

Cite as: Carvalho, E. S. (2015). Analyzing the quality of students interaction in a distance learning object-oriented programming discipline. *Interdisciplinary Journal of e-Skills and Life Long Learning*, 11, 85-99. Retrieved from <http://www.ijello.org/Volume11/IJELLv11p085-099Carvalho0919.pdf>

Analyzing the Quality of Students Interaction in a Distance Learning Object-Oriented Programming Discipline

Elizabeth Simão Carvalho
Portuguese Open University, and
CIAC – Centre for Research in Arts and Communication,
Lisbon, Portugal

ecarvalho@uab.pt

Abstract

Teaching object-oriented programming to students in an in-classroom environment demands well-thought didactic and pedagogical strategies in order to guarantee a good level of apprenticeship. To teach it on a completely distance learning environment (e-learning) imposes possibly other strategies, besides those that the e-learning model of Open University of Portugal dictates. This article analyses the behavior of the students of the 1st cycle in Computer Science while interacting with the object-oriented programming (OOP) discipline available to them on the Moodle platform. Through the evaluation of the level of interaction achieved in a group of relevant selected actions by the students, it is possible to identify their relevancy to the success of the programming learning process. Data was extracted from Moodle, numerically analyzed, and, with the use of some charts, behavior patterns of students were identified. This paper points out potential new approaches to be considered in e-learning in order to enhance programming learning results, besides confirming a high level of drop-out and a low level of interaction, thus finding no clear correlation between students' success and the number of online actions (especially in forums), which reveals a possible failure of the main pillar on which the e-learning model relies.

Keywords: eLearning, student's behavior, distance learning of programming language.

Introduction

Teaching programming is difficult. Much research from many different perspectives has been devoted to the topic during the past couple of decades, but there is still no consensus on what is the most effective way to teach programming (Vihavainen, Paksula, & Luukkainen, 2011). Most universities are still using a traditional format in the introductory programming courses. The traditional format consists of lectures, take-home assignments, and perhaps also demo sessions where model solutions to the exercises are shown.

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

The traditional format consists of lectures, take-home assignments, and perhaps also demo sessions where model solutions to the exercises are shown.

One of the most challenging aspects of any information systems or computer science curriculum involves helping students learn the concepts of computer programming (Shehane & Sherman, 2014). This task involves many com-

Editor: Janice Whatley

Submitted: January 4, 2015; Revised: May 15, July 20, 2015; Accepted July 28, 2015

plexities, including skill development, different learning styles, helping students transition from a problem statement to algorithmic logic, developing programming code from that logic, debugging and resolving the program code, to produce a working program that meets the problem requirements. This learning process is even more challenging when students are also being introduced to a new programming language and development interface. Things can become a little bit more complicated when you have to teach object-oriented programming in a totally e-learning environment.

This article analyses the students' behavior while interacting and using the object oriented programming discipline material available on the Moodle platform of the Open University (UAb) of Portugal, with the undergraduate students of the 1st cycle in the Computer Science degree. The UAb is a University that operates entirely on a distance learning environment. All teaching and learning activities were developed online (with an emphasis on asynchronous communication), and this discipline is taught in the first year of the undergraduate course (second semester). A relevant problem to be investigated is the cause for the high level of drop-outs and overall failure in learning. The numerical analysis of the data retrieved from the Moodle actions report helped the authors to identify some correlations between student's actions on the platform and their overall learning performance in the discipline. The goal is to verify if the actions rate per student (the student's interaction profile) has some correlation with success or failure.

Despite recent advances of electronic technologies in e-learning, a consolidated evaluation methodology for e-learning applications is not available. Maybe the main cause for this is the complexity that the evaluation of an e-learning environment demands. Many different perspectives and, thus, dimensions in the analysis process can be considered, such as the quality of learning, teaching, learning environment, and interaction (Ardito et al., 2006). Each of these dimensions can be evaluated according to a group of pre-defined and chosen indicators. In the case of interaction, we may consider that the quality of students' interactions is one of the most relevant indicators of successful learning.

This paper is organized in four main sections. The first will provide a short background of some relevant studies and work that has been being carried out to understand better the behavior of users' interactions in an e-learning environment, which served as an inspiration for this article. The second section will give a brief description of how the discipline is structured and presented to the student. The third section will describe the data that was used and the analysis carried out. The fourth section will list the main findings, followed by a section discussing the results and proposing some approaches to modify practice.

Background

E-Learning refers to the use of ICT to transform and support the learning process (Kahiigi, Ekenberg, & Hansson, 2007). E-Learning is defined as the acquisition and use of knowledge which is distributed and facilitated primarily by electronic means. Such electronic means may include internet, intranet, extranet, CD-ROM, video tape, DVD, TV, and personal organizers. E-Learning can be carried out in several ways, which include computer based, asynchronous, and synchronous learning (Loncar, Barrett, & Liu, 2014). This facilitates an environment where the students take ownership of their learning.

Although modern technologies allow for synchronous communication (such as Skype), the e-learning model (Pereira, Mendes, Morgado, Amante, & Bidarra, 2007) is based primarily on asynchronous communication as an aid to learning. Asynchronous learning is a student-centered teaching method that uses online learning resources to facilitate information sharing outside the constraints of time and place among a network of people. Asynchronous learning is based on constructivist theory, a student-centered approach that emphasizes the importance of peer-to-peer

interactions. This approach combines self-study with asynchronous interactions to promote learning, and it can be used to facilitate learning in traditional on-campus education, distance education, and continuing education (Northey, Bucic, Chylinski, & Govind, 2015). A special emphasis is given to discussion forums, which aim to promote reflection and knowledge sharing (a higher level of learning). Other asynchronous approaches may be adopted in the future with the development of Web 2.0.

E-learning systems store large amounts of data based on the history of users' interactions with the system. These pieces of information are usually used for further course optimization, finding e-tutors in collaborative learning, analysis of students' behavior, or for other purposes. The interest in analyzing this data is gradually increasing within the academic community. The overall purpose is to enhance the learning experience. A research area, referred to recently as learning analytics, has been created to focus on automatic analysis of educational data (Chatti, Dyckhoff, Schroeder, & Thus, 2012).

Slaninová, Kocyan, Martinovic, Drázdilová, and Snásel (2012) presented the comparison of selected methods of process mining and sequential mining to find the students' behavioral patterns performed in the e-learning system. The patterns were visualized by the methods from graph theory. Their obtained patterns and relations between them are presented using complex networks, such as visualization and pattern clusters extraction optimized by spectral graph partitioning. They have found that the sequences are order dependent and it is better to respect this fact while comparing the sequence similarity. Due to this reason, the methods for finding the longest common subsequence were used.

Hogo (2010) introduces an evaluation methodology for e-learners' behavior that can feed back to decision makers in e-learning systems. His work presented the use of different fuzzy clustering techniques as fuzzy c-means and kernelled fuzzy c-means to find the learners' categories and predict their profiles.

Rovai & Barnum (2007) analyzed nineteen on-line graduate courses in order to determine how perceived learning varies by course and its relationship to active and passive participation by students in on-line discussions. Study results provided evidence that significant differences existed by course, suggesting that quality assurance is an issue in Internet-based instruction. Moreover, female students felt that they learned more than their male counterparts. Only active interaction, operationalized by the number of messages posted by students per week, was a significant predictor of perceived learning. Passive interaction, analogous to listening to but not participating in discussions, and operationalized by the number of accesses to the discussion boards of the e-learning system each week, was found not to be significant.

Sriwardiningsi & Siswono (2014) conducted a survey on 274 e-learning students from Online Binus University and Indonesia Open University (UT). Ten hypotheses were proposed but only some hypotheses were recognized as valid. Variables such as motivation, digital literacy, and satisfaction would affect directly the outcome of understanding of student learning, while the curriculum material and interaction with the e-learning website did not affect the understanding of student learning.

Several case-studies are illustrated by Murphy, Walker, and Webb (2013) in their book about the e-learning experience. They illustrate the key concepts, giving some hints about how the educators should use technology to motivate learners and encourage productive interaction.

Although these studies focused on analyzing e-learning student's behavior or the e-learning experience and practice, none of them actually looked at their behavior in terms of Moodle usage, nor in the context of a typical programming discipline, or even tried to establish any correlation between student's actions and programming learning success. This article focuses on the impact of

an e-learning environment on the success in learning programming and tries to perceive how these students explored the content that is made available to them, especially material most relevant to help them acquire object oriented programming skills.

Discipline Content and Organization

The object oriented programming (OOP) discipline being studied here aims at providing students with fundamental knowledge and practices regarding the principles, main concepts, models, and main techniques related with computer programming based on the object-oriented paradigm. Teaching the discipline syllabus uses analysis of the object-oriented programming paradigm, algorithms, and blocks of code. Different problems are analyzed and provided to be solved using an object-oriented programming approach. The students are stimulated to design and implement new tactics or improvements of existing ones. The programming language and environment adopted in this case are C++ and Eclipse IDE, respectively. The Moodle environment is used to provide material and provision for communication with peers and tutors.

The syllabus assumes a total workload of 156 hours; including 26 contact hours (this contact is not face-to-face, but instead asynchronously, preferably via forums or, in extremis, synchronous, via teleconference). The students' assessment is done through 2 digital written documents (called e-folios) during the semester and a classroom assessment (called p-folio) in the end of the semester. If the student fails to be approved in the p-folio, an extra classroom assessment (called extra p-folio) can be completed during September. The e-folios contribute 40% to the final grade, while the p-folio or extra p-folio is 60%. They can also be assessed through a unique classroom exam. If they fail, they have a last chance of being approved with an appeal exam. The e-folios within the context of this discipline are C++ projects that are implemented for a given problem scenario.

The Moodle environment provided for the discipline is composed of 7 topics, in which the students face an increasing degree of complexity and are asked to execute a learning activity (that does not compute to the final grade and is not obligatory, called AF). Each topic lasts 2 weeks (except the first one) and didactic material (with vast examples of codes and relevant links, and even videos) specially developed for the students, are made available, besides the solution of the learning activities and e-folios, on the Moodle learning environment. All the topics have a forum where support for topic content is guaranteed by the teacher asynchronously. The teacher always answers the questions within 48 hours at the weekend, or on a daily basis from Monday to Friday, besides motivating the students to participate. Table 1 shows in detail how the discipline is organized:

At the end of topics 4 and 6, e-folios A and B respectively were given out to the students. The p-folio was given out after the end of topic 7. The appeal exam took place before the beginning of the next semester (usually September, because the semester is from March to July). All the AFs and e-folios are graded by the teacher and made available to the students after the deadlines applied to each of them.

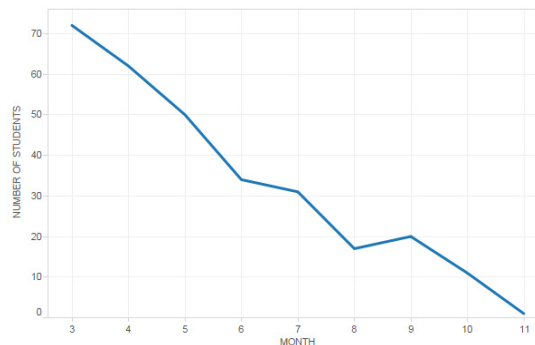
The discipline for this study had a total of 124 students subscribed and two classes, each with 62 students. It took place in the 2014 academic year.

Table 1: Discipline organization

Topic	Content	Goals	Duration
1	Introduction to object oriented programming	Familiarization with the concept of programming and object oriented software quality; installation of Eclipse IDE and configuration;	1 week
2	Programming resources not specific related to OO	Learn the features available in C ++ that are not directly related to OO programming, but inherited in large part from the C language; implement a program in C ++ with the use of functions and all other features not directly related to OOP	2 weeks
3	Definition and creation of classes and their attributes	Familiarization with the concept of class, object and its attributes; use of constructors, destructors, access controls, friends and nested classes	2 weeks
4	Overload, conversion and arrays of objects	Learn to create overload member functions and operators; conversion between objects and simple types; arrays and lists of objects	2 weeks
5	Single inheritance	Creation of classes and methods with single and simple inheritance mechanism	2 weeks
6	Multiple inheritance and polymorphism	Creation of classes with multiple inheritance mechanism; virtual classes	2 weeks
7	Library streams, templates and exception handling	Using and creating templates and exception handling; using the streams library	2 weeks

Analyzing the Data

The data was extracted directly from the Moodle platform, through its reports feature. The activity record report gives detailed information about each student's behavior during the year. It informs what, when (date and time) and from where (IP address) individuals have executed some interaction with the discipline.

**Figure 1: Total of students interacting per month**

Although we had 124 students subscribed (7 female and 117 male, 9 from outside Portugal), only a small number of them actually attended (32%) and were approved in the discipline (passed) (26%). The period of time under consideration is from the end of March to November of 2014. The academic semester is divided in different periods of time: 13-14 weeks of classes, starting usually by the end of February, to the beginning of June; a period of 4 weeks of normal classroom exams, starting in June and ending at the beginning of July; finally a period of 4 weeks of extra

exams, realized during September. Figure 1 shows the total number of students that interacted per month on Moodle. In March, a total of 72 students executed some action, although only about half of them actually tried to be approved and by July (end of the semester) only 32 remained active. Table 2 illustrates the scenario we had and the respective figures:

Table 2: Totals of students in the discipline along time

<i>SCENARIO</i>	<i>TOTALS</i>
Subscribed students	124
Interacting at least once with the discipline	115
Completed e-folio A	35
Completed e-folio B	30
Completed p-fólio	27
Completed only final exam	5
Completed appeal exam	2
Approved	31
Highest number of accesses per student	529
Average number of accesses per student	153
Minimum number of access per student	1

Actions in the Moodle VLE can take the form of any of those listed below. In the context of this discipline, it is expected that all students will have some interaction especially at those actions marked with an asterisk:

- assign submit (*)
- assign view (*)
- assign view all
- assign view submission grading table
- assign view submit assignment form
- book print
- book print chapter
- book view (*)
- book view all
- book view chapter (*)
- choice choose
- choice choose again
- choice view
- choice view all
- course view (*)
- folder view (*)
- folder view all (*)
- forum search (*)
- forum subscribe all
- forum unsubscribe all
- forum user report
- forum view forum (*)
- forum view forums
- imscp view all
- label view all
- page view all
- resource view (*)

- resource view all
- url view all
- user view
- user view all

There are four types of reports in Moodle that the teacher can easily export to other software, including the possibility of applying some filters to it:

- Logs, that are activity reports of the site or course;
- Activity reports that sum the number of views per activity and resource;
- A participation report, which allows tutors to see who has completed a certain action, like a quiz;
- Statistics allowing for tracking the number of posts and views from a certain time frame graphically.

Some reports were produced directly using a tool available in the Moodle platform and exported in .xls format and loaded into an Excel sheet. After extraction, the data was treated to find out totals, average values, and detect potential trends. Based on the numerical analysis, graphs were produced to more easily present the results that were thus being obtained.

Although all of these interactions are important to evaluate the level of interactivity between the students and the platform, before proceeding with the analysis, it is necessary to clarify the meaning of the actions with (*) in the list above and according to the discipline structure, the minimum number expected in each case:

- **Assign view/submit** – E-folios are two programming tasks (more complex than the learning activities) and they contribute to the final grade of the students. An assign submit happens each time the student uploads an e-folio, while an assign view is when he consults the grade and evaluation comment left by the teacher. The student should execute at least 2 assigns views and 2 assigns submit during the semester;
- **Book view** – The discipline is fully described (goals, methodology, evaluation, bibliography, etc.) and organized in a timetable (tasks to be executed by specific dates and weeks each month). The student should consult or download this information at the very beginning of the semester. This action should be executed at least once per student.
- **Course view** – Each time the student enters the Moodle space of the discipline, this action happens, if he is going to do any action further.
- **Folder view** – didactic material is available for self-study. This material includes tutorials specially developed for the discipline by the teacher and several code examples to test and run. The folder view action is executed each time the students view or download the material. Although it is helpful, the students may use other resources to support their studies (books, online tutorials, etc.). A minimum of 7 folders view per student is expected in order to download all the available material;
- **Forum actions** – The asynchronous conversation between students and teacher is one of the most important premises in the e-learning model in UAb. The goal is to achieve a high level of message exchanges between all the students and between the students and the teacher. The students should post their queries and cooperate with other students, exchanging points of view and deliberations. Within these messages, links to other web resources are normally made (videos, articles, etc.). There are a total of 10 forums within the discipline. Each forum may have several threads. It is expected that each student par-

ticipate at least once in each available forum. The minimum number should be 10 forum actions (this number represents a weak level of interaction);

- **Resource view** – Learning activities are proposed along the semester. These activities are programming tasks in C++ (development of a C++ project) and, although they do not contribute to the final grade, they are very relevant to the learning process and are identified as resources. A resource view occurs whenever a student executes an upload of a learning activity or consults the comment left by the teacher about solutions. Each student should at least execute 14 resources view per semester.

Even though they are the most relevant actions, we may state that their relevancy does vary in terms of the programming learning process. Table 3 shows the relevancy of each one of them, classifying it in 3 distinct levels (high, medium and low).

Table 3: Relevancy of the action within the context of programming learning

<i>Action</i>	<i>Relevancy within the context of programming learning</i>
Assign submission	High
Assign view	Medium
Book view	Low
Course view	Low
Folder view	Medium
Forums actions	High
Resource view	High

Based on the structure and timetable of the discipline (total number of topics, folders, learning activities, and forums), each student should perform a minimum number of actions per semester, to be able to download all the material, upload and consult resources and e-folios, consult and post at least once at each forum, and consult the discipline structure and timetable at the beginning. Table 4 summarizes the minimum number of actions each student should perform to show some evidence of participation. Minimum numbers are quite acceptable for all the actions except for the forum and course view actions.

Table 4. Minimum number of actions expected per student

<i>Action</i>	<i>Minimum expected number per student</i>	<i>Minimum per total (162)</i>	<i>Minimum per total approved(31)</i>
Assign submission	2	$(2 \times 162) = 324$	$(2 \times 31) = 62$
Assign view	2	$(2 \times 162) = 324$	$(2 \times 31) = 62$
Book view	1	$(1 \times 162) = 162$	$(1 \times 31) = 31$
Course view	$(2 + 2 + 1 + 7 + 10 \times 2) = 32$	$(32 \times 162) = 5184$	$(32 \times 31) = 992$
Folder view	7	$(7 \times 162) = 1134$	$(7 \times 31) = 217$
Forums actions	10	$(10 \times 162) = 1620$	$(10 \times 31) = 310$
Resource view	$(7 \times 2) = 14$	$(14 \times 162) = 2268$	$(14 \times 31) = 434$

Figure 2 illustrates the totals per action. Course view (38.5%) is the most recurrent action performed by the students followed (in this order) by the book view chapter (17.5%), folder view (12.5%), resource view (10.5%), assign view (7.8%) and book view (3.4%). All the other actions are less significant and have approximately the same total values.

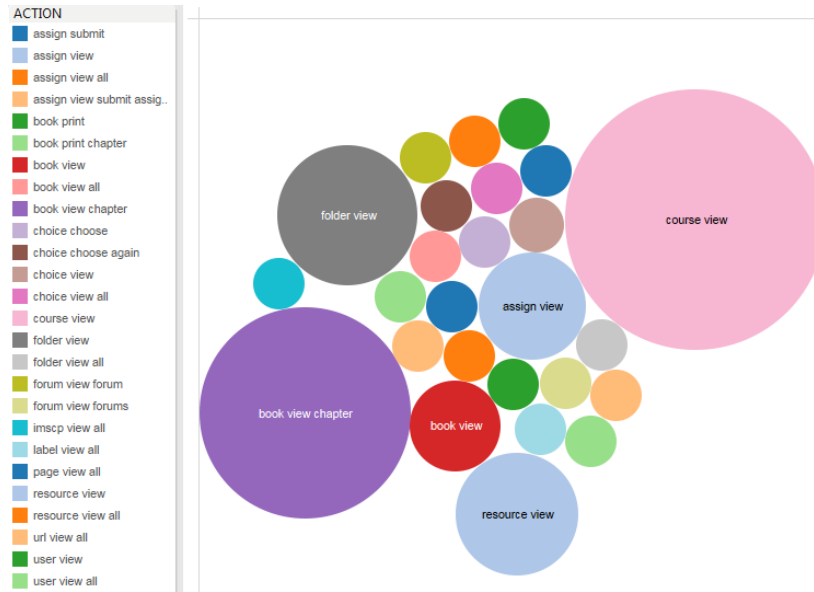


Figure 2: Totals per actions

Figure 3 shows the total number of actions only by weekdays and by weekdays per month. Most of the students in UAb are workers with average age 30-40 years. Surprisingly, the weekend is not when the highest accesses happen, but instead on Monday (24%). Topics are usually made visible to the students on Mondays and e-folios, on Fridays. As time passes we can also notice that the activity decreases significantly and, although Monday is always when more actions are registered, in April (Month 4), Tuesday has slightly more actions, while Thursday, Friday, Saturday or even Sunday, have almost the same number of actions.

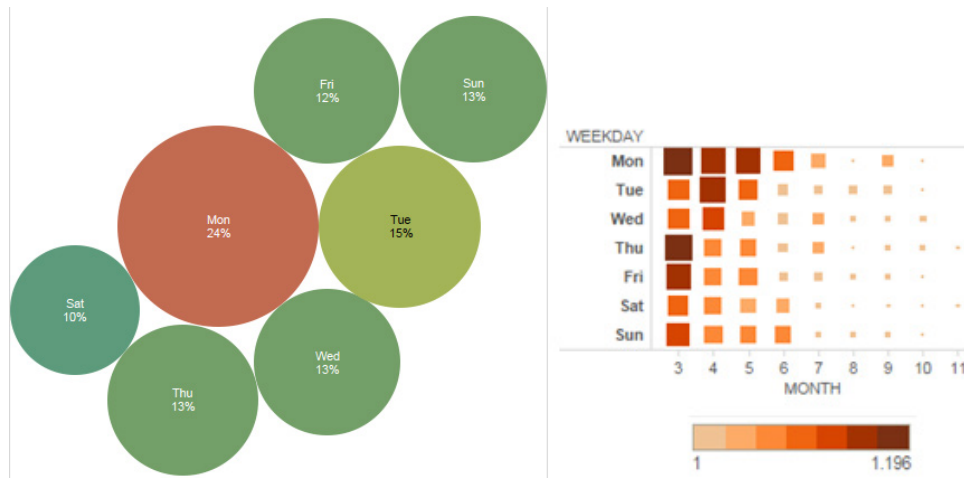


Figure 3: Totals actions per weekdays and per weekdays x months in the period

In terms of time of day, Figure 4 illustrates the total number of actions for each day in the period: first as totals and secondly as stacked totals for each weekday. The students' behavior indicates that they gradually increase their number of actions between 7 to 10 pm, decreasing their activity after that time. There is also a significant increase in lunch time or between 7-10 am, being coincidental with the general profile of our students (employed people) who are more active in less demanding working hours.

Analyzing the Quality of Students Interaction in a distance learning

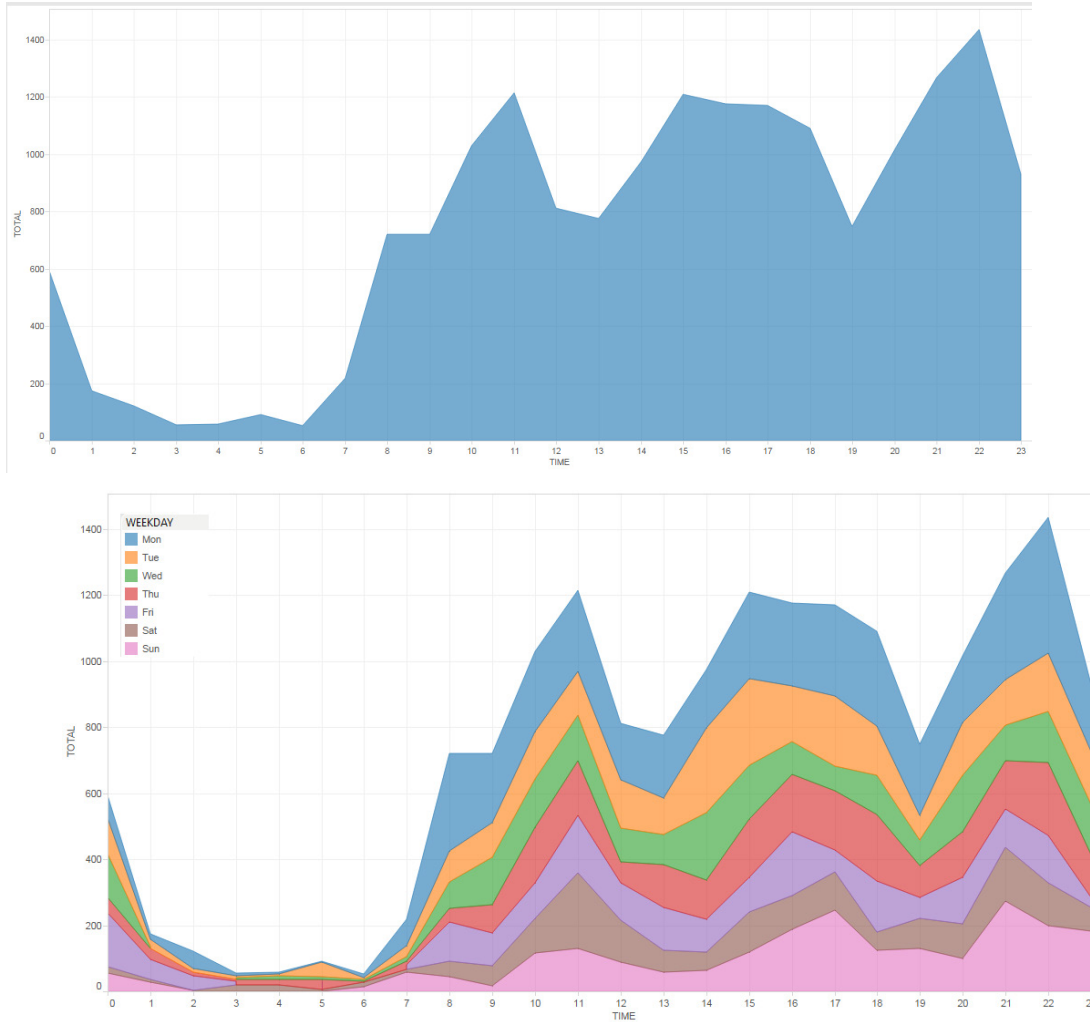


Figure 4: Totals actions per time of day in the period and per weekday

The decrease in participation, and thus interaction of the students, increased significantly with time. Although some of the graphs shown above already give a hint of this, taking a close look at the most repeated action (Course view), it is clear that there is a decrease in the rate of participation of the students. In March, 61% of the students that had accessed the discipline at least once were considered active. In July, this figure drops to only 26%. Figures 5 and 6 give a better view of this happening through highlight tables.

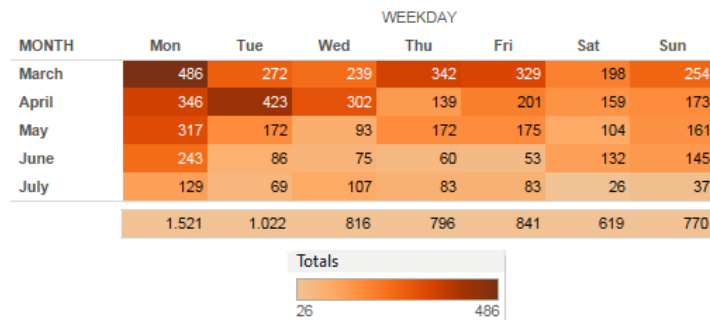


Figure 5: Course view action execution during the period

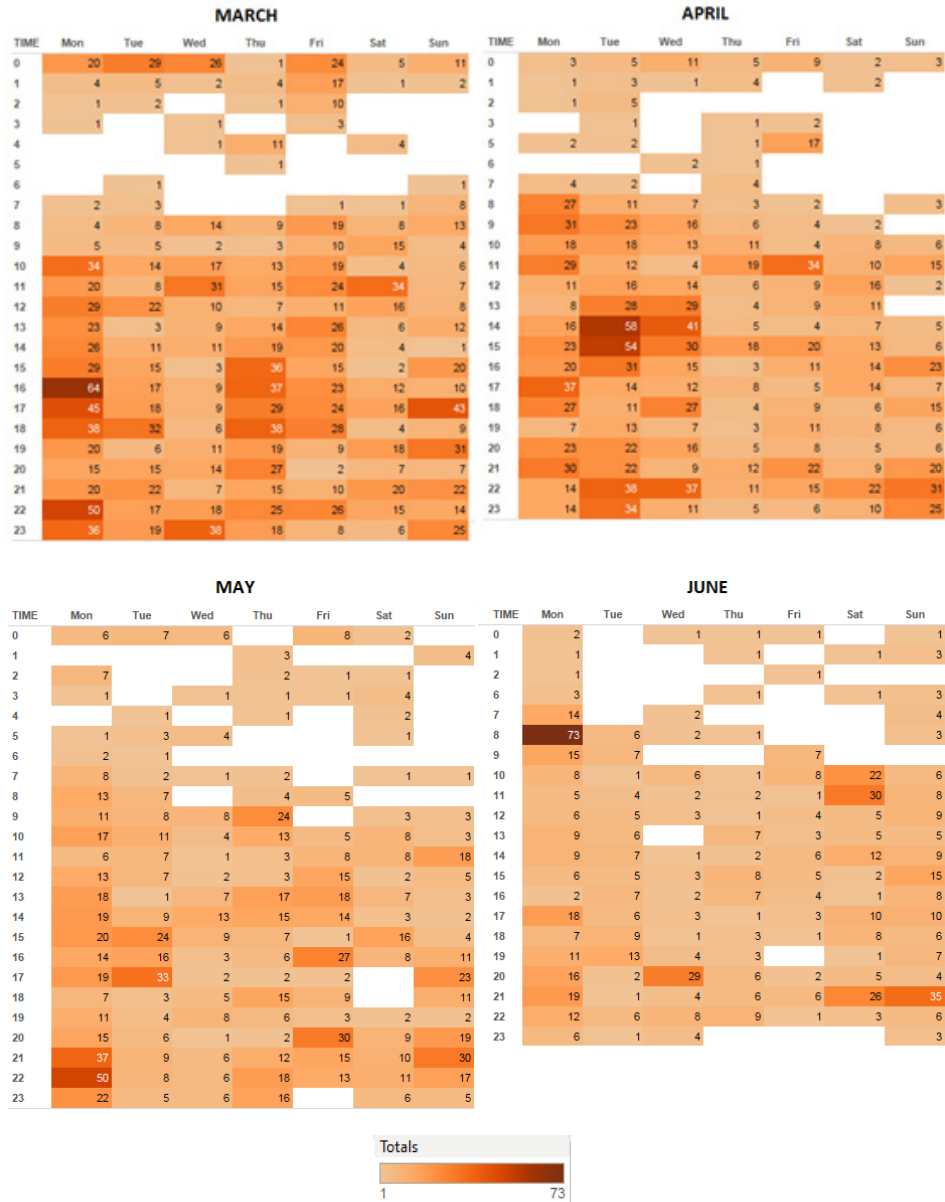


Figure 6: Course view action execution per weekday and time of day during the period

Findings

Based on the results, the drop out of students is very high, although most of them access the discipline and do some minor interaction at least once – there were students that accessed Book view and Course View only once. The quality of their interaction is very low in terms of using (posting or/and reading) the several forums available for asynchronous communication. For instance, only a set of 30 students actually accessed a forum. The highest number of times that the action forum view was executed by a single student (from this set) was 16, where 3 accesses was the average result to the set as a whole. This reveals a possible failure of the main pillar on which the e-learning model relies. In other words, in the UAb e-learning model (Pereira et al, 2007) the process of teaching-learning is continuous, taking place independently of the time and place where the students and teachers are physically. This means that both students and teachers may partici-

pate in the disciplines and learning units from where they are, at any time, and there is no requirement for being online at the same time. This is a fundamental aspect in terms of flexibility – the asynchronous communication, which is provided by the discussion forums.

Another interesting outcome is that only 38 students have accessed the discipline, at least 182 times (in different dates). If we assume that they spent 1 hour per day on the platform, this reveals that these students were more interested in navigating and participating actively than those with lower figures (the other 77). If we assume this rate of usage, this figure also indicates that these 38 students exceeded the expected 156 hour of workload only while interacting with the platform (the highest number of times was 529). In fact, most of this group of students were those that tried to be approved on this object-oriented discipline. Figure 7 illustrates the relation between final grades of the approved students and their percentage of actions with high relevancy (see Table 4) to the programming learning process. We can see a range of behavior amongst the approved students. Two of the students with the highest score had the lowest level of actions. Both of them had less than 5% of the total number of actions.

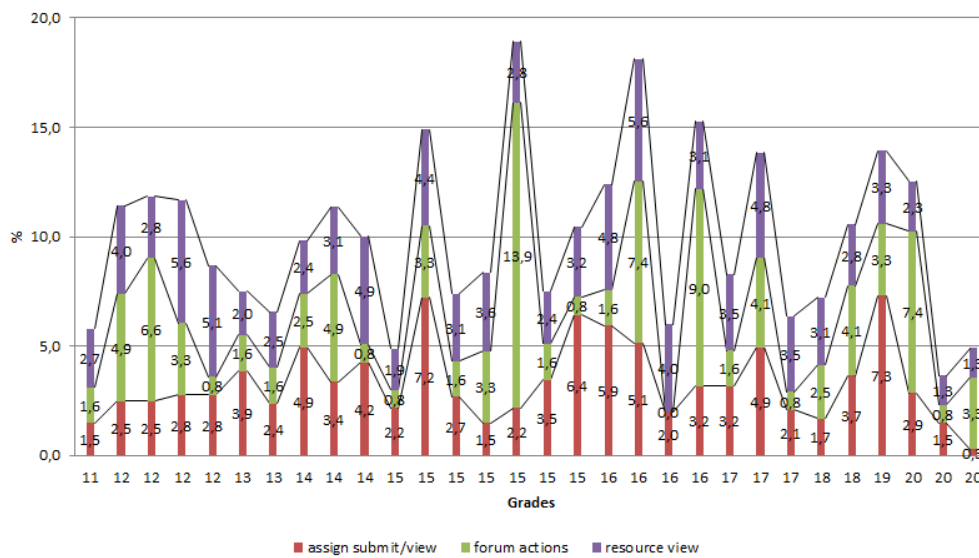


Figure 7: Percentage of the total actions per final grades of approved students

An additional significant outcome is that a divergence occurred between the grades and the number of total forum actions. The interaction through the asynchronous forums is another key premise behind the e-learning model. We can notice clearly in the chart in Figure 8 that the highest grades were achieved by students with low numbers of forum actions.

The students in this discipline presented an irregular profile of usage during the week, with Monday being the most accessed day. Most of the interactions occurred by night, around 10 p.m.

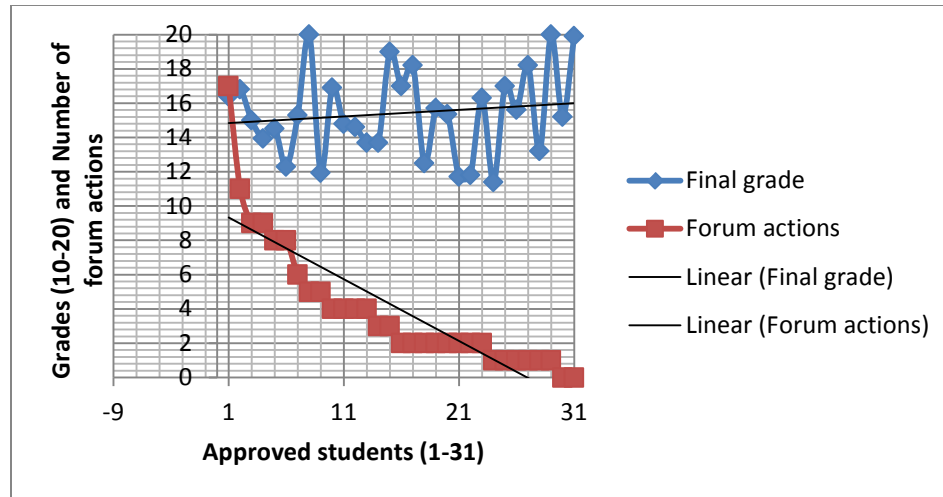


Figure 8: Final grades x total of forum actions per approved student

Conclusions and Further Work

In general, based on the results, we can say that the quality of the students' interaction is lower than would be expected to be ideal. This indicates that possibly the e-learning model is not well tailored to work with disciplines such as those where programming languages are taught, and other didactic and pedagogic strategies have to be added in these cases. Although this is important, we should also take into account that online education has raised some concerns regarding high student attrition rates in online courses and programs. In general, online courses have higher dropout rates compared to face-to-face courses (Bowers & Kumar, 2015).

One very important conclusion is the significant underuse of the forums by all of the students and the lack of evidence that their underuse leads to a failure to be approved. Some of the shortcomings may be caused by the fact that the majority of the students are mature and regular workers, thus have limited time (because of family and work responsibilities) to devote to study. Another pertinent factor is the specificity that learning programming demands. According to Nandi, Hamilton, and Harland (2012), the content for the programming course needs to be more prescriptive, algorithmic, and more narrowly focused; hence, opportunities for direct discussion and asking questions are limited. In their study, often a single solution posted by a student in answer to a problem raised by another student was sufficient, and the instructor ended the discussion at that point. The same situation is applied to assignments. Once the solution is presented to the students, there is little discussion and alternative suggestions for solutions provided. This may explain the low activity in the forums.

The high activity observed during late hours and on Monday is possibly due the mature age of the students and because Monday is always the day when a new topic is made available to the students, generating more accesses.

Taking into account only the 31 students that were approved and actions with a high level of relevancy to the programming learning process, we can identify another interesting outcome, which is that the resource view and assign submit actions had a positive correlation towards the programming learning process. In Table 5 the quality of interaction is evaluated against the minimum number of actions acceptable per student (Table 4) within this discipline and the average of those 31 approved students.

Table 5: Approved students quality of interaction

<i>Average number of actions per approved student</i>	<i>Minimum expected per student</i>	<i>Quality of interaction</i>	<i>Action</i>	<i>Relevance to the programming learning process</i>
4	2	Good	Assign submit	High
25	14	Very good	Resource view	High
4	10	Very weak	Forum actions	High

The introduction of regular synchronous meetings, or the development of multimedia content to teach interactively how to program, may be possible future e-learning tools to explore (e.g., using virtual and augmented reality). Another useful thing would be to run a survey to understand better what limitations and drawbacks may cause so many students to drop out or to have a low quality level of interaction in the distance learning environment in the case of OOP. Also future work suggested is to compare these results against those achieved from other programming disciplines that the Computer Science degree offers in UAb.

References

- Ardito, C., Costabile, M. F., De Marsico, M., Lanzilotti, R., Levialdi, S., Roselli, T., & Rossano, V. (2006). An approach to usability evaluation of e-learning applications. *Universal Access in the Information Society*, 4(3), 270-283.
- Bowers, J., & Kumar, P. (2015). Students' perceptions of teaching and social presence: A comparative analysis of face-to-face and online learning environments. *International Journal of Web-Based Learning and Teaching Technologies (IJWLTT)*, 10(1), 27-44.
- Chatti, M. A., Dyckhoff, A. L., Schroeder, U., & Thüs, H. (2012). A reference model for learning analytics. *International Journal of Technology Enhanced Learning*, 4(5-6), 318-331.
- Hogo, M. A. (2010). Evaluation of e-learners behaviour using different fuzzy clustering models: A comparative study. *International Journal of Computer Science and Information Security (IJCSIS)*, 7(2).
- Kahiigi, E., Ekenberg, L., & Hansson, M. (2007). Exploring the e-learning state of art. In *Conference on E-Learning, Academic Conferences Limited* (pp. 349-368).
- Loncar, M., Barrett, N. E., & Liu, G. Z. (2014). Towards the refinement of forum and asynchronous online discussion in educational contexts worldwide: Trends and investigative approaches within a dominant research paradigm. *Computers & Education*, 73, 93-110.
- Murphy, D., Walker, R., & Webb, G. (Eds.). (2013). *Online learning and teaching with technology: case studies, experience and practice*. Routledge.
- Northey, G., Bucic, T., Chylinski, M., & Govind, R. (2015). Increasing student engagement using asynchronous learning. *Journal of Marketing Education*. First published online on June 4. doi: 10.1177/0273475315589814
- Nandi, D., Hamilton, M., & Harland, J. (2012). Evaluating the quality of interaction in asynchronous discussion forums in fully online courses. *Distance Education*, 33(1), 5-30.
- Pereira, A., Mendes, A. Q., Morgado, L., Amante, L., & Bidarra, J. (2008). *Universidade Aberta's pedagogical model for distance education: A university for the future*. ISBN 978-972-674-534-1. Retrieved from: <https://repositorioaberto.uab.pt/handle/10400.2/2388>
- Rovai, A. P., & Barnum, K. T. (2007). On-line course effectiveness: An analysis of student interactions and perceptions of learning. *International Journal of E-Learning & Distance Education*, 18(1), 57-73.

- Shehane, R., & Sherman, S. (2014). Visual teaching model for introducing programming languages. *Journal of Instructional Pedagogy*, 14(March). Retrieved from <http://www.aabri.com/manuscripts/141823.pdf>
- Sriwardiningsih, E., & Siswono, L. (2014). Interaction e-learning website, curriculum material products, motivation and digital literacy influence to satisfaction and the attitude understanding student learning. *IOSR Journal of Business and Management (IOSR-JBM)*, 16(6), 37-41. Retrieved from <http://iosrjournals.org/iosr-jbm/papers/Vol16-issue6/Version-4/F016643741.pdf>
- Slaninová, K., Kocyan, T., Martinovic, J., Drázdilová, P., & Snásel, V. (2012). Dynamic time warping in analysis of student behavioral patterns. In *Proceedings of DATESO, 12th Annual International Workshop on Databases, Texts, Specifications, and Objects*, April 18th to 20th, Zernov, Rovensko pod Troskami, Czech Republic (pp. 49-59). Retrieved from <http://ceur-ws.org/Vol-837/paper16.pdf>
- Vihavainen, A., Paksula, M., & Luukkainen, M. (2011, March). Extreme apprenticeship method in teaching programming for beginners. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (pp. 93-98). ACM.

Biography



Elizabeth Simão Carvalho is an Assistant Professor at the Portuguese Open University, Department of Sciences and Technology, Lisbon Portugal. She holds a Ph.D in Systems Information and Technologies (2008) by the University of Minho. She is a researcher at CIAC— Research Center for Arts and Communication of the University of Algarve. Her research is concerned with the area of computer graphics with special focus on information and scientific visualization of data.