Introduction:
Thousands of Web-based courses and other educational applications have been made available on the Web. The problem is that most of them are nothing more than a network of static hypertext pages.
A challenging research goal is the development of advanced Web-based educational applications that can offer some amount of adaptivity and intelligence.
The goal of this paper is to provide a brief review of the work performed so far in this area. The review is centered on different adaptive and intelligent technologies.

Web-based educational systems: a review of technologies:
AIES Inherited from : intelligent tutoring systems (ITS) and adaptive hypermedia systems.
This section provides a review of existing technologies grouped by its origin.

ITS technologies in Web-based education
Goal of various ITS is the use the knowledge about the domain, the student, and about teaching strategies to support flexible individualized learning and tutoring.
Three core ITS technologies:
- curriculum sequencing,
- intelligent analysis of student's solutions,
- and interactive problem solving support.

Curriculum sequencing
Provide the student with the most suitable individually planned sequence of knowledge units to learn and sequence of learning tasks (examples, questions, problems, etc.). It helps the student to find an "optimal path" through the learning material.
There are two essentially different kinds of sequencing: active and passive
Active sequencing implies a learning goal . Systems with active sequencing can build the best individual path to achieve the goal.
Passive sequencing 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 correctly. Its goal is to offer the user a subset of available learning material.
High-level sequencing or knowledge sequencing determines next learning subgoal: next concept, set of concepts, topic, or lesson to be taught.
Low-level sequencing or task sequencing determines next learning task (problem, example, test) within current subgoal.
Among active sequencing systems, only a handful of systems such as ELM-ART-II, AST, ADI, ART-Web, ACE, KBS-Hyperbook, and ILESA are able to perform intelligently both high and low level sequencing. Others, like Manic, leave a choice of activity within a topic to the user. Vice versa, some systems, like Medtec, leave a choice of a topic to the user but can generate an adaptive sequence of problems within the topic. Most of the systems supports sequencing with fixed learning goal (equals to the whole course). Only a few systems support adjustable learning goals enabling a teacher (as in DCG) or a student (as in InterBook and KBS Hyperbook) to select an individual goal.

Active sequencing in most of the systems is driven by the students knowledge (more exactly, by the difference between student’s knowledge and global goal).

DCG can perform advanced sequencing of educational material adapted to a learning goal. However, the sequencing is performed before students start working with the system producing a static Web-based course.

SIETTE is an example of a Web-based adaptive testing system. The only kind of learning material it possesses is questions. The only thing it can do is to generate an adaptive sequence of questions to assess student's knowledge. Systems like SIETTE are incomplete by their nature and have to be used as components in distributed Web-based AIES.

**Problem solving support technologies**

Three problem solving support technologies:
- intelligent analysis of student solutions
- interactive problem solving support
- example-based problem solving support

Intelligent analysis of student solutions

Deals with students final answers to no matter how these answers were obtained. 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.

Interactive problem solving support

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

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.
Adaptive hypermedia technologies in Web-based education

two major technologies in adaptive hypermedia: adaptive presentation and adaptive navigation support.

Adaptive navigation support. Adaptivenavigation support (ANS) can be considered as a generalization of curriculum sequencing technology in a hypermedia context. It shares the same goal - to help students to find an "optimal path" through the learning material.

The three ways that are most popular in Web-based AIES are direct guidance, adaptive link annotation, and adaptive link hiding.

Direct guidance implies that the system informs the student which of the links on the current page will drive him or her to the "best" page in the hyperspace. The most popular form of ANS on the Web is annotation. Another popular technology is hiding and disabling link.

Adaptive presentation
The goal of the adaptive presentation technology is to adapt the content of a hypermedia page to the user's goals, knowledge and other information stored in the user model.
2) Adaptive Hypermedia: From Systems to Framework


Introduction

Adaptive hypermedia systems build a model of the goals, preferences and knowledge of the individual user.

Adaptive Hypermedia Systems (AHS) make it possible to deliver "personalized" views or versions of a hypermedia document.

Framework, Methods, and Techniques for Adaptive Hypermedia

A Framework for Adaptive Hypermedia

"knowledge" contained in a hyperdocument is described by

There are three kinds of concepts: atomic concepts or fragments (the smallest information units), pages (composed of fragments) and abstract concepts (representing larger units of information).

Concepts are connected through concept relationships.

An AHS typically performs three functions:

- While a user is "browsing" through an adaptive hyperdocument all user actions are registered. Based on these observations the AHS maintains a model of the user's knowledge about each domain model concept. Typical attributes a user model keeps for each concept are knowledge-value.

- The user model is applied to classify all nodes (pages) into several groups according to the user's current knowledge and interests or goals. The AHS manipulates link anchors within nodes to guide users towards interesting, relevant information.

- In order to ensure that the content of a page contains the appropriate information the AHS will conditionally show, hide, highlight or dim conditional fragments on a page when presenting it.

Adaptive Presentation

Adapting the presentation of information within a page is most often performed as a manipulation of (canned) text fragments.

- Conditional inclusion of fragments:

- Stretchtext:

  stretch or shrink fragments through mouse clicks.

Recently a new technique was introduced, evaluated and found useful, where fragments that are not recommended for a user are grayed out instead of removed or shrunk.
Providing explanation variants: Depending on values in the user model the level of difficulty, the related concepts a page refers to, the length of the presentation, the media type (text, images, audio, video) or other aspects may be changed.

-Reordering information

Adaptive Navigation Support
The manipulation of links that are presented within nodes
- Direct guidance: A "next" or "continue" (link) button is shown. The destination of this link is the node which the AHS determines to be most appropriate.
- Sorting of links:
- Link annotation:
- Link hiding:
- Link disabling:
- Link removal:
- Map adaptation:
3) Adaptive Hypermedia: From Intelligent Tutoring Systems to Web-Based Education


**Introduction**

limitation of traditional "static" hypermedia

present the same static explanation and suggest the same next page to students with widely differing educational goals and knowledge of the subject.

Adaptive hypermedia is an alternative to the traditional “one-size-fits-all” approach in the development of hypermedia systems.

Adaptive hypermedia systems build a model of the goals, preferences and knowledge of each individual user, and use this model throughout the interaction with the user, in order to adapt the hypertext to the needs of that user.

**What Can Be Adapted in Adaptive Hypermedia**

content-level adaptation: -adaptive presentation

link-level adaptation: -adaptive navigation support

The goal of the adaptive presentation is to adapt the content of a hypermedia page to the user's goals, knowledge and other information stored in the user model.

The goal of adaptive navigation support is to help users to find their paths in hyperspace by adapting link presentation and functionality to the goals, knowledge, and other characteristics of an individual user.

- Direct guidance:
- Link sorting:
- Link annotation:
- Link hiding, disabling, and removal:

**From Intelligent Tutoring Systems to Adaptive Hypermedia**

**adaptive presentation:**

- explanation variants
- conditional text: -information divided into chunks
- By choosing appropriate conditions on the knowledge level of the current concept and related concepts represented in the user model we were able to implement several adaptive presentation methods.
adaptive navigation support:

● curriculum sequencing: - Provide 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.

● typical sequencing engine can do more than just selecting the "next best" task. On the way to the "best", such an engine can usually classify all available tasks into nonrelevant and relevant candidates. For example, a task can be considered non-relevant if it was already completed in the past or if it is not ready to be learned due to the lack of prerequisite knowledge and experience. After excluding non-relevant tasks a sequencing engine use some approach to pick up the best of relevant tasks. In a hyperspace of learning material where each learning task is represented by a separate page an ability to distinguish "ready", "not-ready", or "best" tasks is a direct precondition for adaptive navigation support.

● We have used direct guidance in the form of "teach me" button to provide a one-click access to the next best task.
4) The Adaptive Learning System based on Learning Style and Cognitive State


Introduction:

most of the e-learning systems have ignored the individual difference that the learners exist, such as the ability, background, goal, knowledge foundation and learning style, etc. They send the unified teaching material to all learners.

State of the art

The research base of adaptive learning system includes the Intelligent Tutoring System (ITS) and the Adaptive Hypermedia System (AHS). The goal of Intelligent Tutoring System is to provide tailored instruction according to the learner’s individual differences. So Intelligent Tutoring System must reason about the Student Model, the Domain Model and the Teaching Strategies Model. The main function of Adaptive Hypermedia System is the adaptive presentation and the adaptive navigation support.

Learner Model based on learning style and cognitive state

If the domain knowledge is composed of n DCs (dc1,…,dcn), then the cognitive state of a given learner is represented by the set:

Beliefs = {B1,B2,…, Bn},

Each Bi being a belief so defined:

Bi = <DCi, IsReady, HasVisited, HasStudied, HasTested, TestValue>
Architecture of adaptive learning system based on AHAM

User Interface

Adaptive Model
  Adaptive Content
  Adaptive Navigation
  Adaptive Testing
  Tracing and Recording

Course Developer

Learning Goal Hierarchy

Domain Ontology

Domain Model

Instructional Resources

Instructional Resources Description Model

Media Space

Instructional Resources

Instructional Resources Description Model

Media Space

Instruction Model

Content Selection Rules

Media Selection Rules

Learner Model

Cognitive State

Learning History

Learning Style

Learner Profile

Teachers

Learner
5) Dynamic Courseware Generation on the WWW


**Introduction:**

Unlimited access of teaching material on www.
Problems: huge amount of materials and to navigate through links which sometimes are not relevant to their learning goals. Huge hyperspace.
So use prerequisite structure of concepts to suggest which concepts should be learned first in order to understand the teaching goal-concept.
A traditional course is a predefined sequence of teaching materials.
Inflexible to accommodate the wide variety of learners on the WWW, with their possible learning goals, background knowledge and differences in acquiring material.
This problem can be solved by a tool for automatic generation of individualized adaptive courses. According to the learner’s goal and previous knowledge it can dynamically adapt the course according to the learner’s success in acquiring the concepts from the plan.
• automatically assembling courses with different goals from a pool of teaching materials;
• creating different courses for learners with different knowledge;
• changing the course dynamically according to the progress of the learner.
**Main Idea and Architecture of DCG**

Applying AI planning techniques to create a concept(content) plan of a course which achieves a certain goal concept starting from some set of initial concepts. The concept structure is represented as an AND/OR graph where nodes represent concepts and arcs correspond to relationships between the concepts. Given a certain goal-concept that the learner wants to acquire and a student model containing the concepts already known by the learner (initialized with a pre-test), an AI-planner program searches for plans (paths in the graph) that connect the concepts known by the learner with the goal-concept. For each concept, teaching materials are presented from the database. The learner is tested at every concept of the plan and his student model is updated. If the learner is not able to achieve the needed score for a concept that is needed to proceed further with the plan, a new planning takes place to find an alternative plan for achieving the learning goal.

**Authoring with the DCG on the WWW**

Authoring consists of two parts:
- creating/modifying a concept structure for a given domain,
- providing links from each concept / relation to appropriate html-files on the WWW that can be used as teaching materials for this concept.
Learning with the DCG
Learner send request to DCG with desired domain and learning goal - sends preliminary test to initialise student model - learner receives individual course plan and executor program - test the student knowledge and update student model - incase learner unable - executor send student model to planner and request for new course plane.

Components of the DCG on the WWW

The Domain Concept Structure
It is represented as an AND/OR graph consisting of the domain concepts connected with relations. Expressive power of AND/OR graphs which allow representing not only prerequisite, but many different types of relations between concepts. This allows the use of AI-planning techniques to generate alternative courses according to links with different semantic and in this way ensures a wide variety of different teaching goals and possible courses for achieving these goals.
**The Teaching and Testing Materials**
At the authoring stage hyperlinks from the concepts to the desired html-files with teaching material are provided.
Test-items should also be defined for every concept with an indication of their difficulty and a coefficient, showing how much a correct / wrong answer contributes to the overall score of the concept/link in the student model.

**Student Model**
There are two instances of the Student Model. One is dynamic and exists on the client (learner’s) site. A copy of this instance is stored on the DCG-server every time the learner ends a session or re-planning is required. It is used for recording statistics of learner success with different concepts which is useful for improving the teaching materials and domain concept structure. The current state of learner knowledge is system’s estimations that the learner knows a certain concept. An estimation is built using a simple formula from the number and difficulty of successfully solved tests items related to this concept.

**Planner**
On the basis of the AND/OR graph representing the domain concept structure, the learning goal and the set of initial nodes (the Student Model representing the learner’s current knowledge), the Planner generates a plan of the course.
The plan is represented as a sub-graph of the concept structure consisting of concepts and relations that have to be taught during the course.

**Executor**
It takes the plan as input and generates a course from it by searching in the WWW for the appropriate teaching materials. When the learner wishes to be tested on his knowledge of the concept, the Executor selects a test-item, evaluates answer and correspondingly updates the Student Model. It can re-invoke the Planner to create a new plan if the level of knowledge is not satisfied.
6) Ability Assessment based on CAT in Adaptive Learning Systems


INTRODUCTION

Adaptive learning system, in essence, is a kind of online learning environment that supports the personalized learning. And Student Model is the key component of adaptive learning system, which records the learning goal, preference, learning style and the ability level of learner.

In order to more effectively evaluate the learner's ability level and supports the personalized learning, we integrate the Computerized Adaptive Testing (CAT) module into adaptive learning system and analyzed the adaptive testing algorithms based on the Item Response Theory.

COMPUTERIZED ADAPTIVE TESTING

Computerized adaptive testing select the most appropriate test items for learner based on the learner's ability.

1. All the questions in the test item bank are examined to determine which will be the best to ask next according to the current estimation of the learner's ability level.
2. The question is asked, and the learner responds.
3. According to the answer, a new estimation of the ability level of learner is computed.
4. Steps (1) to (3) are repeated until the termination condition defined is met.

INTEGRATING CAT MODULE INTO ADAPTIVE LEARNING SYSTEM
**Item response model**

Item response model is a mathematics model, which is used to express the item characteristic curve. Among the existing item response model, the three parameter logistic (3PL) model is one of the most commonly used models. The 3PL model equation is shown below:

\[
p_i(\theta) = \frac{1}{1 + e^{-1.7a_i(\theta - b_i)}} \quad (1)
\]

**Test item bank**

Each test item includes the content, difficulty parameter, discrimination parameter, and guessing parameter and so on. In order to improve the effect of CAT, test item bank should meet two basic conditions: (1) test item bank must have sufficient test items to supply informative test
items during a test session, and (2) the item bank must provide adequate test items covering different topics and difficulty levels.

For every test item, the test item bank includes the item’s content, correct answer, difficulty, discrimination, and the guessing parameter. In our test items bank, the difficulty is divided into five levels, including the very easy, easy, medium, difficult and very difficult.

**Item generation**

Item information function (IIF):

\[
I_i(\theta) = \frac{[P_i'(\theta)]^2}{P_i(\theta)Q_i(\theta)}, \quad i = 1, 2, 3, \ldots, n \quad (2)
\]

\[
I_i(\theta) = \frac{1.7^2a_i^2(1-c_i)}{[c_i + \exp(1.7a_i(\theta - b_i))][1 + \exp(-1.7a_i(\theta - b_i))]}^2
\]

**Ability estimation**

Maximum Likelihood Estimation (MLE). Under the local conditions independence assumption, we first assume that a learner responds to a set of n test items with response matrix \( U = [u_1 u_2 \ldots u_n] \), if \( u_i | \theta \) represents the conditional probability of response \( u_i \) on test item \( i \) conditioned on the ability \( \theta \), then Likelihood function is shown as follow:

\[
L(u_1, u_2, \ldots, u_n | \theta) = \prod_{i=1}^{n} P_i(\theta)^{u_i} (1 - P_i(\theta))^{(1-u_i)}
\]
**Termination condition**

In general, termination condition is the precision of measurement, the test time, and the number of test item or a combination of them.

\[ SE(\theta) = \frac{1}{\sqrt{TI(\theta)}} \]

Where

\[ TI(\theta) = \sum_{i=1}^{n} I_i(\theta) \]

Where TI (\( \theta \)) is the test information function that is the sum of the value of IIF. Commonly, when the standard error of CAT is about 0.33 or smaller, the test will be terminated.

---

**Figure 3. Process of ability evaluation in CAT**