Pelletier, 1995).

The applications of ITS include computer-based problem-solving monitors, coaches, laboratory instructors, and consultants (Sleeman & Brown, 1982). The effectiveness of an ITS depends heavily on its ability to make good decisions about when and what a leaner should learn. In order to improve the system’s ability to decide, most ITSs use one or more of the following techniques: curriculum sequencing, intelligent analysis of student’s solutions, interactive problem solving support, and

example-based problem solving support (Brusilovsky, 1999). The techniques adopted from ITSs are implemented in the following adaptive systems: adaptive testing by using probabilistic theory (Guzmán, Conejo, & Pérez-de-la-Cruz, 2007), cognitive tutors for collaborative learning by capturing and analyzing learners' behaviors (Harrer, McLaren, Walker, Bollen, & Sewall, 2006), and automatic generation of adaptive courses using dynamic planning techniques in Dynamic Generation of Customized Courses (Brusilovsky & Vassileva, 2003).

ITSs have a significant influence on adaptive e-learning due to the artificial intelligence techniques they use for problem solving support and intelligent solution analysis. However, ITSs have several limitations, which prevent them from becoming truly adaptive e-learning systems. First, the key to providing effective assistance to a learner is an ITS' ability to track the learner’s performance data. These include performance results and some behavior data such as keystrokes, mouse movements, and the amount of time spent on each instruction. However, the collected data may not be used for modeling a learner’s knowledge and skills correctly due to the inconsistent responses and incomplete answers given by the learner, or the unreliable behavior observed by the system. Building a learner model from these observable but uncertain data is a difficult task. Second, most ITSs focus on learners’ performance data rather than their profile data (Sison & Simura, 1998). Since both profile data and performance data are important for characterizing a learner, and these data are interrelated, a learner model that captures less information may degrade a system’s ability to adapt. Third, ITSs present learning contents in a predetermined pattern/structure to its learners. In order to do this, an ITS designer has to predict all possible responses of learners and

decide what content should be presented to learners in advance. This is not always possible in the real life. The fixed and predefined pattern/structure prevents an ITS from delivering personalized instructions to different learners.

Different from ITSs, we use not only the performance data of learners but also their profile data as well as the metadata of contents to evaluate learners so that their knowledge, capabilities and statuses can be more accurately determined. We also tackle the data uncertainty problems by using Bayesian Networks to evaluate the collected data of learners. Instead of using a predetermined pattern of content presentation designed at build-time, GELS allows dynamic binding to learning objects and uses an Event-Condition_probability-Action- Alternative_action (ECpAA) rule-based system to make adaptation decisions at run-time. GELS provides a Bayesian model editor to ease the task of an expert in providing the information needed for deriving the prior distribution of a Bayesian model. Details on GELS' usage of Bayesian Network will be given in Chapter 4.

2.2.2 Adaptive Hypermedia

Adaptive Hypermedia (De Bra, Brusilovsky, & Houben, 1999; Brusilovski, 2001) is a system that contains and manages hypertexts and/or hypermedia. Hypertext is defined as "network information nodes connected by means of relational links" (Parunak, 1990). The goal of Adaptive Hypermedia is to help learners find an "optimal path" through the learning content and to guide learners through the hyperspace of available information. The applications of such a system are in online information/help, information retrieval hypermedia, and educational hypermedia.

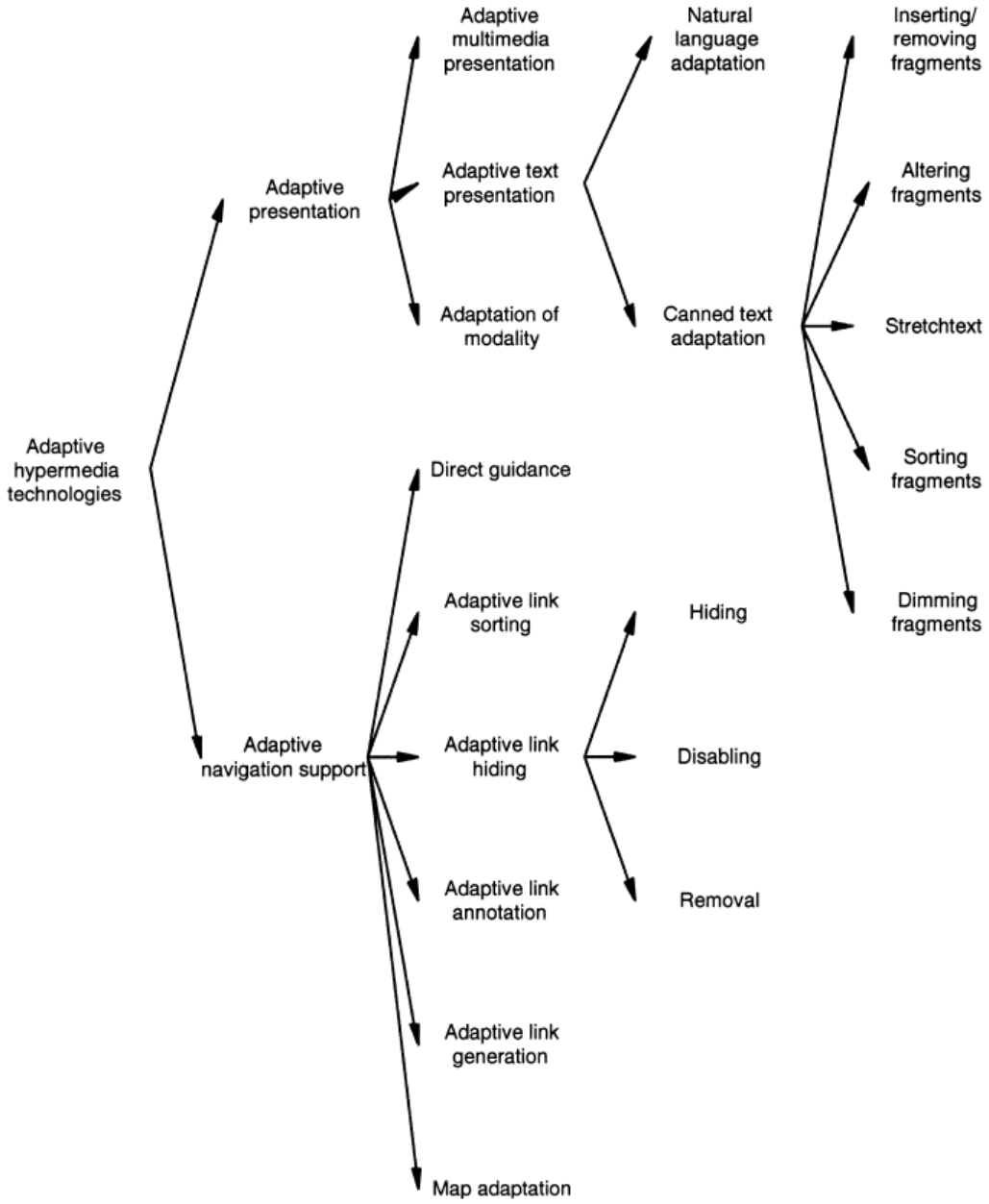

Figure 2-1. Adaptive hypermedia technologies (Brusilovsky, 2001)

As can be seen in Figure 2-1, Adaptive Hypermedia Systems use two different but complementary methods to adapt the content and links of hypermedia pages to suit a learner: adaptive presentation (content-level adaptation) and adaptive navigation (link-level adaptation) (Brusilovski, 2001; Henze & Nejdl, 2004). Adaptive presentation

manipulates content fragments in a hypertext document. Based on the information stored in a learner model, the textual or multimedia content of a page can be adaptively displayed by means of fragment variants (De Bra et al., 1999). The techniques and examples of adaptive presentation include: conditional inclusion/removing/dimming of fragments (i.e., text, figures, etc.) in AHA! (De Bra & Calvi, 1998), stretchtext (technique of using stretched or shrunk text fragments based on learners' interests) in MetaDoc (Boyle & Encarnacion, 1994), altering fragments in Anatom-Tutor (Beaumont, 1994), and sorting fragments in Hypadapter (Hohl, Böcker, & Gunzenhäuser, 1996)). Adaptive navigation, on the other hand, manipulates links in order to guide a learner to choose the best path from the currently presented information to other desired information, depending on the learner's need, level of knowledge, interest, preferences, etc. In the work of Brusilovsky (2001) and Brusilovsky & Nejdl (2005), the explored mechanisms and their examples are: direct guidance with a 'next' or 'continue' button found in most e-learning systems; adaptive link sorting from the most relevant to the least relevant in Hypadapter (Hohl et al., 1996); adaptive link hiding through hiding, disabling, and removing a link in ISIS-Tutor (Brusilovsky & Pesin, 1998); adaptive link annotation (link anchors are presented differently depending on the relevance of the destination) with different font colors in ELM-ART (Weber & Brusilovsky, 2001), different font sizes also in Hypadapter (Hohl et al., 1996), and different font types in InterBook (Brusilovsky, Eklund, & Schwarz, 1998). Recently proposed techniques are adaptive link generation and a navigation map adaptation. Adaptive link generation enables a system to create dynamic links rather than static links for a page (Mitsuhara, Ochi, Lanenishi, & Yano, 2002) and navigation map adaptation provides a graphical representation of a link

structure (Benford et al., 2000). An excellent overview of methods and techniques can be found in Brusilovsky's survey paper (Brusilovsky, 1996).

The contribution of Adaptive Hypermedia Systems to adaptive e-learning is their provision of innovative techniques for adaptive presentation and efficient navigation strategies. However, because these systems control content presentation at the level of fragments within a content, there are several drawbacks. First, although the idea of presenting a single content in various adaptive manners is desirable, the cost of utilizing such technique falls on the content authors. Because Adaptive Hypermedia systems' adaptive presentation techniques include conditional inclusion/removing/dimming of fragments (i.e., text, figures, etc.), a content author must create numerous versions of the same content. The time and cost that are required of content authors can outweigh the benefits of presenting the same content to learners with different formats, combinations of fragments, etc. In our work, we use adaptive techniques to select and present learning objects, which are larger granules of instruction than fragments; thus, less time-consuming and costly to develop the variances of a learning object.

Second, the techniques used in Adaptive Hypermedia Systems can decrease the consistency of how the same content is presented to a learner. Contrary to these systems' original purpose of helping a learner find an "optimal path", applying adaptive techniques too frequently can confuse a learner. For example, if too many adaptive techniques are applied to a hypertext, the presentation of its content may keep on changing every time it is presented to a learner. In our work, we identify a limited number of meaningful stages in processing a learning activity, at which adaptation rules are applied.

Third, an Adaptive Hypermedia system provides learning materials in the form of a hypertext. As the aforementioned definition of hypertext implies, a hypertext is required to have a predefined dependency or precedence (i.e., prerequisite) among ‘network information nodes’, because a hypertext is presented and/or navigated based on a learner’s knowledge status and the dependency information. However, it is not easy to set up these dependencies in advance. If this information is omitted or incorrect, the user may receive inept guidance, which can be worse than no guidance at all. Our system also allows the specification of relationships between learning objects as a part of metadata specification (e.g., a learning object is a part or a version of another learning object). However, the specification of relationships is optional, rather than mandatory, as it is with hypertexts. GELS is able to use other metadata to select learning objects that are most suitable to learners. Lastly, just like in ITSs, the learner model used in an adaptive hypermedia system is response-dependent. A response-dependent learner model is incomplete because the profile data of a learner are not included in the model. It can also be incorrect because the learner data gathered by the system can be inconsistent and inaccurate. In our work, we provide a comprehensive learner model, which captures not only the performance data but also the profile data of a learner. Performance data is collected and continuously updated by the system, while profile data is updated by the learner through a graphical user interface. We also minimize the impact of uncertainties in a learner's data by using Bayesian Network.

GELS adopts some adaptive techniques introduced in adaptive hypermedia systems for achieving adaptive content presentation and navigation. It provides a

comprehensive and up-to-date learner model and evaluates the learner data accurately in the presence of data anomalies discussed in the introduction section.

2.2.3 Group-based Adaptive Systems

This category of systems uses the information about a group of learners to offer individualized assistance to learners and to achieve better coordination and collaboration among them. Examples of this type of systems include *the Intelligent Class Monitoring* systems and *Intelligent Collaborative Learning* systems named in (Brusilovsky & Peylo, 2003). Group information represents a system's knowledge about learners, which can be obtained from Web Logs. An *Intelligent Class Monitoring system* compares the records of different learners to identify those records that are different from those of their peers. In this way, the system can identify the learners in need of help and offer them the proper assistance. HyperClassroom (Oda, Staoh, & Watanabe, 1998) is an example of this type of systems. It uses 'fuzzy mechanisms' to identify learners who need help in a Web classroom.

Intelligent Collaborative Learning systems can be divided into two subcategories: *adaptive group formation and peer help* and *adaptive collaboration support* (Brusilovsky & Peylo, 2003). The first subcategory aims to use a system's knowledge about different learners to form a matching group for different kinds of collaborations. Examples include forming a group for collaborative problem solving at a proper time, or finding the most competent peer to aid a student on a given topic (i.e., finding a person with a model showing good knowledge of this topic). PHelpS (Peer Help System) is a system that uses *the adaptive group formation and peer help* technology in discussion forums (Greer et al., 1998). By using its knowledge about different learners, an adaptive collaboration support system can coach or advise collaborating peers. Examples of this

type of work are COLER (Constantino-Gonzalez, Suthers, & Escamilla De Los Santos, 2003) and EPSILON (Soller & Lesgold., 2003).

While group-based adaptive systems use group information to offer personalized assistance to learners and enable the coordination and collaboration of learners in their learning activities, our system uses the information about a group of previous learners to increase the accuracy of evaluating new learners so that suitable adaptation actions can be taken by the system to deliver individualized instructions. Our group information contains the profile and performance data of those previous learners who have learned from the same learning object. This information is continuously updated as each new learner is going through a learning process. This approach enables our system to continuously improve its ability to accurately evaluate a new learner and deliver individualized instruction to the learner.

2.2.4 Supporting Tools

A variety of tools have been developed in the existing adaptive systems: authoring tools, monitoring tools, and assessment tools. Authoring tools have been developed in MetaLinks (Murray, 2003), AHA! (De Bra & Calvi, 1998) and NetCoach (Weber, Kuhl, & Weibelzahl, 2001) for authoring adaptive hypermedia contents, while an authoring tool has been developed in Algebra Tutor (Ritter, Anderson, Cytrynowitz, & Medvedeva, 1998) for ITS content editors. Our system, GELS, also provides authoring tools for content authors and content composers to construct self-contained, reusable, and distributed learning objects by using educational resources on the Web. These tools can be installed at content authors and composers' network sites in order to promote various types of content creation. CourseVis uses techniques adopted from information visualization and Web log data, and provides instructors with a graphical learner

monitoring tool (Mazza & Dimitrova, 2004). GELS provides a monitoring tool called 'MonitorStudents' for instructors, and an activity tracking tool called 'My CLOs' for learners. Through 'MonitorStudents', an instructor can monitor his/her learners' progress, while a learner can check his/her own activities by using 'My CLOs'. Assessment tools are developed in many systems such as SIETTE (Systems of Intelligent Evaluation using Tests for Tele-education: an adaptive Web-based assessment system by using Item Response Theory) (Rios, Pérez-de-la-Cruz, & Conejo, 1998), AthenaQTI (a tool for authoring personalized assessments) (Tzanavari, Retalis, & Pastellis, 2004), and QuizGuide (Brusilovsky, Sosnovsky, & Shcherbinina, 2004) for creating assessments. We also provide an assessment tool as a part of learning object authoring tools. Our assessment tool supports four different assessment types: namely, 'text', 'interactive', 'multiple choices single-selection', and 'multiple choices multiple-selections'. We also provide an 'autograde' function for the instructors' convenience.

In summary, three categories of existing systems possess adaptive capabilities. Intelligent Tutoring Systems provide a learner model and use various artificial intelligence techniques to deliver individualized instructions to learners. Adaptive Hypermedia Systems provide innovative techniques for achieving adaptive content presentation and navigation. Group-based adaptive systems use the information about a group of learners to monitor their progresses, offer them individualized assistance, and enable their coordination and collaboration in learning activities. Web-based tools have been developed in these systems for authoring contents, monitoring learners' progresses, and assess learners' performances.

GELS adopts and extends the adaptive techniques introduced above. It provides an enriched learner model to capture both profile and performance data of learners, and uses group information represented by prior distributions of Bayesian Models to improve the accuracy of evaluating learners. GELS applies adaptive techniques for content presentation and navigation at six specific stages of processing a learning activity. It also provides virtual e-learning community members with supporting tools to create learning objects, monitor learners' progresses and assess their performances. Different from all the existing adaptive systems, GELS uses an ECpAA rule specification and a Bayesian Model Processor for probabilistic specification and evaluation of the condition clauses of adaptation rules so that the problems of data anomalies and data correlations addressed in the introduction section can be mitigated.

CHAPTER 3
CONTENT CONSTRUCTION AND ADAPTIVE CONTENT SEARCH AND SELECTION

The goal of an adaptive e-learning system is to take into account the diverse needs of learners and provide them with tailored learning materials. To achieve this goal, the system should offer facilities for creating a variety of reusable contents as well as for searching, selecting and presenting the most appropriate content to a learner. In this chapter, we present GELS' method and tools for content construction and the adaptive techniques used for content search and selection.

3.1 Content Construction in Gator E-Learning System

Many educational resources on the Web are highly unstructured, heterogeneous, and scattered. In order to provide Web users with more structured and usable learning contents, there is a need to develop a method and tools that package these resources into self-contained and reusable learning objects for instructional and training purposes.

GELS recognizes three types of content components, as illustrated in Figure 3-1: learning assets, Atomic Learning Objects (ALOs) and Composite Learning Objects (CLOs) introduced in Lee and Su (2006). Multimedia learning assets can be in forms such as text, image, HTML, XML, Audio, Flash, JPEG, and simulation. The two types of learning objects, ALO and CLO, are modeled based on the ideas presented in SCORM's Content Aggregation Model (ADL, 2004), Cisco Learning Institute's Reusable Information Objects and Reusable Learning Objects (Cisco Systems, 1999 & 2003), and LOM (LOM, 2002), but with some important extensions.

3.1.1. Atomic Learning Object

An ALO is a self-contained and reusable unit of instruction defined in XML. It typically consists of *content items*, *practice items* and *assessment items.* A content,

practice, or assessment item can be constructed out of multimedia learning assets. It is either prepared by a content author or accessible through its URL. Practice and assessment items can be optional. Assessment items can be given to a learner both before and/or after the presentation of content items (i.e., pre-assessment and post-assessment). Four different assessment types are currently supported by GELS: namely, 'text', 'interactive', 'multiple choices single selection', and 'multiple choices multiple selections'. The 'interactive' assessment is a special type of assessment method used in our application of GELS for teaching medical personnel in the use and operation of anesthesia machines (see Section 7.2 for details). In this method, GELS presents a problem to a learner, then monitors the learner's interactions with a simulated anesthesia machine to determine his/her understanding of and ability to solve that problem.

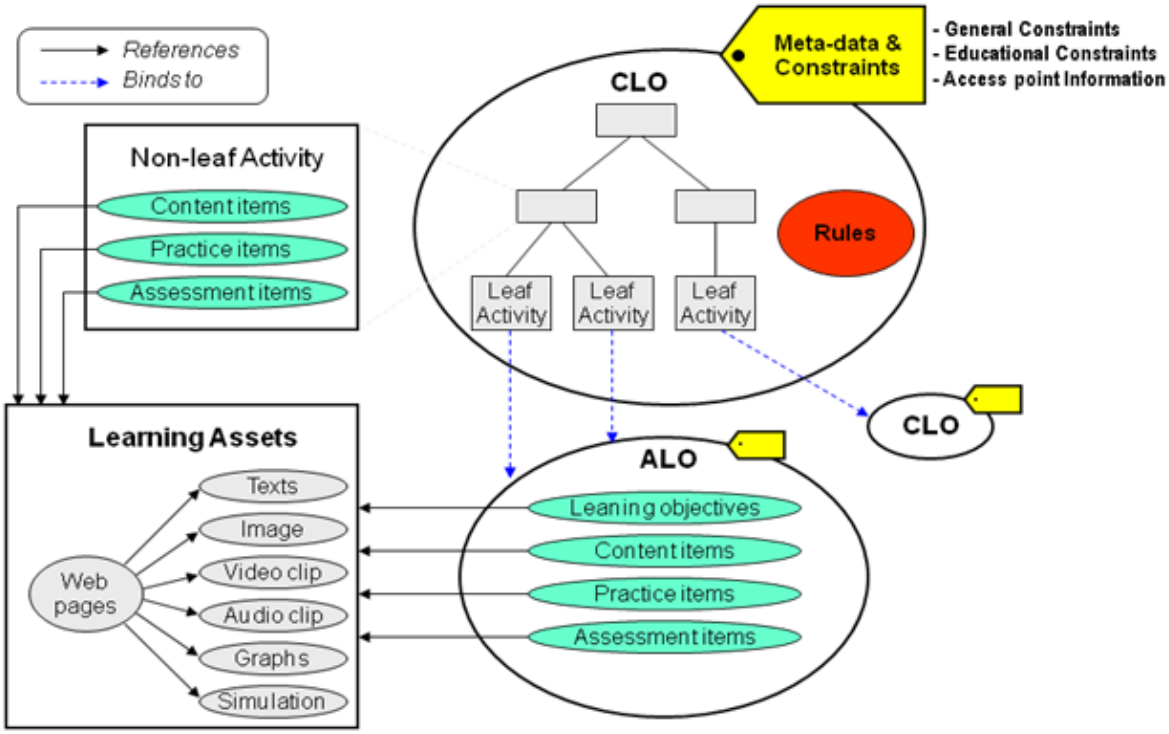

Figure 3-1. The content components of GELS

We also add *metadata* and a *sequence* specification of learning items to an ALO. The sequence specification specifies the delivery order in which content and practice items are presented to a learner when an ALO instance is processed by the system. Metadata are data that describe the learning object. They are used for the registration and retrieval of the ALO through an LO Broker. For example, the metadata of the ALO called "FDACheckStep1" consist of: 'Title' is "FDACheckStep1", 'Keywords' is "{FDACheckStep1, manual resuscitator, Ambu bag}", 'Author' is "Sem Lampotang", 'Language' is "English", and 'Media_Format' is "HTML" and "Shockwave". The detailed meta-model of learning object will be discussed in Section 3.2.

3.1.2. Composite Learning Object

A CLO is composed of some constructed ALOs and CLOs. Like that of SCORM (ADL, 2004), GELS' CLO is defined as an activity tree, which models a learning process and specifies the delivery sequence of learning objects referenced in it. As shown in Figure 3-2, a CLO is an activity tree which consists of *activities, connectors,* and *edges*. This example CLO is taken from our application of GELS mentioned above. Each node of an activity tree is called an "activity". It represents a unit of instruction to be delivered to a learner. There are two types of activities depending on their locations in an activity tree: *non-leaf activity* and *leaf activity*. A non-leaf activity is a node that has child nodes, and a leaf activity is one that does not.

Unlike that of SCORM, GELS' non-leaf activity can have an introduction to and/or a summary of the contents presented in its child activities. A *non-leaf activity* in GELS can also have practice items and assessment items for enhancing and evaluating a learner's knowledge gained from the non-leaf activity and its child activities. Thus,

GELS allows not only a leaf activity but also a non-leaf activity to have content, practice, and assessment items.

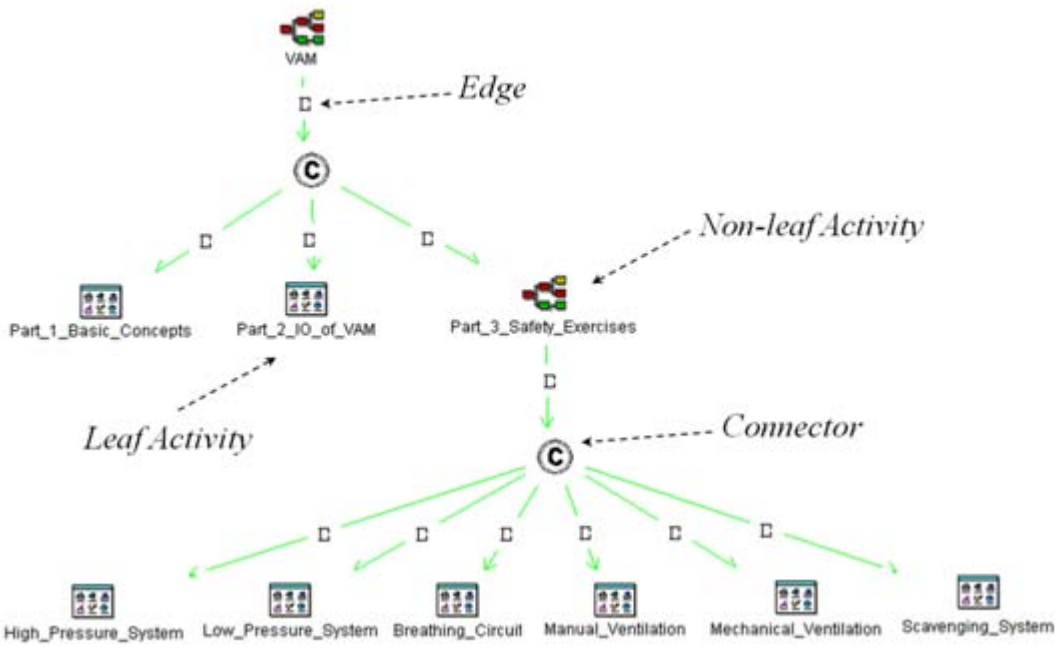

Figure 3-2. The activity tree of a composite learning object

In an activity, a learning objective can be specified to indicate whether a learner has or has not met an instructional goal of the activity. It is adopted from SCORM's Sequencing and Navigation model (ADL, 2004). The status of the learning objective can be represented numerically or marked as unknown, satisfied or unsatisfied.

Additionally, a set of adaptation rules can be specified in both a non-leaf activity and a leaf activity. A non-leaf activity can have six adaptation rules: 'Before-Activity' rule, 'After-Pre-Assessment' rule, 'Drill-Down' rule, 'Roll-Up' rule, 'After-Post-Assessment' rule, and 'Before-End-Activity' rule. A leaf activity can have four adaptation rules: 'Before-Activity' rule, 'After-Pre-Assessment' rule, 'After-Post-Assessment' rule, and 'Before-End-Activity' rule. These rules correspond to different

stages of processing a non-leaf activity or a leaf activity. They are used to select, sequence, navigate, and present content to a learner in an adaptive manner. The details of GELS' adaptation rule specification and execution will be given in Chapter 4.

A *leaf activity* can be statically or dynamically bound to an ALO or a CLO. The provision to bind a leaf activity to a CLO is an important feature of our system. It enables the construction of a large granule of instruction out of modularized instructions, thus increasing content scalability and reusability. For example, in our application of GELS in the instruction of 'FDA's pre-use check of traditional anesthesia machines', the instruction consists of three CLOs: Root CLO, Low Pressure System (LPS) CLO, and Breathing System (BS) CLO. Two of the leaf activities in the Root CLO are bound to the other two CLOs, which cover the pre-use checks of the Low Pressure System and the Breathing System, respectively. The activity tree of Figure 3-3 serves to illustrate that one can aggregate ALOs to form a CLO and the CLO can in turn link to other CLOs to form a more complex structure. ALOs and CLOs are individually packaged objects. They are reusable for any instructional purposes.

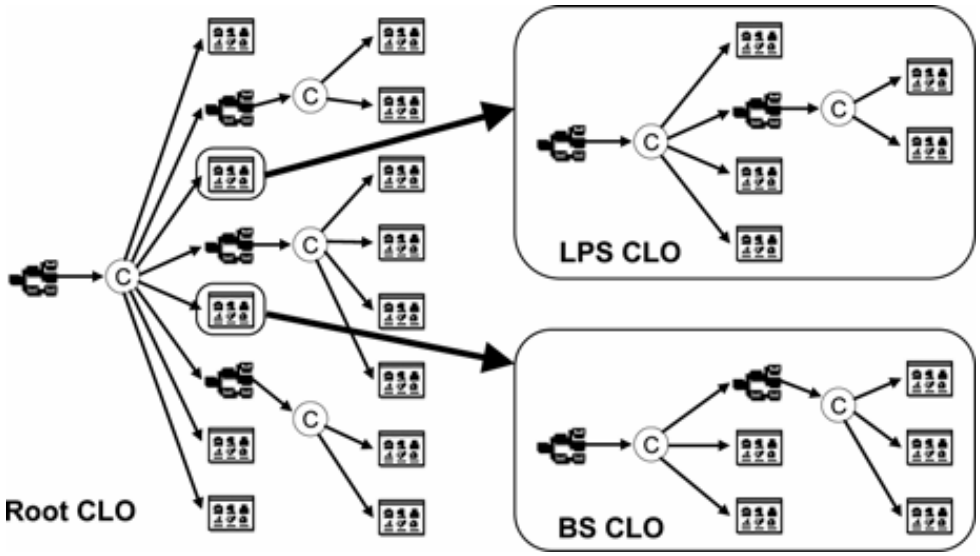

Figure 3-3. Activity tree that contains leaf-nodes bound to CLOs

Two additional modeling constructs are also introduced in an activity tree: connector and edge. A connector, denoted by © in Figure 3-2, connects a parent activity to its child activities. An edge is a link that connects an activity to its connector. The reason that we place a connector in between a parent activity and its child activities is to use it for the specification of a sequencing control mode that determines the sequencing order of the child activities. Any change made to the sequencing control mode as a result of an adaptation action taken by the system will not entail changes to the specification of the parent activity, which is traditionally used to house the specification of a sequencing control mode. The sequencing control modes supported by our system are the ones defined in the Sequencing Definition Model (ADL, 2004): Choice, Flow and Forward-Only. Choice allows a learner to have full freedom to select any of the child activities for processing in any order. Flow allows a learner to select either the next child activity or the previous child activity, while Forward-Only allows a learner to select only the next child activity in sequence. The learner would navigate the child activities according to the sequencing mode specified in the connector.

The activity tree of a CLO can be processed either in a pre-ordered traversal manner or according to the sequencing modes specified in or selected by GELS for the connectors. After creating a CLO, the content author can tag the CLO with metadata just like the metadata of an ALO for its discovery and retrieval. GELS' metadata will be covered in Section 3.2.

3.1.3. Learning Object Authoring Tools and Repositories

In order to ease a content author's tasks of constructing and storing learning objects, GELS reuses the ALO authoring tool, the CLO authoring tool, and the LO Repositories developed by Lee and Su (2006). They are collectively called Learning

Object Tools and Repositories (LOTRs), which can be installed at the network sites of users who serve as content producers and/or composers of a virtual e-learning community. The ALO authoring tool is designed for use by a content producer to define, construct, and register ALOs, and the CLO authoring tool is used by a content composer to aggregate some reusable ALOs and CLOs into a larger granule of instruction. Both authoring tools are also modified to provide a new metadata editor to ease content authors' tasks of specifying the attributes and value constraints that constitute the metadata of constructed LOs. The constructed ALOs and CLOs are stored in their respective LO repositories installed at the network sites where they are defined. Thus, they can be conveniently maintained and updated by their producers/composers. The details on the implementation of LO authoring tools and repositories will be given in Chapter 6.

3.2. Adaptive Content Search and Selection

The deployment of Learning Object Authoring Tools and Repositories at many network sites would promote the creation of many reusable objects for different subjects of learning. Some of these learning objects may cover the same subject but use different content items and different forms of presentation. The ability to search and select a suitable learning object for a learner from a potentially large number of registered learning objects is an important adaptive feature of GELS. GELS achieves adaptive content search and selection in the following way.

3.2.1 Adaptive Content Search

For adaptive search, GELS tags constructed learning objects with metadata and registers them with an LO Broker installed at the host site of a virtual e-learning community. The LO Broker is a metadata repository which provides facilities for

searching and retrieving learning objects. Learners and the system can thus query the Broker for learning objects that meet their instructional purposes.

We adopt selected categories of metadata proposed in LOM (2002) and extend them to suit GELS' specific need because we believe that some categories, such as ‘general’, ‘educational’, ‘relation’, and ‘technical’, are more appropriate for our system than categories such as ‘life cycle’ and ‘meta-metadata’. For example, ‘life cycle’ is a category which “describes the history and current state of a learning object and those entities that have affected the learning object during its evolution” (LOM, 2002, p.16). Since maintaining and updating a learning object are the responsibility of its content author, we believe that the life cycle of an object is not a necessary part of the metadata maintained by our system. Thus, in GELS, the metadata associated with learning objects are categorized into ‘general’, ‘educational’ and ‘access point information’, while the LOM uses nine categories of attributes to describe learning objects. The adoption of LOM and the extensions unique to GELS are shown in Figure 3-4.

The general category is mostly adopted from LOM and describes a learning object’s generic attributes. These attributes are ‘*title*’, ‘*description*’, ‘*keyword*’, ‘*cost*’, ‘*subject*’ and ‘*author*’, of which the latter two are added.

- Title: name given to this learning object
- Description: a textual description of the learning object
- Keyword: a keyword or phrase describing this learning object
- Cost: the cost of using this learning object, if payment is required
- Subject: the topic of this learning object
- Author: the person who created this learning object

The educational category defines the key educational or pedagogical information of a learning object. Among the attributes in GELS’ educational category, ‘*media format*’ and ‘*platform requirements*’ are adopted from LOM's technical category, and

‘*interactivity type*’, ‘*learning resource type*’, ‘*typical age range*’, ‘*difficulty*’, ‘*typical learning time*’ and ‘*language*’ are adopted from LOM's educational category. ‘*Course relations*’ is adopted from LOM’s relation category. Lastly, ‘*LO type*’, ‘*learning objective*’, ‘*number of attempts*’ and ‘*typical audience*’ are added to GELS' attribute set as an extension of LOM. The attributes of the educational category are explained below:

- Media format: technical data type of this learning object. e.g., video, text, etc
- Platform requirements: information about software and hardware requirements. e.g., sound card, adobe, etc.
- Interactivity type: predominant mode of learning supported by this learning object. e.g., active, expositive (passive), or mixed learning
- Learning resource type: specific kind of learning object. e.g., exercise, simulation, questionnaire, figure, exam, etc
- Typical age range: age of the typical intended user
- Difficulty: degree of challenge for the typical intended target audience. e.g., very easy, easy, medium, difficult, and very difficult.
- Typical learning time: approximate or typical time taken to work through the learning object for the typical intended target audience
- Language: language used by the learning object
- Course relations: relationship among this learning object and other targeted resources, if any. Multiple relationships are supported. e.g., ‘IsPartOf’, ‘HasPart’, ‘IsVersionOf’, ‘HasVersion’, ‘IsFormatOf’, ‘HasFormat’, ‘References’, ‘IsReferencedBy’, ‘IsBasedOn’, ‘IsBasisFor’, ‘Requires’, and ‘IsRequriedBy’.
- LO type: type of learning object, either ALO or CLO
- Learning objective: the purpose of the learning object
- Number of attempts: typical number of attempts to achieve the learning objective
- Typical audience: typical intended user

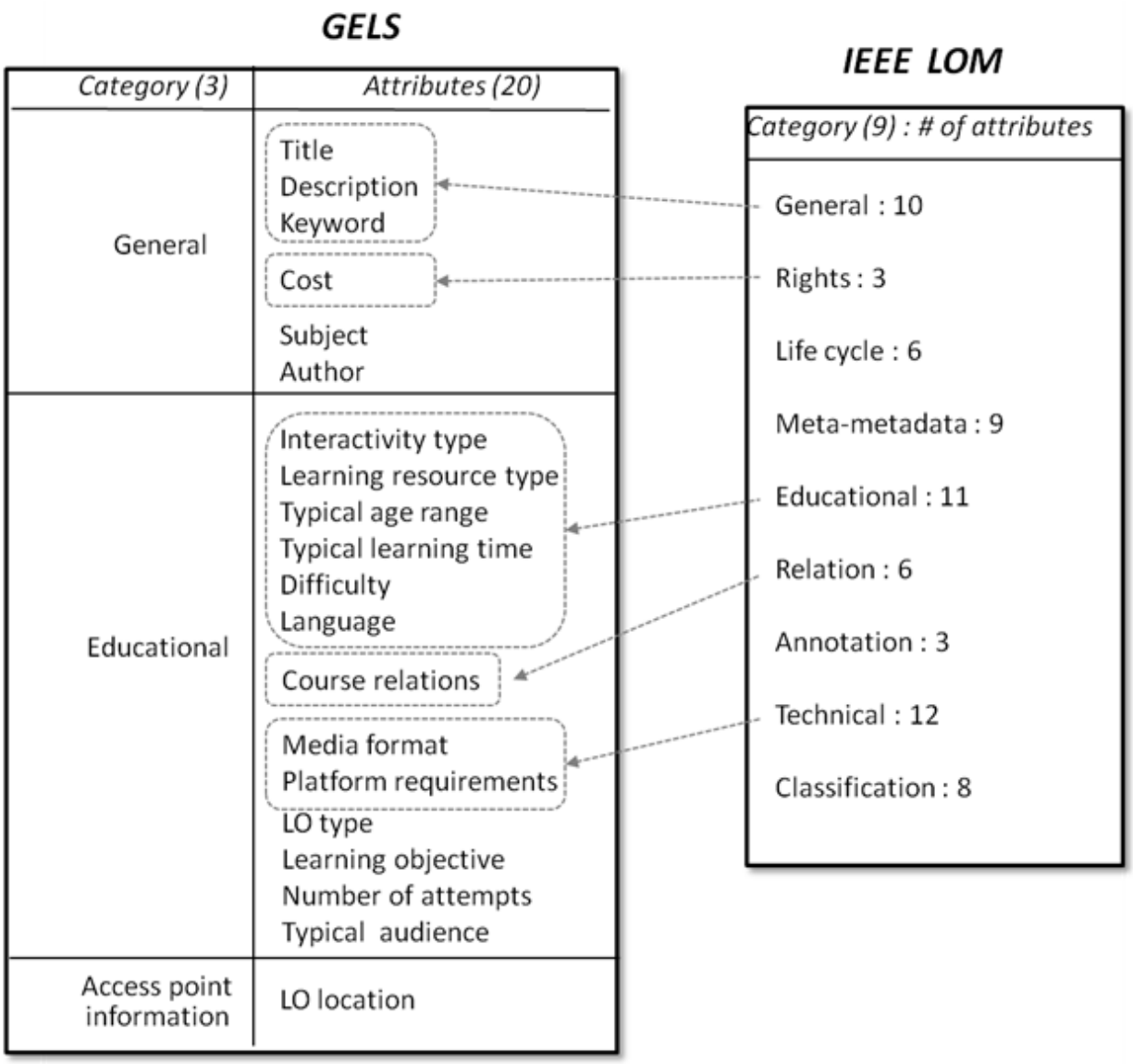

Figure 3-4. GELS' adoption and extension of IEEE LOM's categories of attributes

Although LOM has an attribute called 'location' in the technical category, this attribute only provides a general description of where one may find this resource. GELS adds 'access point information' as a separate category for specifying the access point of a distributed learning object stored in an LO repository. This information is used for adaptive content search and selection. GELS tags constructed objects with metadata for automated retrieval, dynamic binding and adaptive selection as well as for content

sharing and exchange with other systems. For the above stated purposes, the metadata of a learning object are specified using the extended Web Service Definition Language (WSDL) introduced in Lee, Zhang, and Su (2004), and registered with the LO Broker as Web services. The extension we made to WSDL is the inclusion of value constraints associated with metadata attributes. These value constraints are used by the Broker to match with the data conditions given in a search query issued by a learner or the system so that a learning object that best matches with the query can be selected. As an example, a part of the extended WSDL document, which describes the names, data types, and value constraints of the metadata attributes, used to define the learning object, 'Breathing System', is shown in Figure 3-5.

GELS provides content producers with a user-friendly authoring tool for constructing ALOs. The tool provides a user interface for a content author to specify and edit the metadata of an ALO shown in Figure 3-6. The system also provides content composers with an authoring tool for composing CLOs. The tool has a user interface for a content composer to specify and edit the metadata of a CLO as shown in Figure 3-7. The specified attributes and value constraints in Figure 3-7 are then used to generate the extended WSDL file shown in Figure 3-5. GELS also provides a user interface called 'Search LOs', which allows a user to select some metadata attributes from a drop down menu and specify their values to form a search query as shown in Figure 3-8. The data conditions of the query are used by the LO Broker to match against the registered metadata and constraints in a constraint satisfaction processing to select the best matched LO for the user or the system.

```
- <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>Title</name>
  <type>string</type>
  <keyword>eq</keyword>
  <valueList>Ch2 Breathing System</valueList>
  <negotiable>Yes</negotiable>
  <priority>0</priority>
  </service_constraint>
- <service_constraint>
  <name>Subject</name>
  <type>string</type>
  <keyword>eq</keyword>
  <valueList>FDA Checkout Breathing System</valueList>
  <negotiable>Yes</negotiable>
  <priority>0</priority>
  </service_constraint>
- <service_constraint>
  <name>Author</name>
  <type>string</type>
  <keyword>eq</keyword>
  <valueList>Sanghyun Jeon</valueList>
  <negotiable>Yes</negotiable>
  <priority>0</priority>
  </service_constraint>
- <service_constraint>
  <name>Keywords</name>
  <type>string</type>
  <keyword>enu</keyword>
  <valueList>{FDA Checkout, Breathing System, 1993}</valueList>
  <negotiable>Yes</negotiable>
  <priority>0</priority>
  </service_constraint>
- <service_constraint>
  <name>Language</name>
  <type>string</type>
  <keyword>eq</keyword>
  <valueList>English</valueList>
  <negotiable>Yes</negotiable>
  <priority>0</priority>
  </service_constraint>
- <ns1:service_constraint>
  <ns1:name>LO_Type</ns1:name>
  <ns1:type>string</ns1:type>
  <ns1:keyword>eq</ns1:keyword>
  <ns1:valueList>CLO</ns1:valueList>
  <ns1:negotiable>No</ns1:negotiable>
  <ns1:priority>0</ns1:priority>
  </ns1:service_constraint>
- <ns1:service_constraint>
  <ns1:name>Course_relations</ns1:name>
  <ns1:type>string</ns1:type>
  <ns1:keyword>eq</ns1:keyword>
  <ns1:valueList>IsPartOf Ch2CLO</ns1:valueList>
  <ns1:negotiable>No</ns1:negotiable>
  <ns1:priority>0</ns1:priority>
  </ns1:service_constraint>
  </service_constraints>
  <interattribute_constraints />
  </constraints>
  </wsdl:definitions>
```


Figure 3-5. Metadata attributes in extended WSDL document

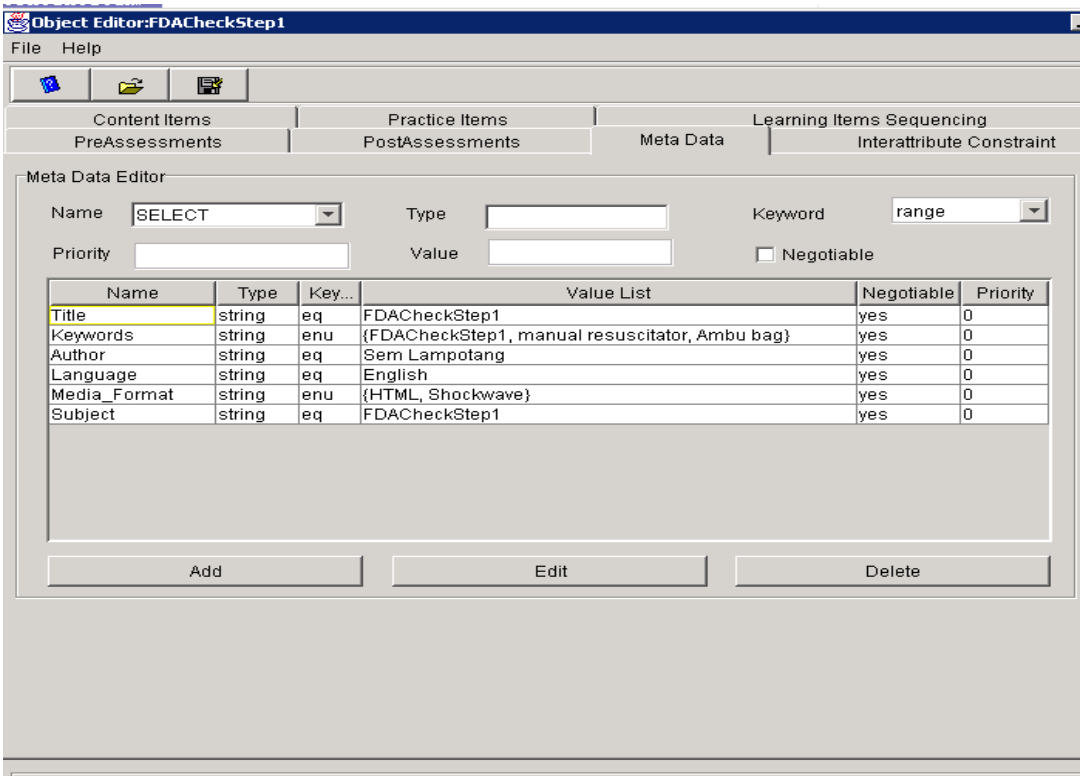

Figure 3-6. ALO metadata editor for adaptive search of ALO

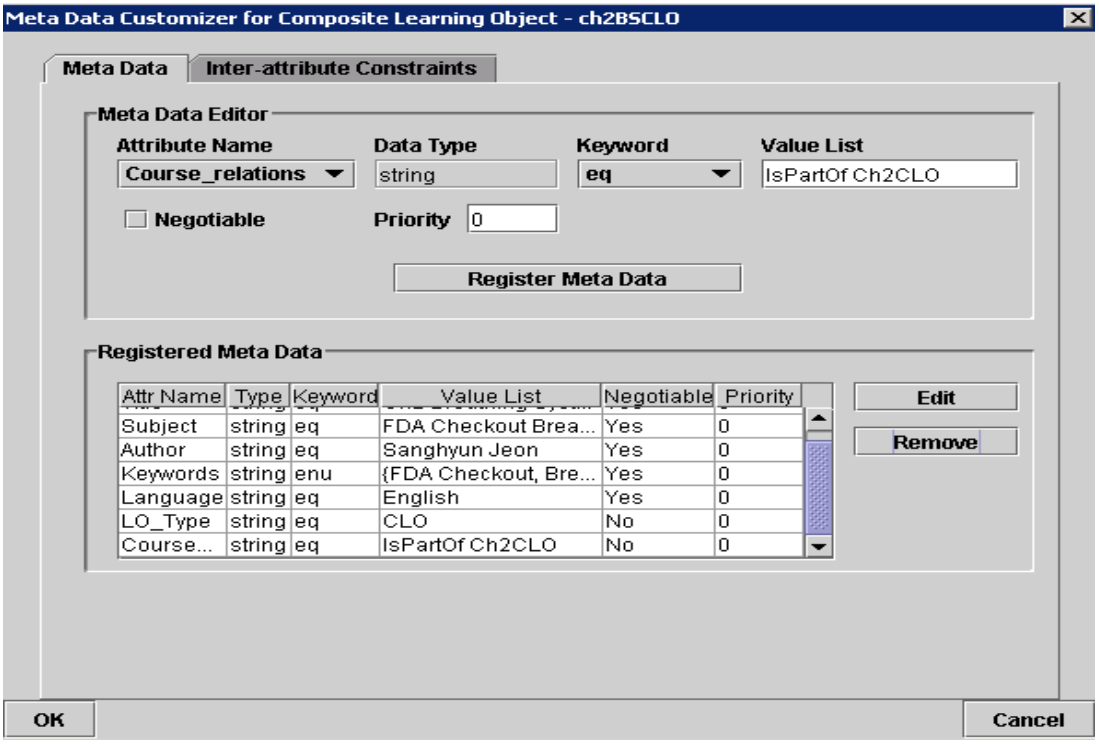

Figure 3-7. CLO metadata editor for adaptive search of CLO

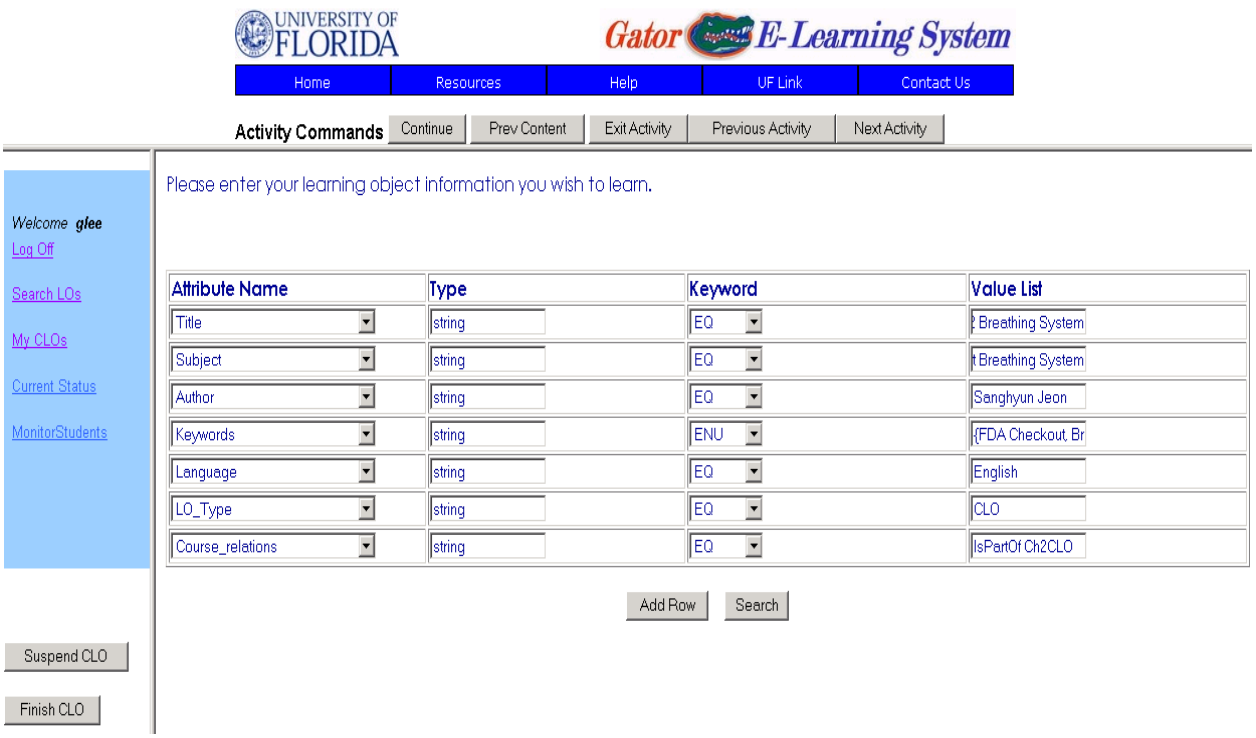

Figure 3-8. User interface for searching LOs

3.2.2. Adaptive Content Selection

GELS enhances a system's adaptive properties by providing an adaptive content selection facility. Content selection is achieved in two ways: static binding and dynamic binding. GELS allows a content composer to specify at design time whether a leaf and/or non-leaf activity of a CLO can be statically or dynamically bound to an ALO or CLO as illustrated in Figure 3-9.

For static binding, the composer names the specific ALO or CLO that should be activated for that activity. Then, the named ALO or CLO is processed at run-time. For dynamic binding, the composer instead specifies a query at design time for accessing a learning object that can satisfy the learning objective of a leaf-activity. The query specification is a logical expression of data conditions, whose attributes are those used to define learner profile and performance, and whose values are taken from the run-time

profile and performance data of the learner who is using the CLO. The query may contain a variable, denoted by 'VAR'. At runtime, the value of this variable is set to the value taken from the learner's profile or performance data at the time of dynamic binding. Thus, the query is dynamically generated.

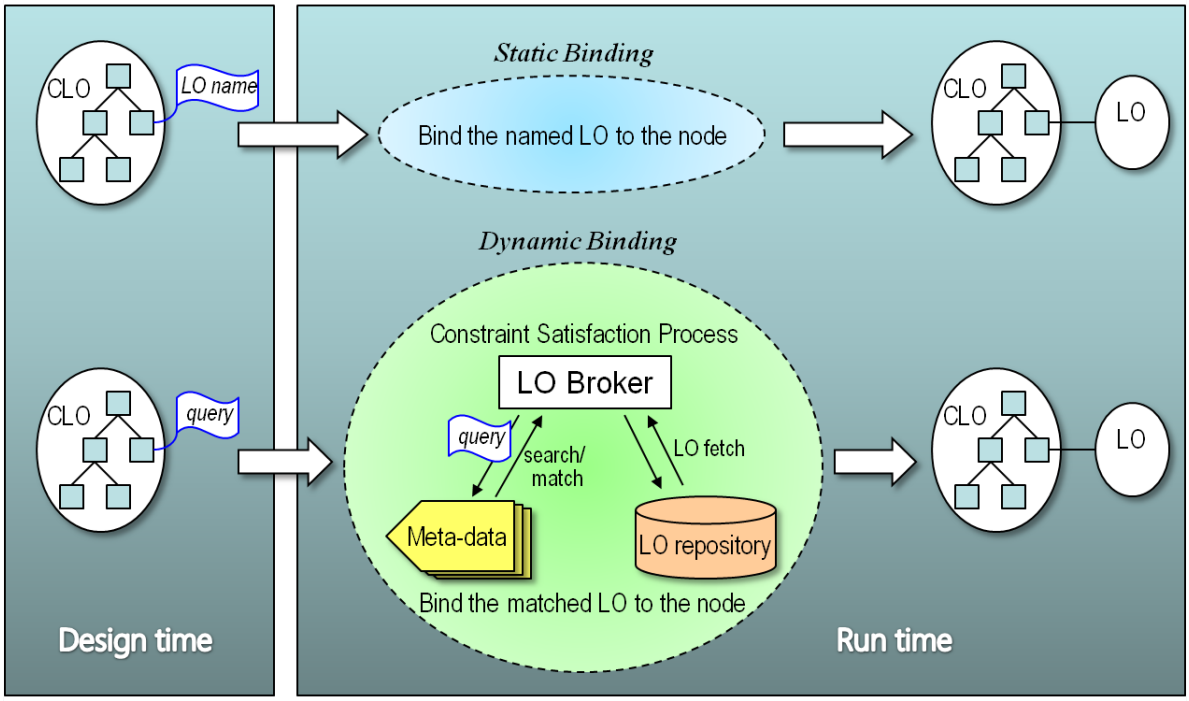

Figure 3-9. Adaptive content selection: static and dynamic bindings

Figure 3-10 shows an example of a query with dynamic binding information, which is specified by the attribute *keyword* having "input output of anesthesia machine" as its value and the attribute *language* having 'VAR:sLanguage' as its value. At run-time, the value of the attribute *language* is set to the value of the learner's preferred language, which is recorded in the learner's profile. Then, the query is sent to the LO Broker for processing. The Broker uses a constraint satisfaction processing technique (Degwekar, Su, & Lam, 2004) to match the query against the registered metadata of learning objects in order to find the one with the best match. If multiple learning objects with the same matching score are found, the matched learning objects and their metadata are

shown to the learner so that he/she can choose one. The chosen learning object is then fetched from a remote LO repository by using its registered access information. Dynamic binding using the constraint satisfaction processing technique allows the system to choose the most suitable learning material for a learner at run-time.

```
.......
<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>.......
```

Figure 3-10. A query example

In summary, GELS reuses previously developed tools and repositories to construct learning objects and provides adaptive features for content search and selection with extended metadata of LOM. The deployment of Learning Object Tools and Repositories (LOTR) at many Web sites will promote the construction of numerous reusable and competitive learning objects to enrich a virtual e-learning community. The use of metadata, the LO Broker, the static binding, and the dynamic binding using the constraint satisfaction processing technique will facilitate the adaptive search and

selection of learning objects to meet the diverse needs of virtual e-learning community members.

CHAPTER 4
PROBABILISTIC RULE MODEL

A popular way of guiding an e-learning system to provide individualized instructions to learners is to use condition-action (CA) rules (De Bra et al., 2003; Duitama et al., 2005). The condition part of this type of rule is a Boolean expression. If the expression is evaluated to be true, the operation(s) specified in the action clause is executed by the system. The way that a condition clause is specified and evaluated has a significant impact on a system's behavior, because the result of the evaluation determines the adaptation actions taken by the system. In light of this, there are several problems associated with the condition specification of existing adaptation rules. In this chapter, we address these problems and show how to resolve them by using an Event-Condition_probability-Action-Alternative_action (ECpAA) rule specification, Bayesian Models, and group data.

4.1 Problems with Existing Adaptation Rules

4.1.1 Existing Rules

A condition-action rule consists of a condition specification and an action specification, and has the format "if [condition] then [action]". When a CA rule is invoked, the condition specification is evaluated against a learner's profile and performance data as well as the metadata of the learning object that the learner is using. The adaptive action(s) specified in the action specification is taken based on the result of the evaluation.

Another popular type of rule in use is called an Event-Condition-Action (ECA) rule. The difference between a CA rule and an ECA rule is that the latter adds an event specification to the rule. In an ECA rule, the occurrence of the specified event will trigger

the processing of the rule condition. The format of ECA rule is “on [event] if [condition] then [action]”. A simple example of an ECA rule is “On the completion of a pre-assessment, if a learner did not take the prerequisite course and his/her assessment result is below a specified score, then the learner is asked to study the content again”.

4.1.2 Problems

As pointed out in Jeon, Su, and Lee (2007b), there are several problems with using CA rules or ECA rules to achieve a system's adaptive capabilities. First, if the data conditions in a condition specification are evaluated against a learner's data that are uncertain or incorrect, the condition specification cannot be evaluated correctly. This will prevent a system from taking the proper adaptation actions. Second, a traditional rule-based system evaluates each data condition independently. The correlations among data conditions are not taken into consideration. Third, most existing systems evaluate the condition specification without making use of the data of previous learners who have used the same learning object. A system's knowledge about previous learners can be very useful in the evaluation of an adaptation rule for a new learner, particularly when the data about this new learner are uncertain. In the following three sub-sections, we address these problems in detail.

4.1.2.1 Data uncertainty problem

The condition specification of a rule can potentially consist of many data conditions, which need to be evaluated against a learner's profile and performance data. The profile data are typically provided by a learner at the time of his/her registration with the system, and the performance data are collected by the system based on the learner's interactions with the system and the results of assessments. These data can be incomplete, inaccurate, and/or contradictory. We collectively call these data

problems the data uncertainty problem. For example, a learner may not be able to tell the system what his/her preferred learning style is. Or, a learner may not be willing to provide a piece of personal information (e.g., disability) because of privacy concerns. Even if he/she provides the system with a piece of information, that information may no longer be accurate as time passes (e.g., a learner's preferred learning style may change with time and with the subject he/she takes.) Also, some profile data may contradict with performance data (e.g., a learner may specify that he/she has certain prior knowledge of a subject which contradicts with his/her actual performance).

These data anomalies can create the following problems for a CA- or ECA-rule-based system in its evaluation of the condition clause of a rule. First, if a value of a data condition is missing, the system either assigns a default value or treats the data condition as false. If a default value is assigned, the same default value is assigned to all learners who fail to provide values for the data condition. This method goes against our overall aim of adaptation. If the missing value causes the data condition to be treated as false, it can negatively impact the evaluation of the entire condition specification. This is especially true if a condition clause is a complex Boolean expression that has many logically ANDed data conditions. A single data condition that is evaluated as false due to a missing value can cause the entire expression to be evaluated as false even if all the other data conditions are true. Second, a system may take wrong actions if the condition clause is evaluated based on inaccurate data provided by a learner. Third, contradictory data conditions may cause contradictory actions to be taken because multiple rules that make references to these data conditions can be activated.

4.1.2.2 Data correlation problem

The condition clause of a CA or ECA rule can be a complex expression that contains many data conditions. When such a clause is evaluated, the system evaluates each data condition independently from the others; the correlation of these data conditions is not considered. In many situations, the truth value of one data condition may affect the truth value of some other data condition(s). For example, the truth value of a pre-assessment result having a high value may depend on the truth value of the average grade of prerequisites having a high value. It is also possible that the truth value of one data condition has more influence to the truth value of the entire condition clause than that of another data condition. We believe that an adaptive system should take into consideration the interdependency of data conditions and their relative importance in order to evaluate a complex condition specification correctly.

4.1.2.3 Problem of not utilizing group profile and performance data

Most adaptive e-learning systems such as ELM-ART (Weber & Brusilovsky, 2001), MLTutor (Smith & Blandford, 2002), and SQLT-Web (Mitrovic, 2003) make use of an individual learner's profile and performance data to determine what and how content should be presented to a learner adaptively (Brusilovsky, Karagiannidis, & Sampson, 2004). We believe that the profile and performance data of those learners who have used a learning object can also be an important source of information for tailoring the instruction of the same learning object to a new learner. An adaptive system should utilize its knowledge of other learners' profile and performance data, "learn" from the accumulated data of learners (i.e., group profile and performance data), and continue to improve its adaptive capabilities.

In the ECpAA rule-based system that we propose, group profile and performance data are used to accurately evaluate the condition specifications of adaptation rules in much the same way as an instructor gains experience from teaching more and more students. Accurate evaluation is important because the result of evaluating a condition clause of an adaptation rule determines the action or alternative action to be taken and thus affects the way the content of a learning object is delivered to a learner, and the degree of freedom given to the learner in navigating the content.

4.2 Probabilistic Rule: Event-Condition_probability-Action-Alternative_action (ECpAA) Rule

The traditional CA or ECA rule-based system cannot evaluate complex and uncertain condition specifications properly as we addressed in the preceding sections. GELS uses an ECpAA rule specification, Bayesian Models, and group profile and performance data to resolve the addressed problems and make proper adaptation decisions. Our ECpAA rule specification allows an expert to probabilistically express the condition part of an adaptation rule. A Bayesian Model then probabilistically evaluates the data conditions given in the defined rule by using a prior distribution to determine if the action or the alternative action should be taken. The profile and performance data of each new learner is used to update the prior distribution so that the accuracy of learner evaluation will continue to improve. In the following sections, we explain our ECpAA rule specification along with Bayesian Models, and also present the technique used to process ECpAA rules with examples.

4.2.1 Event-Condition_probability-Action-Alternative_Action (ECpAA) Rule Specification

In order to resolve the problems associated with the condition clause in the traditional CA or ECA rule-based systems, we introduce a new way of specifying and

processing adaptation rules. The new rule specification is called ECpAA specification, which consists of an event, a probabilistic specification of a condition, and an action/alternative action specification. The ECpAA rule has the format "**on** [Event], **if** [Condition_probability] **then** [Action] **else** [Alternative_action]". We explain its components below.

4.2.1.1 Events: adaptation points

Events are defined and posted at various points in time during the processing of a learning object to trigger the evaluation of rule conditions in order to determine the proper actions that should be taken by the adaptive system. SCORM defines two events: after activity and before activity. At these two points in time of processing each activity of an activity tree, rules can be applied to determine the system's actions. We call these points "adaptation points". It is our opinion that more adaptation points should be recognized so that the system can apply adaptation rules more frequently to tailor an instruction to a learner. For example, adaptation decisions can and should be made at 1) the time to bind a learning object to the activity before an activity is processed, 2) the time after a pre-assessment has been performed, 3) the time before going down the activity tree from a parent activity to a child activity, 4) the time when returning to the parent activity after a child activity has been processed, 5) the time after a post-assessment has been carried out, and 6) the time when a learner is about to exit the activity.

In GELS, events can be defined and posted at the above six adaptation points to invoke adaptation rules. These events are named 'Before-Activity', 'After-Pre-Assessment', 'Drill-Down', 'After-Post-Assessment', Roll-Up', and 'Before-End-Activity'.

They are posted automatically by GELS to trigger the processing of their corresponding adaptation rules at these points.

4.2.1.2 Condition_probability specification

The ECpAA rule specifies the condition clause of a rule not deterministically but probabilistically. Unlike the traditional CA or ECA rule, in which the condition clause is specified to determine a true or false value, the condition clause of an ECpAA rule is specified to determine the probability of being true or false. Therefore, in the ECpAA rule specification, the condition part is specified probabilistically in the form of p(condition specification) ≥ x, where x is a threshold value between 0 and 1. If the probability value derived from the evaluation of the condition specification is greater than or equal to the threshold x, the action part of a rule is processed. Otherwise, the alternative action is taken. The condition specification contains data conditions that make references to attributes selected from those that define the metadata of learning objects and the profile and performance data of learners. The evaluation of each data condition is based on not only the data of a learner, but also the accumulated data of previous learners of the same learning object. These data are used to derive the probability of a data condition being true. The evaluation of the entire condition specification results in an aggregated probability value between 0 and 1.

4.2.1.3 Action / Alternative_action specification

The action or alternative action specification defines the actions taken by the system, which include 1) select a suitable learning object for a learner, 2) present its content, practice, and assessment items in a way or format suitable to that particular learner, 3) determine how the child-activities of the current activity should be sequenced, 4) grant the learner the proper degree of freedom to navigate the contents

of the sub-tree rooted at the current activity, and so on. In processing the action or alternative action clause, GELS employs several existing techniques such as direct guidance, link hiding, and conditional text inclusion which have been described in Section 2.2.

In GELS, ECpAA rules are specified by a content author or instructor when a learning object is designed. Six ECpAA rules, if specified, can be activated at the six adaptation points of processing a non-leaf activity: ‘Before-Activity’ rule, ‘After-Pre-Assessment rule, ‘Drill-Down’ rule, ‘Roll-Up’ rule, ‘After-Post-Assessment’ rule, and ‘Before-End-Activity’ rule. For a leaf activity, there are four rules for controlling its processing: ‘Before-Activity’ rule, ‘After-Pre-Assessment’ rule, ‘After-Post-Assessment’ rule, and ‘Before-End-Activity’ rule. Each adaptation rule is invoked by its corresponding event that has the same name. Rules are applied to both leaf activities and non-leaf activities of an activity tree. Each activity may invoke multiple rules. The detailed about our rule processing technique and examples of ECpAA rules used in GELS will be given in Section 4.2.3 and Section 4.2.4, respectively.

4.2.2 Probabilistic Evaluation of Rule Conditions: Bayesian Model

As we mentioned before, the profile and performance data of a learner may have the data uncertainty problem. Dealing with the uncertainty problem is not new in the area of Artificial Intelligence (AI). Numerous paradigms have been proposed, such as certainty factor in knowledge representation, probability theory, fuzzy logic, and Bayesian Network. Among them, Bayesian Network dominates AI research on uncertainty reasoning because of its efficient representation of and rigorous reasoning with uncertain knowledge in a probabilistic way (Russell & Norvig, 2003).

4.2.2.1 Bayesian Model

A Bayesian Model is a graphical model which specifies the probabilistic relationships among a set of variables (Heckerman, 1995). Each model has a network structure with a unique Directed Acyclic Graph (DAG) and a set of conditional probability distributions. The DAG is used to encode a joint probability distribution which represents the causal correlations among a set of data conditions. A DAG consists of nodes and edges (Russell & Norvig, 2003); a node represents a data condition, such as a learner's current assessment result, a learning goal, or a learning style, while an edge represents a causal or informational correlation between two data conditions. Prior and conditional probability values are assigned to the nodes. The correlation among data conditions is captured by computing the conditional probability of a data variable having one of *m* possible values as the hypothesis H_i given the probabilities of *n* correlated data conditions as the evidences using the following Bayes' rule (Gonzalez & Dankel, 1993):

$$p(H_i \mid E_1E_2...E_n) = \frac{p(E_1E_2...E_n \mid H_i) * p(H_i)}{p(E_1E_2...E_n)} = \frac{p(E_1 \mid H_i) * p(E_2 \mid H_i) * ... p(E_n \mid H_i) * p(H_i)}{\sum_{k=1}^{m} p(E_1 \mid H_k) * p(E_2 \mid H_k) * ... * p(E_n \mid H_k) * p(H_k)}$$

4.2.2.2 The design of Bayesian Models

The design of a Bayesian Model consists of the following three steps: selecting the proper variables, building a network, and assigning probabilities. Typically, not all the data attributes used to define the metadata of learning objects and the profile and performance data of learners are suitable for specifying the condition clause of an adaptation rule. The designer of a Bayesian Model would select a subset of these attributes that are relevant for making the adaptation decision at a particular adaptation point. These attributes form a structure of data conditions, which defines how these data

conditions are correlated and the probabilities of which data conditions can be used to generate the conditional probability of other data conditions. We distinguish two phases in the development and usage of Bayesian Models: the design phase and the usage phase.

Design Phase. In the design phase, the designer of Bayesian Models would design the following six Bayesian Models (BMs) for the six corresponding adaptation rules triggered at the six adaptation points: 1) a *beforeActivityBM* for the 'Before-Activity' rule, which is used to select a suitable learning object, 2) an *afterPreAssessmentBM* for the 'After-Pre-Assessment' rule, which specifies how the next set of activity nodes should be sequenced based on the pre-assessment result, 3) a *drillDownBM* for the 'Drill-Down' rule, which determines if a learner is ready to learn its child activities, and prepares the child activities of a current activity for processing, 4) a *rollUpBM* for the 'Roll-Up' rule, which determines how to set the objective status of the parent activity based on the results of its child activities, 5) an *afterPostAssessmentBM* for the 'After-Post- Assessment' rule, which evaluates the result of a post-assessment and determines a learner's performance in an activity, and 6) a *beforeEndActivtyBM* for the 'Before-End-Activity' rule, which determines whether a learner is allowed to exit the activity or not.

Figure 4-1 shows our design of the *drillDownBM* for the 'Drill-Down' rule. It contains five root nodes, two of which are attributes selected from those that define the profile data of learners (learner's *Competency Level on a Subject* (*CLS*) and *Learning Goal* (*LG*)) and three from those that define their performance data (*Average Grade of Prerequisites* (*AGP*), *Grade Point Average* (*GPA*), and *Current Pre-Assessment score*

(*CPA*)). These attributes are deemed relevant for making a drill-down decision. Four additional non-root nodes are introduced to capture the correlations of root nodes and non-root nodes: *Learning Motive* (*LM*), *Prior-Knowledge Level* (*PKL*), *Learning Capability* (*LC*), and *Drill Down* (*DD*). The nodes are connected by directed edges and the weights (i.e., w) shown on the edges specify the relative strengths of influence that parent nodes have on child nodes. For example, Figure 4-1 shows that the probability value of AGP has more influence on the probability value of PKL than that of GPA (0.6 vs. 0.4).

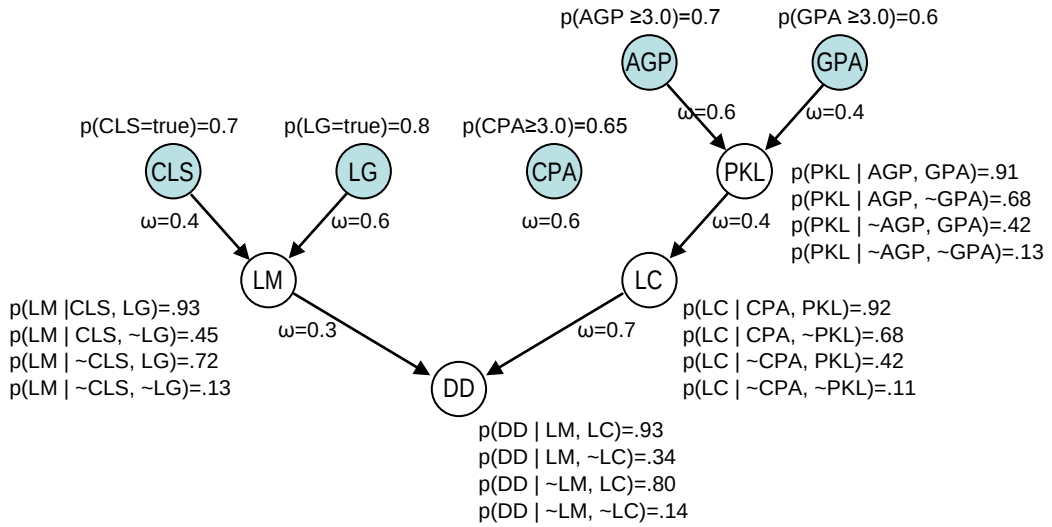

Figure 4-1. *drillDownBM* with conditional probability tables for drill-down decision

The Condition_probability specification of a 'Drill-Down' rule makes reference to the root nodes and specifies a threshold value (e.g., 0.6). Therefore, the Condition_probability specification is defined as follows: if [p(AGP, GPA, CPA, LG, CLS) ≥ 0.60]. When the 'Drill-Down' rule is triggered in a parent activity before going down the activity tree to process its child activities, the corresponding *drillDownBM* is used to

evaluate the Condition_probability specification of the ‘Drill-Down’ rule. The processing of the *drillDownBM* would involve the probabilistic evaluation of all the nodes down to the *DD* node. If the resulting probability value of the *DD* node is greater than or equal to the threshold, the system would goes down the activity tree to process its child activities. Otherwise, all child nodes are hidden from the learner, and the learner can be directed to process the next activity.

Usage Phase. In the usage phase, when a rule is triggered at an adaptation point of processing an activity for a learner, the corresponding Bayesian Model is used to evaluate the Condition_probability clause of the rule. Before its use, two steps are taken to tailor it to the learning object: assigning prior probabilities to the root nodes and computing the Conditional Probability Tables (CPTs) for the non-root nodes. Prior probabilities, which represent the prior knowledge about the states of the data conditions, are assigned to the root nodes by a domain expert (e.g., the content author or instructor), based on the profile and performance data of learners who have been instructed by the expert using the same or a similar learning object.

A root node’s probability table contains the unconditional probability of a data condition. For example, Figure 4-1 shows that the probability of a student having a *GPA* greater than or equal to 3.0 is 0.6 (i.e., p(*GPA*≥3.0). A non-root node’s CPT represents the correlation between the non-root node (i.e., the child node) and its parent nodes. Figure 4-1 also shows the CPTs derived for all value combinations of each non-root node’s parents. Each CPT is computed as follows: The expert provides the system with his/her best knowledge of the root nodes' prior probabilities and the weights for all edges. Then, the system automatically computes the conditional probabilities by using

our proposed formulas and the expert's inputs. These prior and conditional probabilities are, then, used to derive the CPT for each non-root node. For example, given the values of prior probabilities of *AGP* and *GPA* conditions, and the their weight information as shown in Figure 4-1, our proposed formulas derives the probability values of *PKL* and the conditional probabilities of *AGP* and *GPA* given *PKL* or ~*PKL* as the evidence in all four logical combinations and its complements as shown in Table 4-1. Note that each column sums up to 1. Using the above information and Bayes' rule, the CPT of *PKL* given the *AGP* and *GPA* conditions as evidences is derived as shown in Table 4-2. The other CPTs of Figure 4-1 are calculated using the same procedure. All the prior probabilities and CPTs are constantly updated as a new learner uses a learning object, and as each piece of his/her profile and performance data is provided or derived as new evidence. The details about our formulas and how CPTs are computed will be given in Chapter 5.

Table 4-1. Probability values for PKL

Probabilities of PKL	Conditional Probabilities of *AGP*		Conditional Probabilities of *GPA*	
p(*PKL*)=0.66	p(*AGP*|*PKL*)=0.88	p(*AGP*|~*PKL*)=0.35	p(*GPA*|*PKL*)=0.72	p(*GPA*|~*PKL*)=0.36
p(~*PKL*)=0.34	p(~*AGP*|*PKL*)=0.12	p(~*AGP*|~*PKL*)=0.65	p(~*GPA*|*PKL*)=0.28	p(~*GPA*|~*PKL*)=0.64

AGP(Average Grade of Prerequisites); GPA(Grade Point Average); PKL(Prior-Knowledge Level).

Table 4-2. CPT of node PKL

Conditional Probability	*p(PKL|AGP, GPA)*	*p(PKL|AGP,~GPA)*	*p(PKL|~AGP,GPA)*	*p(PKL|~AGP, ~GPA)*
Probability Value	0.91	0.68	0.42	0.13

4.2.3. ECpAA Rule Execution

The system components that support the ECpAA rule evaluation are shown in Figure 4-2. ECpAA rules are executed during the processing of learning activities. When a system component, called Learning Process Execution Engine (LPEE), reaches a particular stage of processing a learning activity (i.e., the occurrence of an event), its Activity Handler calls the ECpAA Rule Engine to execute the specified rule. The ECpAA Rule Engine has two subcomponents: an Event-Trigger-Rule (ETR) Server and a Bayesian Model Processor (BMP). The ETR Server processes a rule, while the BMP processes the Condition_probability specification of the rule. Reaching a specified adaptation point is treated as an event by the ETR Server, which fetches the ECpAA rule that is linked to the event in a trigger specification. The ETR Server then processes the fetched rule. When the Condition_probability specification of the rule (i.e., Cp) needs to be processed, the ETR Server calls the BMP to evaluate the condition specification. The BMP returns a probability value to the ETR Server. Then the ETR Server executes the action specification or the alternative action specification of the rule depending on whether the returned value satisfies the threshold condition.

GELS provides a graphical user interface for a content author or instructor to define ECpAA rules in an activity, after he/she created an activity tree that models a CLO. A system facility is provided for modeling the correlations among the data conditions specified in an adaptation rule to form a Bayesian Model. Another facility is provided for the communication between the BMP and the other components (the ETR Server and the Repository). The implementation details of the system components will be presented in Chapter 6.

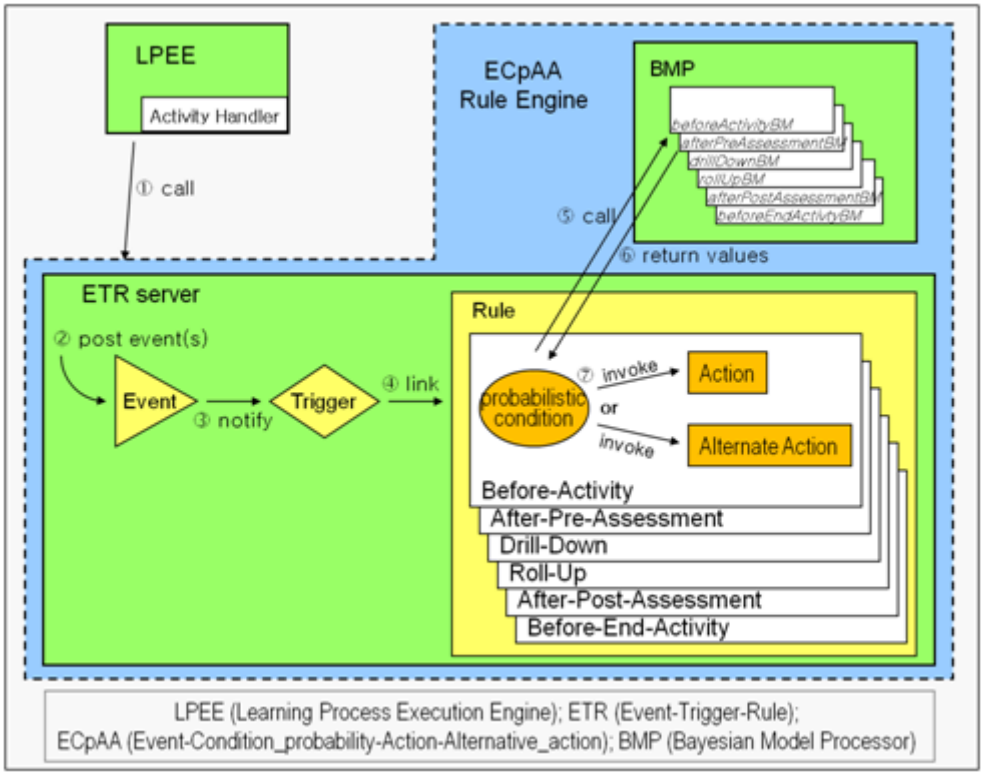

Figure 4-2. ECpAA rule execution

4.2.4 Six Types of ECpAA Rules in Gator E-Learning System

As we mentioned in Chapter 3, a CLO is defined as an activity tree, which models a learning process. To make the CLO adaptive, a domain expert has the option of specifying some adaptation rules to be activated before or after the tasks performed in each activity. Figure 4-3 shows the tasks performed in both non-leaf activity and leaf activity, which are shown in blue, and the places where adaptation rules can be introduced are shown in yellow. As shown in the figure, GELS gives the expert the option of specifying six adaptation rules for a non-leaf activity and four out of the six for a leaf-activity. These rules are explained below.

‘Before-Activity’ rule: invoked before processing an activity. The condition specification of the rule is evaluated against the learner’s profile and performance data.

It is also evaluated against the metadata of learning objects. The ‘Before-Activity’ rule is used to select a suitable learning object for the learner.

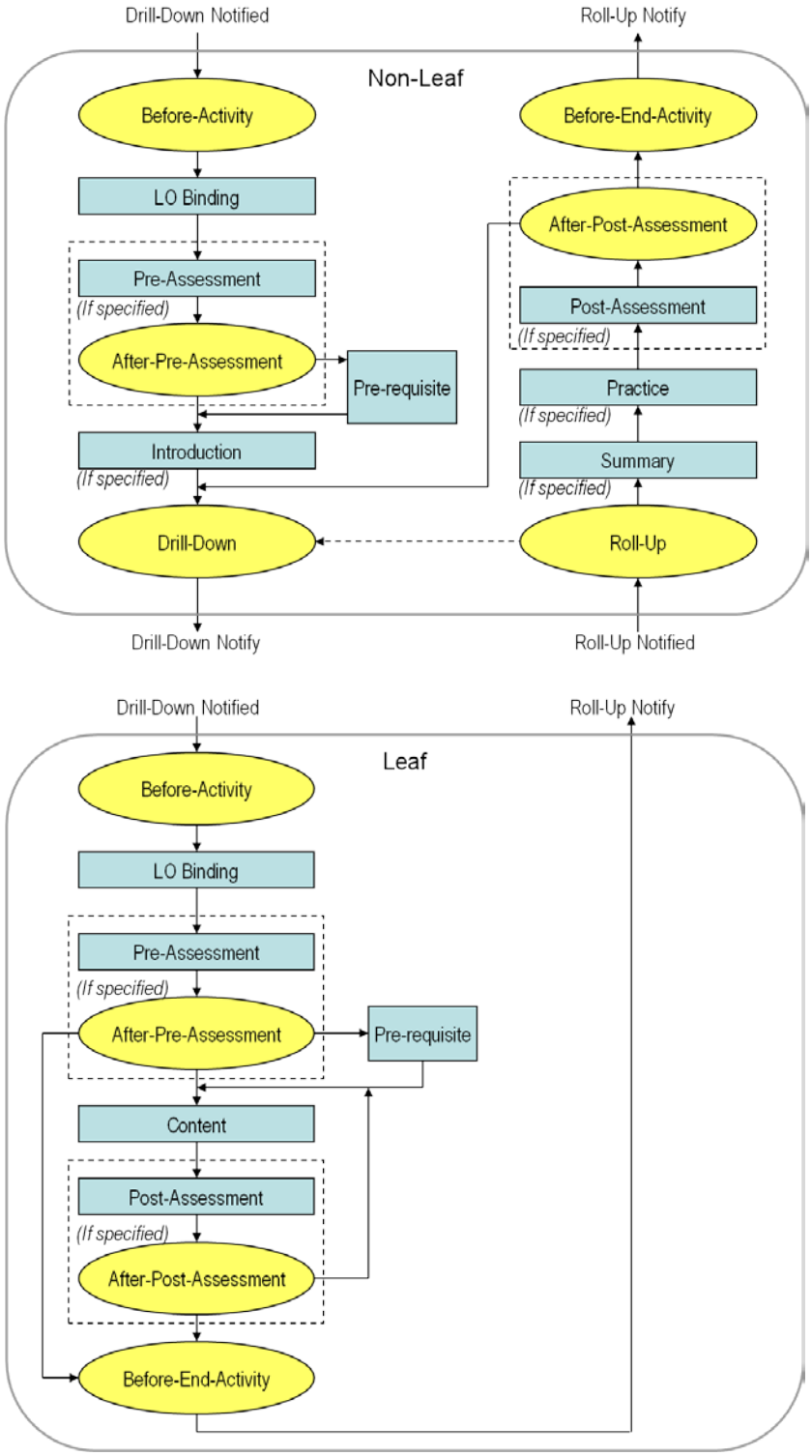

Figure 4-3. Adaptation process of ECpAA rules in an activity

‘After-Pre-Assessment’ rule: invoked after the pre-assessment (if pre-assessment items are given) of a learning activity is performed. The condition specification of the rule is used to check the assessment result and to determine what action should be taken by the system with respect to the activity. Depending on the learner’s profile and performance data, one of the following actions will be taken by the system: process the activity, skip the activity, or ask the learner to take a pre-requisite learning object before learning from the learning object selected for this activity.

‘Drill-Down’ rule: invoked before going down the activity tree from a parent activity to process its child activities. The condition specification is used to evaluate the learner’s readiness to learn from the child activities. Based on the result of evaluating the rule condition, the system will determine which child activity should be selected and how it should be presented to the learner. The processing order of the child activities will be set by selecting one of the Sequencing Control Modes (Choice, Flow, or Forward Only). A ‘Drill-Down’ rule allows the system to enable/disable an activity, hide an activity from choice, sort activities, and deliver activities in a proper order depending on the learner’s profile and performance data. ‘Drill-Down’ rules are applicable only to non-leaf activities.

‘Roll-Up’ rule: invoked after all the child activities of a parent activity have been processed and the system is about to return to the parent activity. The condition specification of a ‘Roll-Up’ rule determines if the learner has adequately learned from the learning objects invoked by the child activities. For example, if [the learner’s performance in child activities is satisfied], then [set ‘Satisfied’ status for the parent activity and skip its post-assessment], else [set ‘Not Satisfied’ status for the parent

activity and perform its post-assessment]. ‘Roll-Up’ rules are applicable only to non-leaf activities.

‘After-Post-Assessment’ rule: invoked after the post-assessment (if post-assessment items are given) of a learning activity has been performed. The condition specification of the rule is evaluated against the learner's assessment result to make an adaptation decision. For example, if the learner’s result is not satisfactory, then the learner is required to “retry this activity”, or “retry selective activities”, or “update the objective of the activity”.

‘Before-End-Activity’ rule: invoked after the current activity has been processed and before the learner is allowed to leave this activity (i.e., before the system terminates the current activity). The condition specification of this rule may check the learner’s performance and the restrictions associated with the activity such as the number of attempts allowed, the time that the learner is allowed on each attempt, and the time that the learner spent on the activity. The result of evaluating the condition specification is used by the system to decide if the learner should be asked, for example, to redo the activity or should be allowed to leave the current activity.

4.3 Resolving the Identified Problems

We believe that the ECpAA rule model can resolve the problems of data uncertainty, data correlation, and the problem of not utilizing group profile and performance data addressed in Section 4.1.2. We present our ways of resolving these problems in the following subsections. They have been published in Jeon et al. (2007b).

4.3.1 Handling of the Data Uncertainty Problem

The use of ECpAA rules and Bayesian Models for evaluating the Condition_probability clauses of these rules can solve the data uncertainty problem. In

the case of missing data, we use the conditional probability distribution of data associated with the data attribute that does not have a value. For example, Learner Y's *GPA*=3.5, but his/her *AGP* value is missing. Since p(PKL | AGP, GPA)=0.91 and p(PKL | ~AGP, GPA)=0.42 (see *PKL's* CPT in Figure 4-1), these values can be weighted by the prior probabilities of *AGP* and ~*AGP* respectively, then take the sum (Gonzalez & Dankel, 1993):

$$p(PKL|AGP=?,GPA) = p(PKL|AGP,GPA)*p(AGP)+p(PKL|\sim AGP,GPA)*p(\sim AGP)$$
$$= 0.91 * 0.7 + 0.42 * 0.3 = 0.763.$$

Although the *AGP* value is not known, as denoted by "?", the Bayesian Model Processor can still derive the conditional probability of *PKL*. Given the additional evidences, *CPA*=3.7, *CLS*=true, and *LG*=true, the system can successfully draw the final *DD* value of 0.83 and inform the ETR Server so that the ETR Server can take either the action or the alternative action for learner Y.

The contradictory data problem can be alleviated by using a Bayes' decision rule. The Bayes' decision rule allows the system to select the data condition with a higher conditional probability while minimizing the posterior error (Duda, Hart, & Stork, 2001). The contradictory data value is replaced by the data value with a higher conditional probability. Suppose learner Y has two contradictory learning styles. One is given by learner Y in his/her profile *(LS_LP)* having "Exploratory (Ex)" as its value, and the other is determined by the system based on his/her performance data recorded in the Learner Performance of the Learner Repository *(LS_SLP)* having a contradictory value "Receptive (Re)". The system needs to determine which learning style (*LS*) to follow in order to continue its processing of the learning activity. Given the following prior and conditional probabilities (Table 4-3), the conditional probability of *LS* having

"Exploratory" as its value and the conditional probability of LS having "Receptive" as its value can be derived below:

$$p(LS=Ex \mid LS\_LP=Ex, LS\_SLP=Re) = \frac{0.6*0.5*0.6}{0.6*0.5*0.6+0.45*0.4*0.4} = 0.71$$

$$p(LS=Re \mid LS\_LP=Ex, LS\_SLP=Re) = 1 - 0.71 = 0.29$$

Since p(LS=Ex | LS_LP=Ex, LS_SLP=Re) > p(LS=Re | LS_LP=Ex, LS_SLP=Re), the system would choose "Exploratory" as the value of Y's learning style. *LS_SLP*'s value is replaced by "Exploratory" for consistency.

The negative effect of an inaccurate data value can also be reduced because the Bayesian Model considers not only the inaccurate value associated with a data attribute but also the values of correlated attributes that are correct and accessible from the CPTs.

Table 4-3. Prior-probabilities and conditional probabilities of Learning Style

Prior-probability	Conditional Probabilities of LS_LP	Conditional Probabilities of LS_SLP
p(LS=Ex)=0.6	p(LS_LP=Ex | LS=Ex)=0.6 p(LS_LP=Re | LS=Ex)=0.4	p(LS_ SLP=Ex | LS=Ex)=0.5 p(LS_ SLP=Re | LS=Ex)=0.5
p(LS=Re)=0.4	p(LS_LP=Ex | LS=Re)=0.45 p(LS_LP=Re | LS=Re)=0.55	p(LS_ SLP=Ex | LS=Re)=0.6 p(LS_ SLP=Re | LS=Re)=0.4

4.3.2 Handling of the Data Correlation Problem

Our approach of evaluating the condition specifications of rules probabilistically by using Bayesian Models can also account for the interdependencies among attributes used in the condition specifications. The structure of a Bayesian Model (DAG) specifies the correlations among the attributes, and the probability distributions derived for the root nodes and non-root nodes represent their interdependencies probabilistically by

showing all possible value combinations of parent nodes used to derive the value of a child node.

4.3.3 Handling of the Problem of not Utilizing Group Profile and Performance Data

GELS uses group profile and performance data to compute the prior probabilities and the conditional probabilities among data conditions of each Bayesian Model. Group profile and performance data contains the profile and performance data of each new learner as well as the data of those learners who have learned from the same learning object. Group profile and performance data are continuously updated as more and more individual learners' profiles and performance data are gathered by the system. With this updating feature, GELS can improve the accuracy of evaluating rule conditions in the following way. Each learner's individual data affects the assignment of prior probabilities to a Bayesian Model's root nodes and the computation of the CPTs of the non-root nodes. These prior probabilities and CPTs produce the posterior probability, which in turn affects the prior distribution of the model. The new prior distribution is then used for Bayesian inference when an adaptation rule is activated for the next learner (Jaynes, 1989). In this way, GELS is able to learn from its experiences with all the previous learners of a learning object and improve its ability to accurately evaluate the next rule condition so that the proper adaptation action can be taken for the next learner. This is like a skilled tutor/instructor who adjusts the way he/she instructs a new student based on the accumulated experience of instructing previous students. GELS is designed and implemented to emulate an experienced tutor/instructor by using continuously updated group profile and performance data.

Our definition of "group" and the usage of the "group data" are different from those used in group-based adaptive systems. "Group" in other adaptive systems is defined as

a group of learners who share some common characteristics and behaviors, and the group model is used to detect a learner who is different from others using “a learner matching technique” (e.g., an outstanding peer who has more information than others will be selected to help others; or a learner who has trouble learning will get some additional help) (Brusilovsky & Peylo, 2003). Our ‘group’ is defined as the group of learners who have learned from the same learning object, and our group data are used for improving the system’s adaptation capability.

4.4 Chapter Summary

In summary, our ECpAA rule-based processing model has several advantages. First, probabilistic rules are high-level specifications. They are easier to understand and modify than the program code that implement them. The defined rules are automatically translated into program code, thus saving program development time and cost. It is commonly known that if rules are “hard-coded” in a computer program, the program code will be very difficult to understand and modify because the person who wants to modify the rules will have to know the programming language that implements these rules, understand many lines of program code, and make sure that the modification of the code will not have any side-effect that damages other code. Second, at run time ECpAA rules can alter the system’s sequencing behaviors and dynamically control a learner’s navigation of contents based on the most up-to-date profile and performance data of the learner as well as the metadata of a learning object. Third, compared with SCORM’s rule-based Sequencing Definition Model (ADL, 2004), GELS applies six ECpAA rules at six adaptation points, while SCORM applies two rules (a sequencing rule applied before starting an activity and before finishing an activity, and a rollup rule) at two adaptation points (before activity and after activity). Our system collects learner

data relevant to each adaptation point and evaluates the current needs of the learner. Thus, adaptation actions can be taken by the system more frequently and more accurately. The six adaptation rules can be used as guidelines by a content producer or composer to design and develop more complex adaptive courses. Fourth, the condition specification of an ECpAA rule can contain a variety of data conditions for examining a learner's profile and performance data, and/or the metadata of the learning object being used, whereas SCORM’s sequencing rules can inspect only three aspects of learner performance: score, completion, and objective satisfaction. Fifth, in contrast to the traditional ECA rule, the ECpAA rule can make proper adaptation even if some learners' profile and performance data are inaccurate, missing, and/or contradictory. Finally, an individual learner’s information is updated continuously. The system uses both the individual and group profile and performance data to improve the accuracy of learner evaluation. All of the above features enable GELS to tune the learning activities of each learner during a learning process, thus providing him/her with individualized instruction.

CHAPTER 5
DERIVING PRIOR DISTRIBUTIONS FOR BAYESIAN MODELS

As presented in the preceding section, we use a Bayesian Model to capture the correlations among data conditions specified in an adaptation rule and use it to probabilistically evaluate the rule's condition specification. Using a Bayesian Model requires setting up a prior distribution for it (Kass & Wasserman, 1996), because Bayesian inference cannot be done without the prior distribution. It has been recognized that obtaining an informative prior distribution is the most challenging task in building a probabilistic network (Druzdzel & Gaag, 1995). In this chapter, we present our way of acquiring the prior distribution. Our prior distribution is automatically derived by using the formulas presented in this chapter together with prior probabilities and weights assigned by the content author or instructor.

5.1. Prior Distribution

Prior distribution represents a system's initial assumption on the data of previous learners (Neal, 2001). Choosing an appropriate prior distribution is the key for a successful Bayesian inference (Gelman, 2002) because the prior distribution is combined with the probability distribution of new learners' data to yield the posterior distribution, which in turn is treated as the new "prior distribution" for deriving future posterior distributions. If the initial prior distribution is not informative, it will take a long time for the e-learning system to "train" the Bayesian Model by using new learners' data so that the proper inference can be made for the next new learner.

Prior distributions can be obtained from different sources and methods. To the best of our knowledge, there is no single commonly accepted method. It would be ideal if a large empirical dataset that contains the profile and performance data of previous

learners was available (Gertner & VanLehn, 2000). However, such a dataset is most likely not available for two reasons. First, there is no accepted standard for data that comprehensively characterize a learner's profile and performance, in spite of the fact that several organizations have been working on such a standard (LIP, 2010; PAPI, 2001). Second, the data conditions that are regarded by one expert as relevant to a Bayesian Model can be different from those of another expert. The lack of an established standard may explain why some existing adaptive e-learning systems (Gamboa & Fred, 2001; Butz et al., 2008; Conati, Gertner, & Vanlehn, 2002; García, Amandi, Schiaffino, & Campo, 2007; Arroyo & Woolf, 2005; Desmarais, Maluf, & Liu, 1995) limit themselves to using only easily obtainable data such as test results, questionnaire results and students' log files instead of using a full range of attributes that characterize learners' profile and performance.

The prior distribution can also be obtained by asking an expert (Mislevy et al., 2001), who has prior experiences in instructing learners on a subject. However, this is time-consuming and error-prone because the expert will have to consistently assign prior probabilities to the root nodes and different combinations of conditional probabilities to the non-root nodes of a Bayesian Model. Reported literature also does not provide all the required probabilities (Xenos, 2004). A considerable amount of data processing and some additional domain knowledge are still required to derive an informative prior distribution (Druzdzel & Gagg, 2000).

The prior distribution consists of prior probabilities and conditional probability tables (CPTs). While assigning prior probability values to root nodes is relatively simple, assigning conditional probability values to non-root nodes is not. This is because the

prior probabilities can be determined by the expert based on the estimated percentages of learners who satisfy the data conditions given in an adaptation rule. On the other hand, the conditional probabilities consist of multiple values computed from different combinations of true/false values of all the parent nodes to form the CPTs. Our challenge is therefore to automatically derive the CPTs for all the non-root nodes using a limited amount of inputs from the expert. Our approach is to ask the expert to assign prior probabilities to root nodes and weights to all the edges of a Bayesian Model through a user interface, and to introduce three formulas to automatically derive the CPTs. The next subsection explains our approach.

5.2. Deriving Initial Conditional Probability Tables

We use a simple example to explain our approach. Figure 5-1 shows that the truth value of a child node (C) is influenced by two parent nodes P_1 and P_2, and the weights assigned to them to show the relative strengths of their influence. Here, the conditional probability is the probability of C being true given the probabilities of P_1 and P_2 being true. Suppose each node has two states: true (shown as P_1) and false (shown as $\sim P_1$). There are eight possible conditional probabilities to quantify the parent-child dependency: $p(C|P_1, P_2)$, $p(\sim C|P_1, P_2)$, $p(C|\sim P_1, P_2)$, $p(\sim C|\sim P_1, P_2)$, $p(C|P_1, \sim P_2)$, $p(\sim C|P_1, \sim P_2)$, $p(C|\sim P_1 \sim, P_2)$, and $p(\sim C|\sim P_1 \sim, P_2)$.

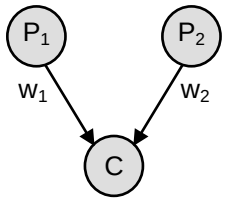

Figure 5-1. Two-parent-one-child relationship with weights

Bayes' rule can be used to compute these conditional probabilities. For example, $p(C|P_1, P_2)$ is calculated as:

$$p(C|\, P_1, P_2) = \frac{p(P_1P_2 \mid C)\cdot p(C)}{p(P_1P_2)}$$

$$= \frac{p(P_1 \mid C)\cdot p(P_2 \mid C)\cdot p(C)}{p(P_1 \mid C)\cdot p(P_2 \mid C)\cdot p(C) + p(P_1 \mid \sim C)\cdot p(P_2 \mid \sim C)\cdot p(\sim C)} \qquad (5\text{-}1)$$

Note that, in order to compute $p(C|\, P_1, P_2)$, we need to know the numerical values of these six terms: $p(C)$, $p(\sim C)$, $p(P_1|C)$, $p(P_1|\sim C)$, $p(P_2|C)$, and $p(P_2|\sim C)$. Calculations of $p(C|\sim P_1,P_2)$, $p(C|P_1,\sim P_2)$, and $p(C|\sim P_1,\sim P_2)$ can be done in a similar way:

$$p(C \mid \sim P_1, P_2) = \frac{p(\sim P_1 \mid C)\cdot p(P_2 \mid C)\cdot p(C)}{p(\sim P_1 \mid C)\cdot p(P_2 \mid C)\cdot p(C) + p(\sim P_1 \mid \sim C)\cdot p(P_2 \mid \sim C)\cdot p(\sim C)}$$

$$p(C \mid P_1, \sim P_2) = \frac{p(P_1 \mid C)\cdot p(\sim P_2 \mid C)\cdot p(C)}{p(P_1 \mid C)\cdot p(\sim P_2 \mid C)\cdot p(C) + p(P_1 \mid \sim C)\cdot p(\sim P_2 \mid \sim C)\cdot p(\sim C)}$$

$$p(C \mid \sim P_1, \sim P_2) = \frac{p(\sim P_1 \mid C)\cdot p(\sim P_2 \mid C)\cdot p(C)}{p(\sim P_1 \mid C)\cdot p(\sim P_2 \mid C)\cdot p(C) + p(\sim P_1 \mid \sim C)\cdot p(\sim P_2 \mid \sim C)\cdot p(\sim C)} \quad .$$

These three equations show that we must know four more terms other than the six terms previously identified. The ten probabilities required to compute the CPT are shown in Table 5-1.

Table 5-1. Ten probabilities required for CPT computation

Probability				
$p(C)$	$p(P_1\|C)$	$p(P_1\|\sim C)$	$p(P_2\|C)$	$p(P_2\|\sim C)$
$p(\sim C)$	$p(\sim P_1\|C)$	$p(\sim P_1\|\sim C)$	$p(\sim P_2\|C)$	$p(\sim P_2\|\sim C)$

The values for the probabilities shown in the upper row of Table 5-1 are complements of the corresponding values shown in the lower row. Within the five probabilities shown in the upper row, there are two pairs, which can be calculated in a similar manner: the method for finding $p(P_1|C)$ is the same for finding $p(P_2|C)$, only with

a different parent. The same goes for $p(P_1|\sim C)$ and $p(P_2|\sim C)$. Therefore, we only need to show how the three highlighted probabilities in Table 5-1 can be derived in order to compute each CPT. In the remainder of this section, we present three formulas used for estimating the values of $p(C)$, $p(P_1|C)$, and $p(P_1|\sim C)$, respectively.

5.3 Three Formulas for Deriving Conditional Probability Tables

5.3.1 Formula 1: Weighted Sum for p(C)

In order to find $p(C)$, the weighted sum is used. Given $p(P_1)$ and $p(P_2)$, $p(C)$ can be found if relative weights w_1 and w_2 are assigned to P_1 and P_2, respectively, where $0 < w_{1,2} < 1$, and $w_1 + w_2 = 1$.

Formula 1: $p(C) = p(P_1)\cdot w_1 + p(P_2)\cdot w_2$

5.3.2 Formula 2: Correlation Coefficient for $p(P_1|C)$

According to the definition of conditional probability, the conditional probability of P_1 given C is:

$$p(P_1|C) = \frac{p(C \cap P_1)}{p(C)} \tag{5-2}$$

Therefore, to find the value of $p(P_1|C)$, we need to know the value of $p(C\cap P_1)$, or "the intersecting probability of C and P_1", which depends on the correlation coefficient of the two.

- If the relationship between P_1 and C is proportional (i.e., if P_1 is true then C is true, and if P_1 is false then C is false), then the correlation coefficient would be in the range of 0 to 1. A correlation coefficient equal to 1 would mean that $p(C\cap P_1)$ has the maximum value.
- If the relationship is inversely proportional (i.e., if P_1 is true, then C is false and vice versa), then the correlation coefficient would be in the range of –1 to 0. A correlation coefficient equal to –1 would mean that $p(C\cap P_1)$ has the minimum value.

- A correlation coefficient equal to 0 means that P_1 and C are independent. In this case, we can compute $p(C \cap P_1) = p(P_1) \cdot p(C)$ based on the probability independence theory.

If we assume that the relationship between P_1 and C is proportional, then the correlation coefficient must be between 0 and 1. Therefore, our task becomes finding a suitable value in the range of 0 to 1.

In the example of "two parents ($P_1$and P_2) and one child (C)", the influence of P_1 on C can be different from or equal to that of P_2. The relative strengths of their influence are represented by the weights assigned to them. Therefore, we can use these weights to determine the proper correlation coefficient values for $p(C \cap P_1)$ and $p(C \cap P_2)$. Let us use $p(C \cap P_1)_0$ to denote the probability of $C \cap P_1$ when the correlation coefficient is 0, and $p(C \cap P_1)_1$ to denote its probability when the correlation coefficient is 1. Then $p(C \cap P_1)_{w1}$ is the probability of $C \cap P_1$ when the correlation coefficient is w_1. As it lies between $p(C \cap P_1)_0$ and $p(C \cap P_1)_1$, we can get $p(C \cap P_1)_{w1}$ by multiplying the difference $p(C \cap P_1)_1 - p(C \cap P_1)_0$ with the weight of P_1 (i.e., w_1) then adding $p(C \cap P_1)_0$. Thus, the probability of $C \cap P_1$ can be derived by the following equation:

$$p(C \cap P_1) = p(C \cap P_1)_0 + \{p(C \cap P_1)_1 - p(C \cap P_1)_0\} \cdot w_1. \qquad (5\text{-}3)$$

Equation 5-3 allows us to use the influence of P_1 on C (i.e., the weight) to express the intersection of P_1 and C (i.e., $p(C \cap P_1)$). The value of $p(C \cap P_2)$ can be derived in a similar fashion by replacing P_1 with P_2 and w_1 with w_2.

Formula 2: $p(P_1|C) = \dfrac{p(C \cap P_1)}{p(C)} = \dfrac{p(C \cap P_1)_0 + \{p(C \cap P_1)_1 - p(C \cap P_1)_0\} \cdot w_1}{p(C)}$,

where p(C) is not equal to zero

5.3.3 Formula 3: Complement Conversion for $p(P_1|\sim C)$

Theoretically, $p(P_1|\sim C)$ can be estimated using the method described in Section 5.3.2. However, it is not desirable to have estimated values if they can be avoided. Since $p(P_1|\sim C)$ can be calculated by using $p(C)$ from Formula 1 and $p(P_1|C)$ from Formula 2, the formula for its calculation is shown below:

$$p(P_1|\sim C) = \frac{p(P_1) - p(C)\cdot p(P_1 \mid C)}{p(\sim C)} \tag{5-4}$$

This formula is proven below:

By definition, $p(P_1|C) = \frac{p(C \cap P_1)}{p(C)}$, where p(C) is not equal to zero.

Similarly, $p(C|P_1) = \frac{p(P_1 \cap C)}{p(P_1)}$, where $p(P_1)$ is not equal to zero.

So, $p(C\cap P_1) = p(C)\cdot p(P_1|C)$, and $p(P_1\cap C) = p(P_1)\cdot p(C|P_1)$.

Since, by definition, $p(C\cap P_1) = p(P_1\cap C)$, we can derive that

$p(P_1\cap C) = p(P_1)\cdot p(C|P_1) = p(C)\cdot p(P_1|C)$.

Similarly, $p(\sim C\cap P_1) = p(P_1\cap \sim C) = p(P_1)\cdot p(\sim C|P_1) = p(\sim C)\cdot p(P_1|\sim C)$. (5-5)

We know from Set theory that

$p(P_1\cap \sim C) = p(P_1 - C) = p(P_1) - p(C\cap P_1) = p(P_1) - p(C)\cdot p(P_1|C)$. (5-6)

We can derive from equations (5) and (6) that $p(\sim C)\cdot p(P_1|\sim C) = p(P_1) - p(C)\cdot p(P_1|C)$.

Formula 3: $p(P_1|\sim C) = \frac{p(P_1) - p(C)\cdot p(P_1 \mid C)}{p(\sim C)}$, where p(~ C) is not equal to zero

Formulas 1, 2, and 3 are used to compute the first three probabilities out of the ten listed in Table 5-1. From those three values, the rest of the probabilities required for the CPT can be derived. By using three formulas given above and Bayes' rule, all CPTs can be automatically computed. The expert only needs to provide the prior probabilities of the root nodes and the weights to all the edges of a Bayesian Model.

5.4 Computation Example for Deriving Conditional Probability Table

A Bayesian Model is designed to probabilistically compute the set of data conditions so that the system can decide on the proper adaptation action to take. As we mentioned in Chapter 4, each Bayesian Model is a structure of data conditions given in the corresponding rule. This structure captures the correlations among these data conditions. Six Event-Condition_probability-Action-Alternative_action (ECpAA) rules, if provided by the expert, are activated at six different stages of processing a learning activity. We shall use the 'Roll-Up' rule given in Figure 5-2 as an example to explain how to derive the prior distribution for the *RollUpBM* of this activity.

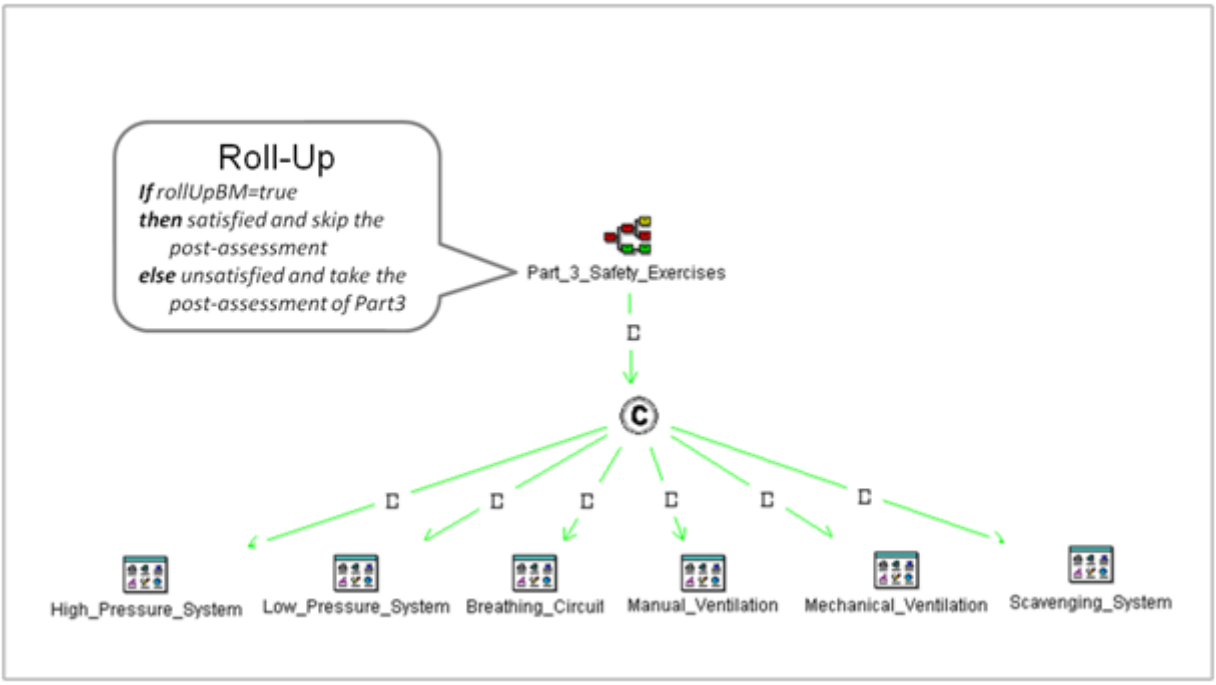

Figure 5-2. Example of 'Roll-Up' rule

The 'Roll-Up' rule is associated with a parent activity and is evaluated based on the learner's performance in its child activities to decide the objective status of the parent. In this example, the rule is shown below:

Event: On the event of returning to the parent activity after a child activity has been processed,

Condition_probability: if [p(PL, AL, NFS, AS) ≥ 0.60], where *PL*, *AL*, *NFS*, and *AS* are shown as the root nodes in Figure 5-3,

Action: set Parent-Summary-Status as 'Satisfied' and skip the post-assessment of the parent activity,

Alternative_action: set Parent-Summary-Status as 'Unsatisfied' and carry out the post-assessment.

The example shown in Figure 5-2 is taken from an implemented learning object for teaching medical personnel in the proper use of a simulated anesthesia machine (Jeon et al., 2007b). The parent activity, Part_3_Safty_Exercises, has six child activities, which are connected to the parent activity by a connector denoted by ©.

RollUpBM is designed to compute p(PL, AL, NFS, AS) given in the Condition_probability specification. As shown in Figure 5-3, *rollUpBM* has four root nodes defined below:

PL (Pass Limit): If four out of the six child activities have an assessment score greater than or equal to 70, then *PL* is true,

AL (Attempt Limit): If the number of attempts does not exceed the number of child activities, then *AL* is true,

NFS (No Failure Score): If none of the assessment results of the child activities is less than 50, then *NFS* is true,

AS (Average Score): If the average score of the attempted child activities is greater than or equal to 70, then *AS* is true, where Average Score = $\frac{Total\ Score\ of\ Child\ Activities}{Number\ of\ Attempted\ Child\ Activities}$.

These root nodes are included in this Bayesian Model because they are deemed important for making the roll-up decision. To specify the correlations among these root nodes, two non-root nodes, Limit Value (*LV*) and Measure Value (*MV*), are introduced to form a structure that leads to the final non-root node named Roll Up (*RU*).

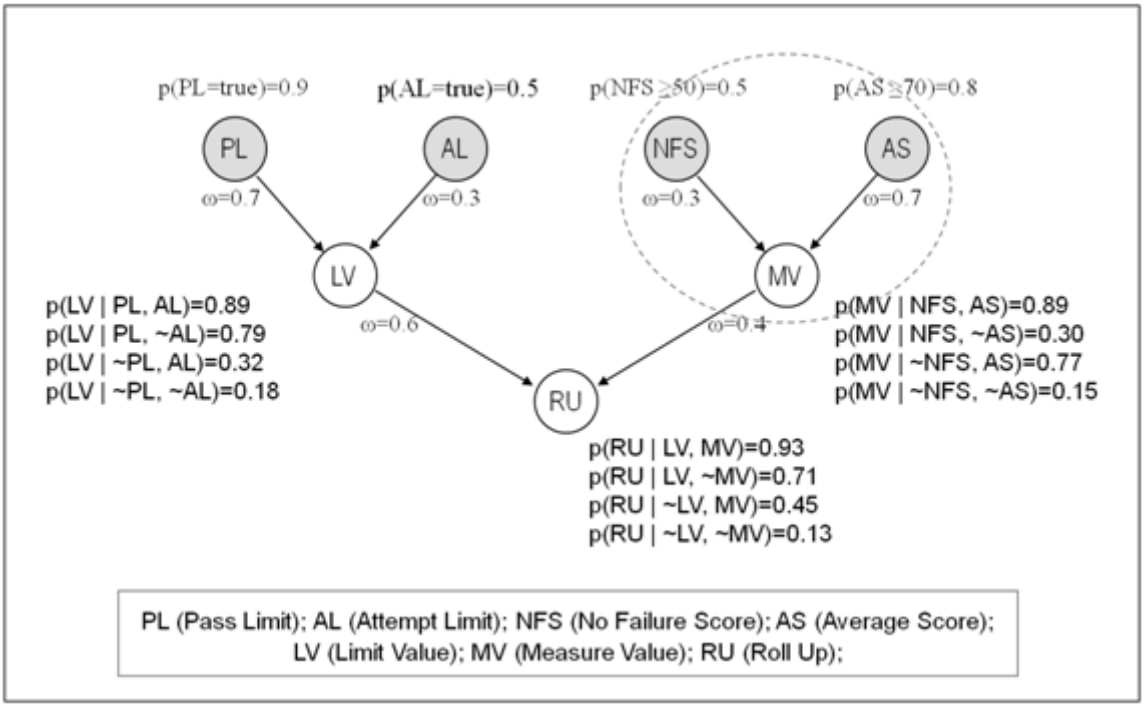

Figure 5-3. Prior probability distribution and weights of rollUpBM

After the specification of the rule's data conditions and the design of the Bayesian Model's structure, the prior distribution must be acquired. The prior distribution is necessary for the system to evaluate its first new learner. The acquisition of prior distribution contains two steps: assigning prior probabilities of the root nodes and computing CPTs of the non-root nodes. Prior probabilities are assigned to the root nodes based on the expert’s knowledge of previous learners. For example, if 90% of previous learners satisfied *PL*, then the probability of *PL* being true is 0.9 as denoted by p(*PL* is true) = 0.9 in Figure 5-3. Additionally, weights can be introduced to the edges

that connect the parent nodes to a child node to specify the relative influences of the parent nodes on the child node. For example, as shown in Figure 5-3, the truth value of *LV* is 70% dependent on the truth value of *PL* and 30% on the truth value of *AL*. The prior probabilities of the root nodes and the weights assigned to all the edges can be used to derive the conditional probability tables for all the non-root nodes. Each table contains entries that show the probability of a child node being true given all the combinations of true and false values of the parent nodes. For example, the probability of *MV* being true, given that *NFS* is true (shown by *NFS*) and AS is false (shown by ~*AS*), is 0.3 as denoted in Figure 5-3 by p(*MV*| *NFS*, ~*AS*) = 0.3. Using this prior distribution, *rollUpBM* can determine the probability value of the RU node; if this value is greater than or equal to the threshold specified in the 'Roll-Up' rule (i.e., 0.60), then the action clause of the rule is processed. Otherwise, its alternative action clause is processed.

We now explain the process of generating CPTs using the *MV* node shown in Figure 5-3 as an example. The terms P_1, P_2, C, w_1, and w_2 from Section 5.3 can now be replaced by *NFS*, *AS*, *MV*, w(*NFS*), and w(*AS*) respectively. In *rollUpBM*, after prior probabilities and weights have been assigned by the expert through the user interface, the system uses these assigned data along with the presented formulas to automatically compute the probability values shown in the right column of Table 5-2. These probability values are then used to derive the CPT for the *MV* node as shown in Table 5-3 by using Bayes' rule (Equation (5-1)). The CPTs of other non-root nodes of *rollUpBM*, *LV* and *RU*, are computed in the same manner, and their results are shown in Figure 5-3. The

derived prior distribution allows our system to aptly evaluate a learner and provide an adaptive e-learning experience to the learner.

Table 5-2. A set of probability terms/formulas for generating probability values for MV

Probability Term	Formula	Probability value	Probability Term	Formula	Probability value
p(MV)[1]	*p(NFS)*w(NFS)+p(AS)*w(AS)*	0.71	p(~MV)	*1 – p(MV)*	0.29
p(NFS|MV)[2]	$\frac{p(NFS \cap MV)}{p(MV)}$	0.56	p(~NFS|MV)	*1 – p(NFS|MV)*	0.44
p(NFS|~MV)[3]	$\frac{p(NFS) - p(MV) * p(NFS \mid MV)}{p(\sim MV)}$	0.35	p(~NFS|~MV)	*1 – p(NFS|~MV)*	0.65
p(AS|MV)[2]	$\frac{p(AS \cap MV)}{p(MV)}$	0.94	p(~AS|MV)	*1 – p(AS|MV)*	0.06
p(AS|~MV)[3]	$\frac{p(AS) - p(MV) * p(AS \mid MV)}{p(\sim MV)}$	0.46	p(~AS|~MV)	*1 – p(AS|~MV)*	0.54

Note: super scripts ([1,2,3]) denote which of our proposed formulas were used (Section 5.3).

Table 5-3. CPT of node MV

Conditional Probability	*p*(*MV*|*NFS, AS*)	*p*(*MV*|*~NFS, AS*)	*p*(*MV*|*NFS, ~AS*)	*p*(*MV*|*~NFS, ~AS*)
Probability Value	0.89	0.77	0.30	0.15

5.5 Chapter Summary

An informative prior distribution is needed for Bayesian inference and is the most difficult task to build a Bayesian Model. In this chapter, we present formulas to derive CPTs. The conditional probabilities are automatically generated based on the presented formulas together with assigned prior probabilities and weights by the expert. Then, CPTs are derived by Bayes' rule. The prior distribution of a Bayesian Model consists of the assigned prior probabilities to the root nodes and the computed conditional probability tables for the non-root nodes. The prior distribution is updated as each new learner's profile and performance data become available, which is then used to evaluate the next new learner. The system thus continues to improve the accuracy of learner evaluation as well as its adaptive capability. We have evaluated our approach of deriving prior distributions and updating the distributions using simulated learner cases and have found that the approach is effective (See Section 8.4). Our system also provides a graphical user interface called Bayesian Model Editor, which allows the system to easily obtain all the information necessary to derive the prior distribution of a Bayesian Model from the expert. The implementation details of the Bayesian Model Editor will be presented in Chapter 6.

CHAPTER 6
ADAPTIVE E-LEARNING SYSTEM AND ITS IMPLEMENTATION

GELS is an adaptive e-learning system designed to enable Web users who have the same interest on a subject of learning to form an e-learning community. People in the community play the following major roles: content author, content learner, and community host. A member of the community can play multiple roles. Content authors develop and register learning objects for the virtual community. Content learners select and learn from learning objects delivered by GELS. The Community host manages software components installed at the host site and communicates with both learners and authors. The general architecture of GELS is shown in Figure 6-1.

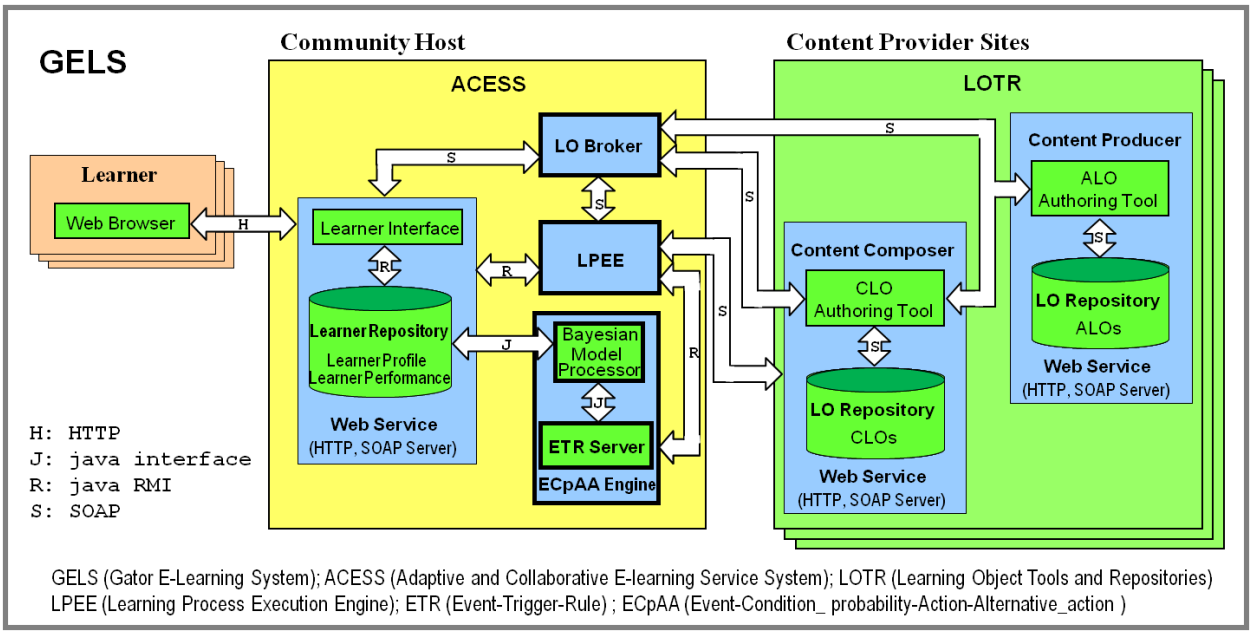

Figure 6-1. The system architecture of GELS

Some system components are newly developed (e.g., Bayesian Model Processor, Bayesian Model Editor, Learner registration interfaces). Others (e.g., LO Tools and Repositories, LO Broker, ETR Server, and Learning Process Execution Engine) were developed in a previous project (Lee & Su, 2006). The newly developed components

are then integrated with the existing components to form the adaptive system GELS. We present GELS' tools and system components in the following subsections.

6.1 Authoring Tools

GELS provides the members of a virtual e-learning community with development tools and e-learning service components to facilitate the authoring, registration, and management of contents and adaptation rules. The primary purpose of these tools is to simplify the authoring process without requiring content authors to have programming skills. These tools provide user-friendly graphical interfaces to make the authoring process more intuitive.

6.1.1 Learning Object Authoring Tools

We extend the previously developed LO Authoring Tools, which facilitate the authoring and registration of ALOs and CLOs, by adding a metadata editor and a rule editor. The extended LO Authoring Tools provide a tabbed dialog box for selecting and performing several functions (i.e., create, add, delete, edit, and register a learning object). In addition to the dialog box, the CLO Authoring Tool provides a frame which has a canvas area for graphically designing an activity tree. A Metadata Editor is included in both authoring tools in the tabbed dialog box for defining metadata. The CLO Authoring Tool has a rule editor within its tabbed dialogue box for optionally creating and editing an adaptation rule for each adaptation point of an activity.

Both LO Authoring tools generate XML files by using Java Swing components. XML parsing, updating, writing, and validating are implemented by using the Java Architecture for XML Binding (JAXB) technology (Oracle, n.d.) based on an XML Schema definition. The user interfaces of the ALO authoring tool and the CLO authoring tool are shown in Figure 6-2 and Figure 6-3, respectively.

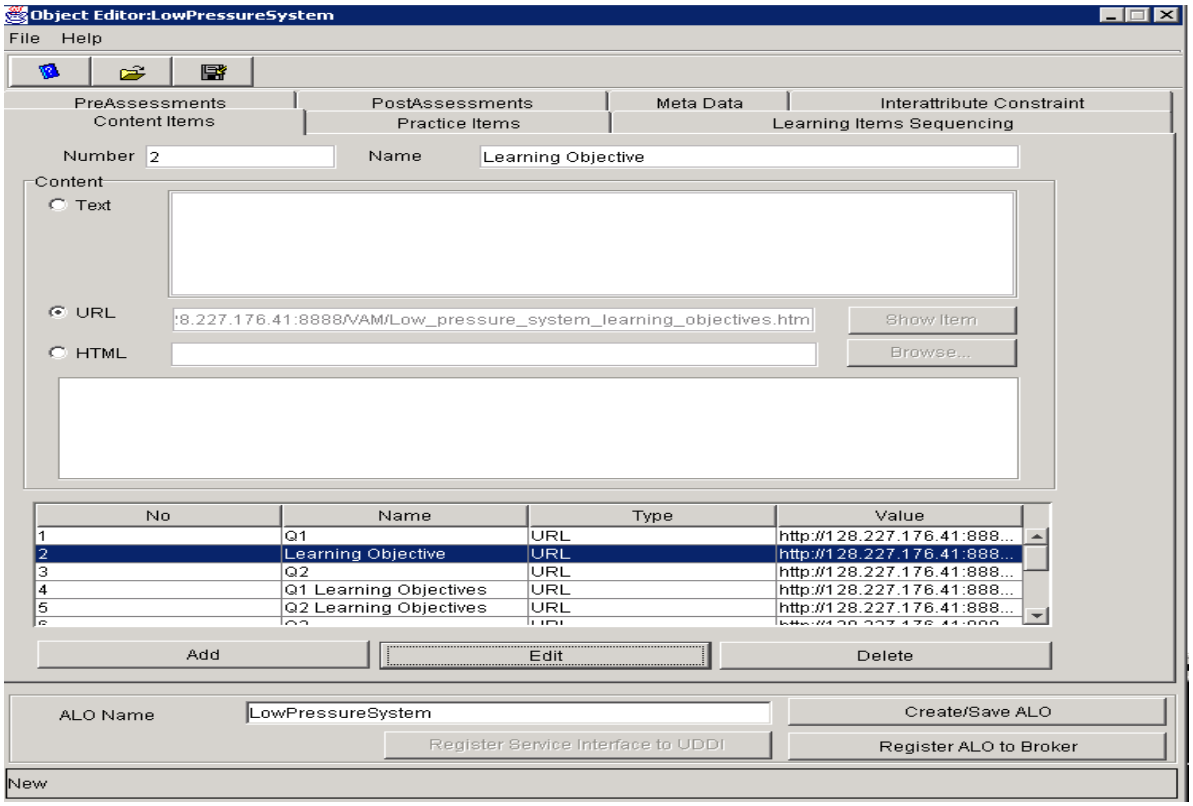

Figure 6-2. ALO authoring tool.

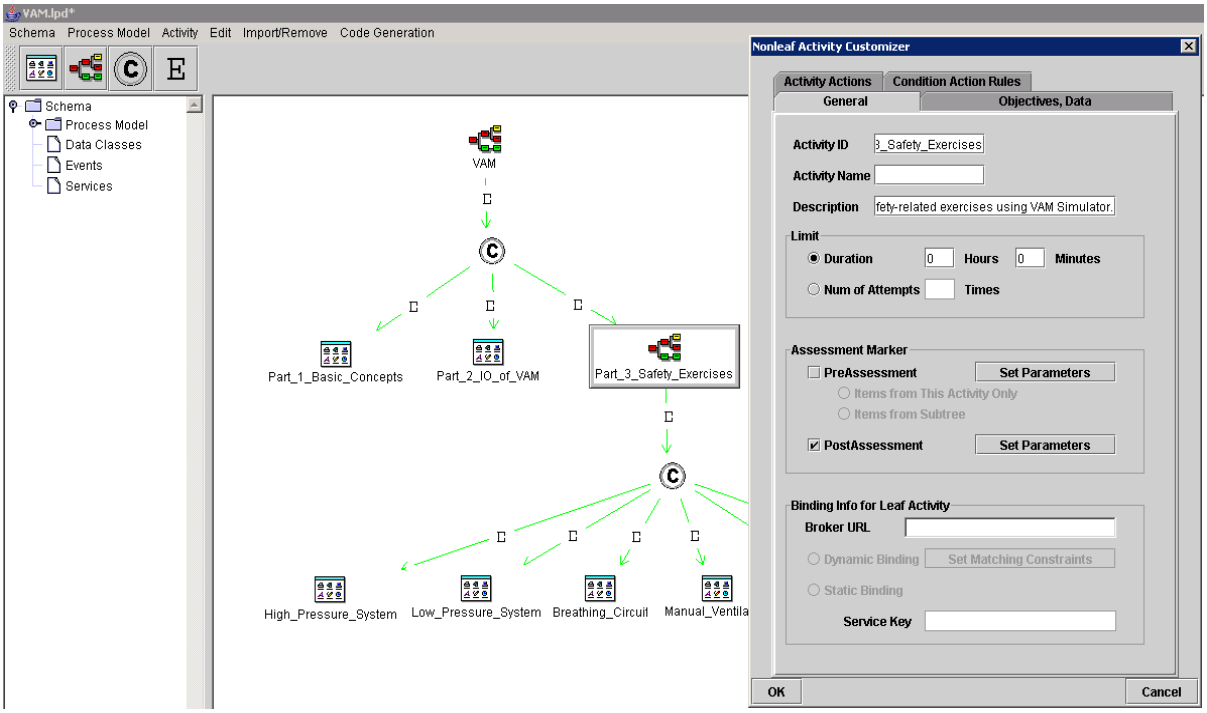

Figure 6-3. CLO authoring tool.

6.1.2 Bayesian Model Editor

The Bayesian Model Editor is a graphical user interface, which allows the expert to provide all the information necessary for the system to derive the prior distribution of a Bayesian Model. As shown in Figure 6-4, the interface provides an image of the Bayesian Model's structure and allows the expert to assign prior probabilities to root nodes and weights to all edges based on his/her best estimation. Since the sum of the weights of the joined edges is 1.0, when the expert assigns a weight to an edge leading from one parent, the interface automatically sets the weight of the edge leading from the other parent. The system uses these assigned data along with the formulas presented in Chapter 5 to automatically compute conditional probability tables. This interface is implemented using Matlab and Java. Figure 6-4 illustrates the user interface showing the assigned values for the example *rollUpBM*.

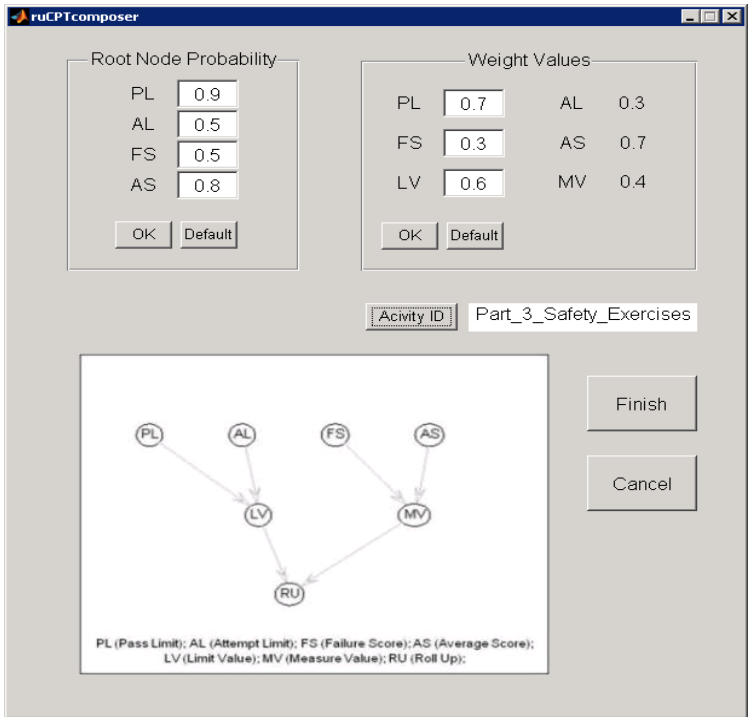

Figure 6-4. Bayesian Model Editor

6.2 Adaptive e-Learning Service Components

GELS' system components are grouped into three sets installed at different network sites of a virtual e-learning community. Shown in the box on the right side of Figure 6-1 are the Learning Object Tools and Repositories (LOTRs) installed at each content developer's site. Shown in the middle of the box is the Adaptive and Collaborative E-Learning Service System (ACESS), which is installed at the community host site. ACESS consists of a LO Broker, a Learning Process Execution Engine (LPEE), a Learner Interface, a Learner Repository, and an ECpAA Rule Engine. All learning activities are processed by ACESS. Shown on the left box of Figure 6-1 is the facility needed at a content learner site. Only a Web browser is needed for accessing the adaptive e-learning services provided by GELS.

6.2.1 Learning Object Tools and Repositories (LOTRs)

As we mentioned in Section 6.1.1, we extended the previously developed LOTRs for the development and storage of reusable learning objects. Each constructed learning object is written in XML. Each LO Repository is an XML database managed by the Web-based XML DBMS, Xindice (The Apache XML Project, 2007). Since an LO Repository is implemented as a Web-service, the Java Web Service (JWS) technology supported by the Apache Axis's Simple Object Access Protocol (SOAP) engine (W3C, 2007) is used to deploy the Web-service of the LO Repository.

6.2.2 Learning Object Broker (LO Broker)

The LO Broker is a metadata repository that supports learning object registration, browsing, and dynamic binding facilities. It is built upon the Universal Description, Discovery, and Integration (UDDI) registry (OASIS, 2005) and the extended Web Service Description Language (WSDL) information model (W3C, 2001). The extended

WSDL document can be posted on the Internet. It describes the metadata of a Web-service description, such as a textual description, category information, and a constraint element.

6.2.3 Learning Process Execution Engine (LPEE)

We reuse the existing LPEE for processing learning objects selected by a learner or the system. When LPEE receives a request for processing a learning object, it fetches the requested learning object from a LO Repository by making a Web-service call. It then creates an instance of the learning object in order to record the execution history of the fetched learning object. The instance of the learning object is written in an XML document and is maintained in the Learner Repository at the host site. LPEE is encapsulated as a JAVA Remote Method Invocation (RMI) server, which is easily accessible to the ECpAA rule engine and the Learner Interface component.

6.2.4 Event-Condition_probability-Action-Alternative_action (ECpAA) Rule Engine

The rule engine processes rules defined in the ECpAA format. It consists of an Event-Trigger-Rule (ETR) Server developed in a previous project, a newly developed Bayesian Model Processor (BMP), and Java / Matlab interfaces. We explain its components below.

6.2.4.1 Event-trigger-rule server (ETR server)

The ETR Server manages and processes events, triggers, and rules. It is an event-trigger-rule processor that provides timely and automated responses to an event by processing a triggered rule. An event type has a data structure and can be posted to represent an occurrence of the event type. It is implemented as a Java class. A trigger is a Java object that links an event to a rule, which is processed by the ETR Server

when the associated event is posted. A rule is defined by a rule designer and is also implemented as a Java class.

6.2.4.2 Bayesian Model Processor (BMP)

The BMP is a probabilistic condition evaluator. It probabilistically evaluates the condition specification of an ECpAA rule upon the ETR Server's request and returns the result of a Bayesian Model to the ETR Server, which takes the proper adaptation action according to the returned result. Bayes Net Toolbox (an open-source Matlab package) is used to build Bayesian Models and perform Bayesian reasoning (Murphy, 2007). JAVA / Matlab interfaces are developed to enable the BMP to communicate with the ETR Server and the Repositories.

6.2.5 Learner Interfaces and Repository (LIR)

The Learner Interfaces are implemented by using the Java Server Pages (JSPs) technology which generates Web pages. Through the Web user interfaces, a learner can start, suspend, finish, navigate, and monitor his/her learning activities. The interactions between a learner and a Learner Interface are stored in the Learner Repositories.

6.2.5.1 Learner interfaces

Learner registration interface. Adaptive learning starts with a 'Log-On' page, which carries out user authentication and the loading of the main GELS page. A first-time user of the system is requested to provide his/her profile information to the system through a learner registration interface. Learners' profile information provides the basis for the adaptation and customization of learning activities to give different learning experiences to learners who have diverse competencies and backgrounds. As we discussed in Chapter 2, our system adopts most of the core elements in the IMS

Learner Information Package Specification (LIP, 2010) and extends it to include some additional features such as the specification of preferred learning style. The figure shown below (Figure 6-5) is the screen snapshot of our learner registration interface. We also provide a tool for updating learner profile information so that a learner can update and maintain his/her data continuously.

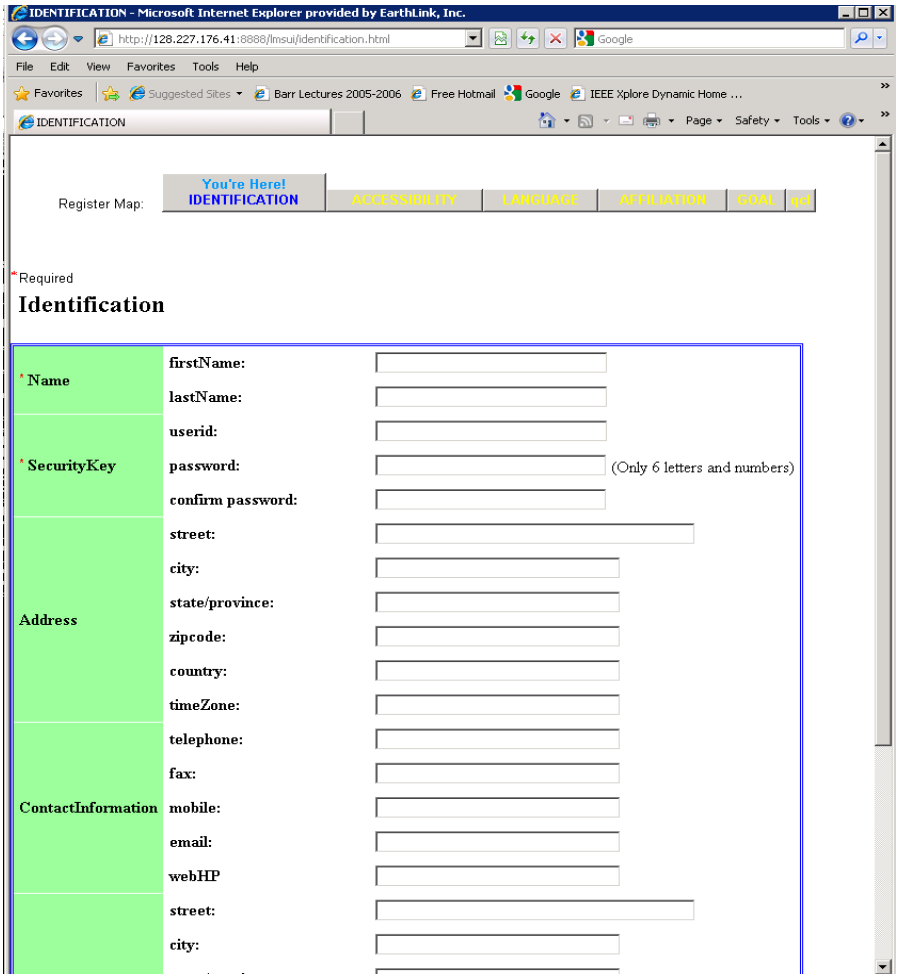

Figure 6-5. Learner registration interface

Learner learning process interfaces. After logging in, a learner can find his/her desired learning objects using the 'Search LO' interface, which is an interface adopted from a previous project and extended by adding metadata. A list of learning objects that satisfies a search is displayed for the learner's selection. The learner can save the

access points of these learning objects, and store those of his/her choice in a learner space, called ‘MyCLOs’, for later execution. ‘MyCLOs’ is an interface adopted from the previous project. It displays the current learning status (“Finished”, “Completed”, “Satisfied/Unsatisfied”, etc) of each learning object and allows a learner to start/resume the saved learning objects. Figure 6-6 shows the ‘MyCLOs’ page. A learner can navigate a learning object by using the activity navigation commands, such as “Continue”, “Previous”, “Next”, “Start/Resume”, “Suspend”, and “Finish”.

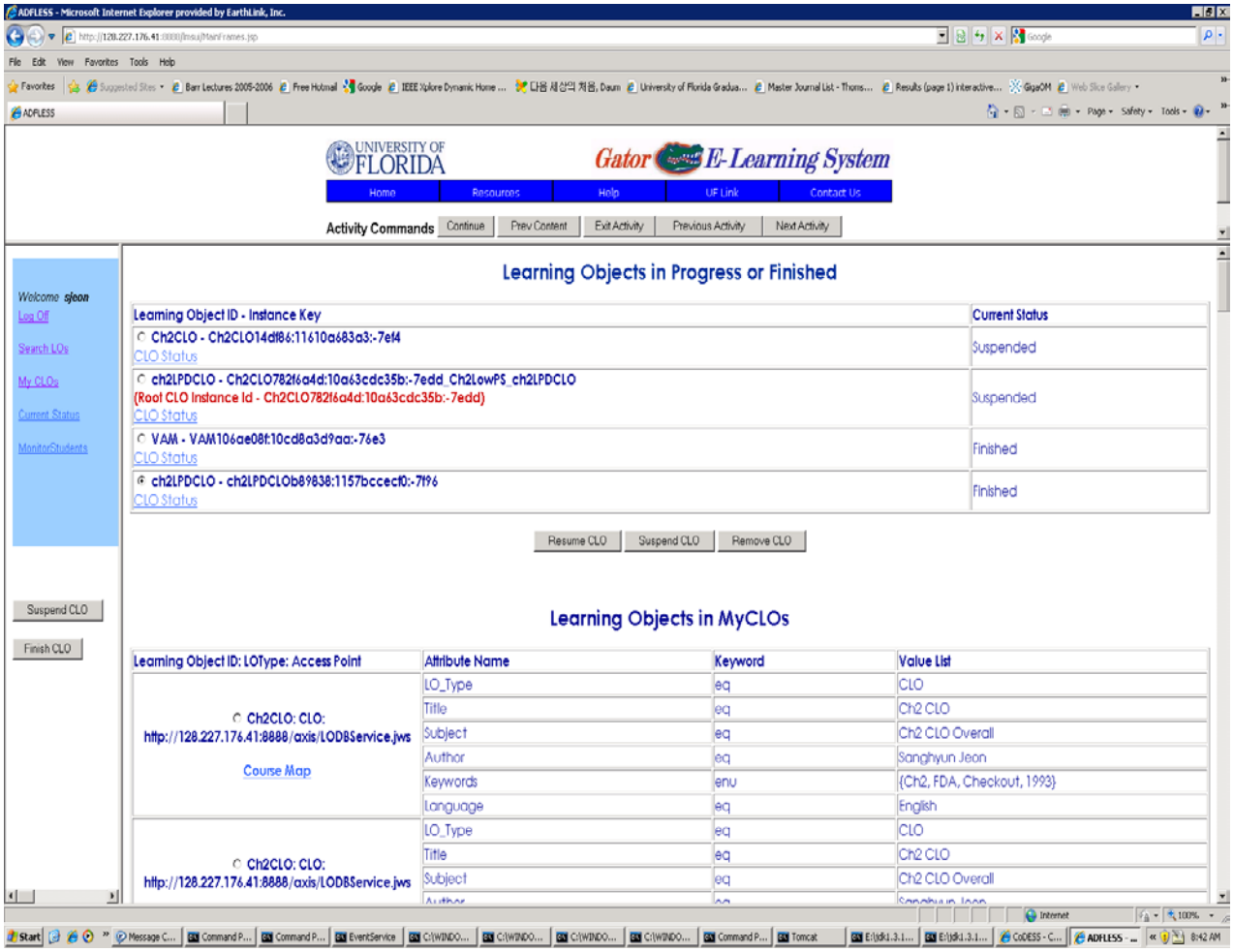

Figure 6-6. MyCLOs

Monitoring interfaces. GELS also provides two monitoring interfaces for virtual community users: a newly developed ‘Monitor Students’ and an existing ‘Current

Status'. 'Monitor Students' allows an instructor to monitor the progress and performance of all the learners who signed up for the instruction of a CLO. This information is also very useful to learning object developers in their design of additional learning objects or their modification of existing learning objects. Figure 6-7 shows a snapshot of 'Monitor Students'.

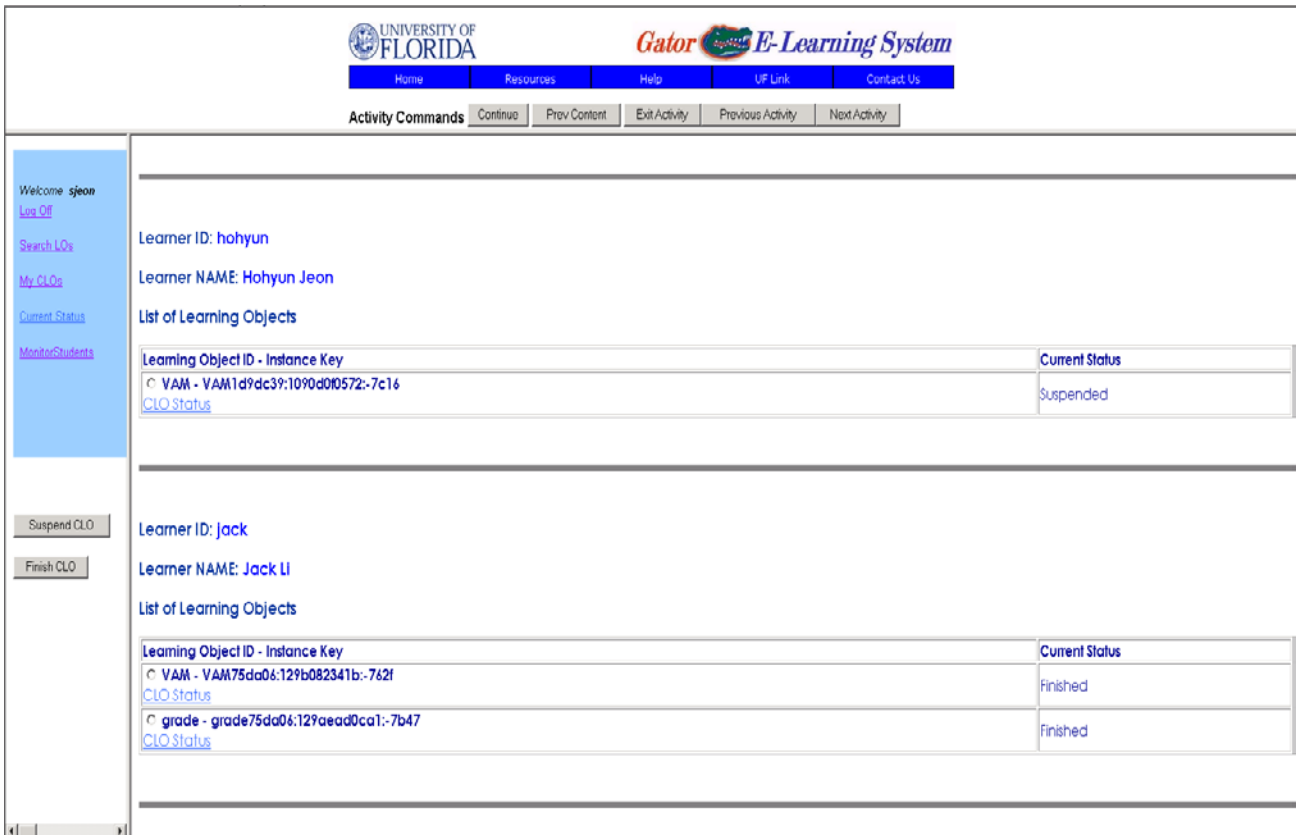

Figure 6-7. Monitor Students

'Current Status' is a CLO Status Monitoring interface, which uses the runtime status model of the CLO instance and provides the current status information of a learning object instance to a learner. A user interface component invokes the monitoring component to create a Web page that presents the status information along with the activity tree of the learning object instance. During the execution of a CLO instance, a leaner can self-monitor the status of a learning object instance through a Web browser.

As shown in Figure 6-8, a current status can display a color-coded map similar to a table of contents. The map shows the activities and their statuses to the learner. The status information can include “Satisfied/Unsatisfied”, “Completed”, “Active”, “Attempted”, “Disabled”, number of attempts, etc.

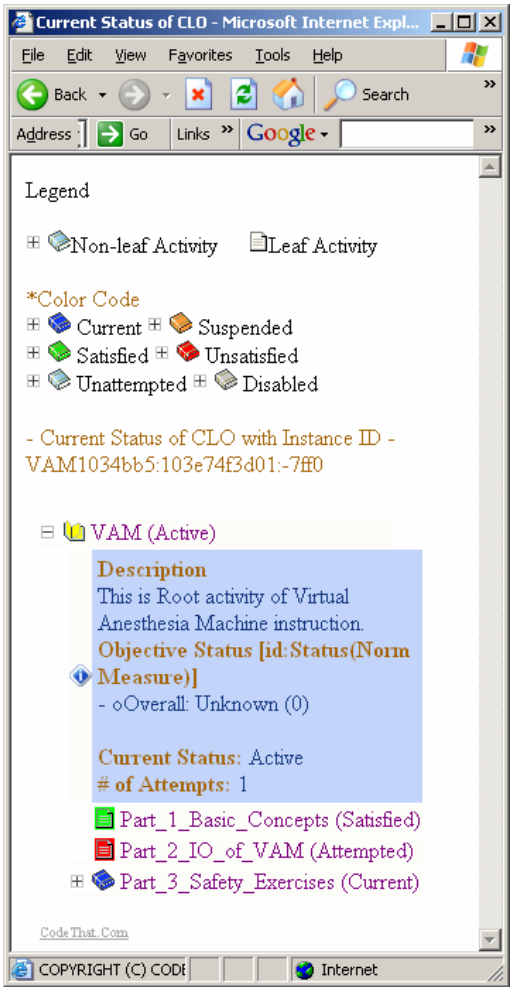

Figure 6-8. Current Status

6.2.5.2 Learner repository

The Learner Repository is an existing XML database managed by the Web-based XML DBMS, Xindice. It consists of two sub-repositories: Learner Profile and Learner Performance. Learner Profile is used to store learner’s profile information gathered

during the learner registration process. Learner Performance stores the learning objects that are fetched from remote LO repositories. These stored learning objects can be processed locally at the host site to achieve better efficiency. Learner Performance is also used to maintain the learner performance information, such as learner's progress, assessment results, and LO runtime status information.

CHAPTER 7
APPLICATIONS OF GATOR E-LEARNING SYSTEM (GELS) IN MEDICAL INSTRUCTION

Hospitals and clinics use complex medical equipment on a regular basis. Patient safety relies on the proper interaction between a skilled practitioner and his/her equipment (Dalley, Robinson, Weller, & Caldwell, 2004). Unfortunately, Weinger (1999) reported that medical practitioners' mistakes in their use of complex equipment are a dominating factor in up to 90% of the problems in medical practices. Errors made by medical professionals draw public attention. It is necessary to teach medical personnel the domain knowledge about equipment as well as their associated skills.

In this chapter, we present two applications of our adaptive e-learning technology in the instruction of a Virtual Anesthesia Machine (VAM). VAM is a Web-based simulator of a generic anesthesia machine developed by the Department of Anesthesiology at the University of Florida (Lampotang, Lizdas, Gravenstein, & Liem, 2006). The first application is designed to teach the medical personnel in the normal functions and operations of anesthesia machines. The second application deals with the instruction of the US Food and Drug Administration's (FDA) pre-use check of traditional anesthesia machines. These applications are developed to show how to convert the available instructional materials into simulation-based learning objects and to demonstrate GELS' adaptation features for achieving individualized instruction.

7.1 Normal Functions of Traditional Anesthesia Machines

In VAM, the instructional material for teaching the functions of anesthesia machines is in the form of a workbook, which is accessible to learners through a web site provided by the Department of Anesthesiology at the University of Florida. The workbook is like a text or a training manual, and was not designed based on learning

object design principles. The content of the workbook is put together in a large structure; it is not partitioned into reusable instructional modules with their corresponding assessment items and well-defined objectives. Although some remarks are given to learners to suggest an order of study, a learner can explore any part of the workbook without restriction. This may be suitable for some learners, but others may need a more structured and guided instruction.

After studying the workbook, a learner can activate the VAM simulator to practice what he/she learned from the workbook by, for example, turning the dials or opening and closing the valves of the machine to see the effect on gas flow. After practicing, the learner is instructed to visit a separate quiz website to be assessed on the knowledge and skills learned. The workbook, simulator, and quiz are separate entities. They are not integrated to provide learners a seamless learning experience.

In their previous work, Lee and Su (2006) made use of the contents provided in an existing workbook and the existing quiz questions to construct a number of ALOs. They then aggregated these ALOs to construct a CLO, which is defined by an activity tree that models the instructional process of VAM as shown in Figure 7-1. The workbook consists of the following three parts. Part 1 introduces the basic concepts related to a general anesthesia machine. Part 2 introduces how the VAM simulation works. Part 3 covers specific safety-related exercises about the six subsystems of an anesthesia machine: the High Pressure System, the Low Pressure System, the Breathing System, the Manual Ventilation System, the Mechanical Ventilation System, and the Scavenging System. In this activity tree of the CLO, the contents of parts 1 and 2 are used as the content items of two ALOs, and the contents of the six subsystems of part 3 and their

associated quiz questions are used as the content items and assessment items of six ALOs. Each of these ALOs contains practice items which describe a demonstration scenario related to the content, and provides a web link for invoking the VAM simulator so that the learner can use it to follow the demonstration scenario. Thus, a learner can study the content of an ALO, use the VAM simulator for practice, and have his/her knowledge and skill assessed in a seamless fashion under the control of GELS. Unlike SCORM's activity tree, GELS allows non-leaf activities of an activity tree to have content and assessment items. The non-leaf activities labeled ‘VAM’ and ‘Part3_safety_exercises’ have content items to introduce and summarize the content presented by the sub-trees rooted at these activities. Their assessment items gauge a learner’s ability to integrate the knowledge and skills learned from the ALOs of the sub-trees.

The previous application focused on converting the instructional materials of VAM into learning objects by using the authoring tools and showed the advantages of the object-based system over the VAM system. In the current application, we add metadata in the specifications of learning objects; all constructed learning objects are tagged with metadata, which are registered with the LO Broker to facilitate adaptive search and selection of learning objects. We also incorporate ECpAA rules in the CLO as shown in Figure 7-1 to demonstrate GELS’ adaptive features. Some rule examples are given below.

1. To select a suitable learning object, a ‘Before-Activity’ rule is invoked before processing an activity. For example, the ‘Before-Activity’ rule selects the most suitable learning object of the ‘Part 1 – Basic Concepts’ and presents it to a learner based on the

rule condition specification, which may specify that, if the learner's language preference is Korean, the Korean version of ‘Part 1 –Basic Concepts’ should be selected.

2. A different learning path is provided to a learner based on his/her prior knowledge level evaluated by *afterPreAssessmentBM*. An ‘After-Pre-Assessment’ rule selects one of the following actions: “process the activity”, “skip the activity”, or “ask the learner to take a prerequisite first”. For example, a rule would state that, if the probability value derived by the *afterPreAssessmentBM* is greater than or equal to 0.9 in the ‘Mechanical_Ventilation’ activity, a learner is allowed to skip this activity.

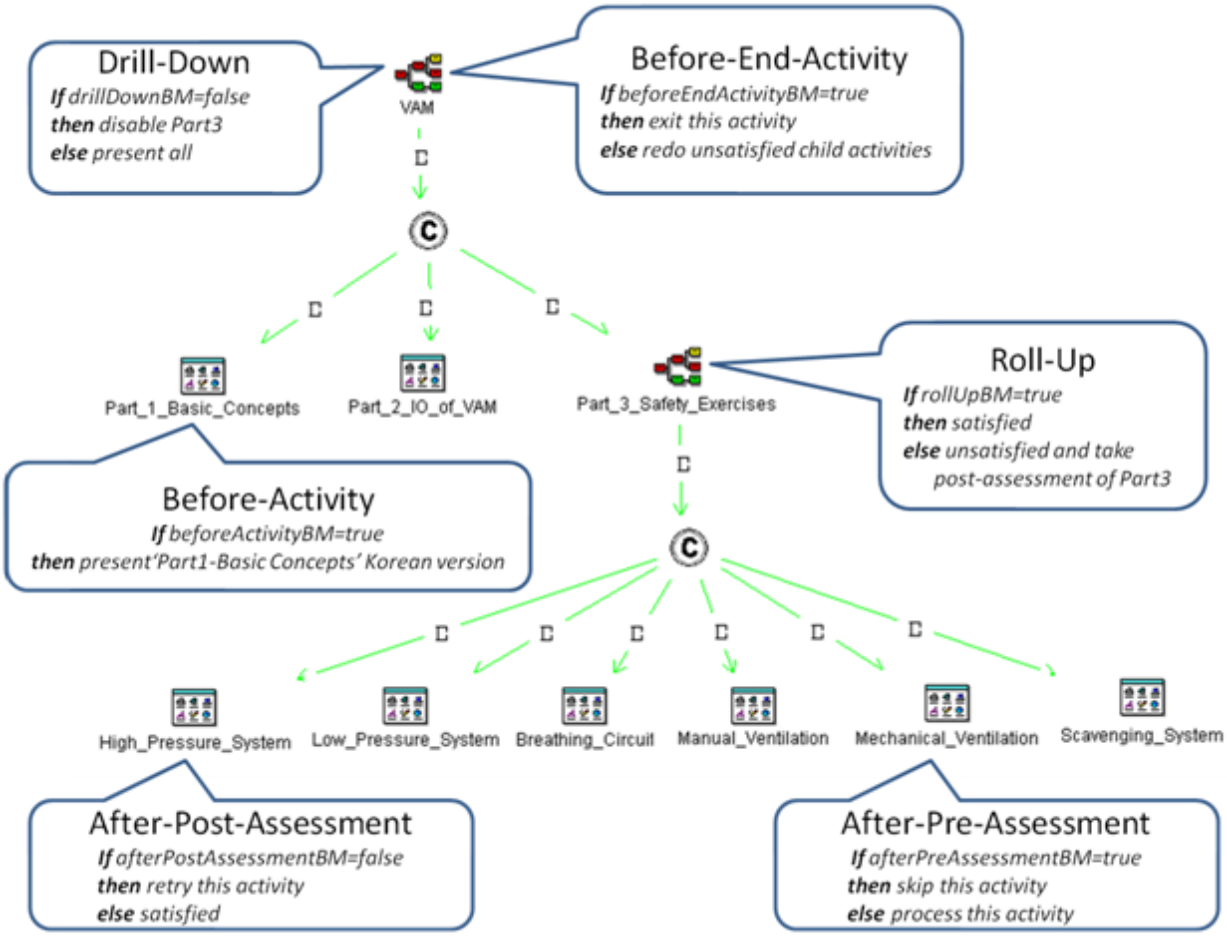

Figure 7-1. Activity tree that models ‘Normal Functions of Traditional Anesthesia Machines’

3. The processing sequence of child activities (e.g. “enable/disable a child activity”, “hide a child activity from choice”, or “sort the child activities”) can be controlled by a ‘Drill-Down’ rule. For example, the Drill-Down rule of the root activity called ‘VAM’ invokes the *drillDownBM* before going down the activity tree to process its child activities. Depending on the resulting probability value of *drillDownBM*, the rule would select one of the sequences for processing the child activities. If the *drillDownBM*’s evaluation result is false (i.e., the probability value derived from the *drillDownBM* is less than 0.6), then the system disables the Part3_Safety_Exercises, which has more detailed information about the sub-system, so that it will be hidden from the learner. Otherwise, all three child activities can be seen by the learner.

4. A ‘Roll-Up’ rule checks the status of each child activity (i.e., a learner’s performance in the child activity) by using the *rollUpBM* and sets the summary status for the parent activity. Depending on the result of the *rollUpBM,* the system may execute either the specified action or an alternative action like the example shown in the ‘Roll-Up’ rule of the ‘Part3-Safety Exercises’. In this example, if *rollUpBM* is true, the system will set the objective status of the ‘Part3-Safety Exercises’ as 'Satisfied'. Otherwise, the objective status of the ‘Part3-Safety Exercises’ will be set as 'Unsatisfied' and since the learner did not successfully study the child activities, the learner is required to take a post assessment in the ‘Part3-Safety Exercises’. A ‘Roll-Up’ rule is used to close the child activities, while a ‘Drill-Down’ rule opens them to the learner. Roll-Up and Drill-Down rules are applicable only to non-leaf activities.

5. The learner’s performance result can be used for making adaptation decisions, such as “retry this activity”, “retry selective activities”, “update the objective of the

activity", etc. For example, if the probability derived by an *afterPostAssessmentBM* is less than 0.7, an 'After-Post-Assessment' rule can require a learner to retry the 'High Pressure System' activity. Otherwise, the system will set the objective status of the 'High Pressure System' activity as 'Satisfied'.

6. Before terminating an activity, the system can check a learner's performance and the learning conditions associated with the activity such as the number of attempts allowed, the period of time that the learner is allowed on each attempt, and the period of time spent on the activity in a 'Before-End-Activity' rule. Such a rule can decide if the learner should be asked to redo the activity or be allowed to leave the activity. For example, a 'Before-End-Activity' rule in the root activity allows a learner to either leave the "VAM" activity or redo some selected child activities of the 'VAM' activity, depending on the probability derived by the *beforeEndActivityBM*.

7.2 Food and Drug Administration's (FDA) Pre-Use Check of Traditional Anesthesia Machines

The second application, 'FDA's pre-use check of traditional anesthesia machines', is a checklist that a person who will operate an anesthesia machine must go through to ensure that the machine is perfectly functional and safe before it is used on a patient. Although the FDA checklist is only one page long as shown in Figure 7-2, each step in the checklist requires that a user perform rather complicated operations on an anesthesia machine. Our task, therefore, is to first design simulation-based learning objects to provide instructions needed for the checklist and then use GELS to find the best way of using simulations to present content, facilitate practice, and perform assessment.

Anesthesia Apparatus Checkout Recommendations, 1993

This checkout, or a reasonable equivalent, should be conducted before administration of anesthesia. These recommendations are only valid for an anesthesia system that conforms to current and relevant standards and includes an ascending bellows ventilator and at least the following monitors: capnograph, pulse oximeter, oxygen analyzer, respiratory volume monitor (spirometer) and breathing system pressure monitor with high and low pressure alarms. This is a guideline which users are encouraged to modify to accommodate differences in equipment design and variations in local clinical practice. Such local modifications should have appropriate peer review. Users should refer to the operator's manual for the manufacturer's specific procedures and precautions, especially the manufacturer's low pressure leak test (step #5).

Emergency Ventilation Equipment

*** 1. Verify Backup Ventilation Equipment is Available & Functioning**

High Pressure System

*** 2. Check Oxygen Cylinder Supply**
- a. Open O_2 cylinder and verify at least half full (about 1000 psi).
- b. Close cylinder.

*** 3. Check Central Pipeline Supplies**
- a. Check that hoses are connected and pipeline gauges read about 50 psi.

Low Pressure System

*** 4. Check Initial Status of Low Pressure System**
- a. Close flow control valves and turn vaporizers off.
- b. Check fill level and tighten vaporizers' filler caps.

*** 5. Perform Leak Check of Machine Low Pressure System**
- a. Verify that the machine master switch and flow control valves are OFF.
- b. Attach "Suction Bulb" to common Fresh) gas outlet.
- c. Squeeze bulb repeatedly until fully collapsed.
- d. Verify bulb stays *fully* collapsed for at least 10 seconds.
- e. Open one vaporizer at a time and repeat 'c' and 'd' as above.
- f. Remove suction bulb, and reconnect fresh gas hose.

*** 6. Turn On Machine Master Switch**
and all other necessary electrical equipment.

*** 7. Test Flowmeters**
- a. Adjust flow of all gases through their full range, checking for smooth operation of floats and undamaged flowtubes.
- b. Attempt to create a hypoxic O_2/N_2O mixture and verify correct changes in flow and/or alarm.

Scavenging System

*** 8. Adjust and Check Scavenging System**
- a. Ensure proper connections between the scavenging system and both APL (pop-off) valve and ventilator relief valve.
- b. Adjust waste gas vacuum (if possible).
- c. Fury open APL valve and occlude Y-pTece.
- d. With minimum O_2 flow, allow scavenger reservoir bag to collapse completely and verify that absorber pressure gauge reads about zero.
- e. With the O_2 flush activated allow the scavenger reservoir bag to distend fully, and then verify that absorber pressure gauge reads < 10 cm H20.

Breathing System

*** 9. Calibrate O_2 Monitor**
- a. Ensure monitor reads 21% in room air.
- b. Verify low O_2 alarm is. enabled and functioning.
- c. Reinstall sensor in circuit and flush breathing system with O_2.
- d. Verify that monitor now reads greater than 90%.

10. Check Initial Status of Breathing System
- a. Set selector switch to "Bag" mode.
- b. Check that breathing circuit is complete, undamaged and unobstructed.
- c. Verify that CO_2 absorbent is adequate.
- d. Install breathing circuit accessory equipment (e.g. humidifier, PEEP valve) to be used during the case.

11. Perform Leak Check of the Breathing System
- a. Set all gas flows to zero (or minimum).
- b. Close APL (pop-off) valve and occlude Y-piece.
- c. Pressurize breathing system to about 30 cm H_2O with O_2 flush.
- d. Ensure that pressure remains fixed for at least 10 seconds.
- e. Open APL (Pop-off) valve and ensure that pressure decreases.

Manual and Automatic Ventilation Systems

12. Test Ventilation Systems and Unidirectional Valves
- a. Place a second breathing bag on Y-piece.
- b. Set appropriate ventilator parameters for next patient.
- c. Switch to automatic ventilation (Ventilator) mode.
- d. Fill bellows and breathing bag with O_2 flush and then turn ventilator ON.
- e. Set O_2 flow to minimum, other gas flows to zero.
- f. Verify that during inspiration bellows deslvers appropriate tidal volume and that during expiration bellows fills completely.
- g. Set fresh gas flow to about 5 L/min.
- h. Verify that the ventilator bellows and simulated lungs fill <u>and empty</u> appropriately without sustained pressure at end expiration.
- i. <u>Check for proper action of unidirectional valves.</u>
- j. Exercise breathing circuit accessories to ensure proper function.
- k. Turn ventilator OFF and switch to manual ventilation (Bag/APL) mode.
- l. Ventilate manually and assure inflation and deflation of artificial lungs and appropriate feel of system resistance and compliance.
- m. Remove second breathing bag from Y-piece.

Monitors

13. Check, Calibrate and/or Set Alarm Limits of all Monitors

Capnometer	Pulse Oximeter
Oxygen Analyzer	Respiratory Volume Monitor (Spirometer)
Pressure Monitor with High and Low Airway Alarms	

Final Position

14. Check Final Status of Machine

a. Vaporizers off	d. All flowmeters to zero
b. AFL valve open	e. Patient suction level adequate
c. Selector switch to "Bag"	f. Breathing system ready to use

* If an anesthesia provider uses the same machine in successive cases, these steps need not be repeated or may be abbreviated after the initial checkout.

Figure 7-2. FDA's pre-use check

Our design of learning objects is guided by these design principle: each ALO represents a self-contained and reusable object, each CLO aggregates a number of related ALOs, these learning objects must be reusable in the instructions of other related medical devices, and they must be easily searched and selected. Dr. Lampotang from the Department of Anesthesiology at the University of Florida develops 115 learning assets. We use these assets and ALOs and CLOs are aggregated to form a Root CLO in order to capture the instructional content of the Checklist as illustrated in Figure7-3. The tree contains four non-leaf activities and thirteen leaf activities. Two of the leaf activities, named Ch2LPS and Ch2BS, in turn link to other LPS CLO and BS

CLO to cover the pre-use checks of the Low Pressure System and the Breathing System, respectively. In this way, a more complex structure of a learning object is formed. The other leaf activities are bound to ALOs, each of which provides content, practice, and assessment items to instruct and assess the pre-use check of one part of an anesthesia machine. All created LOs and their metadata are registered with the LO Broker, which provides browsing and querying facilities.

In the first application described in Section 7.1, we use the VAM simulator to offer learners an environment in which to practice what they learned from the instructional content. In this second application, we use simulations not only to present instructional content, but also to provide opportunities for practice and perform assessment visually and interactively. The processing of each ALO shown in Figure 7-3 is carried out in three steps: See one, Do one and Test one.

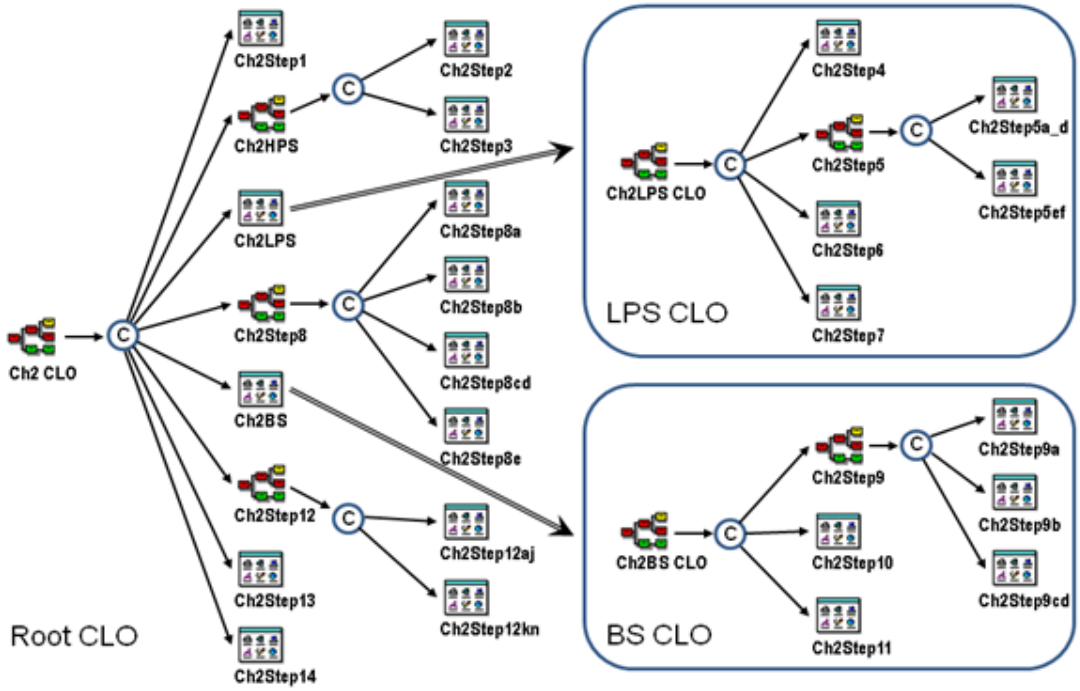

Figure 7-3. Activity tree that models the checklist

The step ‘See one’ displays the VAM simulator that is to be checked. A ‘show me’ (i.e. a ‘play’) button, when clicked by the learner, will show the actions of an expert who operates the machine properly to perform the pre-use check of that part of the machine covered by the ALO. By observing the expert’s actions and the effects of the actions on the simulator, the learner can develop his/her own knowledge and skills pertaining to the machine. Figure 7-4 shows a snapshot of what the learner sees at this first step.

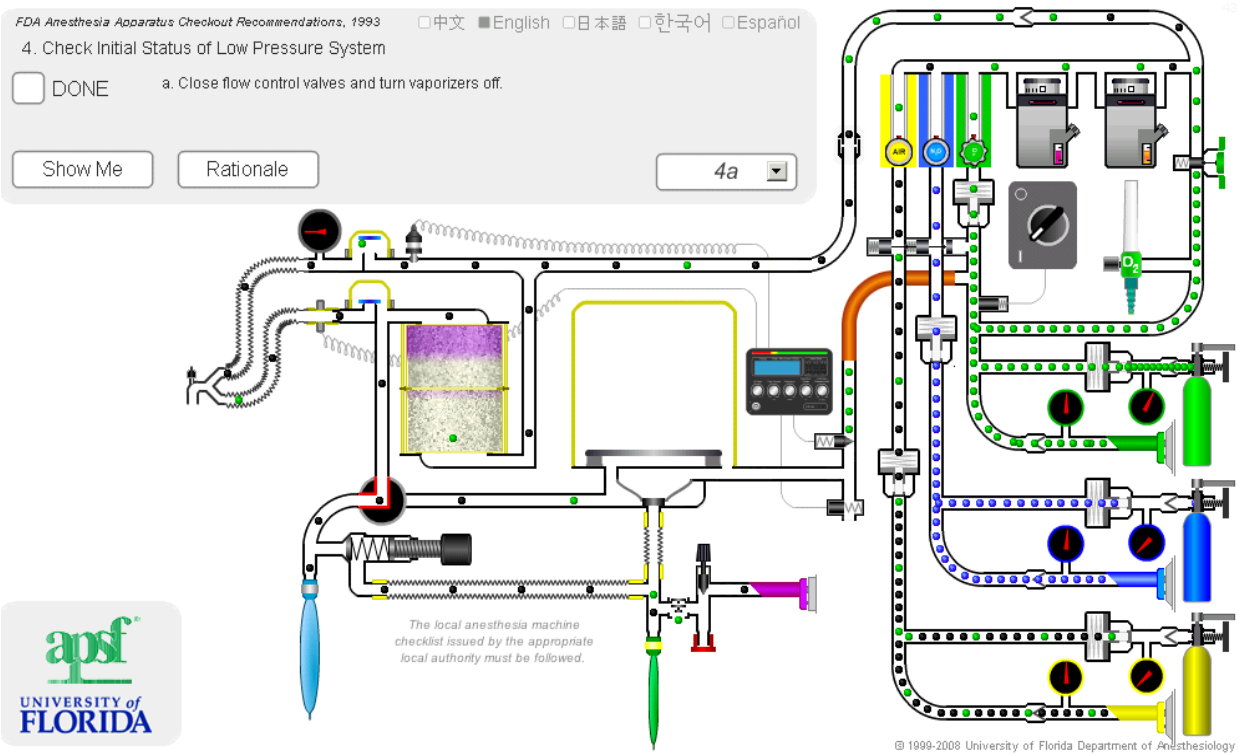

Figure 7-4. VAM simulator at the ‘See one’ step

The step ‘Do one’ is a practice step designed to let the learner redo the steps that ‘See one’ showed. It enables the learner to solve problems on his/her own with the help of some tutorial messages. Although the learner operates on the same simulator as the one shown in the ‘See one’ step, in ‘Do one’, the learner can operate the simulator without any limitation because all icons are activated. The learner is free to try out

different things and find the correct procedure by trial-and-error. When a mistake is made, the learner gets immediate feedback on his/her action. If a learner makes a sufficient number of mistakes, the simulation will take over and show the ‘See one’ operation again to the learner.

‘Test one’ (Figure 7-5) is an assessment stage for testing the learner’s competence and retention of domain knowledge and skills. The simulator tricks the learner into making a mistake by giving either a faulty or a perfect machine at random and then provides the assessment result as delayed feedback after the learner’s work is assessed. Since all parts of the simulator are operable, the learner is free to do anything to solve the problem given to him/her. ‘Test one’ aims to identify and correct the misconceptions acquired or gaps left in earlier steps.

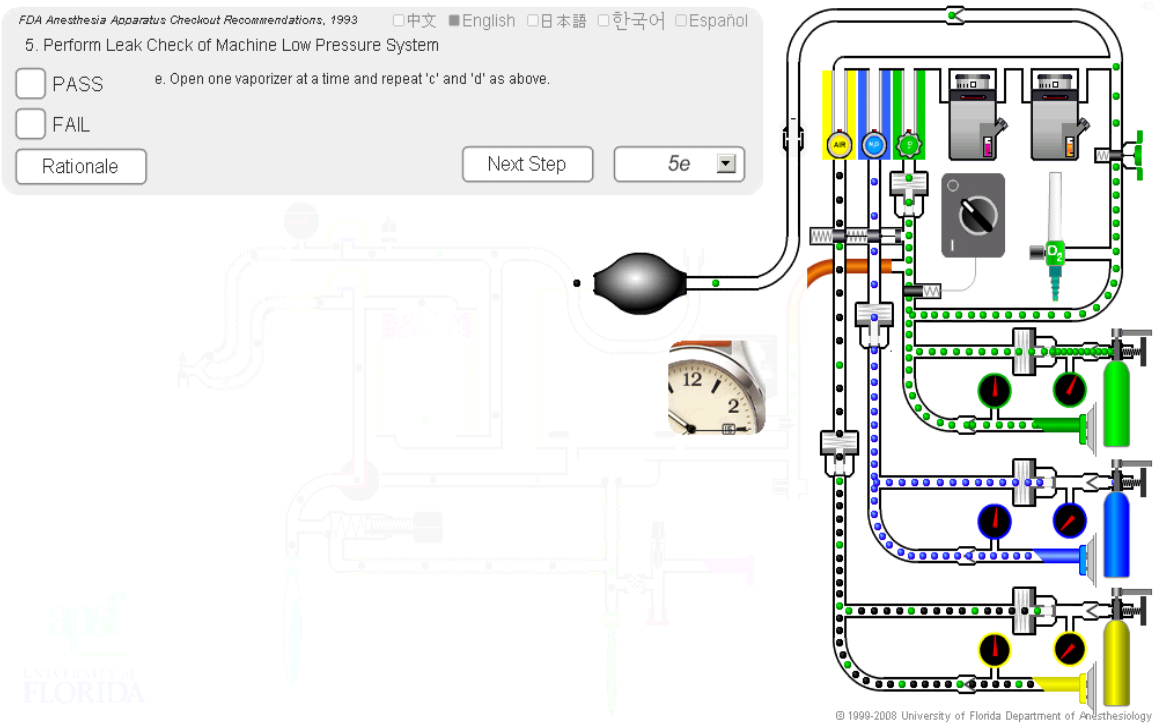

Figure 7-5. VAM simulator at the ‘Test one’ step

Simulation-based assessment gives the learner a chance to review what he/she learned, correct misconceptions, and reflect on his/her progress. The ‘Test one’ step is a mouse-driven interface where the learner’s clicks are tracked by the VAM simulator to determine if the learner is following the correct steps. This method of ‘interactive assessment’ is very different from the traditional approach of simple multiple-choice and fill-in-the-blank questions, because it allows the system to capture all of a learner’s actions taken on the simulator according to the assessment instructions that were given. This is effective in a simulation-based learning environment. The assessment result collected by the VAM simulator is passed to the assessment component of GELS, which records and uses the assessment result to control and carry out the subsequent processing of the CLO developed for the pre-use checklist (Jeon, Lee, Lampotang, & Su, 2007a).

In summary, we have added the extended metadata to the previously developed ALOs and CLOs for training medical personnel in the functionalities and use of traditional anesthesia machines and have developed simulation-based learning objects for the FDA’s pre-use check of traditional anesthesia machines by using the existing instructional materials used in VAM. We have also integrated GELS with the VAM simulator to demonstrate that the integrated system can operate seamlessly to deliver instruction, use the simulator for instruction and/or practice, and perform assessment. By applying ECpAA rules and their corresponding Bayesian Models in the processing of the developed CLOs, we have demonstrated that GELS can deliver these learning objects adaptively to suit diverse learners' profiles and performances.

CHAPTER 8
EVALUATION OF GELS' ADAPTIVE FEATURES AND PROCESSING TECHNIQUES

In this work, we are interested in verifying, if the use of ECpAA rules, Bayesian Models, and group data, as well as the techniques used to implement and process them, gives GELS the desired adaptive features. For this purpose, we design and conduct four simulations. The purpose of these simulations is not to demonstrate the effectiveness of our system in improving learners' ability to learn things better and faster. This would be a very difficult undertaking because there are too many factors involved in determining a learner's ability to learn and is out of the scope of our current research. Rather, the purpose is to verify and demonstrate that, by using ECpAA rules and their corresponding BMs, a system like GELS is able to exhibit some desirable adaptive features and to resolve the data uncertainty and correlation problems addressed in this dissertation. The first three simulations make use of the ‘Drill-Down’ rule and its corresponding *drillDownBM* (Figure 4-1 of Chapter 4), and the fourth simulation makes use of the ‘Roll-Up’ rule and its corresponding *rollUpBM* (Figure 5-3 of Chapter 5) for the evaluation. The first simulation demonstrates that GELS is able to make different drill-down decisions when two learners have different profile and performance data. The second simulation examines the effect and advantage of using continuously updated group data to make a drill-down decision. The third simulation shows GELS’ ability to handle missing data. The fourth simulation shows the effect of using prior probabilities, weights and the proposed formulas to derive the prior distribution of *rollUpBM* in seven simulated user-cases.

8.1 Adaptation Based on Individual Profile and Performance Data

In this simulation, we are interested in knowing if our system is able to make reasonable adaptation decisions for two individuals with different profiles and performances. The main purpose is to show that the techniques used to implement and process ECpAA rules and their corresponding Bayesian Models work. As shown in Table 8-1, Learner X has an *AGP* of 3.5 (i.e., *AGP* is true) and a *GPA* of 3.6 (i.e., *GPA* is true). The conditional probability of *PKL* given "*AGP* and *GPA*" is 0.91 (denoted in Figure 4-1 by p(PKL | AGP GPA) = .91). Since Learner X shows a *CPA* of 3.8, the conditional probability of *LC* given "*CPA* and *PKL*" is 0.92 (denoted in Figure 4-1 by p(LC | CPA PKL)=.92). Even though Learner X shows a low *LM* (i.e., *CLS*=false and *LG*=false), he/she will be allowed to drill-down because the conditional probability value of *DrillDown* (*DD*) comes out to be 0.81, which is above the specified threshold of 0.6. In the case of learner Y, who has an *AGP* of 2.8, a *GPA* of 2.9, and a *CPA* of 2.3, but has a high *LM*, the conditional probability value of *DD* is only .37 which is below the threshold. Therefore, GELS will take different actions based on the different results derived for these two learners.

Table 8-1. Profile and performance data of learners X and Y

Values of input/output node	*AGP*	*GPA*	*CPA*	*CLS*	*LG*
Learner X	3.5 (True)	3.6 (True)	3.8 (True)	False	False
Learner Y	2.8 (False)	2.9 (False)	2.3 (False)	True	True

AGP(Average Grade of Prerequisites); GPA(Grade Point Average); CPA(Current Pre-Assessment score); CLS(Competency Level on a Subject); LG(Learning Goal).

8.2 The Effect of Using Updated Group Profile and Performance Data

For performing Bayesian inference (Jaynes, 1998), an informative prior distribution for a Bayesian model is needed. The initial prior distribution given by a domain expert may or may not accurately represent the profiles and performance data of a group of

previous learners who have taken the same learning object. Thus, updating the prior distribution based on the data of each new learner is very important because it improves the quality of the prior distribution and thus improves the system's ability to correctly evaluate the condition specifications of adaptation rules processed for the next new learner.

This simulation aims to study the effect of continuously updating the probability values of the *drillDownBM*. We use the data of a group of learners whose individual data are randomly generated to compute a prior distribution. Two cases are compared. First, the same prior distribution is used for making the drill-down decision for all new learners. The individual data of each new learner is stored but is not used to update the prior distribution. Thus, it does not affect the system's drill-down decision for the next new learner. Second, the individual data of each new learner is used to update the prior distribution, which is used in making the drill-down decision for the next new learner.

Knowing that different learner data can affect drill-down decisions, we randomly generate individual learner data for three groups of new learners in a controlled manner. Each group has 1000 learners. The first group, called the "mid group", represents learners having an average performance. Based on the data conditions generated for the root nodes, the numbers of true and false values are determined for this data set. Using these truth values, conditional probability tables for the non-root nodes are computed to make the drill-down decision. As shown in Table 8-2, 60.1% of the learners in this group are allowed to drill-down. We then use the same data set to produce a "low group" and a "high group". The former contains 10% fewer true values than the mid group and represents learners having a lower performance; the latter has 10% more

true values than the mid group and, thus represents learners having a higher performance.

Table 8-2. Effect of using group profile and performance data

Prior distribution	Learner Data Type	Drilldown result of Non Updating System	Drilldown result of Updating System
Mid (601/1000)	Mid	601/1000 (60.1%)	601/1000 (60.1%)
	High	740/1000(74.0%)	767/1000 (76.7%)
	Low	458/1000(45.8%)	400/1000(40.0%)

Table 8-2 shows that it is advantageous to continuously update the group data and the prior distribution of the Bayesian Model as each new learner's individual data becomes available. The generated data for the mid group is statistically equal to the prior distribution that primes the model. Thus, the percentage of drill-down is the same for both cases. The drill-down percentage of the high group increases from 74.0% to 76.7%, due to the fact that the continuous update of the model's prior distribution increases the probability values of both the prior probabilities of the root nodes and the values of the CPTs. The drill-down percentage of the low group decreases from 45.0% to 40.0% for the opposite reason. Updating group data and the prior distribution allows the system to make more accurate drill-down decisions. This is analogous to the case of a good instructor who can do a better job in teaching as he/she gains more and more experience from each one of the students he/she instructs.

8.3 Incomplete Data Handling

In this simulation, we want to know how well GELS can make a drill-down decision when a learner's data is incomplete. As we showed in Figure 4-1 of Chapter 4, the *drillDownBM* requires five input values for the five root nodes to produce the drill-down result, and we assume that the missing value can occur in any one of these five input

attributes (*CLS*, *LG*, *CPA*, *AGP*, and *GPA*). Based on the five input values of each learner, the *drillDownBM* derives the drill-down probability for the learner.

8.3.1 Drill-down Results for Incomplete Data Sets Having Different Percentages of Missing Data

By comparing the drill-down result of a complete data set with those of incomplete data sets, Table 8-3 shows us that the number of learners who are allowed to drill-down increases nonlinearly as the percentage of missing data increases. The differences between the numbers of drill-downs of the complete data set and those of 0.02%, 5% and 10% incomplete data sets are 0, 10, and 29, respectively, which are not very significant. For these data sets, GELS is able to derive drill-down results within an acceptable error range (i.e., 0 to 3%). In the case of a data set with 20% of its data missing, the difference is 87, which is quite significant. However, a data set with 20% of values missing means that each of the 1000 learners could have one data value missing. We consider this to be an extreme case that GELS or any other adaptive system should not be expected to handle; therefore, we have chosen to disregard the 20% case and any other case in which the percentage of missing data is higher than 20. Otherwise, the above results show that GELS is able to handle missing data quite well if the percentage of missing values is not exceptionally high.

Table 8-3. Experimental data sets and drill-down results

Experimental Data Set	Complete	0.02% missing	5% missing	10% missing	20% missing
Number of missing node / number of total input nodes	0/5000	1/5000	250/5000	500/5000	1000/5000
Number of learners who successfully drill down / number of total learners	587/1000	587/1000	597/1000	616/1000	674/1000
Difference from the complete data set		0	10	29	87

8.3.2 Incomplete Data Handling Given Different Percentages of Missing Data

Since the *drillDownBM* derives a drill-down probability for each learner, we can show how many drill-down probabilities in an incomplete data set are within certain error ranges when compared with those of the complete data set. The five error ranges we use are: [0.00, 0.05), [0.05, 0.10), [0.10, 0.15), [0.15, 0.20) and ≥ 0.20. To do the comparison, we first derive the drill-down probabilities of all learners using the complete data set. We assume that these are correct drill-down probabilities, since there is no missing value. We then derive drill-down probabilities for each of the four incomplete data sets and compare each learner's drill-down probability with that of the corresponding learner in the complete data set (i.e., the drill-down probability of the tenth learner in the 10% incomplete data set is compared with that of the tenth learner in the complete data set).

The results are shown in Figure 8-1. When a data set has 0.02% missing values, all learners fall in the error range of [0.00, 0.05). None of them differs from its equivalent in the complete data set by more than 0.05, since the data set has only one missing value. When a data set has 5% missing values, 909 learners' probabilities are almost the same as those of the complete data set differing only by [0.00, 0.05), 28 learners' probabilities differ by [0.05, 0.10), 17 learners' probabilities differ by [0.10, 0.15), 12 learners' probabilities differ by [0.15, 0.2), and 34 learners' probabilities differ by more than 0.20. We observe that the number of learners in the error range of [0.00, 0.05) decreases and the numbers in all other error ranges increase as the percentages of missing values increase. However, more than 90% of learners are still within [0.00, 0.05), which shows that GELS can handle missing data quite well.

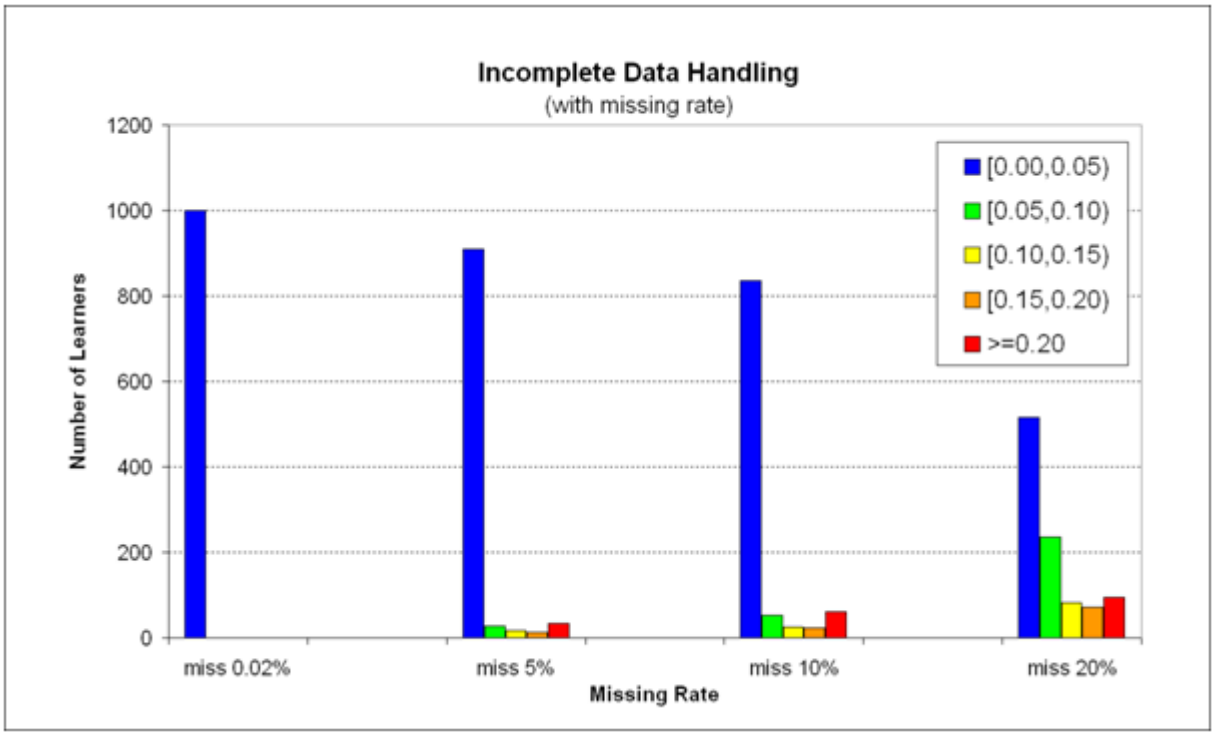

Figure 8-1. Incomplete data handling in a drill-down decision for missing percentages

This trend continues when the percentage of missing values reaches 10%. For this data set, 836 learners' probabilities differ from those of the complete data sets by [0.00, 0.05), 53 learners by [0.05, 0.10), 26 learners by [0.10, 0.15), 24 learners by [0.15, 0.2), and 61 learners by more than 0.20. Given 500 missing values, 84% of learners are within [0.00, 0.05) and only 6% of learners are in the range above 0.20. Finally, when the number of missing values reaches 1000 (i.e., 20% missing values), the result gets worse. The number of learners in the error range of [0.00, 0.05) drops to almost half (52%). The numbers in other error ranges significantly increase: 236 learners in [0.05, 0.10), 83 learners in [0.10, 0.15), 71 learners in [0.15, 0.2), and 94 learners in the range above 0.20. We can therefore make the same conclusion that GELS can handle missing data well when the percentage of missing values is within 10%.

8.3.3 Incomplete Data Handling Given Default Values and Bayesian Model

This simulation compares three approaches to handling missing value: assigning 'True' as the default, assigning 'False' as the default, and using our Bayesian model to derive the missing value probabilistically. We first derive the drill-down probabilities of all 1000 learners using a complete data set. We assume that these are correct drill-down probabilities, since there is no missing value. We then derive drill-down probabilities for each of the incomplete data sets with a missing rate from 1% to 20%. The results are shown in Figure 8-2. The number of learners who drill-down increases as the missing rate increases in the 'Default True' case, whereas the number of learners in the 'Default False' decreases as we expect. Using our Bayesian model gets almost the same result as the complete data set, regardless of the increase in the missing rate. The results thus show that GELS can handle missing data better than the two default cases.

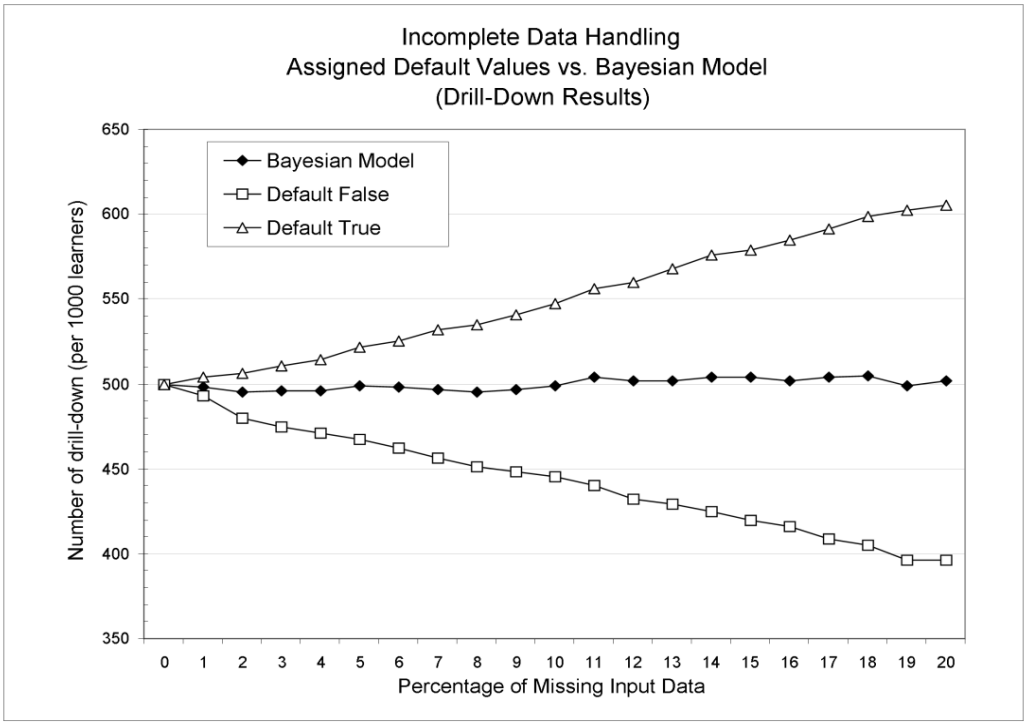

Figure 8-2. Incomplete data handling given assigned default values and Bayesian Model

8.4. Evaluation of the Technique used to Derive Prior Distribution

In this part of our work, we are interested in evaluating the technique used to derive the prior distribution of a Bayesian Model based on the prior probabilities assigned to its root nodes, weights assigned to its edges, and the three formulas given in Chapter 5. For this evaluation, we use the example of Part_3_Safety_Exercises given in Figure 5-2, and the 'Roll-Up' rule given in Chapter 5 and the *rollUpBM* given in Figure 5-3. We use the proposed technique to derive the prior distribution of *rollUpBM* and use it to determine the roll-up probabilities of seven simulated learners who have different performance data. The 'Roll-Up' rule says that, **on** the event of roll-up, **if** [p(PL, AL, NFS, AS) ≥ 0.60], **then** set the objective status of Part_3_Safety_Excercises as 'Satisfied' and skip the post-assessment of Part_3_Safety_Exercises, **else** set Parent-Summary-Status as 'Unsatisfied' and carry out the post-assessment, where *PL* is 'pass limit', *AL* is 'attempt limit', '*NFS* is 'no failure score', and *AS* is 'average score'. Since 'Roll-Up' rule is based on a learner's performance data, the seven learners' performance results of the child activities are given in Table 8-4.

Several notations are used to describe the performances of the learners in detail. The arrow indicates that a learner had to retry a child activity, because the initial score was unsatisfactory. In this experiment, a learner is allowed to retry only once per child node. Boxed numbers indicate satisfactory scores of greater than or equal to 70, whereas shaded numbers indicate failed scores of less than 50. Plain numbers indicate unsatisfactory scores. A summary of the probabilities of *rollUpBM* is provided in Table 8-5. In our simulation, Nicole, Eva and Michael satisfy the pass limit (*PL*) as shown in Table 8-5. Since Nicole satisfies the objectives of her first four child nodes (denoted by *PL* being true in Table 8-5), she is not required to complete the remaining two child

activities. She also has the highest average score (88) and no failed child activities. All of these factors contribute to her high roll-up probability (0.86). Michael has four satisfactory scores with an average score of 70, which is above the threshold. However, his two failed child activities and many attempts result in a roll-up probability of 0.78. His roll-up result is higher than the defined threshold (0.60) because *PL* and *AS* are weighted higher than *AL* and *NFS*.

It is for learners like Jack that our system offers a better adaptive e-learning experience. Jack has an average score of 82, which is almost as high as Nicole's, and has not failed in any child activity (denoted by *NFS* being true). Unfortunately, he cannot satisfy the data condition *PL* (Pass Limit). He would have failed if the correlations among the data conditions were not considered. The *rollUpBM* evaluates his result as 60, which meets the defined *rollUpBM*'s threshold (60), because the system not only considers the *PL* condition but also *PL*'s correlations with other data conditions as shown by the structure of *rollUpBM*. Although *PL* is weighted heavier than *AL* and *LV* is weighted heavier than *MV* as shown in *rollUpBM* of Figure 5-3, our system does not allow *PL* and *LV* to have absolute influence on the roll-up decision. Rather, it takes all of the data conditions and their correlations into consideration to determine that Jack has gained enough knowledge from the instruction given in the child activities and that he can skip the post assessment of the parent activity.

In our user case study, we found that the system can derive the prior distribution of a Bayesian Model based on limited inputs from the expert and the proposed formulas, and use the model to accurately evaluate new learners with different performances. As each new learner's data becomes available, it is used to update the prior distribution of

the Bayesian Model. Thus, the updated prior distribution becomes more and more accurate in representing the performance characteristics of learners. This accumulation of “group data” will improve the accuracy of evaluating the data of the next new learner and continuously improve the adaptive capability of the system.

In summary, we conduct four cases of simulations to evaluate our system’s adaptive capabilities. The first simulation successfully demonstrates GELS’ adaptive capability based on different learners' profile and performance data. The second simulation evaluates the effect of updating group profile and performance data, and shows that the system is able to improve its adaptive capability based on the updated data. The third simulation shows that GELS is able to make adaptation decisions in the presence of missing data as long as the percentage of missing data is reasonable. The last simulation indicates that the technique used to derive prior distributions of Bayesian Models and the idea of using Bayesian Models to aid adaptation decision-making are sound.

Table 8-4. Assessment results and average scores of the simulated learners

Child Activities / Average Score	Nicole	Eva	Michael	Jack	Adam	Steven	Kim
High Pressure System	86	74	60⇨74	100	86	60⇨74	74
Low Pressure System	100	81	46⇨46	100	42⇨42	69⇨81	54⇨69
Breathing Circuit	86	69⇨80	86	86	86	51⇨51	37⇨37
Manual Ventilation	81	58⇨54	42⇨46	69⇨69	46⇨54	42⇨31	31⇨27
Mechanical Ventilation	X	60⇨60	86	66⇨69	86	60⇨80	34⇨40
Scavenging System	X	69⇨81	69⇨81	69⇨69	58⇨69	27⇨42	42⇨54
Average Score	88	72	70	82	71	60	50

(Note) X: no assessment result, ⇨: retry, ☐: satisfied score, ▒: failed score.

Table 8-5. User case evaluation results

Input / Output	Nicole	Eva	Michael	Jack	Adam	Steven	Kim
PL	T	T	T	F	F	F	F
AL	T	F	F	F	F	F	F
NFS	T	T	F	T	F	T	F
AS	T	T	T	T	T	F	F
p(RU)	0.86	0.81	0.78	0.60	0.57	0.42	0.37

CHAPTER 9
SUMMARY, CONTRIBUTION, AND FUTURE WORK

9.1. Summary and Contribution

Among higher education schools, over 90% of four-year public institutions and 88.6% of private institutions offer some form of e-learning (Allen & Seaman, 2004). The advantage of Web-based e-learning systems is that they can deliver multimedia content to learners anywhere and anytime. In spite of the wide-spread popularity of e-learning education and its advantages, one significant limitation is that it provides learners with identical instructional materials without taking learners' diverse knowledge, goals, preferences, and performances into consideration. Learners are different from one another, and not all learners learn things in the same way (Schroeder, 1993). Thus, the "one-size-fits-all" service provided by many existing e-learning systems does not work well for diverse learners. For this reason, recent research efforts on e-learning focus on adaptive e-learning using learners' profiles and performance data as the basis for making adaptation decisions. However, developing an adaptive e-learning system is not a simple task. It requires that such a system be capable of collecting the useful information about each learner, evaluating the collected data correctly, and taking the proper adaptation actions to deliver individualized instruction to the learner.

This research aims to investigate 1) the methodology for the modeling, construction, search, and selection of learning objects, 2) the technique for the specification and processing of adaptation rules, 3) the use of Bayesian Models for specifying the correlations among data conditions given in adaptation rules and resolving the data uncertainty problem addressed in the introduction section, and 4) the method for deriving the probability distributions for Bayesian Models. The above

investigations and results are needed for the development of an adaptive e-learning system.

In this dissertation, we explained in Chapter 1 the motivation, the goal, and the intended contributions of our research. In Chapter 2, we surveyed the standards introduced by the e-learning community as well as the techniques and features of some developed adaptive e-learning systems and tools. We also discussed what we adopted from these existing works in the development of our system, GELS, and how our work differs from them. In Chapter 3, we presented the methodology for modeling and constructing learning objects using the multimedia learning assets available on the Web, and registering these objects with a Learning Object Broker for adaptive search and selection. In Chapter 4, we presented a new way of specifying and processing adaptation rules. We used the ECpAA rule specification instead of the traditional ECA rule specification to define adaptation rules so that these rules can be evaluated probabilistically instead of deterministically. We used Bayesian Models to capture the correlations and the relative weights among the data conditions used in adaptation rules as well as to represent the prior and conditional probabilities of these data conditions based on the profile and performance data of a group of learners. The Bayesian Models are used by GELS to evaluate the condition specification of each adaptation rule that is applied for each learner. In Chapter 5, we presented three formulas for deriving the prior distributions of Bayesian Models automatically by using limited inputs (i.e., prior probabilities and weights) provided by the author or instructor of a learning object. In Chapter 6, the architecture of the adaptive e-learning system GELS and the functions of its system components were described. The system was developed based on the

results of research presented in the earlier chapters. Chapter 7 covered two applications of a system, which integrates GELS with an existing VAM system, used for instructing medical personnel in the operation and use of anesthesia machine and FDA's pre-use check. This effort is a continuation of a previous project. We applied ECpAA rules in the processing of learning objects developed for these applications to demonstrate GELS' adaptive features. In Chapter 8, we used simulations to evaluate and demonstrate GELS' adaptive features and processing technique, and its ability to take proper adaptation actions to tailor instructions to suit diverse learners in the presence of data anomalies. The capability of automatically deriving the prior distributions of Bayesian Models and using them in different user-cases was also demonstrated.

Having summarized what we have presented in this dissertation, we now summarize the key contributions of our research below:

The modeling, construction, search, and selection of learning objects. In modeling learning objects, we have extended the activity tree proposed in SCORM for modeling a composite learning object by allowing non-leaf activities to have content, practice, and assessment items. This is important because learners can learn from and be assessed on the knowledge gained by integrating the lessons learned from the child activities of these non-leaf activities. We also extended the set of attributes proposed by LOM for specifying the metadata of learning objects by including "educational attributes" to describe the educational characteristics of learning objects. These attributes together with others can be used for searching and selecting learning objects that are suitable for learners. For the construction of Web-accessible and reusable learning objects, we

have designed and implemented authoring tools for the construction of ALOs and CLOs, and learning object repositories for their storage. These tools and repositories can be replicated and installed at many network sites for content producers and composers to develop and store ALOs and CLOs. The metadata of these objects are registered with the LO Broker installed at the host site. The learners of a virtual e-learning community as well as GELS can query the Broker to locate and fetch distributed learning objects that suit diverse learners' needs. This effort extends the previously developed LO authoring tools and repositories (Lee & Su, 2006). The added features are the inclusion of additional attributes to the metadata of LOM and the dynamic binding technique, which enables the linking of ALOs and other CLOs to the activity tree of a CLO to form a larger unit of instruction. This technique increases the scalability and reusability of learning objects.

ECpAA rule specification and processing. The first contribution of this effort is to identify problems in existing rule-based systems, such as the data uncertainty problem, the data correlation problem, and the problem of not utilizing group profile and performance data. Second, the probabilistic specification and evaluation of rule conditions are proposed to resolve these problems. The condition specification of an ECpAA rule and its evaluation are handled probabilistically based on the accumulated and continuously updated profile and performance data of a group of learners who have a common interest in a subject of learning. The probabilistic approach of the ECpAA rule specification is better suited for handling data anomalies than the deterministic approach used in the traditional event-condition-action rule specification because the former does not require that the condition specification of a rule be 100% true. The

processing of the Cp part of an ECpAA rule is aided by a Bayesian Model, which specifies the correlations of data conditions given in the condition specification and their relative weights, prior probabilities and conditional probabilities. Third, six adaptation points and their corresponding events are introduced. This allows more adaptation rules than those proposed by SCORM to be applied more frequently to tailor the instruction of each activity to fit the profile and performance of each learner. As a result, GELS is more adaptive than those systems implemented based on SCORM's standard. Lastly, six adaptation rules are presented. These rules are general and can be used to control the selection, navigation, sequencing, and presentation of contents during the processing of a learning object in the presence of data anomalies. The attributes that are suitable for defining the condition specifications of these rules have been identified and proposed. Content providers can use these rules as guides to define their own adaptation rules.

The design and implementation of the ECpAA Rule Execution Engine. This rule engine combines the processing power of an ETR Server developed in a previous project (Lee & Su, 2006) with the newly developed Bayesian Model Processor (BMP). The efforts and the contributions of this part of the work are the design and implementation of the Bayesian Model Processor and its integration with the ETR Server. The Bayesian Model Processor evaluates the condition specifications of ECpAA rules, computes the probability distributions of data attributes used to define Bayesian Models, and update their probability values. The use of probabilistic evaluation of the condition clause by using a Bayesian Model and the Bayesian Model Processor to achieve adaptation is new to the field of e-learning. The ECpAA Rule Engine, which

combines the capabilities of these two components, produces a rule engine that is more powerful than the traditional Event-Condition-Action rule engines because the latter's evaluation of the rule condition specification to true or false is a special case of the former's Cp evaluation.

Formulas for automatic computation prior distribution of Bayesian Models. The contributions of this effort are the introduction of the three formulas for computing the conditional probability tables and the design and implementation of a Bayesian Model Editor for easing the acquisition of the expert’s input data. Using a Bayesian Model requires the acquisition of its prior distribution (i.e., prior probabilities and conditional probabilities of data conditions given an adaptation rule) for Bayesian inference. However, obtaining an informative prior distribution is the most challenging task in building an adaptive e-learning system that uses Bayesian Networks. The three formulas that we introduced enable the system to automatically derive conditional probabilities based on the limited input of an expert (i.e., the prior probabilities assigned to the root nodes and the weights assigned to the edges). As each new learner’s profile and performance data become available, the system uses these data to update the prior distribution, thus improving the accuracy of evaluating the next new learner. GELS provides a user interface to allow the expert to easily provide the prior probabilities and weights associated with the data conditions to be evaluated. Using three proposed formulas and the expert’s inputs, the system can derive the prior distribution of each Bayesian Model and perform the Bayesian inference for probabilistic evaluation of adaptation rules

The handling of three data problems by using Bayesian Networks. Profile data provided by learners and performance data gathered by the system can be incomplete, inaccurate, and/or contradictory. These data problems are handled in our work by the use of Bayesian Models, which capture data correlations by their structures, the relative weights assigned to their structural relationships, and the probability distributions based on continuously updated group profile and performance data. Bayesian inferences based on these models to mitigate these data problems are demonstrated in this work. The approach is new and has not been used in other adaptive e-learning systems.

Implementation of two medical applications and development of simulations. We have used two applications and simulations to evaluate and demonstrate the adaptive features of the e-learning system GELS, which is developed based on the results of this research. The effort we have taken is NOT to demonstrate that adaptive e-learning improves learners' ability to learn, which is a subject well-researched by educators and people in other disciplines. Rather, it is to demonstrate that the system developed based on the methodology, techniques, formulas, rule specification method, Bayesian Models, and algorithms introduced in this work does exhibit desirable adaptive features. For this purpose, we have achieved our objectives.

9.2 Future Work

Two research topics are worthwhile for future investigation: the evaluation of the effectiveness/efficiency of an adaptive e-learning system and the further extension of the current Bayesian Models. After Russell's "Technology wars: Winners and losers" (1997), the effectiveness / efficiency of e-learning has become a controversial issue. In the past several years, several research results have been published. Valcheva and

Todorova (2005) present five general approaches for evaluating the effectiveness of e-learning systems that have appeared in the literature: comparison with traditional e-learning systems (i.e., systems without adaptive features); tools and components evaluation based on user satisfaction measurements; return on investment (ROI) reports; product evaluation by the software developer in perspective of design validation; and learning performance evaluation. Some of the approaches have been popularly used, such as comparison with traditional systems (Boyle & Encarnacion, 1994; Brusilovsky & Pesin, 1994; Höök 1997). Some literature suggests a modular or layered evaluation rather than evaluating a system as a whole in order to reduce the complexity of evaluation (Paramythis, Totter, & Stephanidis, 2001; Weibelzahl & Weber, 2003; Brusilovsky et al., 2004; Paramythis & Weibelzahl, 2005). In this evaluation procedure, they ‘identify’ a module or layer, ‘present’ the evaluation rationale for each module or layer, and ‘propose’ the specific methods and/or techniques that can be employed for its evaluation. Despite these efforts, the studies that address the evaluation of the effectiveness of e-learning systems have been criticized for either the small coverage with respect to the multiple aspects of the complex problem or the inadequacy of the evaluation criteria (Martinez, 2001; Costabile, Roselli, Lanzilotti, Ardito, & Rossano, 2007). The evaluation of the effectiveness of e-learning systems is a very challenging and complex topic, because many factors involved in the evaluation are intertwined, and neither established guidelines nor a methodological paradigm are available (Lanzilotti, Ardito, Costabile, & De Angeli, 2006). In our work, we did not conduct an evaluation on the effectiveness of our developed system in improving learners' ability to learn for the stated reasons. Rather, we are only interested in

verifying that GELS possesses some desirable adaptive features by implementing the results of our research.

Another worthwhile research topic is building more complex Bayesian Models than the models we have. This can be done by assigning multiple values rather than a binary value (True or False) to each data condition, or employing temporal models (e.g., dynamic Bayesian Networks or Hidden Markov Models) in conjunction with the developed Bayesian Models. The former allows the system to further improve the accuracy of data evaluation, while the latter has the potential to alter the data conditions of a Bayesian Model, which may lead to generating dynamic adaptation rules during a learning process. This topic is important because the data conditions about the learners in adaptation rules can be very complex and can change in time.

LIST OF REFERENCES

Advanced Distributed Learning (2004). *Sharable Content Object Reference Model (SCORM) 2004 4th Edition.* Retrieved from http://www.adlnet.gov/Technologies/scorm/Lists/SCORM%20Navigation%20Submenu/AllItems.aspx

Allen, I. E. & Seaman, J. (2004). *Entering the Mainstream: the Quality and Extent of Online Education in the United States, 2003 and 2004*. Retrieved from SLOAN: A Consortium of Institutions and Organizations Committed to Quality Online Education website: www.sloan-c.org/resources/entering_mainstream.pdf.

Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE). (2006). *ARIADNE Foundation for the European Knowledge Pool*. Retrieved from http://www.ariadne-eu.org/index.php

Anderson, J. R., Corbett, A. T., Koedinger, K. R, & Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *The Journal of the Learning Sciences*, *4* (2), 167−207.

Arroyo, I. & Woolf, B. (2005). Inferring Learning and Attitudes from a Bayesian Network of Log File Data. In C. K. Looie, G. McCalla, B. Bredeweg, & J. Breuker (Eds.), *Proceedings of the 12th International Conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology* (pp. 33−40). Amsterdam: IOS Press.

Beaumont, I. (1994). User modeling in the interactive anatomy tutoring system ANATOM-TUTOR. *User Modeling and User-Adapted Interaction, 4* (1), 21−45.

Benford, S.D., Taylor, I., Brailsford, D., Koleva, B., Craven, M., Fraser, M., Reynard, G. T., & Greenhalgh, C. M. (2000). Three-dimensional Visualisation of the World Wide Web. *ACM Computing Surveys Symposium on Hypertext*.

Botsios, S. & Georgiou, D. (2009, November). Using Standards for Adaptive Learning Objects Retrieval. *ADIS Conference* Rome, Italy. Retrieved from http://utopia.duth.gr/~dgeorg/PUBLICATIONS/58.pdf

Boyle. C., & Encarnacion, A. O. (1994). Metadoc: An Adaptive Hypertext Reading System. *User Modeling and User-Adapted Interaction*, *4*, 1−19.

Brusilovsky, P., (1996). Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction, 6,* (pp. 87−129). Kluwer academic publishers.

Brusilovsky, P. & Pesin, L. (1998). Adaptive Navigation Support in Educational Hypermedia: An Evaluation of the ISIS-tutor. *Journal of Computing and Information Technology, 6* (1), 27−38.

Brusilovsky, P., Eklund, J., & Schwarz, E. (1998). Web-based Education for All: A Tool for Developing Adaptive Courseware. *Computer Networks and ISDN Systems (Seventh International World Wide Web Conference)*, *30* (1-7), 291−300.

Brusilovsky, P., (1999). Adaptive and Intelligent Technologies for Web-based Education. In C. Rollinger & C. Peylo (Eds.), *Künstliche Intelligenz*, *4*, Special Issue on Intelligent Systems and Teleteaching, 19−25.

Brusilovsky, P., (2001). Adaptive hypermedia. In A. Kobsa (Ed.), *User Modeling and User Adapted Interaction*, *Ten Year Anniversary Issue, 11*(1-2), 87−110.

Brusilovsky, P. & Maybury, M. T. (2002). From Adaptive Hypermedia to the Adaptive Web. *Communications of the ACM, 45*(5), 30−33.

Brusilovsky, P. & Peylo, C. (2003). Adaptive and intelligent Web-based educational systems. *International Journal of Artificial Intelligence in Education 13* (2-4), *Special Issue on Adaptive and Intelligent Web-based Educational Systems*, 159−172*.*

Brusilovsky, P. & Vassileva, J. (2003). Course sequencing techniques for large-scale web-based education. *International Journal of Continuing Engineering Education and Lifelong Learning, 13* (1-2), 75−94.

Brusilovsky, P., Karagiannidis, C., & Sampson, D. (2004). Layered evaluation of adaptive learning systems. *International Journal of Continuing Engineering Education and Lifelong Learning*, *14*(4/5), 402−421.

Brusilovsky, P., Sosnovsky, S., & Shcherbinina, O. (2004). QuizGuide: Increasing the Educational Value of Individualized Self-Assessment Quizzes with Adaptive Navigation Support. In J. Nall & R. Robson (Eds.) *Proceedings of E-Learn 2004*, (pp. 1806−1813). Washington, DC, USA: AACE

Brusilovsky, P. & Nejdl, W. (2005) Adaptive Hypermedia and Adaptive Web. In: M. P. Singh (Ed.) *Chapman & Hall/CRC Computer and Information Science Series: The Practical Handbook of Internet Computing* (pp. 1.1−1.14). Baton Rouge: Chapman & Hall/ CRC Press.

Butz, C. J., Hua, S., & Maguire, R. B. (2008). Web-Based Bayesian Intelligent Tutoring Systems. *Evolution of the Web in Artificial Intelligence Environments*. *Studies in Computational Intelligence, 130*, 221−242. doi: 10.1007/978-3-540-79140-9_10.

Cisco Systems. (1999). *Reusable Information Object Strategy: Definition, Creation Overview, and Guidelines*. Retrieved from http://www.cisco.com/warp/public/779/ibs/solutions/learning/whitepapers/el_cisco_rio.pdf

Cisco Systems. (2003). *Reusable Learning Object Strategy: Designing and Developing Learning Objects for Multiple Learning Approaches*. Retrieved from http://www.e-novalia.com/materiales/RLOW__07_03.pdf

Conati, C., Gertner, A., & Vanlehn, K. (2002, November). Using Bayesian Networks to Manage Uncertainty in Student Modeling. *User Modeling and User-Adapted Interaction, 12* (4), 371−417.

Constantino-Gonzalez, M. A., Suthers, D. D., & Escamilla De Los Santos, J. G. (2003). Coaching web-based collaborative learning based on problem solution differences and participation. *International Journal of Artificial Intelligence in Education*, *13* (2-4), 261−297

Costabile, M. F., Roselli, T., Lanzilotti, R., Ardito, C., & Rossano, V. (2007). A Holistic Approach to the Evaluation of E-Learning Systems. *Universal Access in Human-Computer Interaction, Applications and Services: Lecture Nodes in Computer Science, 4556*, 530−538. doi: 10.1007/978-3-540-73283-9_59.

Dalley, P., Robinson, B., Weller, J. & Caldwell, C. (2004) The user of high-fidelity human patient simulation and the introduction of new anaesthesia delivery systems. *Anaestheia and Analgesia*, *99*, 1737−1741.

De Bra, P., & Calvi, L. (1998). Towards a Generic Adaptive Hypermedia System. *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia* (pp. 5−11), Pittsburgh, PA.

De Bra, P., Brusilovsky, P., & Houben, G-J. (1999, December). Adaptive hypermedia: from systems to framework. *ACM Computing Surveys, 31*(4).

De Bra, P., Stash, N., & De Lange, B. (2003, May). AHA! Adding Adaptive Behavior to Websites. *Proceedings of the NLUUG 2003 Conference*, Ede, The Netherlands.

Degwekar, S., Su, S. Y. W., & Lam, H. (2004). Constraint specification and processing in Web services publication and discovery. In *Proceedings of the International Conference on Web Services*, San Diego, California. Eduworks, & O. K. I. Leadership

Desmarais, M. C., Maluf, A., & Liu, J. (1995). User-expertise modeling with empirically derived probabilistic implication networks. *User Modeling and User-Adapted Interaction, 5* (3-4), 283−315.

Druzdzel, M. J. & van der Gaag, L. C. (1995). Elicitation of Probabilities for Belief Networks: Combining Qualitative and Quantitative Information. *In Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-95),* 141−148.

Druzdzel, M. J. & van der Gaag, L. C. (2000). Building Probabilistic Networks: Where Do the Numbers Come From? Guest Editors' Introduction. *IEEE Transactions on Knowledge and Data Engineering, 12*(4), 481−486.

Dublin Core Metadata Initiative (DCMI). (2006).*DCMI Metadata Terms.* Retrieved from http://dublincore.org/documents/2006/12/18/dcmi-terms/ /

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification (2nd ed.).* New York: John Wiley & Sons.

Duitama, F., Defude, B., Bouzeghoub, A., & Lecocq, C. (2005). A Framework for the Generation of Adaptive Courses Based on Semantic Metadata. *Multimedia Tools and Applications, 25*(3), 377−390.

Gamboa, H. & Fred, A. (2001, July). Designing Intelligent Tutoring Systems: A Bayesian Approach. *3rd International Conference on Enterprise Information Systems (ICEIS 2001)*, Setubal, Portugal.

García, P., Amandi, A., Schiaffino, S., & Campo, M. (2007). Evaluating Bayesian networks' precision for detecting students' learning styles. *Computers & Education, 49*(3), 794−808.

Gelman, A. (2002). Prior distribution, In *Encyclopedia of Environmetrics,* (Vol.3, pp. 1634−1637). Chichester: John Wiley & Sons, Ltd.

Gertner, A. & VanLehn, K. (2000). Andes: A Coached Problem Solving Environment for Physics. *Intelligent Tutoring Systems: Lecture Notes in Computer Science, 1839*, 133−142. doi: 10.1007/3-540-45108-0_17.

Gonzalez, A. J., & Dankel, D. D. (1993). *The Engineering of Knowledge-Based Systems: Theory and Practic*e. NJ: Prentice-Hall.

Greer, J., McCalla, G., Collins, J., Kumar, V., Meagher, P., & Vassileva, J. (1998). Supporting Peer Help and Collaboration in Distributed Workplace Environments. *International Journal of Artificial Intelligence in Education*, *9*, 159−177.

Guzmán, E., Conejo, R., & Pérez-de-la-Cruz, J. L. (2007, March). Adaptive testing for hierarchical student models. *User Modeling and User-Adapted Interaction*, *17* (1-2).

Harrer, A., McLaren B.M., Walker, E., Bollen, L., & Sewall, J. (2006, September). Creating cognitive tutors for collaborative learning: steps toward realization. *User Modeling and User-Adapted Interaction*, *6* (3-4).

Heckerman, D. (1995). *A Tutorial on Learning With Bayesian Networks.* Retrieved from Technical Report MSR-TR-95-06, Microsoft Research Advanced Technology Division website: ftp://ftp.research.microsoft.com/pub/tr/tr-95-06.pdf

Henze, N. & Nejdl, W. (2004). A Logical Characterization of Adaptive Educational Hypermedia. *New Review on Hypertext and Hypermedia (NRHM), 10* (1), 77−113.

Hohl, H., Böcker, H-D., & Gunzenhäuser, R. (1996). HYPADAPTER: An Adaptive Hypertext System for Exploratory Learning and Programming. *User Modeling and User-Adapted Interaction*, *6*(2-3), 131−155.

Höök, K. (1997) Evaluating the Utility and Usability of an Adaptive Hypermedia System. In: J. Moore, E. Edmonds, & A. Puerta (Eds.), *Proceedings of 1997 International Conference on Intelligent User Interfaces*, ACM, Orlando, Florida.

IEEE LTSC. (2001). *IEEE LTSC 1484.2 -Standard for Information Technology-Public and Private Information (PAPI) for Learners Specification version 6 (PAPI).* Retrieved from http://www.cen-ltso.net/Main.aspx?put=230

IEEE LTSC. (2002). *IEEE LTSC1484.12.1-Draft Standard for Learning Object Metadata*. Retrieved from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

IMS Global Learning Consortium. (2003). *IMS Simple Sequencing Best Practice and Implementation Guide Version 1.0 Final Specification*, Retrieved from http://www.imsglobal.org/simplesequencing/ssv1p0/imsss_bestv1p0.html

IMS Global Learning Consortium (2005). *Learning Resource Meta-data Specification Version 1.3.* Retrieved from http://www.imsglobal.org/metadata/

IMS Global Learning Consortium. (2010). *IMS Learner Information Package*. Retrieved from http://www.imsglobal.org/profiles/

Jaynes, E.T. (1989, September). Prior Probabilities, *IEEE Transactions on Systems Science and Cybernetics*, *4,* 227-241. Reprinted by Rosenkrantz, R. D. (1989). *E.T.Jaynes: Papers on probability, statistics, and statistical physics.* Dordrecht, Boston: Kluwer Academic, 116−130.

Jeon, S. S., Lee, G., Lampotang, S., & Su, S. Y. W. (2007a). An integrated e-learning system for simulation based instruction of anaesthesia machine. *International Journal of Knowledge and Learning (IJKL)*, *3*(1), 106−120.

Jeon, S. S., Su, S. Y. W, & Lee, G. (2007b). Achieving Adaptive E-Learning Using Condition-Action Rules and Bayesian Networks. *Proceedings of the IADIS International Conference, e-LEARNING 2007*, 359−366.

Kass, R. E. &Wasserman, L. (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association, 91* (435), 1343−1370.

Lampotang, S., Lizdas, D. E., Gravenstein, N., & Liem, E.B. (2006). Transparent reality, a simulation based on interactive dynamic graphical models emphasizing visualization. *Educational Technology*, *46*(1), 55−59.

Lanzilotti, R., Ardito, C., & Costabile, M. F., & De Angeli, A. (2006). eLSE Methodology: a Systematic Approach to the e-Learning Systems Evaluation. *Educational Technology & Society*, *9* (4), 42−53.

Lee, G. & Su, S. Y. W. (2006). Learning Object Models and an E-learning Service Infrastructure. *International Journal of Distance Education Technology*, *2*(1), 1-16.

Lee, G, Zhang, X., & Su, S. Y. W. (2004). Learning object models, authoring tools, and learning object registry service for achieving dynamic and collaborative elearning. In *Proceedings of the 7th IASTED International Conference on Computers and Advanced Technology in Education 2004* (pp. 321-327), Kauai, Hawaii.

Martin, J. & VanLehn, K. (1995). A Bayesian Approach to Cognitive Assessment*,* In: P. Nichols, S. Chipman, & R. L. Brennan (Eds.), *COGNITIVELY DIAGNOSTIC ASSESSMENT* (pp. 141−165). Hillsdale, NJ: Erlbaum.

Martinez, M. (2001). Key Design Considerations for Personalized Learning on the Web, *Educational Technology & Society, 4*(1), 26−40

Mazza, R., & Dimitrova, V. (2004). Visualising Student Tracking Data to Support Instructors in Web-Based Distance Education. *Proceedings 12th International World Wide Web Conference on Alternate Track Papers & Posters* (pp. 154−161), New York

METS. (2010). *Metadata Encoding & Transmission Standard*. Retrieved from http://www.loc.gov/standards/mets/

Mislevy, R. J., Senturk, E., Almond, R., Dibello, L., Jenkins, F., Steinberg, L., & Yan, D. (2001). *Modeling Conditional Probabilities in Complex Educational Assessments*. CSE Technical Reoprt 580. The National Center for Research on Evaluation, Standards, Student Testing (CRESST). Center for Studies in Education, Los Angeles: University of California. Retrieved from http://www.education.umd.edu/EDMS/mislevy/papers/ConditionalProbs.pdf

Mitrovic, A. (2003). An Intelligent SQL Tutor on the Web. *International Journal of Artificial Intelligence in Education,* 13(2/4), 171−195.

Mitsuhara, H., Ochi, Y., Kanenishi, K., & Yano, Y. (2002, May). An adaptive Web-based learning system with a free-hyperlink environment. In P. Brusilovsky, N. Henze, & E. Millán (Eds.), *Proceedings of Workshop on Adaptive Systems for Web-Based Education at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH'2002,* (pp. 81−91), Málaga, Spain.

Murphy, K. P. (2007). *The Bayes Net ToolBox for Matlab*. Retrieved from http://code.google.com/p/bnt/

Murray, T. (2003). MetaLinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks. *International Journal of Artificial Intelligence in Education*, *13*(2-4), 197−231.

Neal, R. M. (2001). *Defining Priors for Distributions Using Dirichlet Diffusion Trees*. Technical Report No. 0104, Retrieved from Department of Statistics, University of Toronto website: http://www.cs.utoronto.ca/~radford/ftp/dft-paper1.pdf

OASIS, (2005). *Universal Description, Discovery, and Integration (UDDI) version 3*. Retrieved from http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm#uddiv3

Oda, T., Satoh, H., & Watanabe, S. (1998, August). Searching deadlocked Web learners by measuring similarity of learning activities. *Proceedings of Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems (ITS'98),* San Antonio, TX.

Oracle. (n.d.). *Java Architecture for XML Binding (JAXB).* Retrieved from http://www.oracle.com/technetwork/articles/javase/index-140168.html

Paramythis, A., Totter, A., & Stephanidis, C. (2001, July). A modular approach to the evaluation of Adaptive User Interfaces. In S. Weibelzahl, D. Chin & G. Weber (Eds.), *Proceedings of the Workshop on Empirical Evaluations of Adaptive Systems, held in the context of the 8th International Conference on User Modeling (UM'2001),* (pp.9−24), Sonthofen, Germany.

Paramythis, A. & Loidl-Reisinger, S. (2004). Adaptive Learning Environments and e-Learning Standards. *Electronic Journal of e-Learning, 2*(1), 181−194

Paramythis, A., & Weibelzahl, S. (2005, July). A Decomposition Model for the Layered Evaluation of Interactive Adaptive Systems. In L. Ardissono, P. Brna, & A. Mitrovic (Eds.), *Proceedings of the 10th International Conference on User Modeling (UM2005)*, (pp. 438−442), Edinburgh, Scotland, UK.

Parunak, H.V.D. (1990, January), Reference and Data Model Group (RDMG): Work Plan Status, *Proceedings of the Hypertext Standardization Workshop*. (pp. 9−13) Gaithersburg, Maryland.

Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. CA: Morgan Kaufmann Publishers.

Ramsden, P. (2005). The context of learning in academic departments. In F. Marton, D. Hounsell & N. Entwistle (Eds.), *The experience of learning: Implications for teaching and studying in higher education (3rd (Internet) ed.)* (pp.198−216), Edinburgh: University of Edinburgh, Centre for Teaching, Learning and Assessment. Retrieved from http://www.tla.ed.ac.uk/resources/EoL.html.

Rey-Lopez, M., Fernadez-Vilas, A., Diaz-Redondo, R. P., Pazos-Arias, J. J., & Bermejo-Munoz, J. (2006). Extending SCORM to Create Adaptive Courses. *Lecture Notes in Computer Science, 4227,* 679-684. doi: 10.1007/11876663_72

Rios, A., Pérez-de-la-Cruz, J.L., & Conejo, R. (1998). SIETTE: Intelligent Evaluation System using Test for TeleEducation. *4th International Conference on Intelligent Tutoring System. ITS'98. Workshops papers*, San Antonio, Texas, USA.

Ritter, S., Anderson, J., Cytrynowitz, M., & Medvedeva, O. (1998). Authoring Content in the PAT Algebra Tutor. *Journal of Interactive Media in Education*, 98 (9).

Rumetshofer, H., & Wolfram, W. (2003). XML-based adaptation framework for psychological driven E-learning systems. *Educational Technology and Society,* 6(4), 18−29.

Russell, S. & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach, 2nd edition.* NJ: Prentice-Hall.

Russell, T. (1997). Technology wars: Winners and losers. *Educom Review, 32*(2), 44−46.

Schroeder, C. C. (1993). *New Students-New Learning Styles*, Retrieved from http://www.virtualschool.edu/mon/Academia/KierseyLearningStyles.html.

Sison, R., & Simura, M., (1998). Student modeling and machine learning. *International Journal of Artificial Intelligence in Education*, 9, 128−158.

Sleeman, D. H, & Brown, J. S. (1982). Introduction: Intelligent Tutoring Systems. In D. H. Sleeman & J. S. Brown (Eds.), *Intelligent Tutoring Systems*. (pp.1−11). New York: Academic Press.

Smith, A. S. G. & Blandford, A. (2002). MLTutor: An Application of Machine Learning Algorithms for an Adaptive Web-based Information System. *International Journal of Artificial Intelligence in Education, 3*(2/4), 233−260.

Soller, A., & Lesgold, A. (2003). A computational approach to analysing online knowledge sharing interaction. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *AI-ED'2003* (pp. 253−260). Amsterdam: IOS Press.

The Apache XML Project (Apache) (2007) *Apache Xindice*. Retrieved from http://xml.apache.org/xindice/

Tzanavari, A., Retalis, S., & Pastellis, P. (2004). Giving More Adaptation Flexibility to Authors of Adaptive Assessments. *Adaptive Hypermedia and Adaptive Web-based Systems: Lecture Notes in Computer Science, LNCS 3137*. 549−568. doi: 10.1007/978-3-540-27780-4_47.

Valcheva, D. & Todorova, M. (2005). Defining a system of indicators for evaluation the effectiveness of e-learning. *International Conference on Computer Systems and Technologies - CompSysTech'2005*.

VanLehn, K. & Niu, Z. (2001). Bayesian Student Modeling, User Interfaces and Feedback: A Sensitivity Analysis. *International Journal of Artificial Intelligence in Education, 12*, 154-184.

Van Rosmalen, P., Vogten, H., Van Es, R., Passier, H., Poelmans, P., & Koper, R. (2006). Authoring a full life cycle model in standards-based, adaptive e-learning. *Educational Technology & Society*, *9(1),* 72−83.

W3C. (2007). *Simple Object Access Protocol (SOAP) 1.2*, Retrieved from http://www.w3.org/TR/SOAP

W3C. (2001). *Web Services Description Language (WSDL) 1.1*, Retrieved from http://www.w3.org/TR/wsdl

Weber, G. & Brusilovsky, P. (2001). ELM-ART: An Adaptive Versatile System for Web-based Instruction. *International Journal of Artificial Intelligence in Education*, *12*, 351−384.

Weber, G., Kuhl, H. C., & Weibelzahl, S. (2001). Developing adaptive internet based courses with the authoring system NetCoach. *Lecture Notes in Computer Science LNAI 2266*, 226−238. London: Springer-Verlag.

Weibelzahl, S. & Weber, G. (2003). Evaluating the inference mechanism of adaptive learning systems. *User Modeling 2003: Lecture Nodes in Computer Science, 2702.* 148. doi: 10.1007/3-540-44963-9_21

Weinger, M. (1999) Anaesthesia equipment and human error. *Journal of Clinical Monitoring*, *15*, 319−323.

Xenos, M. (2004). Prediction and assessment of student behaviour in open and distance education in computers using Bayesian networks. *Computers and Education*, *43*, 345−359.

BIOGRAPHICAL SKETCH

Originally from Seoul, Korea, Sanghyun Seo Jeon received a Bachelor of Arts. and an Mater of Arts in English literature and language from Korea University, Seoul, Korea in 1986 and from Ewha Womans University, Seoul, Korea in 1988, respectively. Her thesis was on “Study of the Early Poems of Emily Dickinson”. She also received a Mater of Science in Computer and information science and engineering from the University of Florida, United States of America in 2006.

Her current Ph.D. dissertation research is on the development of the adaptive e-Learning system using rules and Bayesian Networks on top of the existing e-Learning standards, such as IMS Global Learning Consortium, ADL SCORM (Advanced Distributed Learning Sharable Content Object Reference Model), PAPI (Public And Private Information), and AICC (Aviation Industry CBT Committee). More specifically, she focuses on handling data uncertainties within the learner profile and performance information. More generally, her interests lie in the development of personalized learning environments including adaptive e-Learning management system, adaptive collaboration supports, rule-based e-Learning system, standards-based e-Learning system, adaptive content management and delivery system, probabilistic e-Learning system, and intelligent tutoring system.

She was an English researcher at Daekyo Co. in Korea and developed the English educational contents for toddlers through high school students. She was also an English teacher for young kids through adults while she was in Korea. She is currently working at Academic Technology, University of Florida, USA. Her current project is Sakai development and management, which is a new e-Learning project in the University of Florida.