Adaptive Systems for E-Learning

Peter Brusilovsky
School of Information Sciences
University of Pittsburgh, USA
peterb@sis.pitt.edu
http://www2.sis.pitt.edu/~peterb

Overview

• The Context
• Technologies
  – ITS technologies
  – AH technologies
  – Web-inspired technologies
• WWW for adaptive educational systems
Overview

- The Context
- Technologies
  - ITS technologies
  - AH technologies
  - Web-inspired technologies
- WWW for adaptive educational systems

Outline:
- The Context
- Technologies
- Implementation
- WWW for adaptive educational systems
- AWBES and E-Learning
The Context

• Adaptive systems
• Why adaptive?
• Adaptive vs. intelligent

Adaptive systems

Classic loop user modeling - adaptation in adaptive systems
Adaptive software systems

- Intelligent Tutoring Systems
  - adaptive course sequencing
  - adaptive ...
- Adaptive Hypermedia Systems
  - adaptive presentation
  - adaptive navigation support
- Adaptive Help Systems
- Adaptive ...

Why AWBES?

- greater diversity of users
  - “user centered” systems may not work
- new “unprepared” users
  - traditional systems are too complicated
- users are “alone”
  - limited help from a peer or a teacher
Intelligent vs. Adaptive

1. Intelligent but not adaptive (no student model!)
2. Adaptive but not really intelligent
3. Intelligent and adaptive

Overview

• The Context
• Technologies
• Implementation
• WWW for adaptive educational systems
• AWBES and E-Learning
Technologies

- Origins of AWBES technologies
- ITS Technologies
- AH Technologies
- Web-Inspired Technologies

Origins of AWBES Technologies

- Intelligent Tutoring Systems
- Adaptive Hypermedia Systems
- Adaptive Web-based Educational Systems
Origins of AWBES Technologies

Adaptive Hypermedia Systems

Intelligent Tutoring Systems

Adaptive Hypermedia

Intelligent Tutoring

Adaptive Presentation

Curriculum Sequencing

Adaptive Navigation Support

Problem Solving Support

Intelligent Solution Analysis

Origins of AIWBES Technologies

Adaptive Hypermedia Systems

Information Retrieval

CSCL

Adaptive Hypermedia

Adaptive Information Filtering

Machine Learning, Data Mining

Intelligent Tutoring Systems

Intelligent Monitoring

Intelligent Collaborative Learning

Intelligent Tutoring
Technology inheritance examples

- Intelligent Tutoring Systems (since 1970)
  - CALAT (CAIRNE, NTT)
  - PAT-ONLINE (PAT, Carnegie Mellon)
- Adaptive Hypermedia Systems (since 1990)
  - AHA (Adaptive Hypertext Course, Eindhoven)
  - KBS-HyperBook (KB Hypertext, Hannover)
- ITS and AHS
  - ELM-ART (ELM-PE, Trier, ISIS-Tutor, MSU)

Inherited Technologies

- Intelligent Tutoring Systems
  - course sequencing
  - intelligent analysis of problem solutions
  - interactive problem solving support
  - example-based problem solving
- Adaptive Hypermedia Systems
  - adaptive presentation
  - adaptive navigation support
Course Sequencing

- Oldest ITS technology
  - SCHOLAR, BIP, GCAI...
- Goal: individualized
  - “best” sequence of educational activities
    - information to read
    - examples to explore
    - problems to solve ...
- Curriculum sequencing, instructional planning, ...

Active vs. passive sequencing

- Active sequencing
  - goal-driven expansion of knowledge/skills
  - achieve an educational goal
    - predefined (whole course)
    - flexible (set by a teacher or a student)
- Passive sequencing (remediation)
  - sequence of actions to repair misunderstanding or lack of knowledge
Levels of sequencing

- High level and low level sequencing

Sequencing options

- On each level sequencing decisions can be made differently
  – Which item to choose?
  – When to stop?
- Options
  – predefined
  – random
  – adaptive
  – student decides
**Topic sequencing**

- No adaptivity within the topic

**Task sequencing**

Usually predefined order of topics or one topic
Multi-level sequencing

- Adaptive decisions on both levels

Simple cases of sequencing

- No topics
- One task type
  - Problem sequencing and mastery learning
  - Question sequencing
  - Page sequencing
ELM-ART: question sequencing

Sequencing for AWBES

- Simplest technology to implement with CGI
- Important for WBE
  - “no perfect order”
  - lack of guidance
- No student modeling capability!
  - Requires external sources of knowledge about student
  - Problem/question sequencing is self-sufficient
Models for sequencing

- Domain model
  - Network of concepts
- Model of Educational Material
  - Indexing
- Student model
  - Overlay model
- Goal model

Domain model - the key

Diagram showing a network of concepts: Concept 1, Concept 2, Concept 3, Concept 4, Concept N.
Vector vs. network models

- Vector - no relationships
- Precedence (prerequisite) relationship
- is-a, part-of, analogy: (Wescourt et al, 1977)
- Genetic relationships (Goldstein, 1979)
Network model

Indexing teaching material

- Types of indexing
  - One concept per ULM
  - Indexing of ULMs with concepts
- How to get the ULMs indexed?
  - Manual indexing (closed corpus)
  - Computer indexing (open corpus)
Simple case: one concept per ULM

- Random selection if there are no links - Scholar
- Links can be used to restrict the order

Indexing ULMs with concepts
Simple overlay model

Simple overlay model
Weighted overlay model

Simple goal model

• Learning goal as a set of topics
More complicated models

• Sequence, stack, tree

Sequencing with models

• Given the state of UM and the current goal pick up the best topic or ULM within a subset of relevant ones (defined by links)
• Special cases with multi-topic indexing and several kinds of ULM
• Applying explicit pedagogical strategy to sequencing
Intelligent problem solving support

• The “main duty” of ITS
• From diagnosis to problem solving support
• High-interactive technologies
  – interactive problem solving support
• Low-interactive technologies
  – intelligent analysis of problem solutions
  – example-based problem solving

High-interactive support

• Classic System: Lisp-Tutor
• The “ultimate goal” of many ITS developers
• Support on every step of problem solving
  – Coach-style intervention
  – Highlight wrong step
  – Immediate feedback
  – Goal posting
  – Several levels of help by request
Example: PAT-Online

You have just been promoted at PAT-E-GIF Furniture Inc and have received a raise to $50 per hour.

1. How much would you get paid if you worked 5 hours?
2. How much would you get paid if you worked 25 hours?
3. How much would you get paid if you worked 10 1/2 hours?
4. How much would you get paid if you worked 100 hours?
5. How much would you get paid if you worked 200 hours?
6. If you plan on working 10 hours a week during a forty week school year, what would be your total earnings be for the entire year?

For the formula, define a variable for the time worked and use this variable to write a rule for your total pay.

<table>
<thead>
<tr>
<th>Heading</th>
<th>Time worked</th>
<th>Earnings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit</td>
<td>[hours]</td>
<td></td>
</tr>
<tr>
<td>Formula</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Low-interactive technologies

• Intelligent analysis of problem solutions
  – Classic system: PROUST
  – Support: Identifying bugs for remediation and positive help
  – Works after the (partial) solution is completed

• Example-based problem solving support
  – Classic system: ELM-PE
  – Works before the solution is completed
Example: ELM-ART

Problem-solving support

- Important for WBE
  - problem solving is a key to understanding
  - lack of problem solving help
- Hardest technology to implement
  - research issue
  - implementation issue
- Excellent student modeling capability!
Models for interactive problem-solving support and diagnosis

- Domain model
  - Concept model (same as for sequencing)
  - Bug model
  - Constraint model
- Student model
  - Generalized overlay model (Works with bug model and constraint model too)
- Teaching material - feedback messages for bugs/constraints

Bug models

- Each concept/skill has a set of associated bugs/misconceptions and sub-optimal skills
- There are help/hint/remediation messages for bugs
Do we need bug models?

- Lots of works on bug models in the between 1974-1985
- Bugs has limited applicability - problem solving feedback. Sequencing does not take bugs into account: whatever misconceptions the student has - effectively we only can re-teach the same material
- Do not model that you can’t use

Models for example-based problem solving support

- Need to represent problem-solving cases
- Episodic learner model
  - Every solution is decomposed on smaller components, but not concepts!
  - Keeping track what components were used and when - not an overlay!
- ELM-PE and ELM-ART - only systems that use this model
Adaptive hypermedia

- Hypermedia systems = Pages + Links
- Adaptive presentation
  - content adaptation
- Adaptive navigation support
  - link adaptation

Adaptive navigation support

- Direct guidance
- Hiding, restricting, disabling
- Generation
- Sorting
- Annotation
- Map adaptation
Adaptive annotation: Icons

Annotations for topic states in Manuel Excell: not seen (white lens); partially seen (grey lens); and completed (black lens)

Adaptive annotation: Font color

Annotations for concept states in ISIS-Tutor: not ready (neutral); ready and new (red); seen (green); and learned (green+)
Adaptive hiding

Hiding links to concepts in *ISIS-Tutor*: not ready (neutral) links are removed. The rest of 64 links fits one screen.

Adaptive annotation: InterBook

1. Concept role  
2. Current concept state  
3. Current section state  
4. Linked sections state
ANS: Evaluation

• ISIS-Tutor: hypermedia-based ITS, adapting to user knowledge on the subject
• Fixed learning goal setting
• Learning time and number of visited nodes decreased
• No effect for navigation strategies and recall

Adaptive presentation techniques

• Conditional text filtering
  • ITEM/IP, PT, AHA!
• Adaptive stretchtext
  • MetaDoc, KN-AHS, PUSH, ADAPTS
• Frame-based adaptation
  • Hypadapter, EPIAIM, ARIANNA, SETA
• Full natural language generation
  • ILEX, PEBA-II, Ecran Total
Example: Stretchtext (PUSH)

Example: Stretchtext (ADAPTS)
Adaptive presentation: evaluation

- MetaDoc: On-line documentation system, adapting to user knowledge on the subject
- Reading comprehension time decreased
- Understanding increased for novices
- No effect for navigation time, number of nodes visited, number of operations

Models for adaptive hypermedia

- Domain model - same as for sequencing
- Student model - same as for sequencing
- Goal model - same as for sequencing
- Model of the learning material
  - For ANS - same as for sequencing
  - For AP - could use fragment or frame indexing
Indexing of nodes

Domain model

Hyperspace

Concept 1
Concept 2
Concept 3

Concept 4
Concept n

Indexing of page fragments

Concepts

Node

Concept 1
Concept 2
Concept 3

Concept 4
Concept N

Concept 5

Fragment 1
Fragment 2
Fragment K
Web-inspired technologies

• One ITS, many student models: student model matching!
• Adaptive collaboration support
  – peer help and collaborative group formation
• Intelligent class monitoring
  – finding troubled students in HyperClassroom
• Not enough work yet, but seems like overlay and bug models work well

Overview

• The Context
• Technologies
  • Implementation
• WWW for adaptive educational systems
• AWBES and E-Learning
Implementation

• What can make an AWBES?
• Interaction
  – CGI-based interaction
  – Java-based interaction
• Student modeling
• From ITS to WITS

What can make an AWBES?

• AWBES <= AH + problem solving support
• Hyperspace of educational material is an essential part of AWBES
  – Need an access to educational material
  – Hyperspace is natural for WBS
• AH is important for guidance
• Problem solving component is important for both interactivity and student modeling
Interaction technologies

• Common Gateway Interface (CGI)
  – Client to server
    • URLs with parameters
    • HTML forms
  – Server to client
    • HTML pages generated “on the fly”
• Java way
  – Client-server solution!

Classic CGI scripting
Separate application

Web browser
HTML form

HTTP Server

Generated page
CGI request

Service script

Permanently running application

AES core

AES server

Web browser
HTML form

HTTP part

Server

AES part

Generated page
CGI request
Java servlets

- Web browser
  - HTML form
- Generated page
- CGI request
- HTTP Server
- Java servelets

Java-based interactivity

- Web browser
  - Java applet
- Page with an applet
- Direct connection
- HTTP Server
- AES server

- Client side
- Server side
- AES core
Student modeling and adaptivity

- How to register
- How to recognize a user within the session?
  - Part of the URL
  - Cookies
  - Separate process for each user
- How to end the session
- Use what your tool provides

Separate processes
From ITS to WITS

- Consider adding full hypermedia
- Choose relevant architecture
- Replace interface part
- Solve problem of multiple users

Example: ELM-PE to ELM-ART

- AH Lisp textbook has been added
- Since ELM-PE was implemented in Lisp a CL-HTTP based “in-server” solution has been chosen
- GUI has been replaced by CGI/form interface
- Multi-user problem solution: part of URL (some reprogramming required)
Overview

• The Context
• Technologies
• Implementation
  • WWW for adaptive educational systems
  • AWBES and E-Learning

WWW for AES

• Just a new platform?
• Web impact
  – Changing the paradigm
• Web benefits
• Web value
  – New AES technologies
  – What else?
Old AI-CAI Paradigm (1970)

• Goal: replace primitive CAI in transferring knowledge (content) to students

Classic ITS paradigm (1980s)

• Goal: support problem solving
• Classroom context
• No learning material on-line
• No adaptive hypermedia
• No course sequencing
• Interactive problem solving support is the core technology
AWBES: The new paradigm

- Goal: comprehensive support
- Self-study context
- All learning material on-line:
  - presentations, tests, examples, problems
- Curriculum sequencing
- Adaptive navigation support
- Problem solving support

Web benefits

- Visibility and impact
- From laboratories to classrooms
  - Equipment issue
  - Maintenance issue
  - Natural part of WBE
- Testing base and data collection
- Standard technologies and component reuse
Web value

- One tutor, many students ⇒ model matching
- One student, many tutors ⇒
  - Distributed ITS (assembling by design)
    - PAT-InterBook
  - Distributed ITS with reusable components
    - authoring time flexibility
  - Mega-ITS (assembling by request)
    - interaction time flexibility
    - Mega-Tutor (Rowley), Topic Server (Murray)

Agents

- Why agents?
- Agent metaphors
  - Animated agents (ADELE, Wincent, )
  - Pedagogical agents (teacher, troublemaker)
- Agent architectures
  - The issue of granularity
Problems of integration

- Control issue
  - User switches
  - What about proper sequencing?
  - One component asking another to do something
- Student modeling issue
  - A tutor can use information collected by others
  - A tutor can pass collected information to others

Student modeling in DITS

- Student mode exchange (PAT-InterBook)
- Student model servers (Tagus)
- Client-side student modeling?
- Integration and distribution issue
  - Different components need different information about students
  - Information may be contradictory
InterBook communication interface

- Interbook is a *component*
- Communication architecture with shared user model
- PAT - InterBook example
Centralized Student Modeling

Overview

• The Context
• Technologies
• Implementation
• WWW for adaptive educational systems
• AWBES and E-Learning
AWBES and WBE

- Why not now and when?
- What do we need for WBE?
- The contribution of AWBS
- Gradual implementation:
  - Challenges of integration of intelligent tutors

WBE Tools

- The classes of users to serve
  - Web presence for a course
  - Assisting in a real classroom
  - Virtual university and distance education
  - Technical training
- From separate tools to Course Management Systems (CMS)
Course Management Systems

• Modern CMS
  – University-level
    • Cyberprof, Mallard, CM Online...
  – Commercial
    • TopClass, WebCT, LearningSpace, CourseInfo...
  – Consulting
    • eCollege, Eduprise...

• Future
  – Standardization: LOM, CMI, SCORM...

CMI functions

• Course material delivery
• Authoring and maintenance
• Assessment
• Communication and collaboration
• Administration
• Control
Course Material

- Presentation
  - Adaptive presentation
- Assessment
  - Adaptive testing
- Learning by doing
  - Problem solving support
- Authoring and maintenance
  - Concept-based customization and maintenance

Beyond Course Material

- Communication and collaboration
  - Peer help and collaborative group formation
  - Collaboration coach
- Administration
  - Identifying students in trouble
- Control
  - Sequencing
  - Adaptive navigation support
Gradual adoption of AWBES

- Static course sequencing - domain modeling for courseware engineering
- Customized course generation
- Adaptive testing
- Sequencing and navigation support
- Model matching
- Problem-solving support