Supporting Learning Analytics in Computing Education

Daniel M. Olivares
Human-Centered Environments for Learning and Programming (HELP) Lab
School of Electrical Engineering and Computer Science
Washington State University
Pullman, WA 99164 USA
daniel.olivares@wsu.edu

Christopher D. Hundhausen
Human-Centered Environments for Learning and Programming (HELP) Lab
School of Electrical Engineering and Computer Science
Washington State University
Pullman, WA 99164 USA
hundhaus@wsu.edu

ABSTRACT

As is the case for many undergraduate STEM degree programs, computing degree programs are plagued by high attrition rates. This is especially true in early computing courses, in which failure and drop-out rates in the 35 to 50 percent range are common. By collecting learning process data as students engage in computer programming assignments, computing educators can place themselves in a position not only to better understand students' struggles, but also to better tailor instructional interventions to students' needs. We have developed OSBLE+, a learning management and analytics environment that interfaces with a computer programming environment to support the automatic collection of learners' programming process and social data as they work on programming assignments, while also providing an interactive environment for the analysis and visualization of those data. In ongoing work, we are using OSBLE+ to explore two possibilities: (a) leveraging learning and social data to strategically deliver automated learning interventions, and (b) presenting learners with visual representations of their learning data in order to prompt them to reflect on and discuss their learning processes.

CCS Concepts

• Social and professional topics~Computing education
• Applied computing~Learning management systems

Keywords

Computing education research; Learning analytics; Learning Management System; Social Learning; Data collection; Visualizations; OSBLE

1. INTRODUCTION

STEM undergraduate degree programs are notorious for having high drop-out and failure rates. Computing degree programs are among the worst culprits; attrition rates of 35 to 50 percent are common, especially in early courses. Learning analytics constitutes a promising way to address this problem: If we can capture learning process data as students work on computer programming assignments, we can use those data to understand their struggles, and ultimately to better tailor instruction to their needs.

To that end, we have been developing OSBLE+, a learning management and analytics environment that facilitates the collection and analysis of learning process data in computing courses (see Figure 1). These data include not only programming process data such as compilations, edits, and run-time exceptions, but also social data: students' asynchronous discussions as they engage in computer programming. In this poster, we will present the design of OSBLE+, including its software architecture and user interface, and we will highlight our ongoing efforts to further develop OSBLE+ and explore its ability to help facilitate improved learning processes and outcomes in computing education.

LAK '17, March 13-17, 2017, Vancouver, BC, Canada
ACM 978-1-4503-4870-6/17/03.
http://dx.doi.org/10.1145/3027385.3029472

2. RELATED WORK

Within computing education, initial interest in learning analytics was spurred by an interest in building predictive models of learning success. Many such models (e.g., [2, 4, 8]) were based on programming process data collected within an integrated development environment (IDE), where students typically spend large amounts of out-of-class time working on individual programming assignments. An outgrowth of one of these [4] is the Black Box project [1], which makes publicly-available a large corpus of data automatically collected through the BlueJ novice programming environment [6].

While Black Box makes programming process data available, it does not provide a means of analyzing the data. By supporting the interactive visual exploration of learning process data, learning analytics *dashboards* (e.g., see [7]) provide one such means. They can be used to augment face-to-face teaching, online learning, or even blended learning settings [7]. They can be part of LMSs such as Moodle, custom LMSs as seen in [5], or a student centered dashboard that uses of both learning analytics and formative assessment [5]. OSBLE+ contributes to this line of research through its development of a learning analytics dashboard for computing instructors.

3. OSBLE+ DESIGN

Figure 2 presents an overview of the OSBLE+ software architecture, which includes three key components:

- The *Visual Studio integrated development environment* (IDE) is where computing students spend much of their out-of-class time

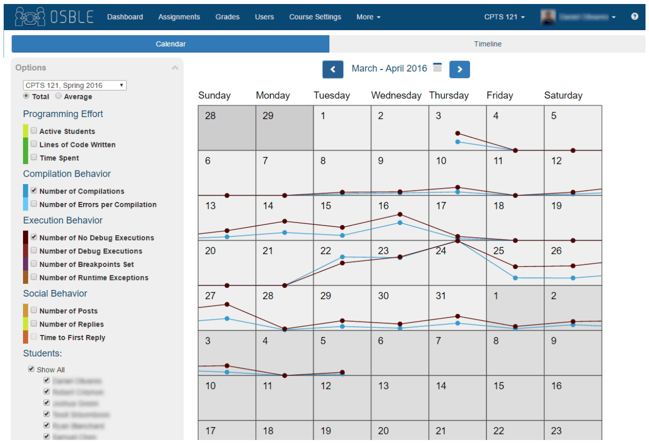


Figure 1. The OSBLE+ *learning analytics* dashboard

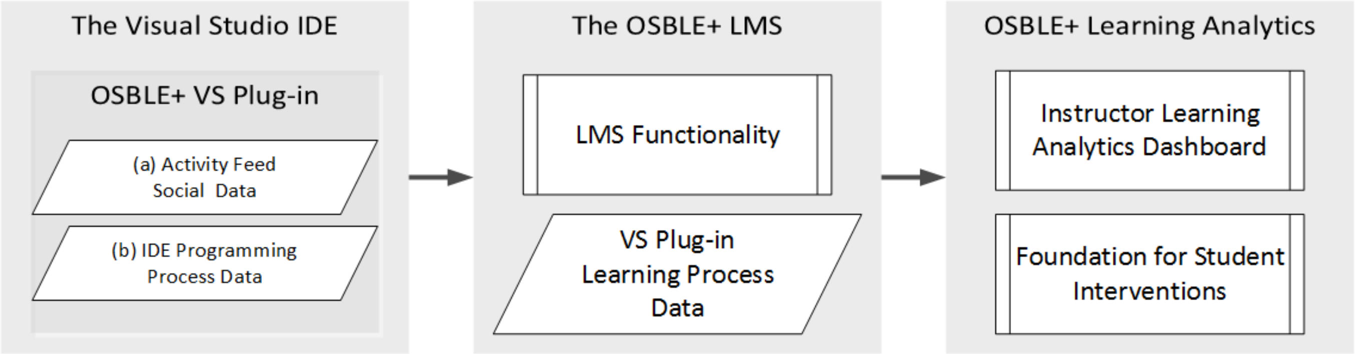


Figure 2. An overview of the OSBLE+ software architecture

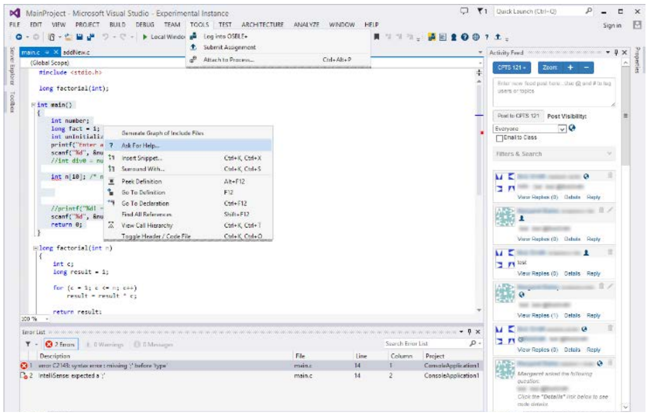
working on individual computer programming assignments, which are typically the centerpiece of computing courses.

- The *OSBLE+ plug-in* to the Visual Studio IDE serves two key purposes: (a) it augments Visual Studio with an *activity stream*, which provides an asynchronous forum for students to discuss issues that arise during the programming process (see Figure 3); and (b) it automatically collects students' programming process data (e.g., edits, compilation attempts, run-time exceptions) and social data (activity stream posts and replies).

- The *OSBLE+ learning management system* (LMS) provides many traditional LMS functions, such as a file system for disseminating course materials, and an assignment system for posting and submitting course assignments. However, because it receives learning process data from the OSBLE+ plug-in, it can also present those data to learners and instructors. In the present implementation, OSBLE+ provides the instructor with a learning analytics environment for exploring student learning process data Within this environment, instructors can select a set of data to visualize, along with a range of dates, and then see line graphs of those data within a calendar view (see Figure 1).

4. STATUS AND FUTURE WORK

To date, we have used various versions of OSBLE+ in large CS1 and CS2 courses at Washington State University. .In an empirical study of the CS2 course, student social participation was strongly correlated with assignment success [3]. Motivated by this result and by social learning theory, we are presently exploring ways to use students' programming and social data as a basis for *automated* interventions—delivered directly within the IDE—that nudge students toward greater social participation at strategic points in the learning process. We are also interested in injecting visual representations of learner data into the activity stream, in order to prompt students to engage in reflective conversations about their learning processes.



```
#include <stdio.h>

long factorial(int);

int main()
{
    int number;
    long fact = 1;
    int uninitializ
    printf("Enter a
    scanf("%d", &nu
    //int div0 = nu

    int n[10]; /* n



    //printf("%d1 -
    scanf("%d", &nu
    return 0;
}

long factorial(int n)
{
    int c;
    long result = 1;

    for (c = 1; c <= n; c++)
        result = result * c;

    return result;
```

	Description	File	Line	Column	Project
1	error C2143: syntax error : missing ';' before 'type'	main.c	14	1	ConsoleApplication1
2	IntelliSense: expected a ';'	main.c	14	2	ConsoleApplication1

Figure 3. The Visual Studio IDE with the OSBLE+ plug-in

5. ACKNOWLEDGMENTS

This project is funded by the National Science Foundation under grant no IIS-1321045

6. REFERENCES

[1] Brown, N.C.C. et al. 2014. Blackbox: A Large Scale Repository of Novice Programmers' Activity. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. ACM. 223–228.

[2] Carter, A.S. et al. 2015. The Normalized Programming State Model: Predicting Student Performance in Computing Courses Based on Programming Behavior. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*. ACM. 141–150.

[3] Carter, A.S. and Hundhausen, C.D. 2016. With a Little Help From My Friends: An Empirical Study of the Interplay of Students' Social Activities, Programming Activities, and Course Success. *Proceedings of the 2016 ACM Conference on International Computing Education Research*. ACM. 201–209.

[4] Jadud, M.C. 2006. Methods and Tools for Exploring Novice Compilation Behaviour. *Proceedings of the Second International Workshop on Computing Education Research*. ACM. 73–84.

[5] Rößling, G. et al. 2008. Enhancing Learning Management Systems to Better Support Computer Science Education. *SIGCSE Bull.* 40, 4 (Nov. 2008), 142–166.

[6] Utting, I. et al. 2012. Web-scale Data Gathering with BlueJ. *Proceedings of the Ninth Annual International Conference on International Computing Education Research*. ACM. 1–4.

[7] Verbert, K. et al. 2014. Learning Dashboards: An Overview and Future Research Opportunities. *Personal Ubiquitous Comput.* 18, 6 (Aug. 2014), 1499–1514.

[8] Watson, C. et al. 2014. No Tests Required: Comparing Traditional and Dynamic Predictors of Programming Success. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. ACM. 469–474.