Development of Intelligent Tutoring System Framework: Using Guided Discovery Learning

Raja Shekhar
10305034
M.Tech-2

Under the guidance of

Prof. Sridhar Iyer

June 25, 2012
Outline

- Existing Systems
- Guided Discovery Learning
- ITS Support for Guided Discovery
- Architecture and Modules
- Integration of the System
- Conclusion and Future Work
Existing Systems

- Developed in: geography, circuits, medical diagnosis, computer programming

Some Example ITSs:
- SQLT-Web: SQL Tutor
- Activemath: Mathematics Tutor
Figure: SQLT GUI
SQLT-Web Tutor- Architecture

User

Web browser

Web Server

Session Manager

logs

Student modeler

pedagogical module

student models
Student id, level, constraints, wrong

constraints base
Constraints Feedback

Domain database
Questions, Solutions
SQLT-Web Tutor- Architecture

User → Web browser → Web Server → Session Manager

Constraints:
- If some keyword appears then another keyword should follow it.
- If "join" keyword is in a select statement then the select statement should also have "on" keyword.

Student modeler → pedagogical module

Student models → Domain database

Constraints base Constraints
SQLT-Web Tutor- Architecture

User

1. user submit solution

Web browser

Web Server

Session Manager

logs

Student modeler

pedagogical module

Student models

constraints base

Domain database

10. receive feedback

2. submitted solution

3. access ideal solution

4. Evaluate

5. access constraints

6. update student database

7. result

8. get feedback

9. send feedback

No data

Find top 5 students of the semester who have taken at least 4 courses
Assessment - CBM

- No data
- Uses Constraint Based Modeling (CBM) for assessment
  - Syntax Verification
  - Equivalent constructs checking - Constraints
- Feedback - Associated to constraints
Our Framework

Problems with existing systems

- Single teaching style

- Subject specific
  - Due to assessment process
Our Framework

- Problems with existing systems
  - Single teaching style
  - Subject specific
    - Due to assessment process

- Our ITS Framework
  - 4 teaching styles
  - Is not subject specific
    - Using MCQ for our ITS
- Existing Systems
- **Guided Discovery Learning**
- ITS Support for Guided Discovery
- Architecture and Modules
- Integration of the System
- Conclusion
- Future Work
Guided Discovery Learning

- Hands-on activities
Hands-on activities

Example:
Goal: array memory allocation concept

What is the output of the following snippet

```c
main()
int a[]={1,2,3,4};
float b[5]={3.2,8.7,8,9.8};
printf("%u %u %u %u %u",sizeof(int),&a[0],&a[1],&a[2],&a[3]);
printf("%u %u %u %u %u",sizeof(float),&b[0],&b[1],&b[2],&b[3]);
```
Figure: Steps in guided discovery learning
• Existing Systems

• Guided Discovery Learning

• ITS Support for Guided Discovery

• Architecture and Modules

• Integration of the System

• Conclusion

• Future Work
ITS Support for Guided Discovery

Course Structure

- Course - C Language
- Topic - Arrays
- Subtopic - 1D Arrays
- Quiz
ITS Support for Guided Discovery

Course Structure
- Course - C Language
- Topic - Arrays
- Subtopic - 1D Arrays
- Quiz

Order of teaching/Pre-requisite relation
1. Topic dependency
2. Subtopic dependency
Quiz

- Multiple choice questions

- 2 types of questions
  1. Guiding questions: Hands-on activities
     Interactive pop-up window
     Not used for assessment
  2. Testing questions: Used for assessment
Steps to be followed by instructor

1. Select/create course
2. Create topic
3. Enter Topic Dependency
Steps to be followed by instructor

1. Select/create course
2. Create topic
3. Enter Topic Dependency

Total number of topics are: 5

Please enter the dependencies for the topic created

**TOPIC DEPENDENCY TABLE**

<table>
<thead>
<tr>
<th></th>
<th>Arrays</th>
<th>Data types</th>
<th>Linked Lists</th>
<th>Pointers</th>
<th>Trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrays</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data types</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linked Lists</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pointers</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trees</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Steps to be followed by instructor

1. Select/create course
2. Create topic
3. Enter Topic Dependency
4. Create subtopic
5. Enter Subtopic Dependency
Steps to be followed by instructor

1. Select/create course
2. Create topic
3. Enter Topic Dependency
4. Create subtopic
5. Enter Subtopic Dependency

Please enter the dependencies for the subtopic created

SUBTOPIC DEPENDENCY TABLE

<table>
<thead>
<tr>
<th></th>
<th>1-D array</th>
<th>2-D arrays</th>
<th>MultiDimensional arrays</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D array</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>2-D arrays</td>
<td></td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>MultiDimensional arrays</td>
<td>□</td>
<td></td>
<td>□</td>
</tr>
</tbody>
</table>
Steps to be followed by instructor

1. Select/create course

2. Create topic

3. Enter Topic Dependency

4. Create subtopic

5. Enter Subtopic Dependency

6. Enter questions

7. Enter threshold value
Steps to be followed by Learner

1. Select course
2. Select topic
3. Select subtopic
4. Use pop-up window
5. Submit answers
6. Reattempt or attempt remaining questions
Steps followed by ITS

1. Topic dependency check
Steps followed by ITS

1. Topic dependency check

**Figure:** Topic dependency check
Steps followed by ITS

1. Topic dependency check
2. Subtopic dependency check
Steps followed by ITS

1. Topic dependency check
2. Subtopic dependency check
Steps followed by ITS

1. Topic dependency check
2. Subtopic dependency check
3. Use adaptation logic
Steps followed by ITS

1. Topic dependency check

2. Subtopic dependency check

3. Use adaptation logic

1) Present guiding question
2) Evaluate
3) Update learner knowledge
4) Repeat steps 1 to 3 until all guiding questions finish
5) if(#correct ans > threshold)
   reattempt/atempt remaining option
else
   display the testing question
6) Evaluate
7) Update learner knowledge
8) Repeat steps 5 to 7 until all testing questions finish
Adaptation levels

- Where is the adaptation applied?
Adaptation levels

- Where is the adaptation applied?
  - Strategy Switching for learner
  - Topic level: Topic Dependency
  - Subtopic level: Subtopic Dependency
  - Promoting to next subtopic
- Existing Systems
- Guided Discovery Learning
- ITS Support for Guided Discovery
- Architecture and Modules
- Integration of the System
- Conclusion and Future Work
Architecture of ITS

User

Web browser

Web Server

Session Manager

Authentication module

Quiz Module

Controller Module

Domain Module

Course module

Topic module

Subtopic module

student models

student1, student2, student_progress

Domain Base

question1, question2, question3

content queries

Stores indexes

files

Physical Storage of resources
Architecture of ITS

User → Web browser → Web Server → Session Manager

Authentication module

Quiz Module

Domain Module

- Course module
- Topic module
- Subtopic module

Domain Base

- student models
- question1, question2, question3

Stores indexes

Physical Storage of resources

content queries

files
Architecture of ITS

User → Web browser → Web Server → Session Manager

Authentication module

Quiz Module

Controller Module

Domain Module

Course module

Topic module

Subtopic module

content queries

files

Stores indexes

Physical Storage of resources

student models

student1, student2, student_progress

Domain Base

question1, question2, question3
Session Manager

- Authentication
- Access Control
  - Php Session Variables
Domain module

User

Web browser

Web Server

Session Manager

Authentication module

Quiz Module

Controller Module

Domain Module

Course module

Topic module

Subtopic module

student models

stuent1, student2

student_progress

Domain Base:

question1, question2,

question3

content queries

Stores indexes

files

Physical Storage of resources
Topic Module

- Content creation
- Topic dependency
Topic Module

- Content creation
- Topic dependency

Total number of topics are: 5

Please enter the dependencies for the topic created

**TOPIC DEPENDENCY TABLE**

<table>
<thead>
<tr>
<th></th>
<th>Arrays</th>
<th>Data types</th>
<th>Linked Lists</th>
<th>Pointers</th>
<th>Trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrays</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data types</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linked Lists</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pointers</td>
<td>☑</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trees</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Submit
Topic Module-Loop Detection

- Content creation
- Topic dependency

Figure: Topic Dependency Graph
Topic Module-Loop Detection

- Content creation
- Topic dependency

**Figure:** Loop formation
Modified DFS Algorithm

1) All nodes are colored white
2) When a node is visited it is turned into red
3) Move on to descendants using DFS algorithm
4) When a node is visited completely it is turned into green
5) If we ever visit a red node during traversal then we have a cycle
Entered topic dependencies lead to a cycle. Please resolve them.
Learner can attempt an independent topic
Learner can attempt an independent topic

Independent topic: If topic is independent of all topics

Topic-A is independent of Topic-B iff
  - No edge from Topic-A to Topic-B or
  - All subtopics in topic-B are completed
Subtopic Module

- Content creation
- Subtopic dependency- Loop detection
Subtopic Module

- Content creation

- Subtopic dependency - Loop detection

- Learner can attempt an independent subtopic

- Independent subtopic: If subtopic is independent of all subtopics

- Subtopic-A is independent of Subtopic-B iff
  - No edge from Subtopic-A to Subtopic-B or
  - Subtopic-B is completed
Quiz Module

User → Web browser → Web Server → Session Manager

Authentication module → Quiz Module

student models: student1, student2, student_progress

Domain Base: question1, question2, question3

Controller Module

Domain Module:
- Course module
- Topic module
- Subtopic module

content queries

stores indexes

files

Physical Storage of Resources
Quiz Module

- Content creation
- Evaluation
- Update student knowledge
- Adaptation logic
Quiz Module

- Pop-up window for hands-on activities

![Pop-up window for hands-on activities](image)

**Figure: Pop-up window**
Controller Module

- Redirects to corresponding strategy’s quiz
- Uses strategy switching logic for student
• Existing Systems
• Guided Discovery Learning
• ITS Support for Guided Discovery
• Architecture and Modules
• Integration of the System
• Conclusion and Future Work
Integration of the System

- Common database
- Developed individual systems
- Controller module
- Strategy switching logic
Integration of the System-Adding new strategy

Adding new strategy
- Implement quiz module
- Edit controller module
- Edit strategy switching logic
**Sequence Diagram for student**

**Figure:** Sequence Diagram for student
Challenges

- Interdisciplinary area
- Non-existing features
- Choosing teaching-learning strategy
- Common database
- Mapping teaching-learning steps to software system
• Existing Systems

• Guided Discovery Learning

• ITS Support for Guided Discovery

• Architecture and Modules

• Integration of the System

• Conclusion and Future Work
Conclusion

- Developed ITS framework using guided discovery
- Integrated 3 strategies

Limitations
- MCQs only
- No collaborative learning
- Response time not considered
Future work

- Improved strategy switching algorithm
- Add more strategies
- Subjective questions- Latent semantic analysis
- Introducing Artificial Intelligence- SmartTutor
- Collaborative learning
- Response time
Thank You
Key References


Patrick Chipman, Andrew Olney, and Arthur C. Graesser. In The Autotutor3 Architecture, University of Memphis, USA.


Jong Suk Kim. The Effects of a Constructivist Teaching Approach on Student Academic Achievement, Self-concept, and Learning Strategy. Asia Pacific Education Review by Education Research Institute, Chungnam National University, Korea.


Richard E. Mayer. Should There Be a Three-Strikes Rule Against Pure Discovery Learning? California, Santa Barbara. University of

Erica Melis and J rg Siekmann. In ActiveMath: An Intelligent Tutoring System for Mathematics, German Research Institute for Artificial Intelligence, Germany.

Key References

Muska Mosston and Sara Ashworth. Teaching Physical Education. 2008.


National Extension Water Outreach Education. Explanation of Teaching Continuum.


Catherine J. Rezak. In Improving Corporate Training Results with Discovery Learning Methodology.