PETCHA -
A Programming Exercises Teaching Assistant

Ricardo Queirós and José Paulo Leal

July 05, 2012

ITiCSE’12
Conference on Innovation and Technology in Computer Science Education
Outline

Introduction
  Context
  Approach

Related Work
  Exercises Authoring
  Automatic Evaluation of Exercises

PETCHA
  Use Cases
  Architecture
  Workflow
  GUI

Evaluation
  Experiment
  Results

Conclusions
Context

- Programming courses have high failure and dropout rates
  - teaching methods
  - subject complexity
  - student motivation

- Teaching and learning computers programming is difficult
  - number of Teaching Assistants (TA) insufficient
  - high number of students enrolled in programming courses
  - extensive course syllabus
  - weak availability of repositories of problems
Many educators claim that "learning through practice" is by far the best way to learn computer programming and to engage novice students.

**Approach:** create an automatic TA for programming exercises

- **Teaching Assistant as a scaffolding tool with 2 facets**
  - bridge between the teacher and the students
    - it helps the teacher in exercise authoring
    - it helps the students in the exercise solving
  - pivot component on a network integrating other e-learning systems
    - automatic evaluation of programs and feedback generation
    - authoring and storing of programming exercises as learning objects
    - managing instruction and learning activities
Exercises Authoring

- Mostly of the programming problems are
  - created manually or using ad hoc mechanisms
  - described with its own exercise format
  - stored in proprietary systems (e.g. online judges)
  - used in programming contests

- Programming problems interoperability
  - lack of a content standard (e.g. FPS, Peach, CATS, MEF)
  - lack of a communication standard (e.g. Evaluate specification)
  - use of conversion web services (e.g. BabeLO)
Assessment Systems

- high number of assessment systems (e.g. Mooshak, Web-CAT, Verkkoke, DOMJudge, Boss)

- main features
  - multi-programming language support
  - evaluation type (static or dynamic)
  - feedback
  - interoperability
  - learning context
  - security
  - plagiarism
Use Cases

- Teacher
  - create exercise
  - deploy exercise in repository
  - configure activity in LMS
  - provide exercises

- Student
  - solve exercises
  - execute activity in IDE
  - select activity in LMS

Architecture

Architecture of the network coordinated by Petcha:

- Learning Objects Repository - to store/retrieve exercises
- Assessment System - to evaluate student exercises
- Learning Management System - to present exercises to students
- Integrated Development Environment - to code the exercises
Integration of Petcha with other systems rely on several specifications

- **Content**
  - programming problem described using PExIL
  - PExIL integrated as a LAO resource in an IMS CC package

- **Communication**
  - IMS LTI - bidirectional communication with LMSs
  - IMS DRI - communication with repositories
  - Evaluate - communication with assessment systems
Workflow

LMS

TA

LOR

AS

Launch()

XML search(URL collectionId)

XML exercises

ERL listCapabilities()

ERL capabilities

{'accepted': False}

ERL Evaluate(URL loid, String attempt, String capability, String lang)

LO Request(URL loid)

LO

ERL Ticket and Report

ERL GetReport(String ticket)

ERL Ticket and Report

Report(URL loid, LOReport rep)

ReplaceResult(Lis_source_id, Grade)
GUI

PETCHA - A Programming Exercises Teaching Assistant

Ricardo Queirós and José Paulo Leal
Experiment

▶ conducted at ESEIG - a school of the Polytechnic Institute of Porto
▶ first-year Mechanical Engineering students of the course Algorithms and Programming
▶ the course is divided in two classes (experimental and control)
▶ the experiment occurred in 6 lab sessions
▶ in each session students had 3 exercises to solve
▶ after each lab session both classes were surveyed
Survey results average

<table>
<thead>
<tr>
<th>Assertions</th>
<th>Experimental group</th>
<th>Control group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercises started</td>
<td>89%</td>
<td>81%</td>
</tr>
<tr>
<td>Exercises completed</td>
<td>83%</td>
<td>74%</td>
</tr>
<tr>
<td>Exercises effectively solved</td>
<td>82%</td>
<td>66%</td>
</tr>
<tr>
<td>Exercises with feedback</td>
<td>59%</td>
<td>62%</td>
</tr>
<tr>
<td>Feedback helpfulness</td>
<td>55%</td>
<td>62%</td>
</tr>
</tbody>
</table>

- using Petcha students start, complete and effectively solve more exercises
- feedback provided by Petcha is less effective than of a human TA
  - clearly room for improving automatic feedback in Petcha
  - it is very difficult to replace a human TA
  - automated feedback is still a remedy for situations where a human TA is not available
Conclusions

Contributions

- concept, design and implementation of a tool acting as a TA for computer programming classes
  - mediate between the teacher and the students
  - act as an integrator of the best-of-bread systems involved in the process of automatic evaluation of programming exercises

Future work

- support for MathJax for displaying math expressions
- improved visualization of evaluation reports and statistical data on student activity
- improve automated feedback
Petcha is currently being used in the practical classes of an undergraduate programming course
More details: http://ensemble.dcc.fc.up.pt

Thanks for your attention

Ricardo Queirós
ricardo.queiros@eu.ipp.pt