Tools and Techniques for Large-Scale Grading using Web-based Commercial Off-The-Shelf Software

Drexel University Programming Learning EXperience (DUPLEX)

Departments of Mathematics[†] and Computer Science[‡]

http://duplex.mcs.drexel.edu

Robert N. Lass[‡], Christopher D. Cera[‡], Nathaniel T. Bomberger[‡], Bruce Char[‡]

Jeffrey L. Popyack[‡], Nira Herrmann[†], Paul Zoski[†], Aparna Nanjappa[‡]

July 2, 2003

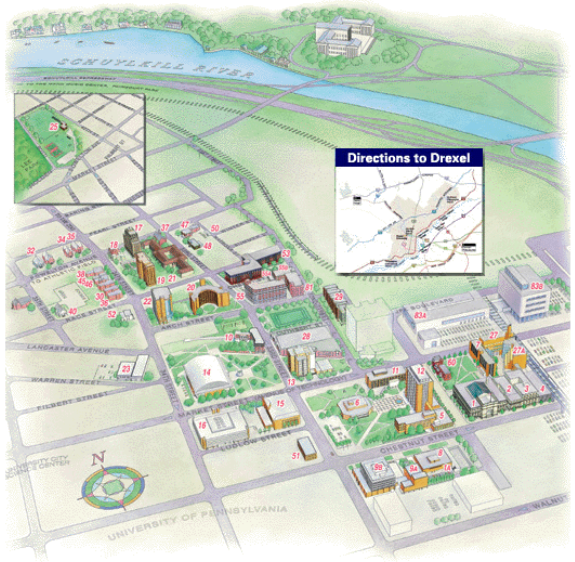
Presented by Christopher D. Cera





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*

Redesigning the Course

- Variety of Majors
  - Computer Science
  - Computer Engineering
  - Information Systems
  - Mathematics
  - Digital Media
- Class
  - 1 x 1 hour lecture
  - 1 x 2 hour lab
- People
  - 250-300 Students
  - 2-3 Professors
  - 10-12 Teaching Assistants
- Emphasis on online materials
- Course Management System (CMS) introduced for:
  - File sharing among staff or between staff and students
  - Centralized repository of materials and student work
  - Chat/Discussion groups
  - Quizzes/Labs
  - Electronic submission, grading, and return of assignments
- Issues: CMS Interface does not handle all course needs
- Solution: Labrador

Generality of a CMS

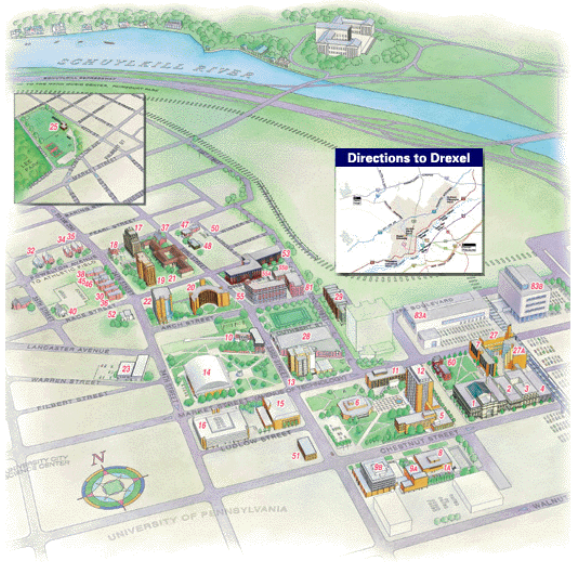
- Cannot yet offer features that would only benefit teaching staff in a limited domain
- For instance: *source code plagiarism detection*
- Interfaces should exist to support interoperability with other systems
- Our discussion of CMSs pertains to features provided by most major vendors

Automation and Management in CS

- "Homegrown" Systems [8, 9, 6, 2, 3]
- Computer-Aided Assessment and Interactive Tutoring [6, 4]
- Our efforts focused writing software to interact with the 3^{rd} party CMS supported by our university
- Can easily write web software to automate repatitive tasks
- This approach conserves our resources, since we only have to administer our client program and not the CMS itself
- Similarly, could interact with open source efforts (such as the Open Knowledge Initiative (OKI) [5])

Roadmap

- Introduction
- Problems and Solution Goals
- Labrador
- Discussion



Before and After Introducing CMS

- Before
  - 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
- After WebCT $_{\text{v3.5} \rightarrow \text{v3.8.3}}$ [10]
  - General Course Website
  - Centralized Administration
  - Labs: Online Quizzes with Automated Grading
  - Homework: Online Assignments Provide Timestamped Submissions
  - Joint Staff–Student Chat
  - Discussion Threads
  - Password Protected Grades

Large Classes and WebCT

- Even with WebCT .... still some difficulties:
  - Bulk download of assignment files and quizzes for grading (clicking for each file is required)
  - Handling select sections requires searching
  - Files submitted in compressed, archived, or encoded formats are tedious to unpack manually
  - Transferring data to other systems (ie. JPlag [7] and Moss [1])

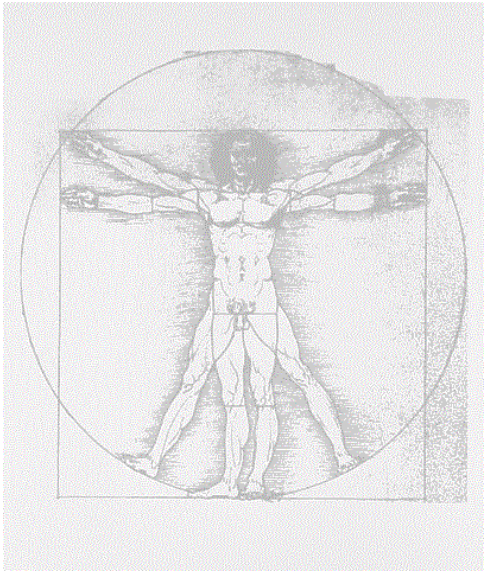


Software Design Goals

- 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
- 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)

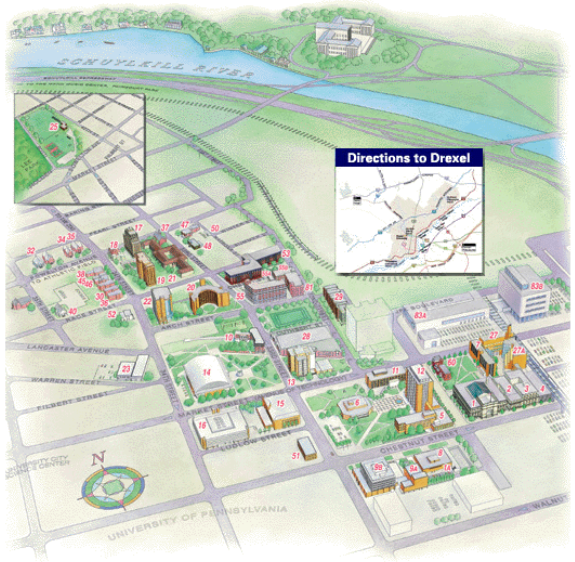
The Bigger Picture

- 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.



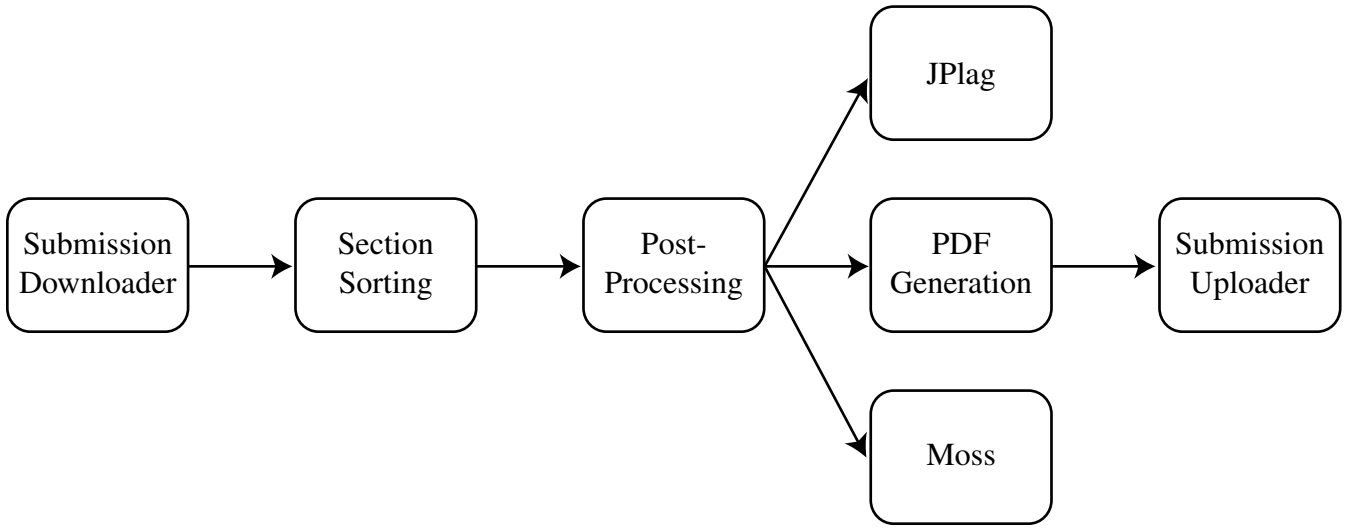
Roadmap

- Introduction
- Problems and Solution Goals
- Labrador
- Discussion



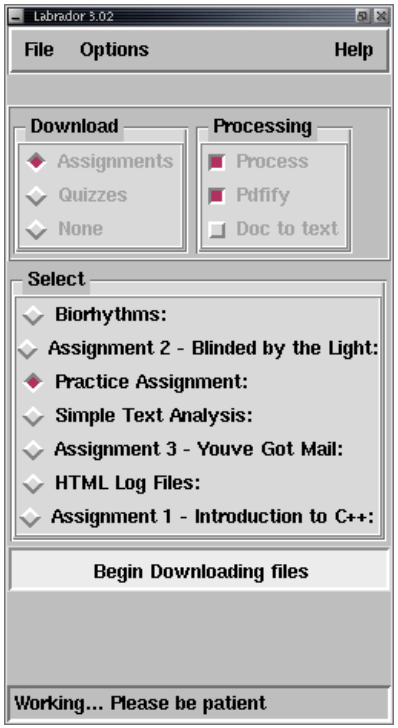
Labrador: Our Solution

- Cross-platform Client-side WebCT Supplement
- Works for users with TA and Designer access to WebCT



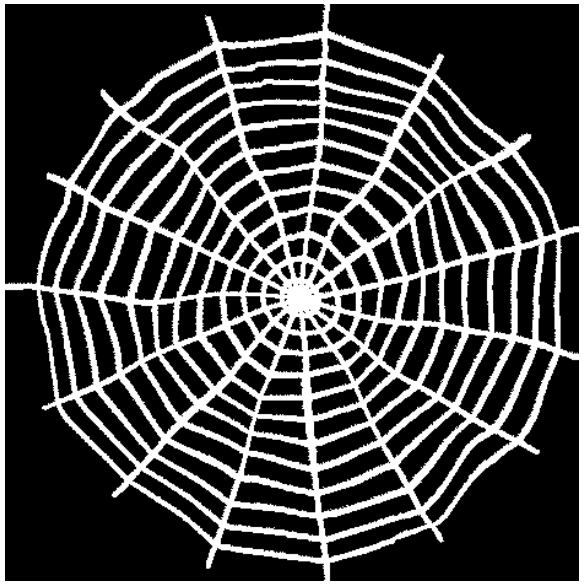
User Interface

- 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



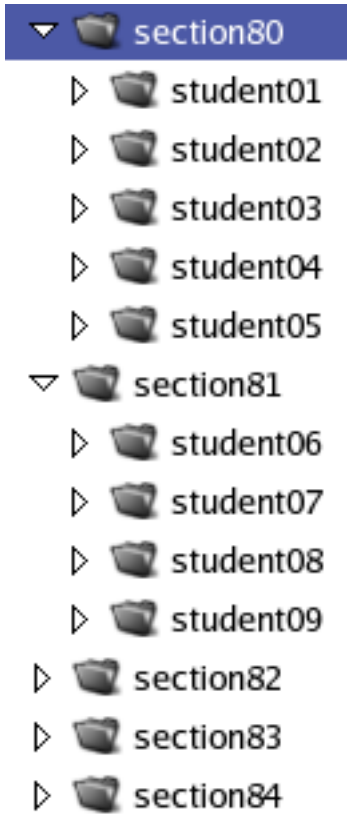
Required Information

http://webct.drexel.edu/SCRIPT/CS164_Fall2002/scripts/designer/dropbox_edit.pl?DROPBOX_ASSN_VIEW+_side_nav++1006285909

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

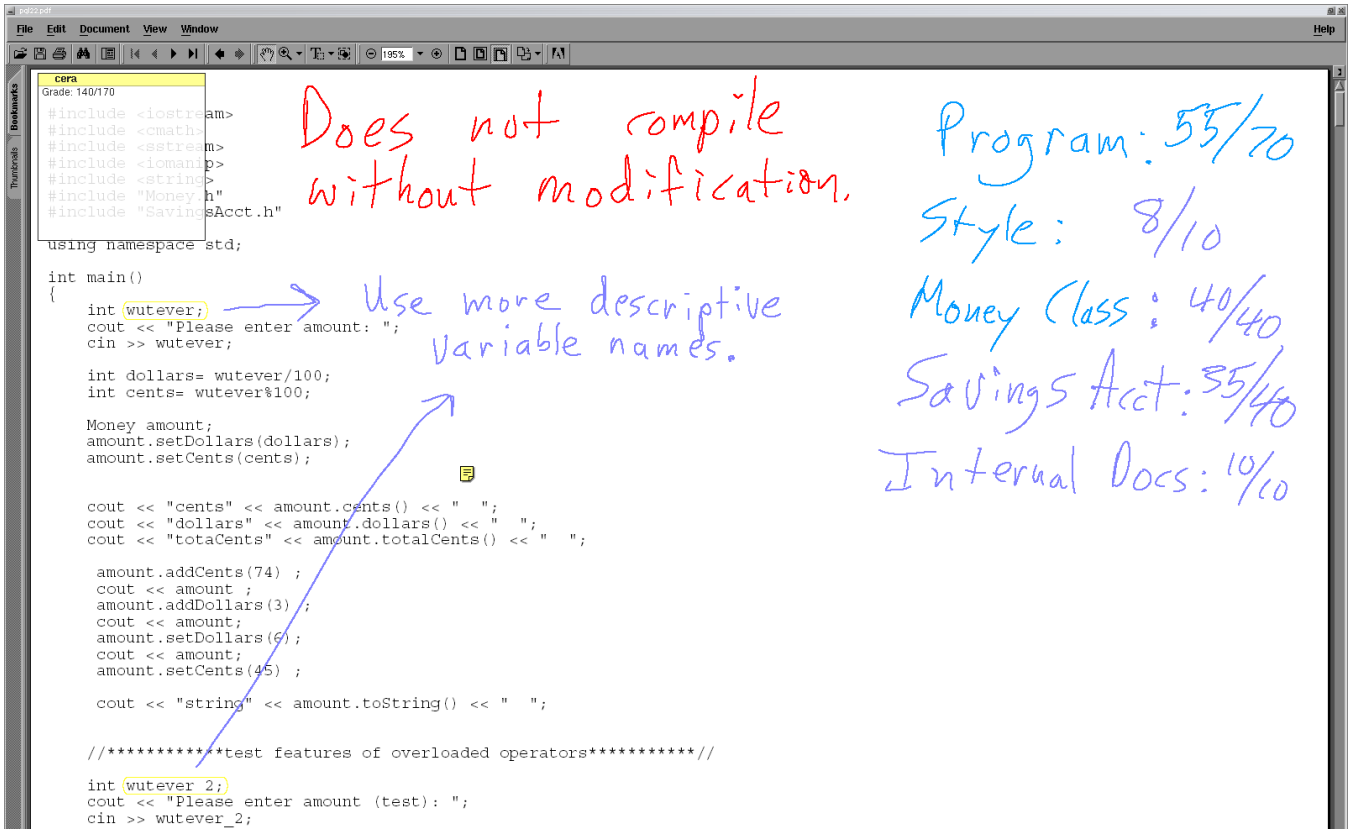








PDF Markup Example



Demonstration



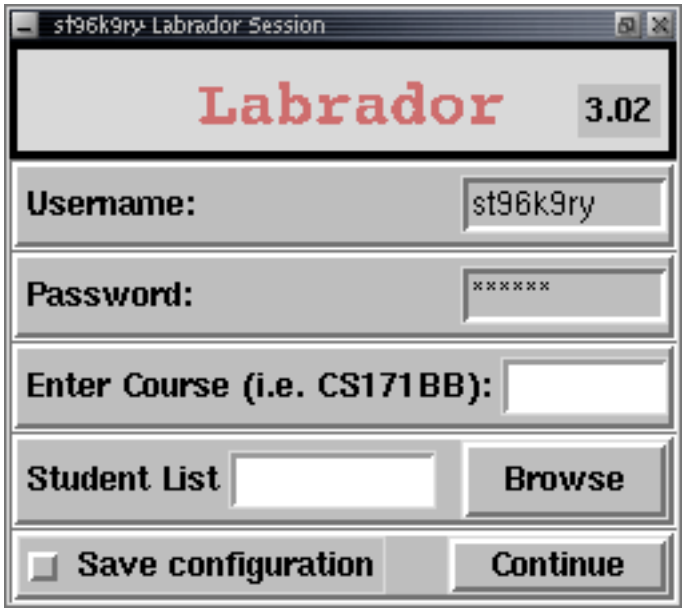
Startup screen prompts for the username and password

Demonstration [continued]



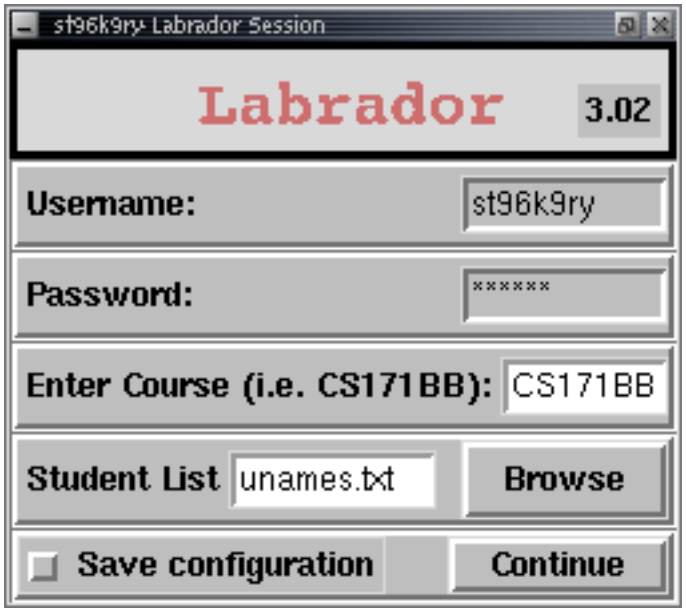
TA enters username and password

Demonstration [continued]



Labrador prompts for the course name and optional student list

Demonstration [continued]



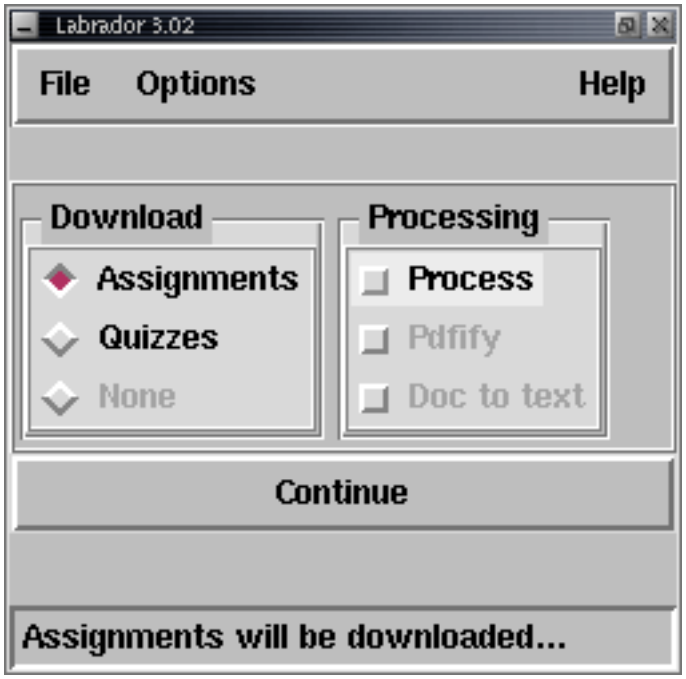
TA enters course name and student list

Demonstration [continued]



Labrador prompts for the submission type

Demonstration [continued]



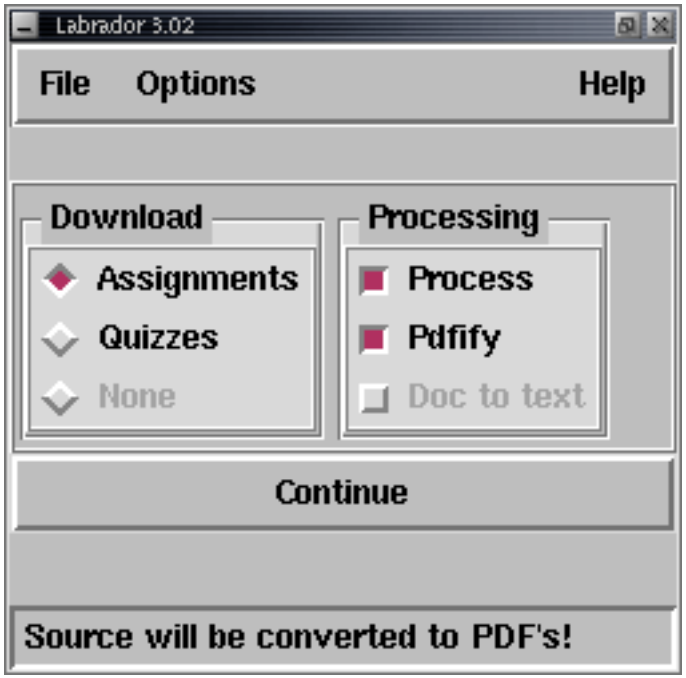
TA selects assignments

Demonstration [continued]



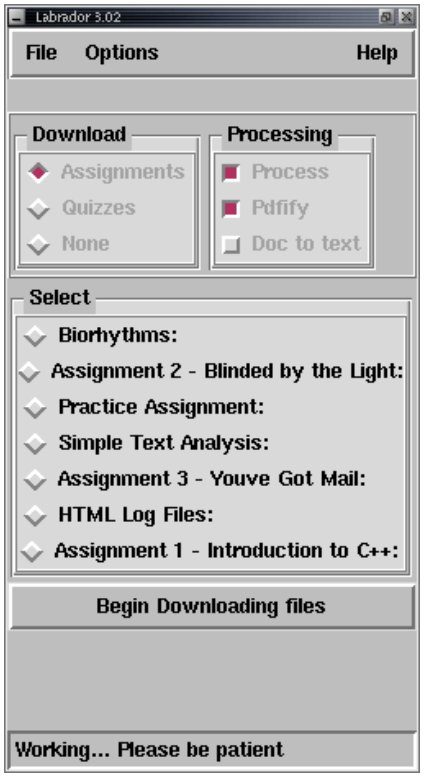
TA selects post-processing

Demonstration [continued]



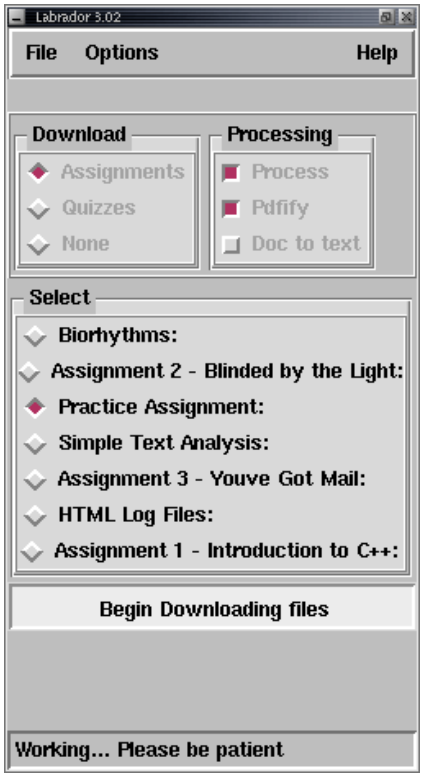
TA selects PDF generation

Demonstration [continued]



Labrador prompts for the specific assignment

Demonstration [continued]



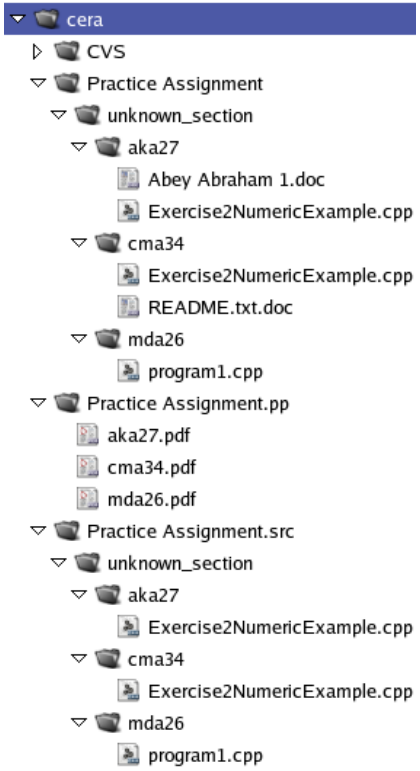
TA selects "Practice Assignment" and begins downloading

Demonstration [continued]



Labrador notifies the TA that the job is complete

Demonstration [continued]



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

	Files	Modification Date	Size
☐	student.pdf	May 19, 2003 1:25am	31.6 kB

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

Upload file | Remove files

```
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 [7] have been used extensively
- Other processes (e.g. PDF generator)
- Future work: Automatic program compiling and testing

Automated Plagiarism Detection

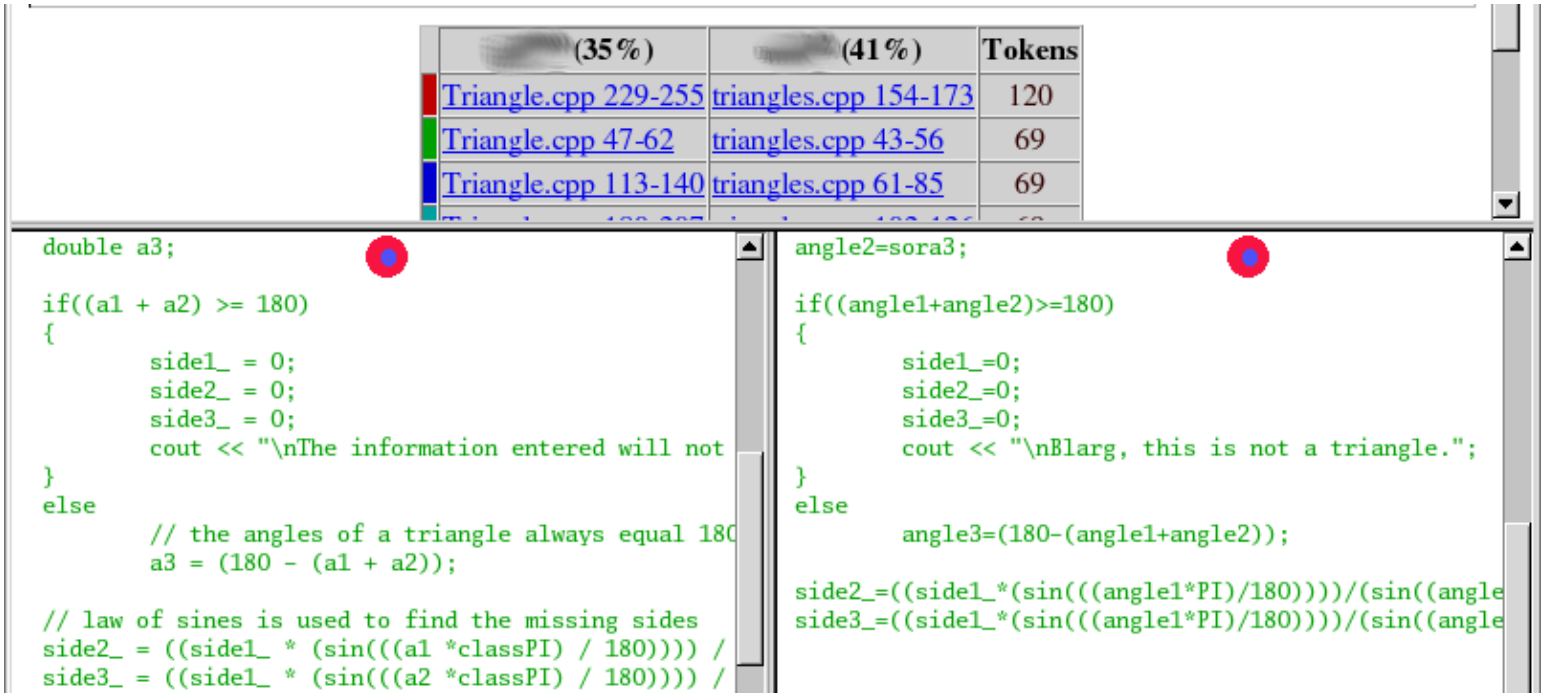
- Digital formats make "borrowing" easy
- Browsing similar works needs a simple and quick user interface.
- Careful review by faculty to assess results and present to students



Moss [1] and JPlag [7]

- Moss
  - C, C++, Java, ML, Lisp, Scheme, Pascal, and Ada
  - Common code feature reduces false positives
  - http://www.cs.berkeley.edu/~aiken/moss.html
- JPlag
  - 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/

Moss [1] Interface



[redacted] (35%)	[redacted] (41%)	Tokens
Triangle.cpp 229-255	triangles.cpp 154-173	120
Triangle.cpp 47-62	triangles.cpp 43-56	69
Triangle.cpp 113-140	triangles.cpp 61-85	69

```
double a3;

if((a1 + a2) >= 180)
{
        side1_ = 0;
        side2_ = 0;
        side3_ = 0;
        cout << "\nThe 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 *classPI) / 180)))) /
side3_ = ((side1_ * (sin(((a2 *classPI) / 180)))) /
```

```
angle2=sora3;

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

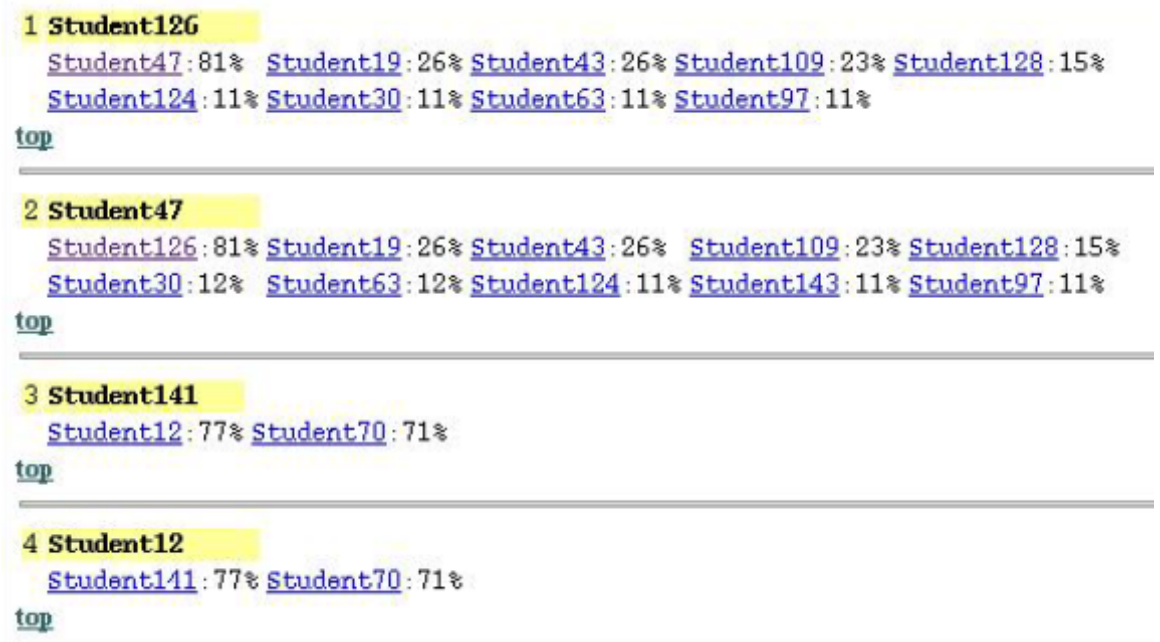
side2_=((side1_*(sin(((angle1*PI)/180))))/(sin((angle
side3_=((side1_*(sin(((angle1*PI)/180))))/(sin((angle
```

JPlag [7] Interface

student41	->	student49 (40.9%)	student86 (40.3%)	student2 (35.8%)	student73 (28.1%)	student151 (25.3%)	student88 (23.0%)
student86	->	student2 (36.0%)	student49 (34.1%)	student73 (28.5%)	student151 (24.5%)	student91 (22.5%)	
student22	->	student75 (30.1%)					
student2	->	student49 (29.5%)	student73 (25.4%)	student151 (24.3%)			
student49	->	student73 (24.9%)	student151 (21.7%)				

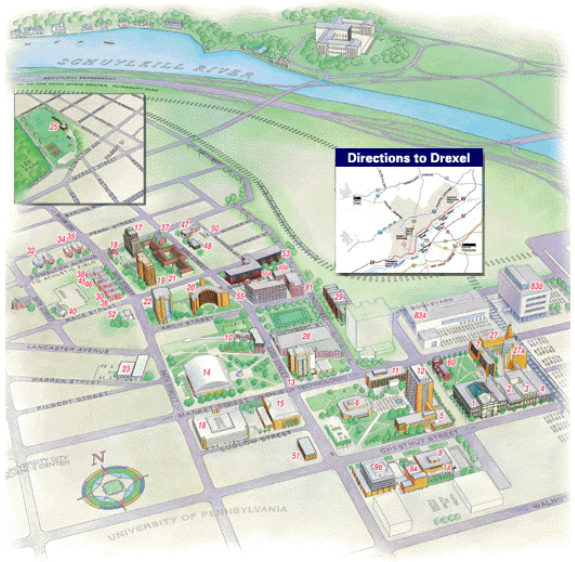
MossCliques Interface

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



Roadmap

- Introduction
- Problems and Solution Goals
- Labrador
- Discussion



The Future of CMS

- 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 CMS 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
- "Semantic Web" efforts or an API would eliminate the need to "screen-scrape" text from web pages

Labrador Availability

- Contact Us

  http://duplex.mcs.drexel.edu



The Group

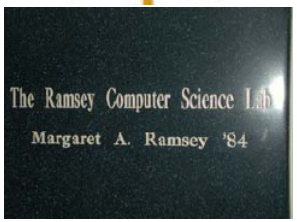
Bruce Char	*Professor, Computer Science*
Nira Herrmann	*Professor and Department Head, Mathematics*
Jeffrey L. Popyack	*Associate Professor, Computer Science*
Paul Zoski	*Instructor, Math and Computer Science*
Christopher D. Cera	*Computer Science Graduate Student*
Robert N. Lass	*Computer Science Undergraduate Student*
Aparna Nanjappa	*Computer Science Graduate Student*
Derek Rosenzweig	*Computer Science Undergraduate Student*
Jasper Zhang	*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

[1] Alex Aiken. MOSS: A System for Detecting Software Plagiarism (Unpublished), http://www.cs.berkeley.edu/~aiken/moss.html.

[2] S. Benford, E. Burke, E. Foxley, N. Gutteridge, and A. M. Zin. A Course Administration and Marking System. In *Proceedings of the International Conference of Computer Based Learning*, 1993.

[3] J. Hyvonen and L. Malmi. Trakla – A System for Teaching Algorithms Using Email and a Graphical Editor. In *Proceedings of HYPERMEDIA*, pages 141–147, 1993.

[4] Thomas Lozáno-Pérez, Eric Grimson, Leslie Kaelbling, Chris Terman, and Patrick Winston. Technologically Enhanced Education in Electrical

Engineering and Computer Science, http://www.swiss.ai.mit.edu/projects/icampus/projects/eecs.html.

[5] Open Knowledge Initiative. http://web.mit.edu/oki.

[6] Abelardo Pardo. A Multi-agent Platform for Automatic Assignment Management. *ACM SIGCSE Bulletin*, 34(3):60–64, 2002.

[7] L. Prechelt, G. Malpohl, and M. Philippsen. JPlag: Finding Plagiarisms Among a Set of Programs. *Technical Report 2000-1, Fakultat fur Informatik, Universitat Karlsruhe, Germany*, March 2000.

[8] Kenneth A. Reek. The TRY System -or- How to Avoid Testing Student Programs. In *Proceedings of the Twentieth SIGCSE Technical Symposium on Computer Science Education*, pages 112–116. ACM Press, 1989.

[9] Michael Richichi. ATTIC: A Case Study of Directory-enabled Course Management. In *Proceedings of the 29th annual ACM SIGUCCS conference on User services*, pages 258–261. ACM Press, 2001.

[10] WebCT. http://www.webct.com.

Related Work

[1] Nira Herrmann, Jeffrey L. Popyack, Bruce Char, Paul Zoski, Christopher D. Cera, Robert N. Lass, and Aparna Nanjappa. Redesigning Computer Programming Using Multi-level Online Modules for a Mixed Audience. In *Proceedings of the Thirty-Fourth SIGCSE Technical Symposium on Computer Science Education*. ACM Press, February 2003.

[2] Jeffrey L. Popyack, Nira Herrmann, Paul Zoski, Bruce Char, Christopher D. Cera, and Robert N. Lass. Academic Dishonesty in a High-Tech Environment (Special Session). In *Proceedings of the Thirty-Fourth SIGCSE Technical Symposium on Computer Science Education*. ACM Press, February 2003.

Related Work [continued]

[3] Jeffrey L. Popyack, Bruce Char, Paul Zoski, Nira Herrmann, Christopher D. Cera, Robert N. Lass, and Aparna Nanjappa. Course Management Systems (Birds-of-a-Feather Session). In *Proceedings of the Thirty-Forth SIGCSE Technical Symposium on Computer Science Education*. ACM Press, February 2003.

[4] Jeffrey L. Popyack, Bruce Char, Nira Herrmann, Paul Zoski, Christopher D. Cera, and Robert N. Lass. Pen-Based Electronic Grading of Online Student Submissions. In *Syllabus fall2002, Technology for Higher Education Conference*, November 2002.

Related Work [continued]

[5] Christopher D. Cera, Robert N. Lass, Bruce Char, Jeffrey L. Popyack, Nira Herrmann, and Paul Zoski. Labrador: A Tool for Automated Grading Support in Multi-Section Courses. In *Proceedings of the Fourth WebCT User Conference, Integrating the Campus: Technical Solutions for Resource Development or Wide Scale E-Learning Deployment*, July 2002.

[6] Jeffrey L. Popyack, Bruce Char, Paul Zoski, Nira Herrmann, and Christopher D. Cera. Managing Course Management Systems (Birds-of-a-Feather Session). In *Proceedings of the Thirty-Third SIGCSE Technical Symposium on Computer Science Education*, page 423. ACM Press, February 2002.