BUILDING AN AFFECT-SENSITIVE PEDAGOGICAL AGENT

ANDALLAZA | JIMENEZ
Computer interfaces capable of interacting with human users in a manner similar to typical face-to-face conversation (Cassel et al.)
INTELLIGENT TUTORING SYSTEM

A computer program that makes use of artificial intelligence to provide learners with individualized instruction
APLUSIX

• An ITS that aims to teach learners arithmetic and algebra

• Text editor allows students to solve items step-by-step
APLUSIX

• An ITS that aims to teach learners arithmetic and algebra

• Text editor allows students to solve items step-by-step
OBJECTIVES

• To have a significant influence in enhancing the learning experience of students when using an ITS such as Aplusix

• To determine what considerations will be needed in order to design, implement, develop, and test a motivational agent that can interact with the student on a real time basis
SIGNIFICANCE

• To contribute to machine emotional learning and apply it in the creation of an ECA to work with an environment such as Aplusix

• To pursue and integrate tools such as interface design, probabilistic learning, artificial intelligence reasoning and others in designing an affective agent
PREVIOUS WORK WITH APLUSIX

• Affect and Learning (Lagud)
  • Relation of learning models to the affective profiles of students

• Detection of Off-task Behavior (Bate)
  • Creation of models that determine when a student goes off-task

• ECA Design (Lim)
  • Pilot study of creating the agent
  • “Wizard-of-Oz” mechanism
METHODOLOGY

ECA ARCHITECTURE
ECA ARCHITECTURE
INPUT MANAGER

• Acquires input from all devices connected to the agent

• Converts input into forms that can be used by the other modules

• Our agent: Aplusix through text logs
DELIBERATIVE MODULE AND HARDWIRED REACTION

• The logical portion of the agent
• Answers the WHAT’s of the agent

DM vs. HR

• DM: prolonged analysis; more input before response
• HR: immediate response; responds per action done

• Our agent: Student models
ACTION SCHEDULER

- Motor controller of the agent
- Coordinates the responses of the agent to make it synchronous
  - Text, face, actions, voice, etc.
- Answers the HOW’s of the agent
- Our agent: Agent Face, Script, and Voice Clips
**ECA ARCHITECTURE**

**Input Manager**
- Provided the significant logs needed for on-task to off-task analysis.
- Provided the significant logs needed to identify affective states.
- Acquisition and filtering of input sent by Aplusix based on the significant logs needed.

**Deliberative Module**
- Provided the significant affective states for reactions.
- Provided the agent reactions based on affective states.
- Provided a means to evaluate on-task and off-task student actions based on logs.

**Action Scheduler Module**
- Provided a basis for common agent reactions and moods.
- Evaluated what specific logs will trigger an immediate reaction.

Lagud's Study, Lim's Study, Bate's Study, Our Study
METHODOLOGY

DESIGN OF THE PROTOTYPE
HOW THE AGENT WORKS

Affective Agent
Interacts with the student by accompanying with the intelligent tutoring system as a separate application.

Aplusix
An intelligent tutoring system that interacts with the student.

Evaluation

Student interacts

Computer
Medium for interaction

Aplusix interacts with Agent response

Student actions
Intelligent Tutoring System

Embodied Conversational Agent
RESULTS

ECA IMPLEMENTATION
RESULTS:
ECA IMPLEMENTATION

- ECA Implementation
  - Input Manager
    - Only takes in input from Aplusix
    - Per-action sending of logs
  - Hardwired Reactions
    - Instant reactions from the agent
      - Quitting the program
      - Solved to Not Solved State
ECA IMPLEMENTATION: LOGS
ECA IMPLEMENTATION: LOGS
### ECA IMPLEMENTATION: LOGS

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Step number</td>
</tr>
<tr>
<td>2</td>
<td>Action duration</td>
</tr>
<tr>
<td>3</td>
<td>Action done</td>
</tr>
<tr>
<td>6</td>
<td>Current expression</td>
</tr>
<tr>
<td>11</td>
<td>Resolution state</td>
</tr>
<tr>
<td>12</td>
<td>Problem type/level</td>
</tr>
</tbody>
</table>
RESULTS:
ECA IMPLEMENTATION

• ECA Implementation
  • Deliberative Module
    • Student models
    • Evaluation according to Bored, Confused, or Flow
    • Evaluation of test level speed and comprehension
    • Generation of logs appropriate for the evaluation
  • Action Scheduler
    • Program Interface
    • Determines when to show the responses depending on which module it receives data from
STATE ARCHITECTURE

Flow Input
Count <= MaxFlowCount
Duration <= MaxFlowDuration

Confused Input
Count <= MinConfusedCount
Duration <= MinConfusedDuration

Bored Input
Count > MinBoredCount
Duration > MinBoredDuration

Flow Input
Count <= MaxFlowCount
Duration <= MaxFlowDuration

Solved

START

FLOW

CONFUSED

BORED

CONGRATULATE

INTERVENE

Solved
<table>
<thead>
<tr>
<th>Code</th>
<th>Script</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>“Good Job!”</td>
</tr>
<tr>
<td>1</td>
<td>“That was fast! Good Job!”</td>
</tr>
<tr>
<td>2</td>
<td>“Keep Going.”</td>
</tr>
<tr>
<td>3</td>
<td>“Good Job!”</td>
</tr>
<tr>
<td>4</td>
<td>“You’re almost there.”</td>
</tr>
<tr>
<td>5</td>
<td>“I can see you worked hard. Good Job!”</td>
</tr>
<tr>
<td>6</td>
<td>“You can do it.”</td>
</tr>
<tr>
<td>7</td>
<td>“Don’t give up just yet.”</td>
</tr>
<tr>
<td>8</td>
<td>“Don’t quit. You have to try and finish every question.”</td>
</tr>
<tr>
<td>9</td>
<td>“Just focus on the problem.”</td>
</tr>
<tr>
<td>10</td>
<td>“Just keep going.”</td>
</tr>
<tr>
<td>11</td>
<td>“You already had the correct answer.”</td>
</tr>
</tbody>
</table>
DEMONSTRATION