Teaching Java with the assistance of harvester and pedagogical agents

D. Mitrović, M. Ivanović, Z. Budimac, Lj. Jerinić

Agenda

- Introduction and background
- Harvesting learning material
- Helpful and misleading pedagogical agents
- Current and future developments
The goal

- Develop an adaptive and intelligent web-based educational system
- Incorporate different recommendation and scaffolding techniques in order to:
  - Increase motivation for learning
  - Suggest additional learning activities
  - Determine the optimal browsing pathways to students, based on their different preferences and levels of knowledge
- Our main objective is to adapt and personalize learning to the needs of a student as much as possible

Segmented personalization

- Currently, we are focused on the segmented personalization style of learning
- Students are grouped into smaller, identifiable and manageable clusters, based on their preferences, survey results, and common attributes (e.g. class and age)
- Instructions are tailored to these groups, and are applied in the same or similar way to all members within a single group
**Code completion tasks**

- The proposed system relies on the *code completion* tasks to assess the student’s progress
- The student is given a code snippet with some parts missing
- He/she is then requested to complete the code in order to meet the program specification
- Code completion tasks are well-suited for both testing and improving the student’s programming skills, because they require a thorough understanding of the underlying programming concepts

---

**Software agents**

- One of the most consistent approaches to distributed computing
- Autonomous software entities
- Various degrees of intelligence
- Capable of exhibiting reactive and pro-active behavior
- More complex agent architectures incorporate mental attitudes, such as beliefs, desires, and intentions
Incorporating agents into e-learning

- For the purpose of this research, we distinguish two types of agents:
  - *Pedagogical agents*: engage, motivate, and guide students through the learning process
  - *Harvesters*: collect learning material from, often, heterogeneous learning resources

- Additionally, we introduce two specialized sub-types of pedagogical agents: *helpful* and *misleading*

---

Agenda

- Introduction and background
- Harvesting learning material
- Helpful and misleading pedagogical agents
- Current and future developments
System overview

Harvesting learning material

- Harvesters: agents that collect learning material from heterogeneous repositories

- With the abundance of programming examples available on the web, harvester agents have been embodied in web crawlers

- To optimize the harvesting process, crawlers rely on a number of agent-based concepts, including inter-agent communication, mobility, etc.
Classifying the source code

- An important feature of the proposed system is that it enables teachers to work with only a sub-set of the harvested learning material at a time.

- This is important because a large pool of Java source code examples may be collected.

- The Classifier module automatically associates each Java source code example with a concrete lecture topic.
  - E.g. “for-loops”, “classes and interfaces”, “inheritance”

Agenda

- Introduction and background
- Harvesting learning material
- Helpful and misleading pedagogical agents
- Current and future developments
Pedagogical agents

- Lifelike characters presented on a computer screen that guide users through multimedia learning environments
- One of the main characteristics of pedagogical agents is interactivity
  - They not only provide answers and additional learning material, but also ask questions and propose solutions
- The agents track a set of information about the student, including the ratio of correct and incorrect solutions to each code completion problem, student's grade for each lecture topic, etc.

Useful and misleading agents

- The helpful pedagogical agent, as expected, provides useful hints for the problem in question
- The purpose of the newly introduced misleading agent is to try and steer the learning process in the wrong direction
- Both agents are hidden behind the same interface
  - The student is never sure with which agent(s) he is interacting
- This novel approach encourages students not to follow the agent’s/tutor’s instructions blindly, but rather to employ critical thinking and, in the end, they themselves decide on the proper solution to the problem in question
### Useful and misleading hints – example 1

```java
class Fib {
    public static void main(String[] args) {
        int[] f = new int[10];
        /* for loop goes here */
        print(f);
    }
}
```

👍 “What should be the starting index? Remember that the first element of the Fibonacci sequence has the index 0, while the expression for calculating other elements is \( f_i = f_{i-1} + f_{i-2} \)"

👎 “What should be the starting index? Hint: the first element of the Fibonacci sequence is often denoted as \( f_0 \)”

### Useful and misleading hints – example 2

```
class Rect {
    private float x, y, width, height;
    public Rect(float x, float y, float w, float h) {
        this.x = x;
        this.y = y;
        width = w;
        height = h;
    }
    /* calculate area */
}
```

👍 “To calculate the area, you need width and height, both of which are available as fields. Remember that a method can access its object’s fields without limitations!”

👎 “To calculate the area, you need both width and height. Make sure your method has access to these values!”
Agenda

- Introduction and background
- Harvesting learning material
- Helpful and misleading pedagogical agents
- Current and future developments

The Acra (sub-)project

- https://code.google.com/p/acra-project/
- Harvester agents implemented as distributed web crawlers (Java, RMI, GWT, GAE)
- Apache License 2.0
- Web app:
  http://acraweb.appspot.com/
Integration into Protus

- Protus – PRogramming TUtoring System
- Uses principles of learning style identification and content recommendation for course personalization, utilizing Semantic web technologies
- Described in details at the 12th Workshop “Software Engineering and Reverse Engineering”

Thank you for your attention

- Questions?
- Suggestions?