The BOSS Online Submission System

Mike Joy and Ian Burnett

Department of Computer Science
University of Warwick

LTSN-ICS Workshop on CAA, 6 April 2001

Copyright © 2001 The University of Warwick
Summary

• What is BOSS?
  – history
  – current architecture
• Using BOSS
  – examples of dialogue
• Issues arising
• Dissemination and future work
Motivation

• Large class sizes
  – up to 300 students
• Programming modules
  – multiple assignments
  – formative
  – require rapid feedback
  – need to run program to mark it
What is BOSS?

- Course management utility
- Online submission
- Online marking
- For programming assignments
  - automatic testing
  - plagiarism detection
Pedagogical Issues

• Must not distract students
  – *not* a CAL tool!

• Must be accurate
  – ensure awarded marks can be reproduced

• Must not infringe academic freedom
  – i.e. must not prescribe marking schemes
Original Architecture (1993-)

- Collection of programs (UNIX - SunOS)
  - compiled C (some setuid)
  - Perl scripts
  - Tcl/Tk GUI
  - mSQL database
- Difficult to maintain or enhance
- Not easily portable
The World has Changed!

- New requirements
  - security
  - networking
- New environment
  - familiarity
  - programming tools
- 1999: decided to re-design and re-code
Problems (1998)

- Reliance on University network
- Bugs in NFS!
- Inextensibility of Tcl/Tk programs
- Data entry for test data
- Testing programs for correctness
  - hard!
  - OK with Pascal
  - Java a bit different
Limitations in I/O

- Lack of support for GUI in student programs
  - e.g. Java programs
- I/O handling under UNIX is messy
  - control characters
  - signal handling
- Actual vs. expected output
  - use of `diff`
  - misspellings, whitespace, ...
Limitations in OS

- Different platforms
  - PC, UNIX, Mac
- Networking not built in
- Multiple installation problems
- Security
  - security is that of host machine
New System Requirements

• Platform-independent
  – Java

• Accessible across the Internet
  – client-server model
  – Java *application* (not Applet)

• Reliable data

• Security

• Standards conformant
Features of New Version

- Java application
- Client-server model
- Runs on Solaris/Linux/Windows
- SSL armoured
- Interface to institution member database
Security

- SSL connection
- RMI
- Separate servers for students and staff
Architecture of Boss2

- Interface
- Server
- Data Store
How is BOSS used?

- Two categories of user
- Student
- Staff
  - administrator
  - manager
  - moderator
  - marker
Students Submitting Work

1) Choose course
2) Choose assignment
3) Choose exercise
4) Choose program files to submit
5) Test program against one data set
6) If OK, submit
7) Await email receipt
Marking

1) Enter automatic test data + other criteria
2) Run (multiple) automatic tests
3) Choose weightings for tests
4) One or more persons mark
5) Lecturer moderates
6) Lecturer enters marks into database
7) Lecturer emails feedback to students
8) Lecturer prints marksheet
Module Management

- Manager can fully edit module details
  - markers/moderators
  - assignments
  - individual problems
  - deadlines
  - extensions
- Plagiarism detection
Automatic Tests

- Marks resulting from automatic tests are incorporated into the marksheet directly.
- If output correct, full marks awarded.
- If output wrong, 0 marks awarded.
- Marker may adjust automatically assigned marks.
Feedback

- Extensive commenting facility is included
- Marker can write note either for moderator or for student
- Moderator may edit feedback to student
- System emails direct to student
- Notes to moderator are confidential
- Comments and marks are retained
Anonymity

• Assignments are marked using ID numbers
• Anonymous marks are stored in a database
• List of final marks by name is produced once marking/moderation is complete
Consistency and Reliability

• Allows double (or multiple) marking on same marksheet
• Allows moderator to view both sets of marks (and original submission)
• System suggests final marks
• All automatic test results are viewed by the markers
Electronic Marksheets

- Graphical marksheet using Swing
- Lecturer specifies categories (and weights) for which marks are awarded
- Integrates marks resulting from automatic tests with those relating to style etc.
The Testing Paradigm

• Specify expected output for given input
• Compare output of student’s program
• In Boss1
  – input and output are text files
  – compare using `diff`
• In Boss2
  – input and output are Java objects
  – use `equals()` to compare
  – use `toString()` to display
Sherlock

- Plagiarism detection
  - for programs
- Compares submitted programs
  - pairwise
  - tokenised, with/without comments
  - neural net analysis
- Indication of possible plagiarism
  - human intervention required!
Some Dialogues

- Marker
- Module Manager
- Just to give you the feel for it!
### Marksheet for CS154 (2000/2001)

Module code: CS154  
Module title: Demonstration  
Academic year: 2000/2001  
Module leader: Alfred N Other

**Department:** Computer Science  
Year of study: 1  
Course: G500 Computer Science

<table>
<thead>
<tr>
<th>Surname</th>
<th>Forename</th>
<th>ID</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>Alice</td>
<td>0015624</td>
<td>67</td>
</tr>
<tr>
<td>Ball</td>
<td>Benjamin</td>
<td>0004563</td>
<td>81</td>
</tr>
<tr>
<td>Copperfield</td>
<td>Charlie</td>
<td>9956312</td>
<td>23</td>
</tr>
<tr>
<td>Duck</td>
<td>Debbie</td>
<td>0020114</td>
<td>54</td>
</tr>
</tbody>
</table>

**Department:** Mathematics  
Year of Study: 2  
Course: G100 Mathematics  
.............
Successes

• No (known!) security breaches
• Handles classes of 300+ students
• Minimal keystrokes for marking
• Easy double-marking
• Speed
• Consistency
• Anonymity
• No paperwork
Interesting Issues

• There are lots of bugs in the Sun JVM
  – processes aren’t always killed
    • sometimes servers hang under heavy load
  – Swing doesn’t work properly over ssh
    • institution network security

• RMI is slow

• Devising tests is hard
  – students *always* uncover loopholes
Future Enhancements

- Public key infrastructure
- Student access to data
  - Data Protection Act ...
- Common specification of database
  - merge with CAL software
  - single course management utility
- Incorporate developments in Java technologies
Availability

- http://www.dcs.warwick.ac.uk/cobalt/
- Documentation online
- Beta code now available
And Finally ...

Questions?
Marker Dialogue (1)

Please choose your role

- Marker
- Moderator
- Manager
- System administrator
Marker Dialogue (2)

<table>
<thead>
<tr>
<th>Status</th>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unmarked</td>
<td>0000072</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000197</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000206</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000210</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000275</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000276</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000342</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000403</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000460</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000517</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000534</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000582</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000629</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000657</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000666</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000749</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0000949</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0001036</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0001161</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0001422</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0001518</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0001521</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0001527</td>
</tr>
<tr>
<td>Unmarked</td>
<td>0002153</td>
</tr>
</tbody>
</table>

Mark:
- Get submission
- Internal note
- Student note

Done
Example on manual page

Running with command: /dcs/acad/msj/usr/doc/course/cs120/2000/Asst2-tests/Test1/run in environment:

  ● WORKING_DIR=/dcs/asg/slave/tmp/tmp798D3987/

Exit code was 1

Expected output (stdout)

`guest4:30807:103:Guest 4:/dcs/guest/guest4/f43d68ca811d4246c41077d6843e4ec0  
msj:1045:102:Mike Joy:/dcs/acad/msj/dd02c7c2232759874e1c205587017bed  
smiles:1542:103:Simon Miles:/dcs/res/smiles:c67a671df23a5f8eaf2208e27f28be39  
websearch:31158:103:websearch:/dcs/guest/websearch:ba8ea2513154e033f4ccfb0eaddf794`

Actual output (stdout)

`guest4:30807:103:Guest 4:/dcs/guest/guest4:36aa3d0f272f0e2f7c631bea8199af2c  
msj:1045:102:Mike Joy:/dcs/acad/msj/dd02c7c2232759874e1c205587017bed  
smiles:1542:103:Simon Miles:/dcs/res/smiles:c67a671df23a5f8eaf2208e27f28be39  
websearch:31158:103:websearch:/dcs/guest/websearch:ba8ea2513154e033f4ccfb0eaddf794`

Streams differ at index 43.

Expected output (stderr)
<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commenting</td>
<td>0</td>
</tr>
<tr>
<td>Indentation</td>
<td>0</td>
</tr>
<tr>
<td>Other style</td>
<td>0</td>
</tr>
<tr>
<td>AUTO: Example on manual page</td>
<td>0</td>
</tr>
<tr>
<td>AUTO: Add user</td>
<td>10</td>
</tr>
<tr>
<td>AUTO: User not in datafile</td>
<td>0</td>
</tr>
<tr>
<td>AUTO: Delete user</td>
<td>10</td>
</tr>
<tr>
<td>AUTO: Verify password OK</td>
<td>0</td>
</tr>
<tr>
<td>AUTO: Help message</td>
<td>10</td>
</tr>
<tr>
<td>AUTO: Add user, non-default datafile</td>
<td>0</td>
</tr>
<tr>
<td>AUTO: Verify password OK different algorithm</td>
<td>10</td>
</tr>
<tr>
<td>AUTO: Bad password</td>
<td>10</td>
</tr>
</tbody>
</table>
Manager Dialogue (2)

[Image of a software interface with a list of extensions and buttons for Add, Edit, Remove, Help, and Done]
Manager Dialogue (4)
Manager Dialogue (5)
Manager Dialogue (6)

Description: Bad password

- Automatic test
- For use by automatic tester

Run native platform command: /dcs/acad/m
Manager Dialogue (7)

Run the following command on the system:
/dcs/acad/msj/usr/doc/course/cs120/Tests2/T

In the following environment:
WORKING_DIR=##

Standard input file
- Empty
- Text

Standard output file
- Empty
- Text

Standard error file
- Empty
- Text
Manager Dialogue (8)