Educational Technologies
WS2008

Pedagogical Components
To complete student modeling: ActiveMath Student Model Structure

Exercise 1

Metadata:
- difficulty:
- for: D
- prerequisite
Exercise-1

- **For:** concept addition of fractions
- **Prerequisite concepts:** fraction, common denominator, ..
- **Difficulty:** 0.7

Exercise-2

- **For:** common denominator of fractions
- **Prerequisite concepts:** fraction, common denominator, ..
- **Difficulty:** 0.4
Student Model, example

Exercise-1

Student solves exercise-1 correct/false (evidence)

Propagation to other Competency nodes

For each Competency C:

   IRT-measure for hypotheses Hi given parameters diff, discr.
      correct then look-up IRT = prob(Hi)
      Incorrect then look-up IRT = 1-prob(Hi)
   Take IRT-measure as weight for each hypothesis Hi in TBM

Exercise-2...

Dempster-Shafer Rule of aggregation => weight of Hi

Output hypothesis with highest weight for Competency
Plan of the Course

28.10.2008 Introduction and overview
04.11.2008: Intelligent tutoring systems (1) - Cognitive Tutors
11.11.2008: Intelligent tutoring systems (2) – ActiveMath
18.11.2008: Student modelling
25.11.2008: Student modelling
02.12.2008: Pedagogical components, instructional planning
  : Error diagnosis and feedback
09.12.2008: Meta-cognitive support (1) – Help
16.12.2008: Meta-cognitive support (2)
13.01.2009: Collaborative learning technologies
20.01.2009: Multi-Media Learning principles
27.01.2009: Web-based systems
03.02.2009: Educational data mining
23.02.2009: Project presentations by students
Overview, Pedagogical Components

▶ General
- Pedagogical strategies
- Realization in systems, exm
  - Mastery Learning
  - Learning by Teaching

▶ More of Intelligent Technologies
- Course planning – GTE
- Adaptive hypermedia -

▶ Course Generator of ActiveMath

▶ Woz study
Reminder: Generic ITS Architecture

Graphical user interface → Action Interpreter → Domain KR → Problem Solver → Feedback Generator → Student Model → Solution Graph → Solution Evaluator → Curriculum Planner → Curriculum Selector → Curriculum Planner

Source: Erica Melis
Educational Technologies WS 2008/09
Supporting the Learning Process

Support

- learning any time, everywhere
- different ways of teaching
- the individual

- Adapt learning materials (Course) with respect to
  - the individual (learning goals, knowledge, emotions, ...)
  - the context (location, ...)

Source: Erica Melis Educational Technologies WS 2008/09
Learning Theory

[Piaget/ Vysotski/ Bettencourt]

### Behaviourist instruction
- Passive learner
- External motivation
- Guidance along 1 solution
- No discussion
- Knowledge in isolation
- Solving competency only
- Static, known goal

### Constructivist instruction
- Active learner, interaction
- Internal motivation
- Responsible learner
- Personalised learning
- Exploration, discovery
- Authentic scenarios
- Knowledge in context
- Collaboration
- Goal finding
Learning Theories

- Goals
- Content sequence
- Strategies, Methods
- Media, tools
- Competencies

Zone of proximal development

- Didactic
- Mastery learning
- Socratic
- Constructivist
- Learning by teaching
- Collaborative*

Exercises: micro-adaptation
- Handling errors*
- Frequent mistakes
- Feedback*
- Multiple solutions

MultiMedia choice*

Szenarios/strategies
Some Pedagogical Tasks (Macro-level)

Pedagogical activity accomplished by instructor during pedagogical process

- teachConcept
- selectAdequateExercise, selectDifficultExercise, selectEasyExercise
- motivate
- helpAboutConcept
- teachPrerequisites
- challenge
- illustrate
Choice of Exercises/Problems

Zone of proximal development [Vygotsky 1978]

distance between actual level (independent ps) and potential level (under guidance or collaboration) development

Figure 3: Amanda’s Sample of Unassisted Writing

Figure 5: Amanda’s Sample of Scaffolded Writing (Teacher-Assisted Stage)
Adaptive Choice of Problems

- Students are not at ZPD if confused, bored, or at impasse.

Source: Eric Melis

Educational Technologies WS 2008/09
Mastery Learning: ActiveMath, Exercise Sequencer: Strategy Train-Competency-Level

1. Start ES
2. Start ES
3. Get Next Exercise
4. Show Exercise
5. Do exercise
6. Give up Or continue
7. ExerciseStepEvents
8. ExerciseStepEvents
Strategy: Train-Competency-Level

- Start ES
- Train same level
  - <= init CL
  - Solved < 3
- Check Competency Level
  - <= init CL
  - Solved >= 3
  - > Init CL
  - Or = 4
- Stop ES
- Train higher level
  - Train higher level
Learning by Teaching: Betty’s Brain [Biswas+ 2001-05]

Betty: You asked me: If Exercise increases, what happens to Heart Disease?
I think that if Exercise increases, Heart Disease decreases a lot. These are the paths I followed to get my answer. If you want me to explain my answer, click Explain.

Betty: I’ll explain how Exercise affects Heart Disease. An increase in Exercise causes Arterial Plaque to decrease a lot which causes Heart Disease to decrease a lot. Click Explain to hear more.
Reasoning techniques

- Relations in concept map
  - **Cause-effect**: Fish eat plants and Plants produce dissolved oxygen. Causal relations are further qualified by increase (+) and decrease (−) labels: eat implies a decrease relation
  - Needs
  - **Hierarchical relations** establish class structures to organize domain knowledge: variety of fish: trout... Relations inherited.

- Simple chaining procedure to deduce the relationship,
- Propagate +/- along path of related entities
  - ’needs‘ has no change effect.
  - Example: effect of more fish?

<table>
<thead>
<tr>
<th>Change</th>
<th>Relation</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>+</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td>−</td>
<td>+</td>
<td>−</td>
</tr>
<tr>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
</tbody>
</table>
Teachable Agents: Principles

- Use explicit and well-structured visual representations
- Enable the agent to take independent actions
- Model productive learner behaviors
- Include environments that support teaching interactions
Betty’s Question panel

- What will happen to B when we add/remove A?
- What will happen when we add/remove A?
- What can cause B to increase/decrease?

Get answer:

Explain result:

Reason 1: [Fish - Plants - Dissolved Oxygen] --> Dissolved Oxygen decreases a bit.

Reason 2: [Fish - Dissolved Oxygen] --> Dissolved Oxygen decreases.
Betty’s Brain 2007
BETTY in Guided-Discovery Video
More Technology: Adaptive Hypermedia
Adaptive Hypermedia

- Conditional text filtering
  - ITEM/IP, PT, AHA!, ActiveMath

- Adaptive stretchtext
  - MetaDoc, KN-AHS, PUSH, ADAPTS

- Frame-based adaptation
  - Hypadapter, EPIAIM, ARIANNA, SETA

- Output format generation via XSLT
  - ActiveMath
Conditional text filtering

- Similar to UNIX cpp
- Universal technology
  - Altering fragments
  - Extra explanation
  - Extra details
  - Comparisons
- Low level technology
  - Text programming

Fragment 1
If switch is known and user_motivation is high

Fragment 2

Fragment K
Example: Stretchtext (PUSH)

In this example, we perform and document an object-oriented analysis of a subsystem. The model should include the abstractions (represented as object types) necessary to understand how the subsystem described by the functional requirements is expressed in an object-oriented world. This analysis will render us a high-level view of the subsystem without any consideration (or at least as little consideration as possible) taken to distribution, persistence aspects or other design and implementation considerations. The goal is a model that clearly describes and gives an understanding of a subsystem without the gory details of design and implementation.

The ideal object model resulting from the ideal object modelling process, is functionally complete in the sense that it covers all areas of the functional specification of a subsystem.
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
Adaptive navigation support: goals

- **Guidance: Where I can go?**
  - Local guidance ("next best")
  - Global guidance ("ultimate goal")

- **Orientation: Where am I?**
  - Local orientation support (local area)
  - Global orientation support (whole hyperspace)
Adaptive navigation support

- **Direct guidance**
- **Restricting access**
  - Removing, disabling, hiding
- **Sorting of links (IR), relevance to current node**
- **Annotation**
- **Generation**
  - Similarity-based, interest-based
- **Map adaptation techniques**
Example: Adaptive annotation

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

- **Educational status for concept**
  - unknown
  - known
  - learned

- **Educational status for sections**
  - not ready to be learned
  - ready to be learned
  - suggested
Adaptive annotation in InterBook

1. State of concepts (unknown, known, ..., learned)
2. State of current section (ready, not ready, nothing new)
3. States of sections behind the links (as above + visited)
Sample index page (annotation)

<table>
<thead>
<tr>
<th>Доступные темы</th>
<th>2 Арифметические выражения</th>
<th>4 Безусловный переход на новую строку</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Общий вид формата</td>
<td>2 Арифметические выражения</td>
<td>6 Выбор позиции в строке</td>
</tr>
<tr>
<td>3 Удаление пустых строк</td>
<td>4 Безусловный переход на новую строку</td>
<td>8 Вывод поля</td>
</tr>
<tr>
<td>5 Переход на новую строку</td>
<td>6 Выбор позиции в строке</td>
<td>10 Безусловный литерал</td>
</tr>
<tr>
<td>7 Печать пробелов</td>
<td>8 Вывод поля</td>
<td>12 Арифметическая функция Val</td>
</tr>
<tr>
<td>9 Понятие MFN</td>
<td>10 Безусловный литерал</td>
<td>14 Арифметическая функция Rsum</td>
</tr>
<tr>
<td>11 Арифметическая функция L</td>
<td>12 Арифметическая функция Val</td>
<td>16 Арифметическая функция Rmax</td>
</tr>
<tr>
<td>13 Арифметическая функция Val</td>
<td>14 Арифметическая функция Rsum</td>
<td>18 Совмещение % и #</td>
</tr>
<tr>
<td>15 Арифметическая функция Rmin</td>
<td>16 Арифметическая функция Rmax</td>
<td>20 Условный литерал</td>
</tr>
<tr>
<td>17 Арифметическая функция Ravg</td>
<td>18 Совмещение % и #</td>
<td>22 Вывод MFN</td>
</tr>
<tr>
<td>19 Совмещение / и #</td>
<td>24 Префиксный условный литерал</td>
<td>26 Нуль-литерал</td>
</tr>
<tr>
<td>21 Повторяющийся литерал</td>
<td>28 Префиксный повторяющийся литерал</td>
<td>28 Префиксный повторяющийся литерал</td>
</tr>
<tr>
<td>22 Строковые выражения</td>
<td>29 Установка режима вывода</td>
<td>29 Установка режима вывода</td>
</tr>
<tr>
<td>25 Суффиксные литералы</td>
<td>30 Совмещение условных литералов с %</td>
<td>30 Совмещение условных литералов с %</td>
</tr>
<tr>
<td>27 Повторяющийся литерал с +</td>
<td>32 Совмещение условных литералов с /</td>
<td>32 Совмещение условных литералов с /</td>
</tr>
<tr>
<td>29 Установка режима вывода</td>
<td>34 Совмещение условных литералов с C</td>
<td>34 Совмещение условных литералов с C</td>
</tr>
<tr>
<td>31 Совмещение условных литералов с #</td>
<td>35 Совмещение условных литералов с M</td>
<td>35 Совмещение условных литералов с M</td>
</tr>
<tr>
<td>33 Совмещение условных литералов с C</td>
<td>36 Режимы L,U в команде M</td>
<td>36 Режимы L,U в команде M</td>
</tr>
<tr>
<td>35 Совмещение условных литералов с M</td>
<td>37 Режим H в команде M</td>
<td>37 Режим H в команде M</td>
</tr>
<tr>
<td>37 Режим H в команде M</td>
<td>39 Режим P в команде M</td>
<td>39 Режим P в команде M</td>
</tr>
<tr>
<td>39 Режим P в команде M</td>
<td>41 Строковая функция Ref</td>
<td>41 Строковая функция Ref</td>
</tr>
<tr>
<td>41 Строковая функция Ref</td>
<td>43 Программы пользователя format</td>
<td>43 Программы пользователя format</td>
</tr>
<tr>
<td>43 Программы пользователя format</td>
<td>Enter — изучить  F4-практ  F6-учи  F8-инд.задач  F9-назад PgDn-следстр.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+ Хорошо изучен  Изучен  Можно изучать  Не готов</td>
<td></td>
</tr>
</tbody>
</table>
Sample index page (hiding)
More Technology: Instructional Planning
Pedagogical Tasks

Pedagogical activity accomplished by instructor during pedagogical process

- teachConcept
- selectAdequateExercise, selectDifficultExercise, selectEasyExercise
- motivate
- helpAboutConcept
- teachPrerequisites
- challenge
- Illustrate
- give counter example
- practice
Course Generation in **active**\textit{math}.

- **“derivation”**

Course Generator

Repositories

- Discover “derivation”
- Train Competencies “derivation”
- Exam Simulation “derivation”
- Illustrate “derivation”
- Motivate “derivation”
Definition of the difference quotient

At a real function $f$ between two different points $P_0$ and $P_1$ one builds the differences $\Delta x = x_1 - x_0$, resp., $\Delta y = y_1 - y_0$, and calls the quotient

$$
\frac{\Delta y}{\Delta x} = \frac{y_1 - y_0}{x_1 - x_0}
$$

the difference quotient (to $x_1$ at $x_0$).
Knowledge Representation:
Ontology of Instructional Objects

• CG reasons about educational resources:
  *Does a resource with the given properties exist?*
    • Introduction for “derivative”
    • Prerequisite of “derivative”
    • Easy example for “sum rule”
    • ...

• Queries to repositories
• Format/Vocabulary of queries?
• CG: system & domain independent
Knowledge Representation:
Ontology of Instructional Objects
Definition

"ε-neighbourhood"

Theorem

"neighborhood-equality"

Proof

Explanation for "neighborhood-equality"

Example

Exercise for "ε-neighbourhood"
Framework for Course Generation

**Hierarchical Task Network planning (HTN)**

- very efficient
- straight-forward way for representing human expert knowledge.

**Planning problem:**

- task networks
- **methods** decompose non-primitive **tasks** into sub-tasks
Basics of Hierarchical Task Network Planning

- **Similar to classical AI planning**
  - World state represented by set of atoms
  - Actions correspond to state transitions

- **Differences to classical AI planning**
  - What it plans for: sequence of actions that perform a task network
  - How it plans:
    - methods decompose tasks
    - down to primitive tasks performed by operators
Representing a Pedagogical Method

Describes how to achieve a task

(:method (teachConcept ?c)
  ((problem ?p (for ?c)))
  (!startSection bookTitle)
  (provideSituation ?p)
  (teachSubConcepts ?p)
  (provideResult ?p)
  (!endSection)))

Goal task
Precondition
Sub tasks
Basics of HTN-Planning:

Tasks: Activities to perform
- Primitive/Compound
  (taskSymbol, terms)
- Generate a course of type “discover” about the concept “derivation”:
  (discover deriv)
- Insert the resource “exc_deriv” in the course: (!insertResource exc_deriv)

Operators: Perform primitive tasks
(task, precondition, delete list, add list)

(:operator (!insertResource exc_deriv) ;; precondition
 ()); ;; delete list
 ()); ;; add list

Source: Erica Melis  Educational Technologies WS 2008/09
Carsten Ullrich – AMSS, 21.4.2008
Basics of HTN-Planning

Methods: Decompose compound tasks

- \((task, \text{precondition}, \text{subtasks})\)
- \(\text{(:method } insertResourceOnce! \ ?r)\)
  - ((not (inserted \ ?r)))
  - ((!insertResource \ ?r)))

Axioms: Infer preconditions not asserted in world state

- \(\text{(:- } \text{head } \text{tail})\)
- \(\text{(:- (known } \?f)\)
  - ((learnerProperty hasCompetencyLevel \ ?f \ ?cl)
    - (>= \ ?cl 3)))

JShop: Call terms allow to use Java functions during planning
Adapting to the Learner’s State

▶ **Mastery:**
  ▶ Train insufficiently mastered competencies
  ▶ Selecting an exercise from the adequate competency level

▶ **Motivation**

```
(:method (motivate ?c)
  ((problem-solving_comp high))
  ((presentEasyExercise ?c))
)
```

```
(:method (motivate ?c)
  ((problem-solving_comp low))
  ((presentEasyExample ?c))
)
```
Generating structure

(:operator (startSection ?type)
  ()
  ()
  ()
  ()
)

(:operator (endSection)
  ()
  ()
  ()
  ()
  ()
)
Accessing Information about Resources

Call term \((\text{GetResources} \ (\text{mediatorQuery} \ ))\)
returns a list of all resources fulfilling the query

\((\text{call GetResources}
\quad ((\text{class Exercise})
\quad \quad \text{(relation isFor def_slope)}))\)

GetRelated, Sort
Reasoning using Learner Properties

Information about the learner: learnerProperty

```prolog
( :- (known ?f)
     ((learnerProperty
       hasCompetencyLevel ?f ?cl)
      (>= ?cl 3)))
```

- allKnownOrInserted, readyAux, removeKnownFundamentals, sortByAlreadySeen, ...
Exercise Selection

• Selecting an exercise, high motivation

\[(\text{learnerProperty hasMotivation ?c ?m}) \\geq \ ?m \ 4)\]

\[(\text{learnerProperty hasField ?field})\]

\[(\text{learnerProperty hasEducationalLevel ?el})\]

\[(\text{learnerProperty hasCompetencyLevel ?c ?cl})\]

\[(\text{equivalent (call + 1 ?cl) ?ex-cl})\]

• Selecting an exercise, adequate competence level

\[(\text{learnerProperty hasField ?field})\]

\[(\text{learnerProperty hasEducationalLevel ?el})\]

\[(\text{learnerProperty hasCompetencyLevel ?c ?cl})\]

\[(\text{equivalent ?cl ?ex-cl})\]
Dynamic Adaptivity

▶ Early course generation supports
  ▶ orientation
  ▶ self-organization

▶ Problem:
  changing capabilities of the learner

⇒ Delayed Task Execution:
  ▶ planning stops at the level of specially marked tasks
  ▶ instantiation at first visit

▶ New possibilities for authors!
Concept Map Exercise on continuity and differentiability

Find examples for the nodes “function”, “continuous”, and “differentiable” and add them to the Concept Map. Connect each example with a for-edge with the correct concept it is for.

Start exercise

Normals of parabolas - step by step

Compute step by step the slope of the normal for a quadratic function.

Start exercise

Computation of the derivative

At which points is the red function differentiable? Compute its derivative in all of these points.

Start exercise
Definition of the difference quotient

At a real function $f$ between two different points $P_0$ and $P_1$ one builds the differences $\Delta x = x_1 - x_0$, resp., $\Delta y = y_1 - y_0$, and calls the quotient

$$\frac{\Delta y}{\Delta x} \ (x_0, x_1) = \frac{y_1 - y_0}{x_1 - x_0}$$

the difference quotient (to $x_1$ at $x_0$).
Definition of the derivative, resp., differential quotient

Rehearse > Definition of the derivative, resp., differential quotient > Definition of the derivative, resp., differential quotient

Do you still remember what the goal content is about? Here you can have a second look at it.

Definition of the derivative, resp., differential quotient

A function $f$ is called **differentiable at** $x_0$ if the limit

$$
\lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0}
$$

exists. This limit is called the **derivative of** $f$ at $x_0$, resp., the **slope of the graph of** $f$ at $x_0$, and is denoted by $f'(x_0)$ (say: “$f$-prime of $x_0$”):

$$
f'(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0}.
$$
Tutorial Planning in a Learning Environment

- NextBest
- Learner
- OLM
- XYZ

Tutorial Tasks

Control

Table of Content

Presenter

Tutorial Planner

Selects which request to present

Tutorial Task
Open Challenges

Open challenges:

- Determine instructional type automatically
- Mobile learning: using the learner’s location
- Using the learner’s private memory
- Recognizing learning situations automatically
Scenario “Discover”

(discover deriv)

Introduce

:method (discover ?f)
()
((!startSection Discover ?f)
 (descriptionScenarioSection ?f)
 (learnFundamentalsDiscover ?f)
 (reflect ?f)
 (lendSection)))

Develop

:method (learnFundamentalDiscover ?c)
()
((!startSection Title (?c))
 (introduceWithPrereqSection ?c)
 (developFundamental ?c)
 (proveSection ?c)
 (practiceSection ?c)
 (showConnectionsSection ?c)
 (lendSection)))

Prove

Practice

Connect

Carsten Ullrich – AMSS, 21.4.2008
Scenario “Discover”

(introduceWithPrereqSection! deriv)

Carsten Ullrich – AMSS, 21.4.2008
Scenario Discover

(:method (motivate! ?c)
  ((learnerProperty hasEducationalLevel ?el)
   (learnerProperty hasAnxiety ?c ?an)
   (?an <= 2)
   (GetElement
     ((class Exercise)
      (class Introduction)
      (relation isFor ?c)
      (property hasLearningContext ?el)
      (property hasDifficulty very_easy)))))

((insertAuxOnceIfReady! ?element)))

Carsten Ullrich – AMSS, 21.4.2008
Wizard of Oz

- Apply teaching strategy/adaptations suited for learning a concept
- How intuitive is adaptation for learner?
- How useful is adaptation for learner?