A Mooshak extension to return detailed testing feedback to all users

Helder Daniel

Departamento de Engenharia Electrónica e Informática
Faculdade de Ciências e Tecnologia
Universidade do Algarve

October 10, 2020

Abstract

Mooshak (Leal and Silva [2003]) was designed as an open source programming contest manager, but its use cases has long moved beyond just programming contests, also into the programming classroom. In fact with suitable extensions (Daniel [2019], Daniel [2020]) it can be used also for real world testing of complex applications.
Due to the original design as a contest manager, only privileged users can access report of the tests performed. To make it useful in the classroom as a pedagogic tool to train programming, or in a production environment it is desirable that the user which submits the source code can access the feedback on the tests performed. The Mooshak extension proposed here, allows any user to access the report of its own submissions, from Mooshak's GUI, without any changes on the original GUI.

1 Introduction

Although Mooshak (Leal and Silva [2003]) was designed as an open source programming contest manager, according to International Collegiate Programming Contest rules (ICPC [2019], its use cases spread rapidly into the programming classroom.
The ability to automatically perform black box testing on submitted source code makes it valuable as an aid to training and assessment. Since the testing report is available only to privileged users, the teacher can use it as a valuable mean to assessment. However this invalidates that during training students can be aided with detailed feedback on the testing. This extension gives Mooshak the

ability, if chosen, to give also students access to the testing report on their own submissions.

Furthermore, if this extension is combined with other extensions (Daniel [2019], Daniel [2020]), that allows Mooshak to perform not only black box testing but also unit testing, on complex java applications and distributed applications such as client-server ones, makes it suitable for real world application testing not only in the classroom but also in production environments.

The extension that is presented in the next sections was developed to enhance Mooshak without changes in Mooshak's GUI. Users can access the testing report through the GUI page where the submissions' status are shown, just clicking on the submission number.

2 Mooshak testing operation and reporting

Mooshak's 1.x.x software testing process follows 2 steps:

1. If needed, the submitted **solo source code** file is compiled or assembled first.
2. The **solo file** application is executed and tested.

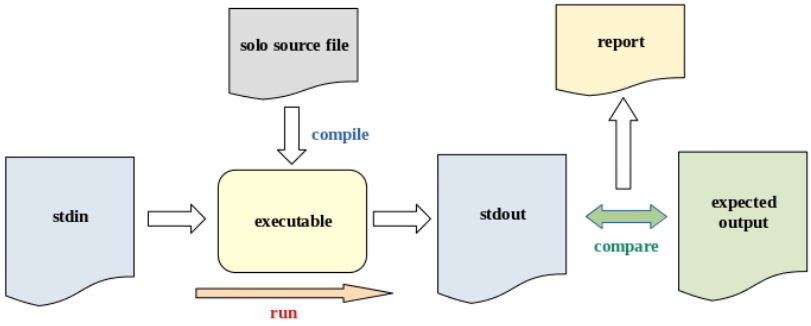

Figure 1: Mooshak default testing operation on solo files

Mooshak performs black box software testing, by executing the compiled application for each test case, comparing the application stdout with the expected and generating a report (Daniel [2019]).

This report, for each test case, presents the expected output of the test, along with the obtained and the differences, as shown in figure 2. Clicking in the

test number, in this figure the T00, T01, T02, at the upper left corner of each test case, it is possible to see the test case input file that was presented at the application stdin.

Test	Expected	Obtained	Differences
T00 T=0.124 sec E=0 sec M=772 Kb Presentation Error	28\n XXXXXX\n X++++X\n XXXXXX\n XXX+XX\n XXX+..\n	28\n XXXXXX\n X++++X\n XXXXXX\n XXX+XX\n XXX+..\n \n	28\n XXXXXX\n X++++X\n XXXXXX\n XXX+XX\n XXX+..\n \n
T01 T=0.120 sec E=0 sec M=772 Kb Presentation Error	16\n XXXX\n X+XX\n XXXX\n ++++\n	16\n XXXX\n X+XX\n XXXX\n ++++\n \n	16\n XXXX\n X+XX\n XXXX\n ++++\n \n
T02 T=0.116 sec E=0 sec M=772 Kb Presentation Error	30\n ++XXXX++++\n XXX+XXXXXX\n +XXXX+++++\n	30\n ++XXXX++++\n XXX+XXXXXX\n +XXXX+++++\n \n	30\n ++XXXX++++\n XXX+XXXXXX\n +XXXX+++++\n \n

Figure 2: Mooshak test report

This is the report that is presented to the privilege user at the submissions status page, by cliking the submission number in blue at the left of figure 3. However unprivileged users does not get nothing when clicking the submission number, even for their own submissions,

#	Contest Time	Country	Team	Problem	Language	Result	State
S996	3604:51:59		students a64583	G	Java zipped	0 Wrong Answer	pending
S995	3574:38:08		students a64587	G	Java zipped	0 Wrong Answer	pending
S994	3574:34:38		students a64587	G	Java zipped	0 Wrong Answer	pending
S993	3574:31:24		students a64587	G	Java zipped	0 Wrong Answer	pending
S992	2764:57:25		students a64644	E	Java zipped	0 Accepted	final
S991	2488:56:15		students a64644	G	Java zipped	0 Wrong Answer	pending
S990	2321:29:19		students a68011	F	Java zipped	0 Accepted	final
S989	2288:02:17		students a61269	G	Java zipped	0 Wrong Answer	pending
S988	2287:36:05		students a65638	G	Java zipped	0 Runtime Error	pending
S987	2287:32:15		students a61269	G	Java zipped	0 Wrong Answer	pending
S986	2287:30:25		students a64545	G	Java zipped	0 Accepted	pending
S985	2287:27:30		students a68026	G	Java zipped	0 Runtime Error	pending
S984	2287:27:19		students a64545	G	Java zipped	0 Accepted	pending
S983	2287:25:20		students a68026	G	Java zipped	0 Compile Time Error	pending
S982	2287:25:06		students a67984	G	Java zipped	0 Runtime Error	pending

Figure 3: Mooshak's submission status page

3 Extending Mooshak to return testing feedback to all users

Enabling the access to the reports to any user by clicking the submission number will solve this issue, but Mooshak source code should be further tailored

and changed to allow this.
Since this extension aims to have none or at least the minimum impact in Mooshak's 1.6.4 source code, the support to make the report accessible to any user was implemented adding surgically just one new line of code in file:

```
<mooshak>/packages/classes/Submission.tcl
```

This line of code is added to the end of function:

```
Operation Submission::analyze
```

the one that analyzes the submission, performs the testing, and records the result to produce a report. This way after the report be produced it can be handled by an external script, by calling it:

```
exec reportext.sh ${_Self_}  ${_Self_}/$Program  ${_Self_}/$Report  $Contest/pro
```

To the script are passed the submission folder information, the report file and the problem folder. The report is stored in the submission folder with name **1.html**.

To return the report to the user it is used what Mooshak already does. When a privilege user click the submission number in the submission status page, figure 3, along with the report is returned also the submitted code.
For an unprivileged user it is returned only the submission code. However the returning mechanism is working,it just strips the report. What the external script does is just to put back, along with the source code the **1.html** report file. Also the input files, used for each test case, which are stored in the problem folder, is included along with the source file. Then all this files are compressed and made available to user download, by clicking the submission number.

This way all the information about the testing performed is accessible to all users, however in a different way for privilege and unprivileged users. For the latter the information came in a compressed file and not in Mooshak's GUI itself, as for the former, but the information is the same.
As shown in figure 4 unpacking the file, the user have access to the report file: **1.html** and the test cases input files: **T0input**, **T1input**, ...

There is however a tweak to hide some test cases input files if required if desired to do not all the test cases to the user.

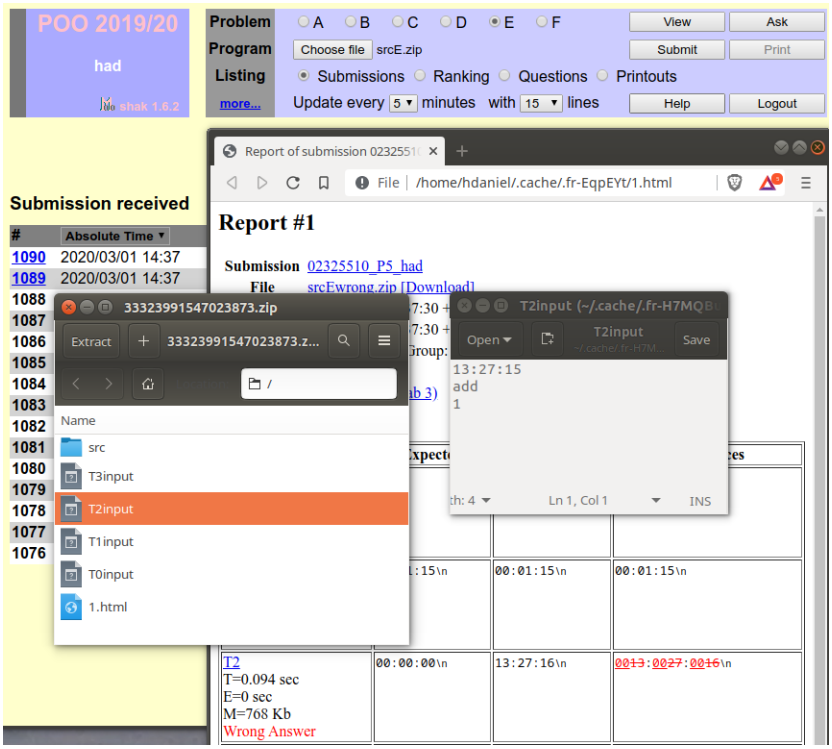

Figure 4: Accessing Mooshak's reports with the extension

4 Conclusion

This extension add features to the basic Mooshak programming contest management system that enhances its capabilities towards giving detailed feedback on the software testing. These, along with the extensions proposed in (Daniel [2019], Daniel [2020]), enhances Mooshak usability in real world production environments or in programming training programmes. The extension was developed with the aim to avoid any changes in the Mooshak's GUI. Enabling the extension, users can access the testing report through the GUI page where the submissions' status are shown, by just clicking on the desired submission number.

References

Helder Daniel. Automatic testing of java applications, 2019. URL https://github.com/helderdaniel/MooshakExtensions/blob/

main/javaext.pdf. [Online; accessed 7-June-2019].

Helder Daniel. Automatic testing of client-server applications, 2020. URL https://github.com/helderdaniel/MooshakExtensions/blob/main/csext.pdf. [Online; accessed 3-September-2020].

ICPC. International collegiate programming contest, 2019. URL https://icpc.global/worldfinals/rules. [Online; accessed 2-May-2019].

José Paulo Leal and Fernando Silva. Mooshak: a web-based multi-site programming contest system. *Software Practice & Experience*, 33(6):567–581, 2003. URL http://www.ncc.up.pt/mooshak/.