# Intelligent Technologies in e-Learning

Four core intelligent technologies in e-Learning are: curriculum sequencing, intelligent analysis of student's solutions, interactive problem solving support, and example-based problem solving support [Peter Brusilovsky].

1. ***Curriculum Sequencing***

The goal of the *curriculum sequencing* technology (also referred to as instructional planning technology) is to provide the student with the most suitable individually planned sequence of knowledge units to learn and sequence of learning tasks (examples, questions, problems, etc.) to work with. In other words, it helps the student to find an "optimal path" through the learning material. There are two essentially different kinds of sequencing are active and passive.

* ***Active sequencing*** implies a *learning goal* (a subset of domain concepts or topics to be mastered). Systems with active sequencing can build the best individual path to achieve the goal.
* ***Passive sequencing*** (which is also called *remediation*) is a reactive technology and does not require an active learning goal. It starts when the user is not able to solve a problem or answer a question (questions) correctly. Its goal is to offer the user a subset of available learning material, which can fill the gap in student's knowledge of resolve a misconception.

For active sequencing systems, it makes sense to distinguish systems with fixed and adjustable learning goal. Most of existing systems can guide their students to the fixed learning goal - the whole set of domain concepts. A few systems with adjustable learning goal let a teacher or a student to select a subset of the whole set of concepts as the current learning goal.

In most of e-Learning systems with sequencing it is possible to distinguish two levels of sequencing are high and low.

* ***High-level sequencing***or***knowledge sequencing****:* determines next learning sub goal: next concept, set of concepts, topic, or lesson to be taught.
* ***Low-level sequencing*** or ***task sequencing****:* determines next learning task (problem, example, and test) within current sub goal.

High and low level sequencing are often performed by different mechanisms. In many intelligent e-Learning systems only one of these two mechanisms are intelligent, for example, a lesson is selected by a student, while learning tasks within this lesson are adaptively selected by the system. Some systems can only manipulate the order of tasks of one particular kind: usually problems or questions. In this case it could be also called problem or question sequencing.

Active sequencing in most is driven by the students knowledge (more exactly, by the difference between student's knowledge and global goal). A few systems and projects, however, experiment with the use of students’ preferences on the type and media of available learning material to drive sequencing of tasks within a topic [14; 15; 45].

1. ***Problem Solving Support technologies***

For many years, problem solving support was considered as a main duty of an ITS system and a main value of an ITS technology. We have identified three problem solving support technologies: intelligent analysis of student solutions, interactive problem solving support, and example-based problem solving support. All these technologies can help a student in a process of solving an educational problem, but they do it by different ways.

* ***Intelligent analysis of student solutions***deals with students' final answers to educational problems no matter how these answers were obtained. To be considered as intelligent, a solution, analyzer has to decide whether the solution is correct or not, find out what exactly is wrong or incomplete, and possibly identify which missing or incorrect knowledge may be responsible for the error (the last functionality is referred as knowledge diagnosis). Intelligent analyzers can provide the student with extensive error feedback and update the student model.
* ***Interactive problem solving support***is a more recent and a move powerful technology. Instead of waiting for the final solution, this technology can provide a student with intelligent help on each step of problem solving. The level of help can vary: from signaling about a wrong step, to giving a hint, to executing the next step for the student. The systems which implement this technology (often referred to as *interactive tutors*) can watch the actions of the student, understand them, and use this understanding to provide help and to update the student model.
* ***The example-based problem solving***technology is the newest one. This technology is helping students to solve new problems not by articulating their errors, but by suggesting them relevant successful problem solving cases from their earlier experience (it could be examples explained to them or problems solved by them earlier).

This paper presents a proposed model of utilizing different intelligent technologies in a proposed e-Learning components model. Components model lists the components that author believe are necessary for any e-Learning System. Proposed e-Learning components model is a service based model; as a result of the advantages of utilizing such services based model in overcoming system’s and software architecture challenges. Based on the previous classification, the two intelligent technologies; curriculum sequencing, and problem solving support can be utilized. Figure 1 presents a proposed utilization of the intelligent features. Services marked with a black circle at the upper right corner represents proposed areas of utilizing curriculum sequencing, while services marked with a black triangle at the lower right bottom highlights services proposed to utilize one or more types of intelligent problem solving intelligent techniques. Proposed e-Learning service based model components as shown in Figure 1 can be categorized into three categories:

* ***Component that Deal with Instructor:*** this component include services that instructors access to manage educational functionalities and capabilities in the learning process. Mainly, to create and edit learning materials, define learning domains, pedagogical features, manage educational situations, and other educational activities. This component includes two main services: Authoring Tools and Learning Tools. In proposed model, this component is presented by external applications that provide capabilities that are out of scope for authors. Examples of those applications are: Adobe Captivate.
* ***Components that Deal with Students:*** this component include services that students access to achieve Personalized learning, and services that students access to fulfill other learning activities.
  + ***Personalized Learning Component:*** It acts as the interface between Student and proposed e-Learning model, and this is where the proposed e-Learning model will address Intelligent Curriculum Sequencing technologies. Services include: Student Model, and LOs Recommender.
  + ***Educational Services Component:*** Include services that fulfill learning process requirements. Services include: Exam, Collaboration, Learning, and Assessment. This component includes services that will address the intelligent problem solving technologies in the learning process; mainly Learning, and Assessment services. Collaborative services are mentioned in [Book Chapter].
* ***Component that Doesn’t Deal directly with users, but deals with other Components:*** this component is important in supporting and defining the other components’ services’ requirements and functions. It includes services to manage pedagogical and define domain aspects of LOs and assistance needed for students during problem solving scenarios. This component services include: Domain, and Pedagogy services.

**Figure 1: Different Intelligent Techniques Utilization in Proposed e-Learning Model**

# Intelligent LO’s Recommender in e-Learning

## Introduction

Recommender system in an e-Learning context is a software agent that tries to “intelligently” recommend actions to a learner based on the actions of previous learners. This recommendation could be an online activity such as doing an exercise, reading posted messages on a conferencing system, or running an on-line simulation, or could be simply a web resource [Osmar R. Zaiane, 2002]. This paper presents different scenarios for different intelligent techniques utilization in Intelligent LOs Recommender as a Service. Figure 2 illustrates the general proposed model of LOs Recommender as a Service.



Figure 2: Intelligent Recommender Process Model

What distinguishes the data that result from web-based activities in general and e-Learning activities in particular, is the sheer complexity of the information and the vast size of the data collected, as well as the fact that simple information extraction is not possible. Information must be deduced by interactive data mining and associated visualization techniques. However, it is not clear how to analyze and visualize web usage data involving long sequence of on-line activities without losing the big picture [Osmar R. Zaiane, 2002].

## First Scenario: Fuzzy Logic and Co-Occurrence Matrix based Intelligent LOs Recommender

First Scenario of Intelligent LOs Recommender as a Service is the one that makes use of different technologies and techniques presented by A. M. Riad et al. (A. M. Riad et al. (c)) in utilizing Co-Occurrence Matrix based recommender to present suitable learning materials to students. Modifications to previous recommender were required to fit the new recommendation scenario. LOs Recommender as a Service relies heavily on LO’s SCORM Meta Data. Figure 3 presents an overview of LOs Recommender as a Service recommendation process. Student; the main initiative of the learning process asks for suitable LOs that fits the learning objectives of the course topics. S/he finally gets three different lists of LOs: Ranked LOs list, list of LOs from Google Scholar, and an Instructor Recommended List.



Figure 3: Intelligent LO's Recommender as a Service Model Utilizing Fuzzy Logic and Co-Occurrence Matrix Intelligent Techniques

Student request can be identified either implicitly through LOs Recommender as a Service observation of students’ navigation through learning topics or explicitly through an ordinary search using Proposed Adaptive System Web site’s interface. On receiving a new request, LOs Recommender as a Service generates a set of recommendations; a list of ranked suggested LOs to view based on content similarity with student request side by side with results from Google Scholar.

At each request, LOs Recommender as a Service identifies the requested LO, Session ID, Student ID, and Referrer LO; that is the LO the student was viewing before the requested one. Based on students’ access behavior, Recommender as a Service updates the underlying Co-Occurrence Matrix as depicted in figure 4. Co-Occurrence Matrix is the mind of the Proposed LOs Recommender as a Service and it is reactive because it reflexes students’ access to LOs as it updates weight between LOs based on students’ navigation model between LOs. Co-Occurrence Matrix is a Matrix

M = N \* N

Where

N = No. of LOs in LCMS

Each element in the Co-Occurrence Matrix presents a Weight W(u,v). The equation used to update Co-Occurrence Matrix values is

W(u,v) = N(u,v) / Max (Nu,Nv)

Where

W(u,v) = Weight of adjacent LOs (u,v)

N(u,v) = Number of Students’ Sessions in which both LO(u) and LO(v) are visited

N(u) = Number of Sessions in which LO(u) is visited

N(v) = Number of Sessions in which LO(v) is visited



**Figure 4: Update Co-Occurrence Matrix Steps**

Intelligent LOs Recommender as a Service relies on a Co-Occurrence Matrix that is stored in the database between different LOs. Building this Co-Occurrence Matrix is a student reactive process because it is mainly based on student navigation model between LOs. LOs Recommender as a Service gathers data required to recommend LOs to students from different data sources, and then builds Un-Directed Weighted Graph between the appropriate LOs based on the stored Co-Occurrence Matrix. The Clustering Algorithm is applied to this Un-Directed Weighted Graph to define different clusters and Generate the Recommended List of LOs. This list is then ranked based on the Ranking Algorithm and returned to the user.

* 1. **Building Un-Directed Weighted Graph**

The Un-Directed Weighted Graph is built programmatically based on the equation:

G = (V, E)

Where

V = Set of Vertices; that contains LOs Identifiers

E = Set of Weighted Edges between Vertices

An example of the Un-Directed Weighted Graph between LOs is the one presented in figure 5. Un-Directed Weighted Graph doesn’t highlight the order of viewing LOs; because this ordering will complicate the Clustering process.



Figure 5: Example of Un-Directed Weighted Graph between LOs

* 1. **Applying Clustering Algorithm**

LOs Recommender as a Service finds clusters based on Usage Data Analysis by Partitioning the built Un-Directed Graph to its Connected Components, assigning each component; that would be an LO into a different cluster based on a threshold value that is used to exclude poor correlated edges. The List of Recommended LOs contains LOs that coexist within the same cluster of the LOs of the requested learning topics. Proposed LOs Recommender as a Service utilized clustering algorithm is Depth First Search (DFS), as depicted in figure 6. DFS is an algorithm for traversing or searching a tree, tree structure, or graph, and it is used to find graph’s connected components. Threshold value shall be used to eliminate poor correlated edges. A start value of 0.4 is used in the beginning and it is a matter of change for later uses based on LOs Recommender performance.



Figure 6: Flow Chart of Implemented Depth First Search Algorithm

## Intelligent LOs Recommender as a Service based on Neural Networks Utilization

**Figure 7: Intelligent LO’s Recommender as a Service Model Utilizing Neural Networks**

## References

**Peter Brusilovsky,** “Adaptive and Intelligent Technologies for Web-based Education”

**Osmar R. Zaiane,** *“Building a Recommender Agent for e-Learning Systems”*, Proceedings of the International Conference on Computers in Education (ICCE’02), IEEE 200