Proceedings of the ASME 2012 11th Biennial Conference On Engineering Systems Design And Analysis
ESDA 2012
July 2-4, 2012, Nantes, France

ESDA2012-82141

MARKUS, AN OPEN-SOURCE WEB APPLICATION TO ANNOTATE STUDENT PAPERS ON-LINE

Morgan Magnin
Guillaume Moreau
LUNAM Université
Ecole Centrale de Nantes
1 rue de la Noë, BP 92101
44321 Nantes Cedex 3
France
Email: morgan.magnin@ec-nantes.fr
guillaume.moreau@ec-nantes.fr

Nelle Varoquaux
École Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan cedex
Email: nelle.varoquaux@gmail.com

Benjamin Vialle
Mobile Devices Ingenierie
100, avenue de Stalingrad
94800 Villejuif
France
Email: benjaminvialle@gmail.com

Karen Reid
University of Toronto
40 St. George St.
Toronto, Ontario M5S 2E4
Canada
Email: reid@cs.toronto.edu

Mike Conley
Mozilla Corp.
Canada
Email: mike.d.conley@gmail.com

Severin Gehwolf
University of Toronto
40 St. George St.
Toronto, Ontario M5S 2E4
Canada
Email: severin.gehwolf@utoronto.ca

ABSTRACT

A critical component of the learning process lies in the feedback that students receive on their work that validates their progress, identifies flaws in their thinking, and identifies skills that still need to be learned. Many higher-education institutions have developed an active pedagogy that gives students opportunities for different forms of assessment and feedback. This means that students have numerous lab exercises, assignments, and projects. Both instructors and students thus require effective tools to efficiently manage the submission, assessment, and individualized feedback of students' work. The open-source web application MarkUs aims at meeting these needs: it facilitates the submission and assessment of students' work. Students directly submit their work using MarkUs, rather than printing it, or sending it by email. The instructors or teaching assistants use MarkUs's interface to view the students' work, annotate it, and fill in a marking rubric. Students use the same interface to read the annotations and learn from the assessment. Managing the students' submissions and the instructors assessments within a single online system, has led to several positive pedagogical outcomes: the number of late submissions has decreased, the assessment time has been drastically reduced, students can access their results and read the instructor's feedback immediately after the grading process is completed. Using MarkUs has also significantly reduced the time that instructors spend collecting assignments, creating the marking schemes, passing them on to graders, handling special cases, and returning work to the students.

In this paper, we introduce MarkUs' features, and illustrate their benefits for higher education through our own teaching experiences and that of our colleagues. We also describe an important benefit of the fact that the tool itself is open-source. MarkUs

has been developed entirely by students giving them a valuable learning opportunity as they work on a large software system that real users depend on. Virtuous circles indeed arise, with former users of MarkUs becoming developers and then supervisors of further development.

We will conclude by drawing perspectives about forthcoming features and use, both technically and pedagogically.

INTRODUCTION

Assessing and providing feedback on student work is a critical and time consuming component of any course. It is especially true in computer science training. Receiving timely, high quality feedback is a important pedagogic goal of any course. MarkUs is designed to simplify the process of submitting work, provide an intuitive interface to annotate students work and record grades, and reduce the administrative task of the instructors.

With all types of work submitted by students there are a number of steps involved.

- Submissions. Students use some mechanism to submit their work either on paper or electronically.
- Assessment. Some form of grading scheme is used to evaluate the students' work.
- Feedback. The graders provide feedback in the form of general comments or annotations on the students' work.
- Distribution. The work is returned to the student with the grade and the feedback.
- Record keeping. The instructor collects the grades to determine a final evaluation.

Of these steps the most challenging is to provide high quality feedback efficiently. While there are many advantages to submitting work electronically, it is challenging to provide the kind of simple efficient feedback achieved by circling incorrect work or drawing a few arrows directly on a paper submission. One of our previous attempts simplified the management and evaluation of work submitted electronically by students through the use of Tablet PCs [1]. Although this approach facilitated annotating their work, the management of students' submission still remained a hot topic: in the past, we received files by e-mail, temporarily stored on the instructor's computer then returned them corrected to the sender(s). This method is error prone, time consuming and only works for small to medium class sizes. Another solution to collect students' work was the use of the submissions tools provided by elearning platforms such as Claroline [2] or Moodle [3]. Yet practical to handle numerous submissions and marking, these tools could not be used to handle annotation of the students' work thus motivating us to turn to a new relevant and effective tool for the assessment of students' work: MarkUs [4].

MarkUs was created at the University of Toronto, Canada. Since summer 2009, students at Ecole Centrale de Nantes have joined the MarkUs development team. First used at the University of Toronto in September 2009, and at the University of Waterloo (Canada) in May 2010, MarkUs was deployed at Ecole Centrale de Nantes in September 2010. In this paper, we present MarkUs' features, and illustrate their potential assets for higher education through our own feedback. We will not only provide an overview of current uses and future but also emphasize the benefits of open-source for this kind of tool.

MarkUs takes into account the needs expressed by teachers and students. Encompassed in a single web-application, the primary features of MarkUs follow:

- Students submit documents using MarkUs. This has the advantage that graders and instructors do not need to manage submissions via email, or copy submissions around. Documents are versioned (i.e. successive versions of each file are stored), so that teachers can track the progress of students over time [5]. Students can easily see exactly what they submitted, allowing them to check their work.
- Graders view student documents and add annotations within MarkUs. They also fill in a grading scheme created by the instructor.
- A standard set of reusable annotations may be created ahead of time, and graders may save annotations for later reuse.
- The software manages the deadlines that students must respect to submit their work. It is possible to specify additional features, like "grace period credits" or "penalty formulas";
- When the grading of an assignment or piece of work has been completed, the instructor can "release" the grades to the students.
- Students view their grades and feedback through a read-only view of the same interface graders used to generate the feedback.
- Instructors may download grades for further processing.

Additional features are described in the next section.

In this paper, we will first describe the core MarkUs features as well as more recent enhancements. Then we will detail how MarkUs has been deployed in various Canadian and French institutions, allowing us to analyze the impact of this new software in the whole pedagogical process. Finally, we will preview the forthcoming features we plan to integrate in the tool, allowing us to overcome current limitations of the software.

THE MARKUS TOOL

History

Initiated during the summer 2005 as Online Marking Tool (OLM) [6], the project aimed at improving the effectiveness and efficiency of marking lower year Computer Science assignments. Built using the Python web framework Turbogears, OLM provided an environment to facilitate grading programming assignments online including the ability to annotate student work, and

fill in a marking rubric. During its three-year deployment at the University of Toronto, we learned the strengths and weaknesses of OLM. The "grader view", which allowed Teacher Assistants to annotate and assess the student submissions was very successful, but the administrative overhead was high and we could see many opportunities to improve the user interface for graders. Our plans to extend and enhance OLM were hampered by a decaying code base, lack of adequate tests, and a sense that modern web programming frameworks would be more advantageous to the project.

In the summer of 2008, it was decided to rewrite the whole application in Ruby on Rails. We kept the best features of OLM as MarkUs took shape, but MarkUs included many new features. One year later, MarkUs was deployed for the first time as a replacement to OLM. Thanks to more careful software development practices, including extensive testing and code reviews, the first deployment of MarkUs went very smoothly.

Free software philosophy

MarkUs is free software, i.e. a software whose use, study, editing and duplication for distribution are permitted, technically and legally, to guarantee a wide range of freedoms to the end-user. Licensed under the MIT Open Source License, it is developed mainly by Canadian and French students.

Presented at several conferences, MarkUs receives various outside contributions. Students and teachers from other universities send patches that are integrated with the main branch of development. The presentations in conference can also expand the range of returns on software, and identify additional needs.

MarkUs benefits from the integration of many components that are widely used in the free software community: a version control module to collect the student code, the use of a well-known javascript library to annotate images and PDF documents, the integration of annotations in LaTeX and MathML to adapt the tool to the research environment.

Community

MarkUs is the subject of ongoing developments. Each semester, a dozen new students participate in the implementation of its features. They are co-supervised by teachers and technical mentors (that often are former students who continue to participate in the evolution of the software).

Since 2008, more than 45 undergraduate students have participated in the development of MarkUs; some as full-time summer interns, but most working part time on MarkUs as a project course. The fact that we have have uncovered so few major bugs, and that MarkUs has been so well-received by instructors is a testament to the high quality work of these students.

In September 2010, MarkUs' code moved to the public hosting web-service, GitHub. This has allowed any contributor to patch and improve the project's code base. As a result, MarkUs has been opened to a wider community.

As MarkUs has grown to include developers from different countries, our development process has evolved. We rely on an issue tracker to keep track of outstanding feature requests and bugs or flaws in the system. Good communication is key both to helping new developers get up to speed and to ensure that the quality of the code is maintained. We depend on a mailing list and IRC chat room for discussions of all sorts. The best addition to the development process has been the adoption of Review Board [7], a code-review tool. Every change to the system first undergoes peer review before it is pulled into the production system.

As the development of MarkUs profits from the dynamics of project-based learning, we have been able to establish virtuous circles: students, who were former MarkUs users, are invited to become contributors, and then act as technical mentors for new projects once they have been graduated. This positive cycle has been extensively described in [8].

Features

As a web-application, MarkUs is cross-platform (Windows, Mac OS X, Linux, etc.). Since MarkUs can be used through any web browser, and is well-suited to the variety of situations that students and faculty members may encounter in their daily life. MarkUs features are summarized in Figure 1.

Each course has its own instance of MarkUs. This simplifies administration and means that if there is a problem with one instance of MarkUs, only that course is affected. At the beginning of a course, the instructor configures MarkUs by adding students, graders, and possibly co-instructors.

An instructor creates an assignment and configures it by setting the submission deadline as well as the default penalties for late submission. The instructor determines if students will work alone or in groups. The instructor can upload a file that assigns students to specific groups, or may allow students to form their own groups with a specified minimum and maximum group size.

Once the assignment has been created, students may begin submitting files. Students may submit multiple times before the deadline, and can look at the files they submit to confirm that they have submitted correctly. If students form their own groups, they may only begin submitting files once the group has the minimum number of members, and invited members may only view their group's submissions after they have explicitly accepted the invitation.

Before the graders begin marking, the instructor configures a marking scheme, creates annotation categories, and optionally creates a set of pre-defined annotations that may be used by the graders. The instructor also determines which graders mark each submission and/or which parts of the assignment.

The instructor monitors the progress of the graders and when

the graders have completed the marking, releases the results to the students. The instructor may also download the grades for each assignment.

A major goal of MarkUs is to streamline the workflow of the graders. Each grader sees a list of the students or groups that that grader is to mark. Figure 2 illustrates how the grader carries out his or her work. The grader sees the file or files submitted by a student in the left half of the screen and can create new annotations or select a predefined annotation from one of the existing categories. On the right side the the screen is a marking scheme that the grader fills in. The grader can also switch to view a list of all annotations and add some general comments, or see a summary of the group's results. When a grader has finished marking one student or group's work, the grader sets the status to complete.

When the instructor releases the results, students see a read-only version of the grader view, and read the annotations and marking scheme. Students may also download a PDF version of their results for their archives.

More recent features include a simple grade spreadsheet that can be used to enter marks for tests or labs where a full assignment interface is not required, charts and summary information about each assignment, and the ability to annotate PDF files.

The general philosophy of MarkUs is a permissive one, allowing users to perform tasks that might not be allowed, but to provide warning messages. For example, if a student tries to submit a file after the deadline, the student will see a prominent message explaining that the submission will not likely be graded, but the student is allowed to proceed. This approach makes it more straightforward to handle special cases without extra effort, while still enforcing deadlines.

THE DEPLOYMENT OF MARKUS

MarkUs has been used in Computer Science related courses since fall 2009 in Canada and since 2010 in France. In this section, we will detail how students and teachers learn to use the tool.

An overview of the use of MarkUs in Higher Education

MarkUs is currently used at the University of Toronto, University of Waterloo and Ecole Centrale de Nantes:

- At the University of Toronto, MarkUs was first deployed in the fall of 2009. In the fall of 2011 it was used in 8 different courses (15 sections) in Computer Science and Engineering for a total of more than 1200 students. The largest course had 650 students in 5 sections.
- At the University of Waterloo, MarkUs has been used since the fall of 2010 in two large first year courses with more than 800 students each term.
- In Ecole Centrale de Nantes, MarkUs has been used since 2010 in the two computer sciences courses of the common engineering core (around 350 students each) but also in the Computer Science major and in other elective courses for a gross total more than 1200 students over one year and a half so far.

MarkUs impacts both undergraduate and master students, in Computer Science related courses like: algorithmics, C, Java and Python programming, databases, modeling languages (UML).

Training for teachers and graders

MarkUs is easy to handle, and does not require much training. A manual for each type of user (administrators, graders, and students) guides the user through the basic operation. Documentation is provided in both French and English and helps users understand the underlying model as well as some of the less obvious operations.

In our institutions, there are one or two teachers that provide general support for the software, meaning that every encountered issue can be forwarded to them. Thanks to their knowledge about MarkUs, they help their fellow teachers better understand how to use the system. When the issue is an unknown one, they can contact the development team, that may open a ticket. It is fundamental to have these people to liaise with the development team, allowing the development team to synthesize feedback and prioritize requests for additional features.

THE IMPACT OF MARKUS

The introduction of a new tool in a pedagogical process needs to be monitored in order to get a relevant overview of its advantages and drawbacks. In this section, we will describe the key factors we have identified and how they have been impacted by MarkUs.

Analysis criteria

A study at Ecole Centrale Nantes evaluated the benefit of MarkUs according to several criteria:

- Quantitatively: how much does MarkUs increase the speed of the assessment process, how does it enhance the number of individual comments on each students' paper, do students' final evaluation improve thanks to the use of MarkUs during courses;
- Qualitatively: through annual surveys of teachers and students to validate the adequacy of the tool to teachers' needs, the improvement of teaching and the efficiency in the assessment process.

Using these analysis processes, we have collected the following key figures:

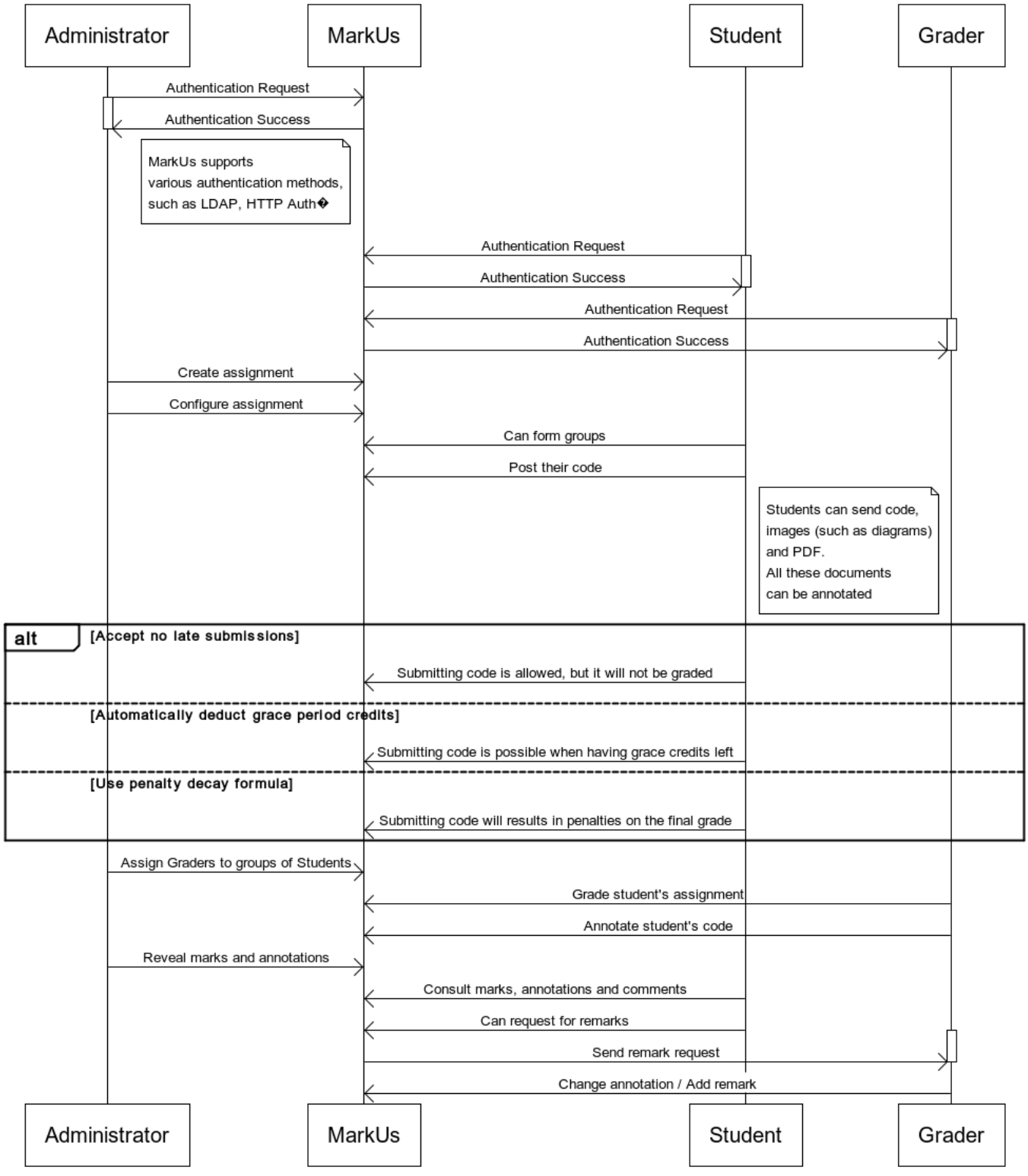

FIGURE 1. MarkUs features.

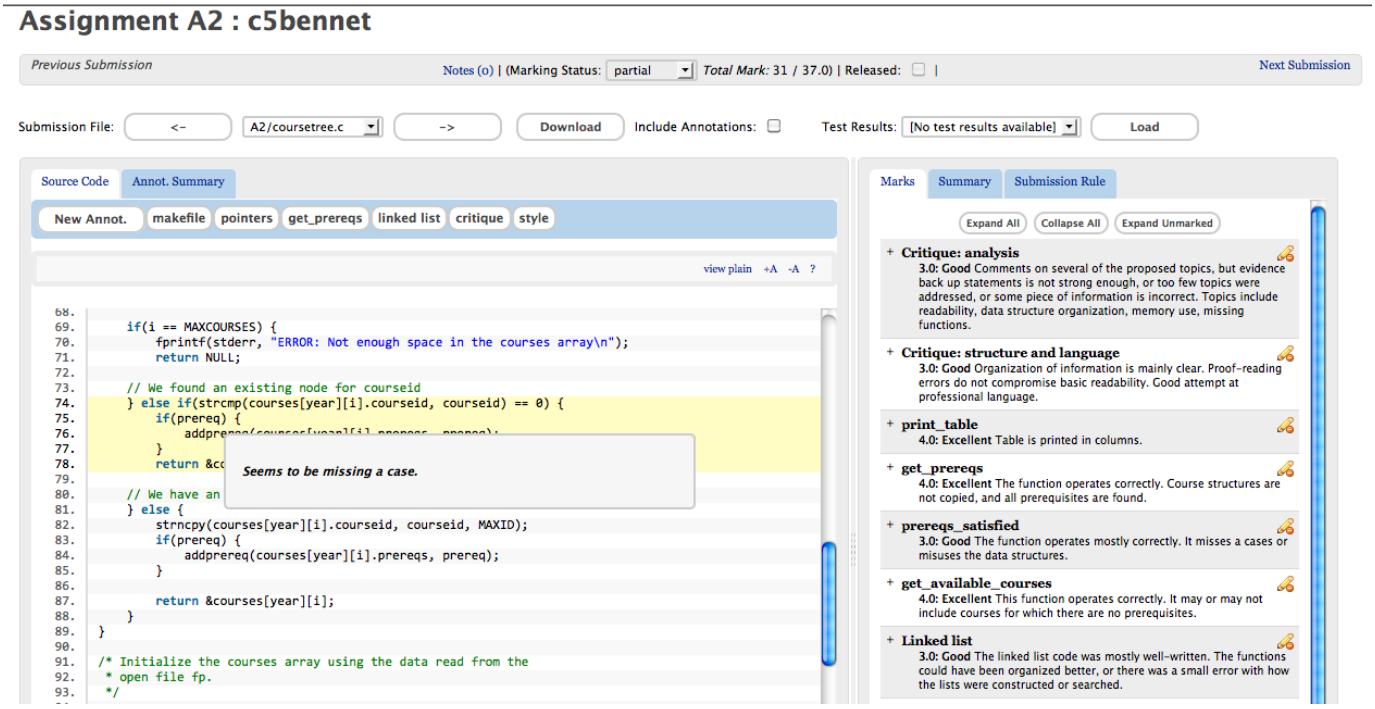

FIGURE 2. The grader interface

- Thanks to MarkUs, all teachers spend less time to assess the same quantity of students' papers. This time has decreased between 14% and 50%, depending on the teacher.
- The number of students submitting their work after the deadline has drastically decreased: before MarkUs has been adopted, this number was between 15% and 20% (students argued there was a misunderstanding about the final deadline, or that they have problems with their emails, . . .). Now, this number is between 5% and 10%. As students know the submission deadline is now an automatic process, they no longer try to cheat with this constraint.
- Since adopting MarkUs, the number of students' practical work reports that are returned to students after the final exam has been cut in half - a dramatic reduction. As the whole grading process has been made easier and faster, teachers are more motivated to do this assessment than before.

These factors have reinforced the idea that MarkUs was responding to needs that have been expressed for many years. It naturally fits into the learning context, it is easy to handle and, finally, it provides important feedback to the students on their work, and helps them understand the expectations.

While we have not carried out a similar study at the University of Toronto, we frequently solicit informal feedback from our users. Instructors confirm the above findings that the work involved in collecting and grading student submissions has significantly decreased.

Results

MarkUs is a tool that makes life easier for students and graders thanks to a wide range of useful features, such as: the centralized and versioned submission of the papers, assessment tools, the annotation of the code and images/PDF documents directly from the web interface, the ability to download the documents on his own computer to test the code of the students, etc.. As all these features have been developed specifically for the educational context, MarkUs improves the global pedagogical quality: annotations can be shared among multiple teachers and incorporates a correction (and notation) grid defined by the head-teacher. It also provides an additional organizational element: it is easy and quick to make corrections on large cohorts (e.g. by making it easy to reuse the same annotations on recurrent mistakes with just a click or two of the mouse) and it is therefore more motivating for teachers. As a noticeable side effect compared with traditional organization of practical sessions in computer science course, there has been a drastic decrease in the amount of paper that is used.

Analysis of the impact

Students have become much more respectful of deadlines than they were before, thanks to the fact of not being able to submit the papers once the due date is past. Student appreciate the ability to submit their work from their home computer and being able to double check exactly what they have submitted. MarkUs

has raised more interest to the annotations left on the platform. In addition, students appreciate receiving prompt feedback and correction from their teachers.

As far as the educational team is concerned, MarkUs has helped to unify the criteria for marking groups: even if a large number of students do not have the same teachers, their work is assessed according to the same marking criteria.

Dissemination

MarkUs is developed through a strong international collaboration between three institutions of higher education. These institutions include research laboratories. The philosophy of the research is rooted in the DNA of the software. MarkUs follows a development process perfectly framed, reinforced by rigorous and efficient quality assurance methods. Moreover, we spend much effort to disseminate our results through publications and conferences. The software, its functionality and its impact on the landscape of higher education were presented at various events. MarkUs received the special mention prize at the third "Trophées des Technologies Educative" of the french "Salon de l'Education / Educatice".

The availability of source code from MarkUs on the collaborative and social platform GitHub gives an international visibility to the project. In addition, we are currently helping teachers from various European institutions in their preparatory work in deploying MarkUs.

CONCLUSION AND FURTHER WORK

Until recently, students submitted their work by printing it on paper or by emailing it to their teachers. MarkUs greatly simplifies these steps by centralizing these documents. It manages classes and and allows students to work in groups. The software allows teachers to get an overview of intermediate work by students thanks to versioning features. As the papers and the corresponding personal assessment are centralized on a single platform, every student can readily access their work and the related comments, notes and evaluations of their teachers. This is a major step forward compared to the traditional use of paper. The integration of MarkUs in the learning environment results in numerous advantages, for both students and teachers.

Current work focuses on the incorporation of a testing framework into MarkUs. This means the code submitted by students would be immediately compiled and tested using criteria defined by teachers. We are also investigating the integration of a plagiarism detection tool into the application, to get automatic hints about the similarity of a submission compared to others. Some of these results would be displayed to students, which may deter them from plagiarizing. Finally, we continue to study how MarkUs could be extended to meet the needs of research, especially during the peer-reviewing process.

Based on a policy of innovation both demanding (through the quality assurance process) and ambitious (working directly with teachers and students to continually improve both the features and the user interface), the success MarkUs lies in how closely it meets the needs of both teachers and students.

REFERENCES

[1] Magnin, M., and Moreau, G., 2010. "Retour sur l'usage de tablet pc à l'ecole centrale de nantes : autonomie et initiatives". In 7ème Colloque International Technologies de l'Information et de la Communication pour l'Enseignement (TICE).

[2] Claroline Consortium, 2004. Claroline open source elearning and eworking platform. Available at http://www.claroline.net/.

[3] Dougiamas, M., 2007. Moodle course management system. Available at http://www.moodle.org/.

[4] MarkUs team, 2011. Markus. Available at http://www.markusproject.org/.

[5] Reid, K. L., and Wilson, G. V., 2005. "Learning by doing: Introducing version control as a way to manage student assignments". In SIGCSE'05, ACM Press, pp. 272–276.

[6] OLM team, 2008. Online marking tool. Available at https://stanley.cdf.toronto.edu/drproject/csc49x/olm.

[7] Review Board, 2011. Review board. Available at http://www.reviewboard.org/.

[8] Magnin, M., and Moreau, G., 2011. "La professionnalisation des étudiants en informatique par les projets libres". In 6ème Colloque Questions de pédagogies dans l'enseignement supérieur (QPES).