Informatics in Education, 2010, Vol. 9, No. 1, 19–36

Students' Activity Focus in Online Asynchronous Peer Learning Forums

Alessio GASPAR, Sarah LANGEVIN, Naomi BOYER, William ARMITAGE
University of South Florida
3334 Winter Lake Rd, 33803 Lakeland, Florida, USA
e-mail: {alessio,sarah,naomi,armitage}@softice.lakeland.usf.edu

Received: April 2009

Abstract. This qualitative study explores how using Peer Learning Forums (PLF) in an online asynchronous computer programming course can be analyzed to derive information about Student Activity Focus (SAF) for adult Information Technology students. Three instruments are proposed to assist instructors classify questions posted by students on these forums, obtain easy quantitative measures of their SAF, and use it to gain a better understanding of the type of learning barriers they are encountering. In addition, we also look at the difference in SAF between passing and failing students based on their exam performance. The PLF learning activity and the classification instruments are easily adaptable to other disciplines or courses and allow instructors and students alike to gain a better understanding of what kind of participation is helpful in online asynchronous discussion forums.
Keywords: CS-1, introductory programming courses, peer learning forums, online asynchronous course, blackboard forums.

1. Introduction

This section presents the original motivations which led us to work on this study, details the specifics of the pedagogical intervention we used and relates it to existing literature, and establishes the rationale behind its design.

1.1. *Motivation of This Study*

A computing professional will devote a significant portion of their professional life adapting to new technologies in a self-directed manner. While professional certification training and adult learning opportunities are helpful, a substantial amount of this learning will be autonomous, based on written technical documentation and interactions with peers. The capability to learn from such sources is therefore critical and often expected from new hires or students pursuing a graduate education. However, as was stated by Henry Walker in the "Classroom Issues" column "Reading and class work" of the SIGCSE Bulletin (vol. 39:2, p. 13), this skill is seldom fostered, let alone taught, in typical undergraduate computing curricula. This type of "curricular paradox" is an expression of

a curricular-level failure to adhere to what is known, at course-level, as the constructive alignment theory (Biggs, 1999). Simply stated, our learning objectives are not aligned with our evaluations, thus providing students with opportunities to be "successful" undergraduate computing students without acquiring the skills required in the workforce. In practice, a selection dynamic often replaces an educational one; instead of modifying our curricula so as to develop the target skills in our students, only those who are already gifted with such capabilities will become successful professionals or graduate students. This paper presents observations resulting from our first attempts at fostering the above-mentioned skills by using Peer Learning Forums (PLF).

1.2. *Pedagogical Intervention – Peer Learning Forums*

PLF were originally designed as a way to help IT students develop the capability of learning from written technical documents and interaction with peers, two regular professional practices in this discipline. This learning activity has been already used in various courses of the Information Technology (IT) curriculum at the University of South Florida Polytechnic (USFP). Most of its deployments were focused on online asynchronous offerings but it has also been tested as a learning activity between sessions in traditional face-to-face delivery methods. This study focuses on data collected during an online asynchronous offering of "IT Program Design" (see Section 2.2) over 10 weeks during summer 2008. Each Monday, a new online module was released. These modules were comprised of reading assignments, a graded quiz to be taken between one to two weeks following the release of the module and a series of practice exercises along with step-by-step video solutions. These so-called "apprenticeship exercises" were based on the benefits of cognitive apprenticeship (Gaspar and Langevin, 2007) and "live coding" in programming pedagogy (Kolling *et al.*, 2003). With each module, a new PLF was released as a Blackboard discussion forum. Students had two days to read the material and post questions to that module's forum. Their participation in this phase of the activity was graded to reward *relevant* questions. One point was awarded for each PLF if the student posted such a question in a timely manner. Students who were confident they had absolutely no questions about the material were instructed to post at least one "challenge question" to invite their peers to think about the most difficult aspects of the chapter. Off topic or not understandable questions were flagged by the moderator to invite their authors to resubmit. During the second phase, students had two days to revisit the forum and attempt to respond to their peers' questions with the help of the material presented in the reading assignments. The pedagogical logic of this activity is that, after reading the material, students should be able to address any question, assuming it was different than their own, posed by others. If not, they would have to review the material in an active manner by seeking responses to the specific questions they were unable to answer, rather than re-reading it superficially. This approach is meant to motivate students to take a more active stance when learning from readings, by providing them with contextualization; students revisit the material in search of answers to specific problems – their peers'. The following Monday, the instructor wrapped up the PLF activity by posting responses to each question and commenting on the various responses that were proposed. In previous

offerings of this programming course, we used face-to-face or online synchronous (via Elluminate, refer to `http://elluminate.com/` for more information) class meetings, during which the instructor would "lecture" on the various misconceptions which surfaced in the week's forum activities. In the specific offering during which data was collected for this study, such wrap ups were conducted solely by posting on the forums and, when relevant, by releasing short video lectures or demonstrations of the most difficult misconceptions expressed through students' posts.

1.3. *Rationale for Designing Peer Learning Forums*

From a pedagogical perspective, our attempt at developing the above-mentioned skills in our students boiled down to motivating an *active* reading of the material by inviting them to question the knowledge presented to them as they attempt to answer their peers' questions. The content relied on the use of a textbook for technical readings (Deitel and Deitel, 2006), thus keeping the content accessible to students. It also relied on guiding them through both the material and the process of reading it by enabling interactions with their peers and instructor. The intent of our pedagogical intervention being stated, let us discuss how the specifics of its implementation were influenced by various educational strategies. First, online discussion forums are essential to fostering a "learning community" in settings which are not as conducive to social interaction as face-to-face ones (Picciano, 2002). Second, having students attempt to respond to each others' questions introduces a *peer learning* dynamic meant to bring benefits associated with *social constructivism* (Vygotsky, 1978) to online settings. Our hypothesis is that the mean difference in proficiency between any two students, with respect to the material being taught, will be less than the mean difference between any one student and the instructor. In such a situation, the difficulty of students' questions should (1) naturally be more representative of the specific learning barriers of the particular student population and (2) be more likely to be located in other students' zone of proximal development (ZPD) (Gaspar *et al.*, 2008). The ZPD refers to the gap between current student knowledge and the next level of understanding that can be traversed through teaching strategies that attend to an individual's learning needs This suggests that peer learning dynamics are a natural implementation of social constructivism and provide information which can be used to tailor the teaching to the learning barriers encountered by a specific student population, as opposed to basing our pedagogies solely on personal experience or results published about radically different student populations. The main limitation of this working hypothesis is that student populations, in any given offering, are unlikely to be homogeneous. In our experience of teaching small classes with less than 25 students, this doesn't constitute a major obstacle. Students are divided into three "rough" categories:

1. *Students with preliminary programming experience or outstanding skills who are generally taking the lead in answering their peers' questions.* These students serve as models for generating the kind of interaction expected in the forums and drive other students to improve their peer learning communication skills. Such students often benefit from bringing up topics they mostly understood but in which the instructor can often add some extra clarification on the more ambiguous scenarios.

2. *Active students who are new to the material and working on overcoming significant learning barriers.* These are generally active participants in the learning activities and forum discussions, generating genuine questions on the specific topics they are struggling with at that moment.
3. *Inactive students who are new to the material and encounter significant learning barriers and/or time management issues.* These students keep their participation to a minimal level, oftentimes posting questions irrelevant to the learning barriers they are struggling with just for the sake of getting participation rewards. For such students, the peer learning interaction is rarely successful, mostly due to a lack of available time or motivation to commit to the regular learning routine.

An ideal student population would have at least a few students in the first category and as few as possible in the last. Our experience has been that, in small offerings, categories 2 and 3 are generally balanced with a few students in category 1. Although this setting seems to be sufficient to leverage the benefits of social constructivism, it will have to be further studied in larger settings.

1.4. *Paper Organization*

The remainder of this paper will be organized as follows; Section 2 will provide the necessary context to enable the reader to put our observations and findings in an appropriate perspective. We will provide descriptions of the USFP IT program, its online components, the specific course used for this study, as well as the students who were enrolled in it. Section 3 will discuss the focus which guided our exploration of how PLF were used by our students and how it led us to design three specific instruments to gather relevant information from available forum archives. Section 4 will present the data collected using these instruments from two complementary perspectives: the topic of the question posts and their cognitive level with respect to a version of the revised Bloom Taxonomy for the cognitive domain, adapted to the discipline-specific elements of our offerings. We then provide a discussion on these results aimed at identifying the differences in the measures obtained on both passing students' question posts and failing students' ones. We conclude in Section 5 with a discussion of our main findings and future work.

2. Study Parameters

Studies such as this one require "thick descriptions" to enable the reader to interpret the observations and findings in the larger perspective of the context in which they occurred. To this end, this section will present information about the USFP Information Technology online program, its students, the specific offering where data was collected and the pedagogical approaches which were leveraged.

2.1. *The USFP IT Online Program*

USFP has offered an undergraduate degree in Information Technology (IT) since 2002. Students take mandatory core computing courses covering material shared with other cur-

ricula such as computer science. The emphasis, in line with the polytechnic philosophy, is on application and hands-on learning. The department has pioneered laboratory infrastructure, funded by National Science Foundation grants from the Course Curriculum and Laboratories Improvement (CCLI) program, allowing hands-on learning to become an integral component of courses where it is often problematic, e.g., operating systems, networking, online courses. Students also participate in an IT senior project capstone experience and, optionally, an IT practicum allowing them to tackle a larger project and gain experience working in teams. Five electives allow them to acquire a specialization of their choice before graduation. Electives require department approval, but are not restricted to IT courses. The program became available in its entirety online in 2007 as part of a campus-wide initiative to research, develop and apply best practices in online education. As part of this effort, IT faculty members have been involved in inter-disciplinary research groups.

2.2. *IT Program Design*

IT Program Design is a junior-level programming course for students who have completed an introduction to programming. It uses the C language in a Linux environment to further develop students' programming discipline and their understanding of lower-level programming concepts (e.g., stack, pointers, and explicit memory allocation) in preparation for system-oriented upper level courses. During summer 2008, the course was offered in an asynchronous online format over ten weeks. Most interactions were through emails and Blackboard forums. Office hours and appointments were available online synchronously (via Elluminate) and face-to-face. Three exams were administrated, each at the end of a course segment. Each exam was cumulative and consisted of a majority of programming assignments with only a few questions not involving writing programs (e.g., fill in the blank in existing programs, multiple choice questions...). The overall grade for each student was computed from the final exam (25%), participation in peer learning forums (15%), and best of the two remaining exams (60%). This policy was used to address technical difficulties, which surrounded the second exam. The practice and exams were hosted on a Linux server which provided students with remote access to virtual desktops. During exams, many students complained about networking difficulties making the remote system less responsive than it should have been. Given the difficulty assessing the nature of these networking transient issues, it was decided to assign students the best of the two exam grades while an alternative was researched for the final exam. All exams were administrated using Blackboard. Each required students to write several programs from scratch on our Linux server. The program specifications were released when the student chose to take that specific part of the exam, which was available for about a week. From that moment, a timer indicated the remaining time before submission, generally one to two hours. Each exam was comprised of several parts requiring students to either write a program or debug an existing one.

From the perspective of the pedagogy of content, a text from Deitel Associates was used (Deitel and Deitel, 2006). The first week was devoted to allowing students to become acquainted with class procedures as well as to install and test the necessary software. The remainder of the course was structured as three three-week segments, each

followed by an exam. The first segment revisited fundamental programming principles while easing the transition to a new language by introducing data types, compiler errors and the development environment (module 101), conditional statements (module 102) and iterative statements (module 103). The second delved deeper into the specifics of the chosen programming language through arrays (module 201), pointers and memory management (module 202) and strings (module 203). The third prepared students for the IT Data Structures course by introducing user-defined data types (module 301), application of pointers to self-referential data structures (module 302) and files (module 303). Learning outcomes were organized into four groups as described by L1, L2, L3 and L4 in Section 3.

2.3. *Surveyed Populations*

Our population consisted of students enrolled in an Information Technology (IT) program, with 15 students enrolled in the online asynchronous IT Program Design offering during summer 2008. This sample was comprised of thirteen males and two females who each took at least one previous programming course (one student reported programming in a professional setting). Our sample had one student was in the 18–20 range, ten in the 21–30 range, one in the 31–40 range, one in the 41–50 range and two over 51 years old. Their course load is summarized in Table 1. In this sample, two students were part time employees, five were full time and four were not employed. Most of these students have transitioned from a community college into the university following the $2 + 2$ model established within the State of Florida. Others may be more mature adults returning for further education beyond technician type credentials. Most students are therefore non-traditional, as seen in age groups, full time employment, etc. Our findings are therefore applicable only to similar student populations. It has been traditional for our department to have a majority of its online students matching the "overcommitted adult learners" profile – registering for multiple online courses while employed full time and shouldering other responsibilities such as a family.

3. Study Focus and Methods

Given the specific context of the presented study, a further description of the data collection and analysis methods follow. Peer Learning Forums served as both a data collection

Table 1
Sample's characteristics summary

	Number of Courses				
	1	2	3	4	5
#students	3	4	2	4	2

tool and the studied learning intervention. The PLF were utilized to capture relevant information about how students interact and gain knowledge from one another as well as independently process the presented content.

3.1. *Methods*

Understanding the potential of this study requires seeing it through appropriate methodological lenses. We did not engage in a quantitative study whose findings could be generalized to other junior-level programming courses nationwide. In fact, very few statistically significant quantitative studies should boast such claims, as the specifics of the student populations being studied often dictate the reliability of the results, e.g., a commuter campus vs. a residential university, self-selected students vs. entry exams. Instead, our study is based on case study methodology which allows us to focus on a specific, well-described if small population and leverages an inductive approach to *generate hypotheses*. The validity of our observations is assessed by use of a multitude of instruments, each providing a complementary perspective on the observed phenomena, which are then triangulated to validate one another. By doing so, our study adhered to well-established qualitative case study methodologies (Merriam, 1998; Creswell, 1998; Creswell, 1994).

3.2. *What are We Exploring?*

This study is meant to help us gain a better understanding of how students decided to use the peer learning forums. While participation was mandatory and graded, students were rewarded for simply posting a single relevant question and a single relevant response within the specified timeframes. The relevance of their posts was measured in a pass/fail manner; the only students not receiving full credit for timely posts were those posting syllabus-related questions or those whose posts weren't comprehensible. This resulted in wide variety in the quality, quantity and focus of the questions and responses posted. *Identifying the differences and variety in PLF usage patterns between students who successfully passed the course and those who failed it is the primary focus of this study.* Such patterns have value in helping forewarn future students of various learner strategies and their efficiency but also in helping instructors identify early on, students who are making inefficient use of the PLF.

To this end, we opted to leverage the availability of blackboard forum archives to examine the nature of student posts. For this specific study, we limited ourselves to analyzing the questions posted in the PLF. The overarching idea is to extract, from the list of questions posted over the entire semester, a picture of the *Students' Activity Focus – SAF, defined as the students learning choices and focus throughout the learning process*. Examples of SAF would be: whether or not a student reads the chapters and practices the exercises, their struggles within the learning process, etc. By examining to what aspects of the material students devote their posts, we hope to gain insights on the nature of the learning barriers that are most troublesome to them.

It is important to stress that, in the authors' opinion, the ready availability of such artifacts is one of the major overlooked benefits of online delivery methods. It is not

practical, in a face-to-face offering, to record, quantify and later analyze every aspect of student participation taking place during the teaching/learning transaction. In an online asynchronous setting, such information is implicitly and effortlessly archived. Given the extensive documentation, *Students Activity Focus (SAF)* becomes an interesting construct to extract and study. In order to extract SAF indicators from student posts, we developed three distinct instruments, each allowing us to observe a specific aspect of student participation in the PLF. The following sub-section will describe these instruments.

3.3. *SAF/LO (Student Activity Focus Relative to Learning Outcomes)*

The first instrument is intended to allow us to categorize student question posts in the weekly peer learning forums based on their relevance with respect to the course's learning outcomes. Following in the footsteps of previous studies (Gaspar *et al.*, 2008), we therefore tracked the questions posted for each module and classified them as follows:

- *L0 – Irrelevant Posts*. This category includes questions which were not rewarded for participation, e.g., syllabus-related inquiries, incomprehensible questions, out of topic questions.
- *L1 – Programming concepts*. This category includes questions regarding fundamental programming principles. Most of these principles had already been introduced to students during a prior introductory programming course. They encompass the notion of variables, usage of loop and conditional constructs and usage of functions. Each of these topics is reviewed during the first few weeks of the course but students are expected to focus on the difficulties involved by re-learning these in a new programming language rather than to discover them for the first time.
- *L2 – Program Design*. This category includes questions regarding algorithmic problems. Students, while they are expected to have already been exposed to the main programming principles during their introductory programming courses, often still encounter difficulties developing, from scratch, solutions to trivial algorithmic problems. Elementary software engineering principles and iterative program refinement and testing are generally introduced at this step of their learning. Questions on these topics as well are classified L2.
- *L3 – Implementation*. This category includes questions focused on issues with implementing specific designs. These questions are usually about syntactical rules, compiler options or usage.
- *L4 – Troubleshooting*. This last category includes questions aimed at helping students validate or debug a specific implementation.

We refer to this classification of student questions as SAF/LO insofar that it allows the quantification of Student Activity Focus relative to Learning Outcomes. This indicator offers a valid approximation only under the assumption that students posted questions related to the specific learning barriers they encountered. While this is the intent of the PLF, and how it was introduced to students in the syllabus, unexpected posting behaviors interfered with this somewhat idealistic picture (e.g., posting irrelevant questions merely for credit).

3.4. *SAF/Sources (Student Activity Focus Relative to Information Sources)*

In addition to mapping student questions with respect to learning objective, we also classified them according to their relevance with respect to the sources of information the student consulted before asking them. More specifically, we distinguished three sources of information, to which all student questions would map.

- *S1* – *Reading Assignments and videos*. Each module provides reading assignments from the textbook and mini-lecture videos. The latter are released either along with the reading assignments or during the wrap up phases. Students get credit for questions posted before the deadline.
- *S2* – *Apprenticeship Exercises*. Other students post questions which emerge when they are either attempting the apprenticeship exercises or when they are watching step-by-step solution videos.
- *S3* – *Non-assigned material*. Other questions were based on external sources of information. Some students asked for help understanding information gathered from web sites, references, other courses' web sites or even code examples. S3 also reflected the questions posted by students attempting to overcome difficulties introduced in non-assigned exercises on which they worked in order to gain more practice.

This SAF/Sources instrument therefore allows us to estimate Student Activity Focus in terms of the type of learning material with which they experienced difficulty. Like the preceding Instrument, measures obtained by SAF/Sources are only as valid as the assumption that student questions are genuine.

3.5. *SAF/Bloom (Student Activity Focus Relative to Bloom's Taxonomy)*

We also categorized questions posted to the forums based on what each post told us about the level at which students are thinking about the material. This approach is quite different from the two previous instruments. A survey of the computing education literature revealed that Bloom's taxonomy of the cognitive domain (Bloom, 1956) has been used to evaluate novice programmers (Lister and Leaney, 2003), identify programming learning outcomes in general (Oliver *et al.*, 2004) or with respect to specific programming activities (e.g., debugging (Xu and Rajlich, 2004), program comprehension (Buckley, 2003)). Based on critiques and reviews of the application of Bloom's taxonomy to computing (Fuller *et al.*, 2007), we used a revised version (Anderson *et al.*, 2001) which had already been adapted to programming (Thompson *et al.*, 2008). We did not use the sub-categories of this taxonomy but focused instead on the broader categories and how to adapt them to the task at hand. Instead of describing learning outcomes at the cognitive level, we needed to redesign them to describe learning barriers encountered by students as revealed in the formulation of the questions they posted in the PLF. This led us to adapt categories of Cognitive Activity (CA) from Anderson *et al.* (2001) and Thompson *et al.* (2008) as described in Table 2. The highest level of the revised taxonomy ("create") is not mentioned, as this course did not focus on developing student problem solving skills nor enabling

Table 2

Revised Bloom's taxonomy adapted to classify PLF questions as SAF/Bloom

Taxonomy	"The question indicates that the student . . ."	Sample questions
B1 Remember	. . . lacks *knowledge* of material (facts, concepts, processes) explicitly covered in the lecture or reading assignment. . . .requests help in a generic manner without providing indications that the student has an idea of what they are having difficulty with. . . .requests opinions on some hypothetic scenario which can be trivially checked by referring to the reading material or implementing a tiny program.	*"Can you re-explain Section 3.2?"* *"What happens if I run the following program?"* *"Do double and float variables both hold floating point numbers?"*
B2 Understand	. . . identifies specific issues with a portion of the material and requests complementary *explanations* about these. . . . has a non-trivial hypothesis to validate regarding his/her understanding of the concepts and processes described in the material. . . . has a problem *differentiating* two constructs but the question phrasing indicates that this learning barrier is due to a *misunderstanding* of each individual construct (otherwise, see B4). . . . needs complementary examples illustrating the material or the application of concepts/constructs to practical scenarios.	*"In Section 3, I don't understand why x has to be negative"* *"If I don't initialize my loop counter variable, is it set to 0 by default?"* *"What are post/pre increment operators used for? Are they necessary?"* *"I know XXX works but why?"*
B3 Apply	. . . needs help in applying a known concept (remember) to a (potentially unfamiliar instance of a familiar task (Thompson *et al.*, 2008; p. 158). . . . realizes a misunderstanding of a process he/she *remembers* while *applying* it to a given scenario. . . . needs help better understanding better the task, the process or the alternatives. . . . understands the idea and its application;, wants to verify a hypothesis about alternative applications.	*"I understand example 3 which reads 10 values from the user using a for loop but how do I make it read until the user enters -1?"* *"When I run example 3, it outputs 23 but it should print 99 instead if I understand Section 6 correctly."* *"What would be a practical use of a linked list?"* *"Can xxx be also used to yyy?"*
B4 Analyze	. . . thinks in terms of a task when considering tools/processes to apply. . . . doesn't ask for straightforward application examples, but rather for examples which help *differentiating* between which tools to use for a given task. . . . exhibits capability to identify several applicable solutions but needs help *differentiating* them. . . . exhibits knowledge of both task and alternatives while needing help with defining standards or judgment criteria in order to *differentiate* them. . . . tries to gain an understanding of how things work "under the hood".	*"Is there another way to do TASK X rather than using XXX?"* *"How does the language know where to return when a function is done running?"* *"When would you use a for loop rather than a while loop?"* *"Are #define constants actually typed?"*

(To be continued)

Table 2 (continuation)

Taxonomy	"The question indicates that the student . . ."	Sample questions
B5 Evaluate	. . . exhibits knowledge of both alternatives being considered and judgment/evaluation standards/criteria knowledge to be applied. However, student needs help in their use in evaluating alternatives by them. . . .fails to connect the erroneous output of a program with the area of the source which might be responsible for it (blind modifications and tests (Gaspar and Langevin, 2007)). . . . fails to be able to generate tests to isolate specific bugs or validate the programs. . . . has design-level issues with a program. Similar to B3's first example but not as straightforward (direct application of a known pattern).	*"My program sorts the 3 variables XYZ by comparing X and Y then Y and Z, do I need to compare X to Z as well?"*

them to design algorithmic solutions to arbitrary problems. These are tackled in the next course, IT Data Structures; IT Program Design prepares students at the task level (e.g., implementing, understanding and troubleshooting a reasonably well specified solution to a problem). We refer the reader to previous work by Lister and Leaney (2003) for a more detailed discussion of the distinction between programming tasks and problem solving.

4. Measures and Observations

4.1. *Data Analysis and Results*

Reports on data collected from the Peer Learning Forum usage in the IT Program Design online asynchronous offering during summer 2008 follow. Data are presented from two perspectives: (a) SAF/LO and SAF/Sources were used to categorize question posts based on their topics, and (b) SAF/Bloom were used to categorize questions based on the cognitive level demonstrated. Once each post was classified according to the three instruments, we grouped students in two sub-populations. The four (out of ten students who didn't drop the offering) *passing* students were defined as being students whose normalized performance, based solely on exams, exceeded 70%. The six (out of ten) *failing* students were the remainder of our population. We established this distinction (basing performance solely on exams) in order to not let participation rewards alter our perception of students' capability to reach the learning outcomes. Participating in rewarded learning activities is often driven by students' willingness to comply with the course's syllabus. It is not an indicator of performance in terms of understanding the material. In our observations, we observed that some excellent students opted out of participating in PLF as soon as they became confident they had already reached a passing grade. Other students participated regularly in the PLF and received full credit, but only submitted a minimal number of questions, which were not focused on learning barriers they encountered. For these reasons, participation reward points would have only made it more difficult to establish students' objective performance.

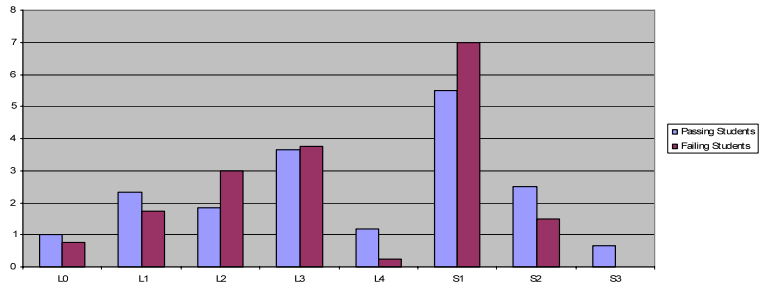

Fig. 1. SAF/LO and SAF/Sources – average numbers of posts per students in each category.

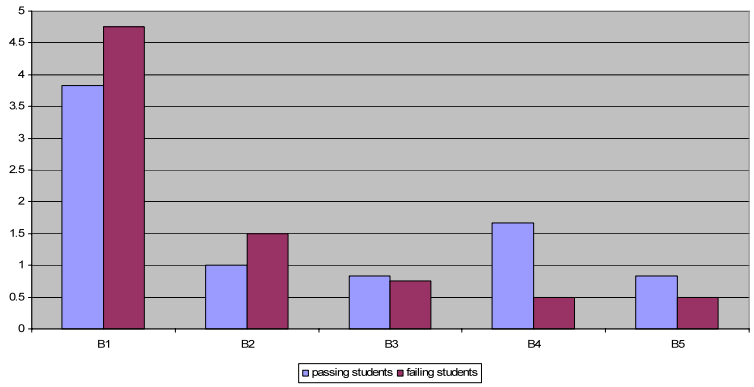

Fig. 2. SAF/Bloom – average numbers of posts per student in each category.

Fig. 1 summarizes the measures based on question topics, using both SAF/LO and SAF/Sources instruments. For each linkage to course outcomes, e.g., L0, L1, L2, L3 or L4 category, or sources of information, e.g., S1, S2 or S3 category, we plotted the average number of questions posted by "passing" and "failing" students.

Fig. 2 summarizes our measures based on Bloom levels of questions. Again, we separated the count of questions for each B1 to B5 category for passing and failing students.

4.2. *Analysis – Irrelevant Factors*

Figs. 1 and 2 reveal that several of our categories were irrelevant in pinpointing differences in Student Activity Focus between passing and failing students. The irrelevant post category L0 is, not surprisingly, one of these. Both types of students (*passing* and *failing*) occasionally posted questions which were not directly relevant to PLF expectations. Generally, students posted such questions (instead of relevant ones) toward the beginning of the semester and, after receiving feedback, posted irrelevant questions in addition to those for which they received participation points. Students were indeed encouraged to ask about just anything, but told that they would get participation points for only those questions relevant to the PLF.

L3, the syntax-related content category, did not distinguish passing from failing students as clearly as we anticipated. We expected that students with good performance on exams would assimilate the syntactical aspects of a new language without many problems. It turned out that both passing and failing students had about the same number of questions about syntax and implementation aspects. In retrospect, this makes sense as many of the concepts studied in this offering were already familiar to students from their first introductory programming course. The most notable difficulties were rooted in the learning of a new syntax and the first discovery of lower-level concepts, such as activation records and pointers, which would most likely lead them to ask questions categorized as L3 (implementation). It is therefore cautioned that the focus of our students on syntax-related difficulties should not be generalized to just any second programming language but might be specific to the importance of implementation-level details and syntax in this particular programming language. The question of whether syntax-related questions are more common for passing or failing students in second programming offerings using a higher level language is still open.

From our adapted Bloom taxonomy perspective, questions related to the "apply" B3 level do not seem to distinguish between passing and failing students and are roughly equivalent in number for both graoups. This suggests that students encounter comparable difficulties in leveraging their newly acquired knowledge.

4.3. *Analysis – On What did Passing Students Focus?*

We now take a look at categories in which the number of questions from passing students exceeded the number of questions from failing students. As previously stated, under the assumption that student questions are genuine and meant to help them overcome learning barriers with which they are struggling, this illustrates on what specific difficulties each group of students was working.

The first such category is L1, questions about programming fundamentals. While the difference is not as salient as the other categories we will discuss below, it indicates that passing students tended to ask more questions about core programming concepts. Since passing students were asking as many syntax-related questions as failing students, this might simply indicate that, *in addition* to learning lower-level details, they also took the time to reflect and ask questions about these higher-level principles. It is interesting that this took place in a second programming course, where the fundamentals of programming were revisited with a lower-level language. This might be interpreted as a need, even for performing students, to be exposed to these principles over the course of at least two semesters in order to be able to integrate and actively reflect upon them. This might indicate that taking the time to reflect on the bigger picture is essential to success in programming. Of course, that passing students ask such questions in addition to L3-level ones might reveal a more significant time commitment on their part that allows them to go beyond questions which come to mind when reading the textbook for the first time.

Both L4 and S2 categories are similar in that they indicate a hands-on approach to learning programming on the part of students asking these questions. L4 includes troubleshooting questions and S2 denotes questions based on actual apprenticeship exercises.

In both categories, passing students dominated significantly. This suggests that a good portion of these students' study time was devoted to not only reading the assigned material and comprehending it but also putting into practice this newly acquired knowledge in a hands-on manner. Students were reminded early in the semester that the intent of the course was not to learn *about* programming but to learn *how* to program. To this end, the importance of practice was stressed repetitively but, for lack of available study time, not all students were able to devote sufficient time to practicing exercises. Most notably, some students would read exercises and immediately watch the video solutions instead of first attempting to solve them. In previous studies, we referred to this as a potential aggravating factor leading to a "loss of intentionality" when programming (Gaspar *et al.*, 2009). Some students tend to tackle a programming course by simply memorizing facts, syntax, working programs, "typical" exam questions and their solutions. This obliterates the creative aspects of programming and prevents students from developing genuine programming skills, which are instead replaced by attempts at pattern-matching problems in a "big book of all solutions" which will provide a listing ready to regurgitate. Also, the mere action of looking for external/additional sources indicate that those students are attempting to synthesize an understanding of the topics at hands from multiple perspectives, rather than simply regurgitating verbatim what the official textbook has to say. This suggests critical thinking and a higher cognitive approach to learning. It is interesting to note that the results in L4 and S2 validate our hypothesis that one of the major learning barriers encountered by failing students is that of understanding that programming is not a matter of memorizing solutions.

S3, questions related to self-assigned exercises or external sources of information, is another category which significantly differentiated the activity focus of passing and failing students. This category consisted only of posts from passing students, suggesting that they engaged in learning activities beyond those which were rewarded by points. The ability of students to engage in self-directed learning is an important quality for their lifelong professional development. Based on our observations, this quality seems prominent in successful students and totally absent in failing ones. We have begun a more detailed investigation of this hypothesis in recent work (Gaspar *et al.*, 2009).

The various perspectives we have discussed converge on the concept that being successful at programming entails more than memorizing solutions, but requires more of a critical thinking process. This is further validated when looking at the question counts in categories B4 and B5, which respectively include questions at the "analyze" and "evaluate" levels in our adapted Bloom taxonomy. Successful students post more questions related to both cognitive levels and, comparatively, fewer questions at the lower Bloom levels such as B1, "remembering", or B2, "understanding". Overall, the profile of a successful student's question posts in peer learning forums entails the presence of many indicators of their activity focus being on higher level cognitive activities. These learning activities, which transcend mere memorization/regurgitation of knowledge, are achieved by hands-on practice and synthesis of multiple sources of information.

4.4. *Analysis – On What did Failing Students Focus?*

The L2 category, which includes design-related questions, is the first in which failing students asked significantly more questions than passing students. This observation has to be considered in the context that the studied course does not introduce many new design principles when compared to the first introductory programming course all students had already taken as a pre-requisite. This suggests that failing students might be trying to overcome learning barriers not tackled during their first course, and therefore lack the study time to confront those specific to this offering. Without a solid grasp of fundamental programming principles, "revisiting" them with a lower level language doesn't result in strengthening students' programming skills but rather makes them realize the existing weaknesses in their understanding of the material studied in their first programming course.

The number of S1 questions, related to the readings and videos, is dominant in failing students compared to passing students. This suggests that the former might spend more time in overcoming difficulties understanding the reading material itself than in (a) putting it into practice through exercises or (b) taking a more critical look at it during a second reading, while answering their peers' questions, or (c) comparing their reading to an alternate reference. Inadequate time dedicated to weekly learning activities could explain this observation. It is also possible that these students need more time on the text itself due, once again, to weaknesses in their understanding of programming principles. The Deitel textbook we are using is meant as an introductory programming text. As such, it takes a very progressive, slow paced approach that does not account for the fact that the course is the second in a programming sequence. This text was chosen to allow students to have time to catch up even though the course requires an introduction to programming as a pre-requisite. Unfortunately, this opportunity does not seem to have been sufficient for students (or sufficiently *utilized* by students) to make up for material they might have forgotten – for example, if they took their introductory programming course a while ago, or its content was not fully grasped originally. The fact that some first-time programming courses put an emphasis on regurgitating solutions on exams or filling the blanks in already functioning programs, rather than design them from scratch, might contribute to this situation.

As indicated in the previous Subsection 4.3, another characteristic of failing students' activity focus, examined from the adapted Bloom taxonomy'6s perspective, is a strong focus on lower-level cognitive engagement with the material. The questions in B1 and B2 suggest that failing students spend more time than passing ones on remembering and understanding their readings. This strengthens the observations made in this sub-section based on other instruments.

5. Discussion

We conclude with the findings resulting from this study, their limitation, and the methodological context within which this study should be interpreted.

5.1. *Major Findings*

The most salient differences in the nature of questions posted by both passing and failing students revealed the impact of the first programming experience on success in "IT Program Design." While syntax did not seem to be a distinguishing focus between passing and failing students, design was. Similarly, passing students seemed to post more questions indicating a higher cognitive level in their struggle with the material. These same students were observed to be self-directed in their learning as opposed to engaging only in learning activities which were mandatory and rewarded by participation points.

Beyond direct observations used to identify passing vs. failing student PLF participation profiles, this study also confirmed that the perspectives offered by the instruments we developed allow an instructor to gain a valid picture of students' activity focus. This can be leveraged to improve teaching practice by providing students with explicit information on what constitutes good or bad practices when leveraging the PLF. This can also be leveraged by establishing profiles of individual student activity foci, and warning early those students who are adopting bad practices.

5.2. *Limitations*

One of the strongest limitations of this study is the inability to separate genuine activity in the PLF from "faked" learning activity. The participation point structure rewards students for posting in time but does not require them to post more than one question/response per PLF. It is also possible for students to read the material superficially and ask a question for the sole purpose of getting participation credit, rather than using the PLF to help them with actual learning barriers.

Our population is also not composed of traditional junior year college students. The majority of our USFP IT students are full time workers and a significant portion are overcommitted by enrolling in too many online offerings. This limits the available time for engaging in meaningful learning activities and, sometimes, the expectation of a less time-consuming, "drive through" education. This context has occasionally led particularly good students to reduce their efforts toward the end of a semester, as they already reached what they considered an adequate grade. This makes quantitative participation in the PLF independent of the student's performance. Also, the relative lack of quality control over students' first programming course experience, despite its impact on their performance in IT Program Design, suggests that a first day evaluation might help students realize the need for catching up in parallel with the first week of this offering.

The number of participants in this study limits the assumptions, generalizations, and implications of the findings. Additional research with a larger sample will either serve to confirm or refute the results of this study.

References

Anderson, L.W., Krathwohl, D.R., Airasian, P.W., Cruikshank, K.A., Mayer, R.E., Pintrich, P.R., Raths, J., Wittrock, M.C. (Eds.) (2001). *A Taxonomy For Learning and Teaching and Assessing*: *A Revision of Bloom's Taxonomy of Educational Objectives*. Addison Wesley Longman.

Biggs, J. (1999). *Teaching for Quality Learning at University*. SRHE and Open University Press, Buckingham.

Bloom, B.S. (1956). *Taxonomy of Educational Objectives*: *The Classification of Educational Goals*. Handbook I. David McKay Co, Inc. New York.

Boyer, N., Langevin, S., Gaspar, A. (2008). Self direction and constructivism in programming education. In: *Proceedings of the ACM Special Interest Group in IT Education Conference*, 16–18 October 2008. Cincinnati, OH.

Buckley, J.E. (2003). Blooms' taxonomy: A framework for assessing programmers' knowledge of software systems. In: *11th IEEE International Workshop on Program Comprehension* (*IWPC'03*), p. 165.

Creswell, J.W. (1994). *Research Design – Qualitative and Quantitative Approaches*. SAGE publications, Thousand Oaks, London, New Delhi.

Creswell, J.W. (1998). *Qualitative Inquiry and Research Design – Choosing Among Five Traditions*. SAGE publications, Thousand Oaks, London, New Delhi.

Deitel, H., Deitel, P. (2006). *C How to Program*. 5/e, Pearson Education, Prentice Hall, Upper Saddle River NJ 07458.

Fuller, U., Johnson, C.G., Ahoniemi, T., Cukierman, D., Hernán-Losada, I., Jackova, J., Lahtinen, E., Lewis, T.L., Thompson, D.M., Riedesel, C., Thompson, E. (2007). Developing a computer science-specific learning taxonomy. *SIGCSE Bull.*, 39(4), 152–170. `http://doi.acm.org/10.1145/1345375.1345438`

Gaspar, A., Langevin, S. (2007). Restoring "Coding with intention" in introductory programming courses. In: *Proceedings of the International Conference of the ACM Special Interest Group in Information Technology Education* (*SIGITE 2007*), Orlando, FL.

Gaspar, A., Langevin, S., Boyer, N. (2008). Redundancy and syntax-late approaches in introductory programming courses. *The Journal of Computing Sciences in Colleges* (in print).

Gaspar, A., Langevin, S., Boyer N., Armitage, W. (2009). Self-perceived and observable self-direction in an online asynchronous programming course using peer learning forums. *Journal of Computer Science Education* (in print).

Kolling, M., Quig, B., Patterson, A., Rosenberg, J. (2003). The BlueJ system and its pedagogy. *Journal of Computer Science Education*, 13(4).

Lister, R., Leaney, J. (2003). Introductory programming, criterion-referencing, and bloom. In: *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education* (*SIGCSE '03*), Reno, Navada, USA. ACM, New York, NY, pp. 143–147.
`http://doi.acm.org/10.1145/611892.611954`

Merriam, S.B. (1998). *Qualitative Research and Case Study Applications in Education* (2nd edition). Jossey-Bass publisher, San Francisco.

Oliver, D., Dobele, T., Greber, M., Roberts, T. (2004). This course has a Bloom Rating of 3.9. In: Lister, R., Young, A. (Eds.), *Proceedings of the Sixth Conference on Australasian Computing Education*, Dunedin, New Zealand, Vol. 30. *ACM International Conference Proceeding Series*, Vol. 57. Australian Computer Society, Darlinghurst, Australia, pp. 227–231.

Picciano, A.G. (2002). Beyond student perceptions: Issues of interaction, presence, and performance in an online course. *Journal of Asynchronous Learning Networks*, 6(1), 21–40.

Thompson, E., Luxton-Reilly, A., Whalley, J.L., Hu, M., Robbins, P. (2008). Bloom's taxonomy for CS assessment. In: Hamilton, S., Hamilton, M. (Eds.), *Proceedings of the Tenth Conference on Australasian Computing Education*, Wollongong, NSW, Australia, Vol. 78. *Conferences in Research and Practice in Information Technology Series*, Vol. 315. Australian Computer Society, Darlinghurst, Australia, pp. 155–161.

Vygotsky, L.S. (1978). *Mind and Society: The Development of Higher Mental Processes*. Harvard University Press, Cambridge, MA.

Wilson, B., Lowry, M. (2000). Constructivist learning on the web. In: *New Directions for Adult and Continuing Education*, Vol. 2000(88). Jossey-Bass, Wiley company, pp. 79–88.

Xu, S., Rajlich, V. (2004). Cognitive process during program debugging. In: *Proceedings of the Third IEEE International Conference on Cognitive Informatics*, pp. 176–182.

A. Gaspar is associate professor in the Information Technology Department at University of South Florida Polytechnic (USF). He received his PhD in computer science in 2000 from the University of Nice Sophia-Antipolis (France). He then worked as visiting professor at the ESSI Polytechnic School (Sophia Antipolis, France) and EIVL Engineering School (University of Tours, Blois, France). He later worked as a postdoctoral researchers in the Computer Science Department of the University of Fribourg (Switzerland) before joining USF in 2002.

S. Langevin is a student at the University of South Florida. She is making researches and writing papers together with Alessio Gaspar, Naomi Boyer, and William D. Armitage at the University of South Florida, Lakeland, USA. She is member of Constructivist Apprenticeship Group. This project revolves around the introduction of so-called Antagonistic Programming Activities as an implementation of constructivist principles in a programming course.

N. Boyer holds a PhD in interdisciplinary education. As the assistant VP of the Office of Extended University at the University of South Florida Polytechnic, she is responsible for teaching and learning technology programs, extended university/continuing education opportunities on the campus, and international partnerships and global engagement projects. Dr. Boyer teaches in the adult education program and maintains research interests in self-directed frameworks in online settings, the impact of technology on educational settings, and the faculty development programs. Dr. Boyer is listed on a number of NSF grants with an emphasis on innovative pedagogy, technology integration, and online material development.

W. Armitage, assistant professor of information technology at the University of South Florida's Polytechnic campus, received his BA from Rhode Island College (1966) and his MS in computer science (1978) and PhD in electrical engineering (1999) from the University of Rhode Island. He has previously held several information technology staff positions in higher education, and has served as a consultant in systems analysis and network design and implementation for many agencies and commercial enterprises. He also founded and managed one of the earliest computer retailing firms (in 1976).

Studentų veikla nukreipta į interneto asinchroninius bendraamžių mokymosi forumus

Alessio GASPAR, Sarah LANGEVIN, Naomi BOYER, William ARMITAGE

Šiame straipsnyje pateikiamas kokybinis tyrimas, kuris atskleidžia, kaip tiriant bendraamžių mokymosi forumus (anglų k. Pear Learning Forums – PLF) internetiniame asinchroninio programavimo kurse, galima gauti informacijos apie studentų veiklos sutelkimą (anglų k. Student Activity Focus – SAF), kuris praverstų vyresniems informacinių technologijų studentams. Siūlomos trys priemonės, padedančios dėstytojams klasifikuoti klausimus, kuriuos forumuose pateikė studentai. Remiantis SAF ir gautais rezultatais siekiama gauti kiekybinių duomenų ir geriau suprasti iškylančių mokymosi kliūčių tipus. Straipsnyje nagrinėjami SAF stiprių ir silpnų mokinių, skirtumai, atsižvelgiama į jų egzaminų rezultatus. PLF mokymosi veikla ir klasifikavimo priemonės yra lengvai pritaikomos kitiems dalykams ar kursams, leidžia dėstytojams ir studentams geriau suprasti naudingiausią bendrininkavimo būdą naudojant internetinius asinchroninius diskusijų forumus.