Insper Active Handout: creating and publishing class activity

Abstract—**Active teaching demands that the student engages in his own activities and reflects on the results, the student often needs to ask himself if his ideals / results obtained are correct and then continue in the activity, a frequent way to deliver these activities is through the use of handouts. This kind of student-centered methodologies can benefit from the use of computational tools to improve student autonomy and the instructor's perception of the classroom's progress and main points of attention in the script. This can be achieved by inserting a mechanism for interacting with the student in the handout, with quizzes, text questions, code exercises and collecting telemetry data of the student's progress through the material. In this article we propose an open source tool for creating interactive handouts and general course materials using the Insper active-handout, a tool created with focus on the ease of creating content and activities, with the ability to use different types of plugins that enables interaction on the handout and students telemetry. Different from other tools, all course material is located in a single git repository as text files and the content written completely in Markdown, this provides the instructor with the ability to work as a team on the course material while keeping track of course progress. The material is published on a well designed web page that can be easily hosted for free on github and customized to suit the discipline. Many plugins can be activated to add the interaction on the handout: Multiple Choice questions and Short/Long text questions can be used inline to the content to check specific issues of student understanding. A progress button can be added to force the student to complete tasks before proceeding through the script. Instructors can include an external task to the handout that can be added in order to make it clear to the student what he should do and many more. Every interaction in the handout (quizzes, progress buttons, . . .) collects information that is sent to a database that can be processed by the instructor through a provided tool. We will analyze the different uses of the tool in eight disciplines of an undergraduate computer engineering course and students' perception of the use of the same.**

Index Terms—**Assessment tools, Educational software**

I. Introduction

Student-centered learning is an educational approach in which students influence the pedagogical strategy of classes by directing content, activities and materials. The main focus of this approach is to place students as the central point of teaching, as they are the authors of the action of learning. Student-centered learning is the opposite of teacher-centered learning [1], but this does not make the instructor expendable from the course, on the contrary, the instructor starts to have a new primordial role: it is he who creates and defines the activities, engages and motivates students, monitors and provides feedback on the learning process [2].

This active learning is based on constructivist approaches where the student is invited to participate in the learning process during classes [8]. There are many strategies that can be used to engage and make the student active in the process, the best known methodologies are: Problem-based learning (PBL); Cooperative/Collaborative Learning; Flipped classroom; Learning by doing, among others [3].

The use of tools that help the student and the instructor during an activity can contribute with relevant information about the individual progress of the student (who is the center of attention), the group or the class [4], being able to serve as an independent observer and transparent progress of an activity. The data generated by the use of the tools can help the instructor to identify the points of difficulty and provide extra information about the activity, such as: software quality and student understanding of their program [5], [6]. The data can also be used to validate improvements in the discipline and collaborate with learning management, identifying bottlenecks and student behavior patterns [7], [8], or even suspected plagiarism [9]. Studies indicate [10] that data collected from students' interaction with content available online in a discipline can contribute to the improvement of the material.

Regarding the current development of these educational tools, it is necessary to consider some aspects, most of the time, it is up to the instructor of the discipline to design, prototype and develop the tools that end up being customized for their course. The development, in addition to implying an overload in teaching work, makes it difficult for new instructors to adhere to the tools and, consequently, their dissemination and validation in different environments. This generates a great demand for general tools, which have been developed and made available in an open way. As an example, we can mention PrairieLearn [11]–[13] which is a dynamic problem creation system, developed and maintained by Illinois instructors, or the A+ LMS [10], [14] which allows the instructor to create class material that mixes different types of content and activities (text, activity, code).

Some of these tools integrate or can be made with systems such as learning management system (LMS) [15], but usually their adherence and availability depend on the university or department. They are complex and rigid, do not provide an easy way to share material between courses and require IT to release access to new plugins, students need to be logged in and usually lose access to material at the end of the course or only have access to the version they attended.

High quality materials that are pleasant to use are hard to create from scratch. In this article, we present a workflow for creating interactive handouts and general course materials using the Insper Active-Handout [16], a tool created by professors of computer engineering at insper with focus on the ease of creating content and activities, with the ability to use different types of plugins that enables interaction on the handout and students telemetry.

II. Active Handout overview

Different from other tools, all course material is located in a single git repository [17] as text files and the content written completely in Markdown, this provides the instructor with the ability to work as a team on course material while keeping track of course progress. Markdown is a markup language very popular in code documentation [18] and supported by various tools. The material is published on a well designed web page that can be easily hosted for free on github [19] as a static web page. This flow is very favorable for instructors who already make use of these tools to deliver materials and receive student activities.

The core of the flow is based on opensource tools already well established for project documentation or code. For the conversion of markdown files into html we use mkdocs [20] a tool created in python with several extension plugins, the selected theme was Material [21]. The set of these two tools offer great features for hosting course material such as: 1) abbreviations; 2) code and text annotations; 3) code blocks; 4) data tables; 5) footnotes; 6) and more; . For the build and hosting process we use github tools such as github actions for the automated build process and gitub.io for hosting, with this set there is no need for the instructor to create and maintain such tools.

In order to add functionality that makes sense for a discipline and keeping the student on the same platform, we have developed a series of plugins that integrate with the ecosystem described above, the plugins allow the instructor to add: 1) multiple Choice questions and Short/Long text questions that can be used inline to the content to check specific issues of student understanding; 2) a progress button can be added to force the student to complete tasks before proceeding through the script; 3) instructors can include an external task to the handout that can be added in order to make it clear to the student what he should; 4) an easy way to add class slides and videos embedded on the page; 5) a progress bar that allows the student to follow their progress in the handout; 6) a class calendar that allows to associate calendar activities with handout pages; 7) authentication: allows the instructor to choose whether or not the student must be logged in to have access to the material, if not logged in the telemetry is anonymous; 8) notification system that shows up a message on the page indicating that the current content displayed is out of date; 9) telemetry plugin. Fig 1 is an example of an discipline handout page, in this image we can see the final page rendered, and some plugins activated.

The flow uses two types of plugins: one in python, which is used in the course material build process, and another in node (javascript/typescript) that are used to display the components, they are kept in a dedicated repositories that is included in the

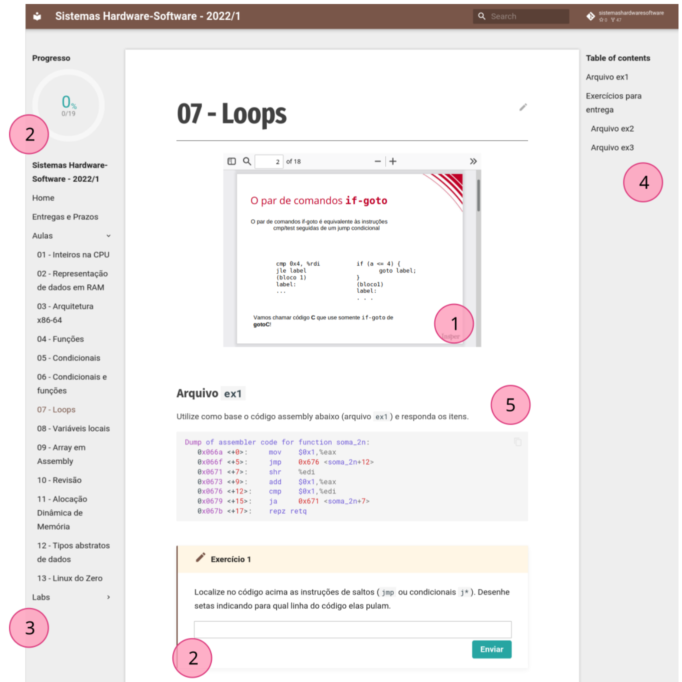

Fig. 1. page example overview of a real handout material, the main elements are highlighted and are: (1) slide plugin; (2) body of the page, where the content will be displayed; (3) short question plugin; (4) progress plugin; (5) course materials separated by category; (6) table of page contents;

project in the build stage so any active handout updates are automatically published to all instructors, without the need for them to update manually, the extra css that are needed by the plugins are also in a separate repository. Fig. 2 has an overview on of the proposed system, instructors only need to modify the highlight blocks which includes a single configuration file and the course and the course material written in markdown. In order to keep the data between student accesses, the plugin data is saved locally in the students' browser, thus keeping the answers already made between different student accesses.

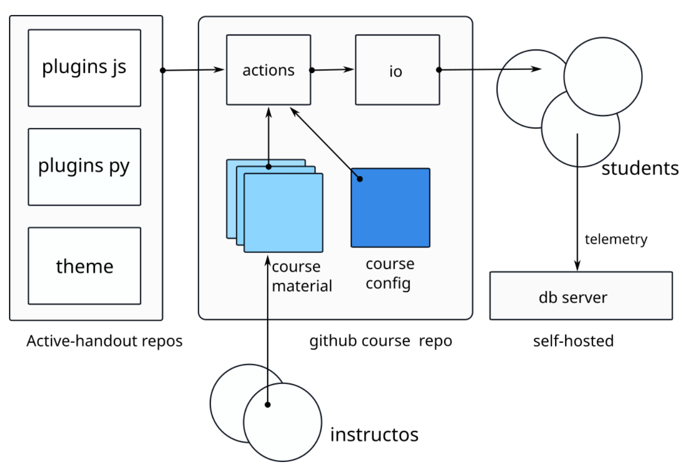

Fig. 2. General system diagram with the main components, the instructor only needs to modify the highlighted items: course material and configuration file.

Listing 1. Multiple choice quiz example

```
!!! exercise choice
    ```c
    void get_temp (int &temp) {
        _arduino_get_temp(temp);
    }

    void main (void) {
        int *temp;
        get_temp(temp)
    }
    ```

    The `temp` variable from the previous code is:

    - [x] local from main.
    - [ ] global as it was declared in main
    - [ ] local from `get_temp`

    !!! answer
        In C `main` is a function like any other!
        A variable declared inside it is visible only
        to the main function, unless it is passed as
        a reference as in the previous example.
```

Every interaction in the handout (quizzes, progress buttons, . . .) collects information that is sent to a database that can be processed by the instructor through a provided tool. The central idea of this telemetry data is to provide the instructor with different information that can help identify problem points in the handout or monitor the performance of the class during the execution of a task. To receive the telemetry data, we created a small server in Django that receives the information directly from the students' browser. There is no need for the instructor to configure the server to receive data from new questions, everything is done automatically

To keep everything in markdown we chose the admonition format as a way of using the plugins in the material, as shown in the Listing 1. In this example a multiple choice quiz is created and the result for the student is shown in the Fig. 3.

III. Results

The tool is used across several semesters of the course by seven disciplines from insper's computing department, the estimated number of unique students accessing the material per semester is over 300. Currently, three professors actively participate in the development and maintenance of the tools.

In a questionnaire sent to students who take courses that use the tool (N=42) in different semesters, in order to understand their behavior with the class materials. 90% said that they do not print the class material, 7% that they would like to print it and the rest already print it. Regarding whether students miss receiving printed material, 89% do not care and the rest miss it. 2% of students said they would prefer to have access to the class material in *.pdf* format instead of 72% who prefer the material as a web page and the rest are indifferent. A question was asked to understand how students behave in relation to the interactive components of the site and if they answer the questions just to be able to continue on the script, on a scale of 1 (always answer anything) to 3 (try to answer correctly) they said that: 32% answer anything just to continue in the handout; 44% responded that they sometimes respond to continue and

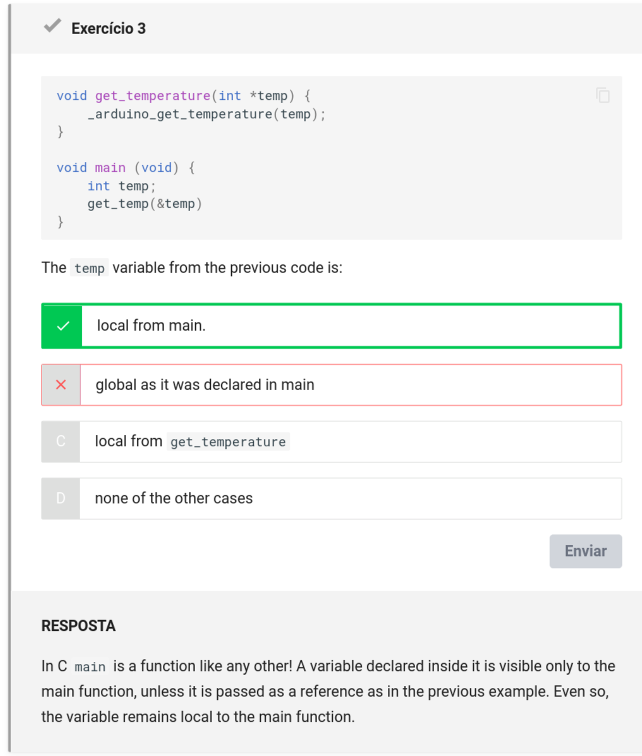

Fig. 3. Image resulting from the multiple choice exercise (Listing 1), here the student has already submitted an wrong answer and the correct answer is displayed along with an explanation.

only 25% respond by thinking and trying to get it right every time.

IV. Discussion and Future work

The tool is already used by many instructors and students and it already helps a lot in the creation and hosting of classroom materials, the concept that everything is available in one place is very practical, and it facilitates the contribution of the different authors of a course, even the students contribute to the improvement of the material. But much still can and should be done to improve and improve the tool, the following are ideas for future work:

We want to better understand the use of active-handout by students, for this we intend use eye tracker systems with students while accessing a material with the intention of understanding and improve the user (student) experience, we know from experience that many students do not update the page even after being notified by the plugin that the material being displayed is out of date, but many other unexpected behaviors may be taking place. How to prevent the student from answering the questions just to continue in the handout? We want to use this information as knowledge checkpoints, but from the students' responses to the forum this has not happened most of the time.

The project intend to expand the use of the tool with instructors from other areas and institutions, for this we need to map new instructors demands in order to create plugins or improve the existing ones, carry out the implementation and measure the results: measure perception of the instructor and student regarding the adoption of the tool. Contribution to the tool is also important, we want instructors to request new features or point out bugs, but that doesn't happen. No instructor apart from the three who do the development have opened an issue in the active-handout repository or suggest a code modification. We want to study how to create a community of active instructors who contribute to the tool they use as if they were your own tools.

New plugins that expand the capacity of active-handout are already under development and testing, we can mention the plugin that integrates vscode with handouts and allows the instructor to suggest programming exercises with unit tests for students, keeping the concept that all course material is been hosted in a single repository, the code execution status is displayed on the page, as well as the submitted code.

One line that we want to contribute is to create tools that assist the instructor in analyzing student data, such as detecting bottlenecks in the execution of an activity, or quizzes that students are having difficulty with, all this in an automated and integrated way. With the tools implemented, we want to revisit the course materials, propose improvements and analyze the results (are students spending time where expected?).

REFERENCES

[1] S. Pedersen and M. Liu, "Teachers' beliefs about issues in the implementation of a student-centered learning environment," *Educational Technology Research and Development*, vol. 51, no. 2, pp. 57–76, 2003.

[2] J. Michael, "Where's the evidence that active learning works?" *Advances in physiology education*, 2006.

[3] S. C. dos Santos, P. B. Reis, J. F. Reis, and F. Tavares, "Two decades of pbl in teaching computing: a systematic mapping study," *IEEE Transactions on Education*, vol. 64, no. 3, pp. 233–244, 2020.

[4] A. Ju and A. Fox, "Teamscope: measuring software engineering processes with teamwork telemetry," in *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 2018, pp. 123–128.

[5] T. Lehtinen, A. L. Santos, and J. Sorva, "Let's ask students about their programs, automatically," in *2021 IEEE/ACM 29th International Conference on Program Comprehension (ICPC)*. IEEE, 2021, pp. 467–475.

[6] R. Cardell-Oliver, "How can software metrics help novice programmers?" in *Proceedings of the Thirteenth Australasian Computing Education Conference-Volume 114*, 2011, pp. 55–62.

[7] J. H. Hayes, A. Dekhtyar, and D. S. Janzen, "Towards traceable test-driven development," in *2009 ICSE Workshop on Traceability in Emerging Forms of Software Engineering*. IEEE, 2009, pp. 26–30.

[8] A. Bogarín, C. Romero, R. Cerezo, and M. Sánchez-Santillán, "Clustering for improving educational process mining," in *Proceedings of the fourth international conference on learning analytics and knowledge*, 2014, pp. 11–15.

[9] L. Yan, N. McKeown, M. Sahami, and C. Piech, "Tmoss: Using intermediate assignment work to understand excessive collaboration in large classes," in *Proceedings of the 49th ACM technical symposium on computer science education*, 2018, pp. 110–115.

[10] T. Sirkiä and L. Haaranen, "Improving online learning activity interoperability with acos server," *Software: Practice and Experience*, vol. 47, no. 11, pp. 1657–1676, 2017.

[11] M. West, N. Walters, M. Silva, T. Bretl, and C. Zilles, "Integrating diverse learning tools using the prairielearn platform," in *Seventh SPLICE Workshop at SIGCSE*, 2021.

[12] M. Silva, M. West, and C. Zilles, "Measuring the score advantage on asynchronous exams in an undergraduate cs course," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 2020, pp. 873–879.

[13] S. Mahmood, M. Zhao, O. Khan, and G. L. Herman, "Caches as an example of machine-gradable exam questions for complex engineering systems," in *2020 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2020, pp. 1–9.

[14] V. Karavirta, P. Ihantola, and T. Koskinen, "Service-oriented approach to improve interoperability of e-learning systems," in *2013 IEEE 13th International Conference on Advanced Learning Technologies*. IEEE, 2013, pp. 341–345.

[15] M. Machado and E. Tao, "Blackboard vs. moodle: Comparing user experience of learning management systems," in *2007 37th annual frontiers in education conference-global engineering: Knowledge without borders, opportunities without passports*. IEEE, 2007, pp. S4J–7.

[16] I. S. Montagner, R. C. Ferrão, and A. T. Kurauchi, "Creating and publishing active learning handouts."

[17] M. A. Angulo and O. Aktunc, "Using github as a teaching tool for programming courses," in *2018 Gulf Southwest Section Conference*, 2019.

[18] G. A. A. Prana, C. Treude, F. Thung, T. Atapattu, and D. Lo, "Categorizing the content of github readme files," *Empirical Software Engineering*, vol. 24, no. 3, pp. 1296–1327, 2019.

[19] T. Griffin and S. Seals, "Github in the classroom: Not just for group projects," *Journal of Computing Sciences in Colleges*, vol. 28, no. 4, pp. 74–74, 2013.

[20] T. Christie. Mkdocs is a fast, simple and downright gorgeous static site generator that's geared towards building project documentation. documentation source files are written in markdown, and configured with a single yaml configuration file. it is designed to be easy to use and can be extended with third-party themes, plugins, and markdown extensions. [Online]. Available: https://www.mkdocs.org/

[21] M. Donath. Write your documentation in markdown and create a professional static site in minutes – searchable, customizable, for all devices. [Online]. Available: https://squidfunk.github.io/mkdocs-material/