Labrador: A Tool for Automated Grading Support in Multi-section Courses

Drexel University Programming Learning EXperience (DUPLEX)

Departments of Mathematics† and Computer Science‡

http://duplex.mcs.drexel.edu

Christopher D. Cera‡, Jeffrey L. Popyack‡, Bruce Char‡, Robert N. Lass‡,
Nira Herrmann†, Paul Zoski†, Aparna Nanjappa‡

May 19, 2003
Roadmap

• Introduction

• Problems and Solution Goals

• Labrador

• Discussion
The Duplex Project: An Overview

- Take advantage of advances in Information Technology to improve instruction and reduce costs for computer programming courses

- Modular Structure
  - Multiple Entry Points
  - Multiple Audiences
  - Multiple levels of knowledge (Bloom’s Taxonomy)

- Computer Supported Cooperative Work (CSCW) in student labs

- Online Services - Today’s Topic
Course Redesign

- Emphasis on online materials

- WebCT introduced for:
  - Delivering online course materials
  - Chat/Discussion groups
  - Quizzes/Labs
  - Electronic submission, grading, and return of assignments

- Issues: WebCT Interface does not handle all course needs

- Solution: Labrador
Who Am I?

- First TA in MCS to experiment with WebCT in December 2000
- Migrated course content to WebCT from previous course website
- HTML assignments and labs into question database
- Gradebook maintainer
- Wrote demo’s and documentation to train other TAs
- Developer of software supplements to WebCT to support course administration
Roadmap

- Introduction
- Problems and Solution Goals
- Labrador
- Discussion
Before WebCT

- Hundreds of people were involved in paper exchanges of handwritten assignments and quizzes
- Testing programs required floppy exchanges
- No Chat
- No newsgroup-style threads
- Feedback and grades are not online
WebCT

- Service from Information Resources and Technology (IRT) (since v3.1)
- MCS started Dec. 2000 (v3.5)
- Currently using version 3.8.3
Features

- General Course Website
- Centralized Administration
- Labs: Online Quizzes with Automated Grading
- Homework: Online Assignments
- Joint Staff–Student Chat
- Discussion Threads
Large Classes and WebCT

- Even with WebCT ..., still some difficulties:
  - Bulk download of assignment files and quizzes for grading
  - Handling select sections requires searching and clicking
  - Files submitted in compressed, archived, or encoded formats are tedious to unpack manually
  - Transferring data to other systems
Software Design Goals

- Bulk Assignment Downloading (prior to 3.7)
- Bulk Quiz Downloading
- Section Sorting for bulk downloads, or only one section
- Post-Processing student files
  - Archive Extraction (tar, zip)
  - Decompress (gz, zip)
  - Decode (uue)
- Minimal staff intervention when transferring submissions between systems, eg. Plagiarism Detection Systems
Software Design Goals  [continued]

- Automatically collate source code
- Generate electronic documents to facilitate grading and archiving
- Upload grades and marked-up documents
- Remote Execution downloading to computer $x$ (on campus) while operating at computer $y$ (off campus)
The Bigger Picture

- Course-Specific Tasks

- Not all processing should be done on the server

- Select files have to be transferred to a different system for further processing and analysis by staff

- Client-side support is needed to perform this, preferably automated and not necessarily using a web browser.
Roadmap

- Introduction
- Problems and Solution Goals
- Labrador
- Discussion
Labrador: Our Solution

• Client-side WebCT Supplement

• Cross-platform

• Works for users with TA and Designer access to WebCT
Labrador Execution Timeline

- Submission Downloader
- Section Sorting
- Post-Processing
- PDF Generation
  - JPlag
  - Moss
- Submission Uploader
User Interface

• Different users / Different UI preferences

• GUI

• Command-line

• Interactive

• Configuration File
Bulk Downloader

- Web-Crawler: simulates clicks of actual staff member
- Parses HTML to find desired text or URLs to crawl to next
- Works on assignments and quizzes
- Only component in Labrador which interacts with WebCT
Required Information


Username/Password               st96k9ry

Server URL                       webct.drexel.edu

Course ID                        CS164_Fall2001

Submission Name or ID            Recursion II or 1006285909

Optional Username List          unames.txt
Organizing Submissions by Section

- Organizes each student’s submissions into a separate folder for each section.

- How to tell Labrador the sections:
  - Creating a Section column in gradebook
  - “Username, Section” CSV File
  - Username file
PDF Generation for Electronic Mark-up

- Adobe Portable Document Format is available on all major platforms
- With Adobe Acrobat, PDFs can be annotated by graders
- Sony VAIO Slimtop PC (PCV-LX920)
- Toshiba Tablet PC: Protege 3500
- Acer Tablet PC
- Wacom Pen Tablet
Sample Mark-up Options
PDF Markup Example

```
#include <iostream>
#include <cmath>
#include <string>
#include <iomanip>
#include <sstream>
#include <stdio.h>
#include "Money.h"
#include "SavingsAcct.h"

using namespace std;

int main()
{
    double wutever;
    cout << "Please enter amount: ";
    cin >> wutever;
    int dollars = wutever/100;
    int cents = wutever%100;

    Money amount;
    amount.setDollars(dollars);
    amount.setCents(cents);

    cout << "Dollars" << amount.getDollars() << " ";
    cout << "Cents" << amount.getCents() << " ";
    cout << "TotalCents" << amount.getTotalCents() << " ";
    amount.addCents(56);
    cout << amount;
    amount.addDollars(3);
    cout << amount;
    amount.setCents(0);
    cout << amount;
    cout << "string" << amount.toString() << " ";

    //*************test features of overloaded operators*************/
    int wutever2;
    cout << "Please enter amount (test): ";
    cin >> wutever2;
```
Demonstration

Startup screen prompts for the username and password
Demonstration [continued]

TA enters username and password
Demonstration [continued]

Labrador prompts for the course name and optional student list
Demonstration  [continued]

TA enters course name and student list
Labrador prompts for the submission type
Demonstration [continued]

TA selects assignments
Demonstration [continued]

TA selects post-processing
Demonstration [continued]

TA selects PDF generation
Demonstration [continued]

Labrador prompts for the specific assignment
TA selects “Practice Assignment” and begins downloading
Demonstration [continued]

Labrador notifies the TA that the job is complete
Exploded view of TA’s folder
Redistribution

- How can we return annotated PDF’s back to students using WebCT?
- Version 3.8 addresses this issue
- Labrador supports this upload feature

<table>
<thead>
<tr>
<th>Files</th>
<th>Modification Date</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>student.pdf</td>
<td>May 19, 2003 1:25am</td>
<td>31.6 kB</td>
</tr>
</tbody>
</table>

To upload the graded files for this student, click Upload file.

Graded by Chris Cera

3 out of 3
Labrador Applications

- Interface between WebCT documents and other software
  - Decompressing files
  - Reformatting files for grading (PDF)
  - Submission to Plagiarism Detection Software (Moss/JPlag)
  - Other third party software programs
  - Returning processed/graded documents to WebCT

- **Primary Issue**: Compatibility with heterogeneous systems
Heterogeneous Systems

- Each system requires data to be packaged in a different way
- Plagiarism Detection Systems
- Moss [1] and JPLAG [2] have been used extensively
- Other processes (e.g. PDF generator)
- Future work: Automatic program compiling and testing
Automated Plagiarism Detection

- Digital formats make "borrowing" easy
- Browsing similar works needs a simple and quick user interface.
- Careful review by faculty to assess results and present to students
Moss [1]

- C, C++, Java, ML, Lisp, Scheme, Pascal, and Ada
- Common code feature reduces false positives
double a3;
if((a1 + a2) >= 180)
{
    side1_ = 0;
    side2_ = 0;
    side3_ = 0;
    cout << "The information entered will not
   
} else
   // the angles of a triangle always equal 180
   a3 = (180 - (a1 + a2));

   // law of sines is used to find the missing sides
   side2_ = ((side1_ * (sin(((a1 * PI) / 180))) / sin((angle1 + angle2))));
   side3_ = ((side1_ * (sin(((a2 * PI) / 180))) / sin((angle1 + angle2))));

   angle2 = sora3;
   if((angle1 + angle2) >= 180)
   {
       side1_ = 0;
       side2_ = 0;
       side3_ = 0;
       cout << "This is not a triangle."
   } else
   angle3 = (180 - (angle1 + angle2));

side2_ = (side1_ * (sin(((angle1 * PI) / 180))) / sin((angle1 + angle2)));
side3_ = (side1_ * (sin(((angle2 * PI) / 180))) / sin((angle1 + angle2)));

Moss [1] Interface
MossClique Interface

http://duplex.mcs.drexel.edu/software/MossClique.zip

1. Student126
   - Student47: 81%
   - Student19: 26%
   - Student43: 26%
   - Student109: 23%
   - Student128: 15%
   - Student124: 11%
   - Student30: 11%
   - Student63: 11%
   - Student97: 11%

2. Student47
   - Student126: 81%
   - Student19: 26%
   - Student43: 26%
   - Student109: 23%
   - Student128: 15%
   - Student30: 12%
   - Student63: 12%
   - Student124: 11%
   - Student143: 11%
   - Student97: 11%

3. Student141
   - Student12: 77%
   - Student70: 71%

4. Student12
   - Student141: 77%
   - Student70: 71%
JPlag [2]

- C, C++, Scheme, and Java
- For plain text files, it matches a user specified number of words appearing in succession
- Could be used for any course grading written (text) documents
- http://wwwipd.ira.uka.de:2222/
### JPlag [2] Interface

<table>
<thead>
<tr>
<th>Student ID</th>
<th>Similarity (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>student11</td>
<td>40.9%</td>
</tr>
<tr>
<td>student86</td>
<td>40.3%</td>
</tr>
<tr>
<td>student2</td>
<td>35.8%</td>
</tr>
<tr>
<td>student73</td>
<td>28.1%</td>
</tr>
<tr>
<td>student151</td>
<td>25.3%</td>
</tr>
<tr>
<td>student88</td>
<td>23.0%</td>
</tr>
<tr>
<td>student96</td>
<td>36.0%</td>
</tr>
<tr>
<td>student49</td>
<td>34.1%</td>
</tr>
<tr>
<td>student73</td>
<td>28.5%</td>
</tr>
<tr>
<td>student151</td>
<td>24.5%</td>
</tr>
<tr>
<td>student91</td>
<td>22.5%</td>
</tr>
<tr>
<td>student22</td>
<td>30.1%</td>
</tr>
<tr>
<td>student75</td>
<td>29.5%</td>
</tr>
<tr>
<td>student49</td>
<td>29.4%</td>
</tr>
<tr>
<td>student73</td>
<td>24.3%</td>
</tr>
<tr>
<td>student151</td>
<td>24.3%</td>
</tr>
<tr>
<td>student49</td>
<td>24.9%</td>
</tr>
<tr>
<td>student73</td>
<td>21.7%</td>
</tr>
<tr>
<td>student100</td>
<td>22.3%</td>
</tr>
<tr>
<td>student28</td>
<td>21.4%</td>
</tr>
<tr>
<td>student01</td>
<td>21.6%</td>
</tr>
<tr>
<td>student107</td>
<td>21.6%</td>
</tr>
</tbody>
</table>
Roadmap

- Introduction
- Problems and Solution Goals
- Labrador
- Discussion
Recent WebCT Enhancements

- Relevant to this talk:
  - 3.7 Addressed bulk download issue for assignments
  - 3.8 Attaching documents to an assignment
Future WebCT Enhancements

- Power users will need functionality not yet supported
- Every domain will also require additional functionality
- Not feasible for all domain-specific functionality to run on the WebCT server
HTTP: Insufficient For Data Interchange

- Was designed for visual content
- Heavy client interaction
- An HTTP based approach is sensitive to the exact location of web pages, and format of text within them
- One possible solution is ...
API for non-administrators

- Stateful protocol so clients can be built by 3rd parties
- Could be coupled with efforts of the “semantic web” (e.g. RDF)
- This would eliminate the need for Labrador to “screen-scrape” text from web pages
Labrador Availability

• Contact Us

http://duplex.mcs.drexel.edu
The Group

Bruce Char
Nira Herrmann
Jeffrey L. Popyack
Paul Zoski
Christopher D. Cera
Robert N. Lass
Aparna Nanjappa
Derek Rosenzweig
Jasper Zhang

Professor, Computer Science
Professor and Head, Mathematics
Associate Professor, Computer Science
Instructor, Math and Computer Science

Computer Science Graduate Student
Computer Science Undergraduate Student
Computer Science Graduate Student
Computer Science Undergraduate Student
Computer Science Undergraduate Student
Project Support

- National Science Foundation, Division of Undergraduate Education, *DUE* — 0089009

- The Pew Learning and Technology Program at the Center for Academic Transformation

- The Ramsey-McCluskey Family Foundation, Margaret Ramsey, ’84

- Drexel University
References


Related Work


Related Work  [continued]


