# DEVELOPING A WEB-BASED ENVIRONMENT FOR LEARNING TO SOLVE PROBLEMS WITH THE LINUX COMMAND LINE: THE PROBLEM-POSING APPROACH

C. Mihci
Department of Computer Education and Instructional Technology, Trakya University Faculty of Education, Turkey
canmihci@trakya.edu.tr

A.F. Satici
Department of Computer Education and Instructional Technology, Marmara University Ataturk Faculty of Education, Turkey
ahmet.satici@marmara.edu.tr

## ABSTRACT
Computers are problem-solving tools at heart. Without an operating system though, they would be rendered useless. Most computer users are unaware of the inherent capabilities of the bare OS and the problems it can solve for them. This comes naturally in an era of commercialized solutions in the form of apps and software packages. However, gaining understanding of the built-in functions of an OS hidden behind the graphical user interface may let students discover new, personally relevant problems. In turn, attempts at formulating and solving these could help students gain complex cognitive skills in IT. This study explores the notion of problem-posing and proposes a Web-based learning environment; where users may create guides and tutorials in addition to posing their own educational problems pertaining to GNU/Linux Command Line use. Problems are stored in a repository and may be rated, reviewed, discussed or solved by peers. A virtual environment running a Linux distribution may be configured by users to accompany a posed problem and help create a complete sandbox reflecting the problem's story and structural elements.

## INTRODUCTION
One of the most important goals of education is to prepare the individual for daily life, which frequently involves situations that require solving problems (Jonassen, 2011, p.17; Krathwolh, 2002; van Merriënboer, 2013). Therefore, it is very important for educational settings to help foster problem-solving skills pertaining to daily and professional lives (Gagne, Briggs, & Wager, 1992). Considering the cognitive skills hierarchy however, problem-solving is usually considered a complex task. This, and the fact that problems come in all sizes and shapes, renders the teaching of how to solve problems a great challenge.

Many researchers have spent effort on classifying educational attainments by grouping learning outcomes contextually. For example, Gagne (1992) groups learning outcomes in a manner that associates simple problems (problems that may be solved by applying a single rule) with the notion of rule-learning, while explaining that the solution of more complex problems is the highest level of intellectual attainment. Another classification is made in Bloom's taxonomy, which displays problem-solving at multiple levels of the top-down hierarchy: whereas the solution of certain problems fall into the application level, others may correspond to higher order thinking skills found in the levels of analysis, synthesis or even evaluation. The taxonomy was revised in the year 2001 (Krathwohl, 2002) and in the revised version, it may be seen once again that problem-solving requires skills ranging from the level of application to creation (Thompson, Luxton-Reilly, Whalley, Hu, & Robbins, 2008).

The reasons for representing problem-solving in multiple levels of a cognitive skills hierarchy may be evident upon inspecting the works of Jonassen on the notion of problem-solving learning (Jonassen, 2011). Jonassen explains that problems encountered in school, or in our daily lives come in all shapes and sizes and thus assigns them into different categories. There are many factors that determine which category a problem falls into. The most important one among these, however, may be considered as the so-called problem-structuredness. From an educational perspective, well-structured problems are those that are usually encountered in schools. At the end of each chapter in every textbook, examples for this kind of problems, which most likely have a single correct answer and offer all prerequisite information for their solution within the problem text, may be found. And mostly, solving these require skills that fall under the category of Gagne's rule-learning (Gagne, Briggs, & Wager, 1992), or the application category in Bloom's taxonomy. However, there also exists another type of problems referred to as ill-structured problems, which are encountered more frequently in daily lives. These usually do not have a single correct answer, nor the information required for their solution may always be readily available. Examples to ill-structured problems include "Which car should I purchase?" or "How may I develop a

Web application?" or "What should be design specifications or technical features of a new smartphone be?". All in all, it may be understood that solutions of ill-structured problems are represented as higher-level cognitive skills by both Gagne and Bloom.

A glance at the literature shows many researchers suggesting the use of worked examples in teaching how to solve well-structured problems (Sweller & Cooper, 1985; Paas & van Merriënboer, 1994; Renkl & Atkinson, 2003). It has been proposed that worked examples reduce cognitive load especially in novice learners and help them develop strategies in tackling educational problems. Researchers also advise that the transition into problem-solving may be smoother by using backward completion problems and that taking a large step into problem-solving learning by disregarding the use of worked examples is not recommended (Renkl & Atkinson, 2003). No matter how difficult or time-consuming a well-structured problem may be, or however many different parameters its solution needs to take into account, familiarity with the rules that govern the problem's internal structure may become transferrable in time, rendering the act of problem-solving a routine task. This would reduce the cognitive effort it would take for subject matter experts working on similar, well-structured problems, taking the task to the level application in Bloom's taxonomy (Thompson, Luxton-Reilly, Whalley, Hu, & Robbins, 2008). However, the same could not be said for ill-structured problems since there may always be multiple paths to multiple outcomes that may pass as solutions. Nievelstein, van Gog, van Dijck & Boshuizen (2013) have shown in their study the reverse effect of expertise arises quickly in the solution of well-structured problems, as opposed to ill-structured ones where such an effect is not observed. This Expertise Reversal Effect has been defined by Kalyuga (2007) as "the reversal in the relative effectiveness of instructional methods as levels of learner knowledge in a domain change". In working with well-defined problems, it is the limited size of the problem space that leads experts to experience this phenomenon. In the case of ill-structured problems, however, there is usually a wider problem space that may result expertise reversal to not occur even in the most experienced and knowledgeable individuals. In short, the effect is experienced in problem solving tasks that are reduced to the lower cognitive skill level of application, whereas it is not displayed in problem-solving that requires higher order skills. And even though the use of worked examples in learning how to solve well-structured problems has been extensively researched, the notion using these in teaching how to solve ill-structured problems seems to be overlooked (Nievelstein, Van Gog, Van Dijck, & Boshuizen, 2013; Frerejean, van Strien, Kirschner, & Brand-Gruwel, 2016). Jonassen (2011) explains that cases as problems to solve may be used in teaching how to solve ill-structured problems just as they are used with well-structured problems, since students may investigate cases and use these as analogies later on, in addition to gaining prerequisite information and developing an alternative perspective in the investigation process. Needless to say, the cases presented to students as problems to solve need to display certain educational qualities (Nicol & Krykorka, 2016). Jonassen (2011) therefore concludes that a good educational problem should be (a) interesting and relevant to the solver, (b) complex and/or ill-structured, (c) authentic and rooted in real life, and (d) designed in a fashion that takes into account the context, representation mode and manipulation space.

Jonassen (2011) also expresses that the motive behind problem-solving learning activities is not limited with the need for finding and implementing an acceptable solution. These activities also prepare a student to more easily recognize and solve future problems she may encounter and help her develop skills -metacognitive or otherwise- that reduce the mental effort in doing so. As previously stated, the main goal is for students to successfully transfer their problem-solving skills to real life scenarios. In a relevant study, Perkins and Salomon (1992) list the requirements for learning transfer to occur as: 1) thorough and diverse practice 2) explicit abstraction 3) active self-monitoring 4) arousing mindfulness and 5) using a metaphor or analogy. It might be said that, in order to have the skill of problem-solving be transferred to real life scenarios, mundane problem solving tasks may not be sufficient. Learning environments may need diverse educational activities in problem-solving education where existing problems serve as analogies for others and where higher order cognitive and meta-cognitive skills are also used in the progress. One such example for these diverse activities is problem-posing.

Problem-posing is a concept popularized in academic literature by Brazilian educator Paulo Freire. Freire's approach to problem-posing is rooted in the tenets of Critical Pedagogy and it suggests an approach where the student is playing a more active role in the educational setting. In his Problem Posing Education, Freire envisions a classroom where students may sincerely and openly share with their peers problems that affect their personal lives and are not reluctant to argue on or take action towards the solution of these. Voiced by students themselves, the collective attempts at solving these problems of personal or societal nature turns education into liberatory praxis. In other words, this progressive concept aimed to utilize education to benefit students by enabling them to develop some kind of sensitivity towards their own problems, as well as those of others and of the society, thus breaking down the invisible wall between "school" and "real life". Thus, problems posed by students in a classroom where the pedagogical approach of Freire is pursued are expected to be quite different than, well-structured problems such as algorithms or story problems; often illustrating "cases" that are relatively more complex and that may have multiple solutions or none at all.

In time, the concept of problem-posing found its way into educational domains where problems are typically well-structured, such as mathematics or natural sciences. Especially beginning with the work of Silver (1994), the notion of "mathematical problem-posing" has been popularized in the academic field of mathematics education. Freeing itself of the "mathematical" prefix in time, this approach was simply referred to as "problem-posing", "question-posing" or "student question generation" (Khansir & Dashti, 2014; Lai & Liang, 2017; Lan & Lin, 2011; Shakurnia, Aslami & Bijanzadeh; 2018, Yu & Chen, 2014; Yu, Liu & Chan, 2003) and defined as a technique based on creating new questions about a given scenario (Mishra, 2014) or the act of posing problem situations in order to generate or discover new information (Mishra, 2014; Mishra & Iyer, 2013). Nevertheless, student efforts on posing their own problems has been claimed to be beneficial in the way of developing higher order cognitive skills; regardless of whether the problems are ill-structured ones derived from actual, distressful circumstances in life, or; well-structured ones that are artificial, abstract and most-likely forgotten once the textbook cover is closed. Indeed, it has been suggested that one of the main goals of problem-posing activities is for problems posed by students to display a higher level of cognitive ability of the poser (Profetto-McGrath et al. 2004). It has also been claimed that students posing their own questions/problems in a given subject helps them improve their understanding, motivation, problem-solving skills, cognitive and meta-cognitive strategies, group communication skills, active learning behavior, flexible thinking and positive attitude towards subject matter (Kopparla et al., 2018, Yu & Chen, 2014). For this reason, including problem-posing activities in a problem-solving education context may facilitate the progress of learning to transfer problem-solving skills to real life.

**Training for the GNU/Linux Operating System**

Goldweber (2015) claims that the principal and most important learning outcome for students in fields such as IT and programming is not the display of mastery over the rules and syntax of a programming language but a general understanding of the computer is capable of doing and what kind of problems it can solve for the student.

Operating Systems is one of the indispensable courses in the curricula of most college-level educational programs offering degrees in (or related to) Computer Science. Although this course is widely offered, it has been observed that the instructional process in this course generally limited to helping a student gain understanding of operating systems and their components at a general knowledge level (Yodaiken, 1996; Yang & Wei, 2007). In other words, de facto learning outcomes of this course are usually associated with lower levels of cognitive skills in Bloom's hierarchy. However, the actual purpose of this course should not be limited to enabling the student to gain a holistic understanding of the concept of operating systems or to understand the intricate relationships between its sub-components. There also exists the fact that operating systems are the essence of a computer, without which the device would lose its identity as a problem-solving tool. Therefore, students completing the operating systems course should be expected to perceive the operating system as a problem-solving tool, gain some kind of sensitivity towards daily life problems that may be solved using it and lastly, learn to use it effectively to overcome the said problems.

Offering a large amount of transparency and universality, the highly popular family of GNU/Linux (or shortly, Linux) operating systems have been claimed to be a popular choice in problem-solving oriented computer science curricula (Adams & Erickson, 2001; Chong, 2008). In addition to the educational value it possesses, expertise in Linux use is a highly sought after skill in today's IT industry, and universities may do well to teach the general use and system administration of the Linux family of operating systems to computer science students (Rogers, 2000).

A review of the literature has failed to reveal attempts at discovering novel methods in teaching the use of the Linux operating system. However, one study worthy of note has been carried out by Dall and Nieh (2015) in Columbia University. In this research, a Web-based assignment grading environment named GradeBoard has been developed in order to facilitate teaching the subject of Linux Kernel Development. Making use of the Git version control system, it enables students to submit their homework assignments for review by their instructors, who provide them with feedback over the same environment. However, to the understanding of the authors, this system does not make use of peer-interaction or problem-based pedagogies. It has been noted in a recent study that education in the use of Linux operating system needs to include hands-on problem solving activities just as much as they include presentation of rote knowledge and yet; it is a difficult process to design and choose educational problem cases for use in this manner (Xu et al., 2014).

There exist numerous design based research efforts in the literature, where educational software solutions are used alongside problem-based pedagogies. Some of these also cover the domain of problem-posing. However, Lai and Liang (2017) reports that the limited number of studies that aim to develop a software system to assist problem-posing activities have certain limitations such as a) being developed with a generic mindset and not with a purpose of supporting a specific course subject and b) incorporating only multiple-choice type questions and not open-ended ones. To the author's information, there is currently no study investigating the specific domain of operating systems education in the context of software systems assisting open-ended and ill-structured problem-posing activities.

## PURPOSE

This descriptive case study investigates the implementation of a web-based learning platform that is still under development for the purpose of employing problem-based pedagogies in teaching GNU/Linux operating system fundamentals to college students in an operating system course (the Platform) within the context of the problem-posing cognitive process initiated by students during their interaction with the platform. Both qualitative and quantitative methods are employed to produce the most convincing body of evidence. The data collected with both methods are used to answer the following questions:

1. How does the quality of educational problems posed by students change after the implementation of the Platform within the Operating Systems course in the proposed manner?
2. Is there a correlation between student academic achievement and the quality of student-posed authentic educational problems?
3. What were student opinions on the use of the Platform, whether their satisfaction changed through the course, if yes how?
4. What were students' opinions on the problem posing activities, and how did they engaged with the Platform to complete problem posing duties?
5. According to the students, how could the design and implementation of the Platform be improved to attract students and better scaffold student generation of higher quality educational problems?

## METHOD

This descriptive case study comprises preliminary work and the first iteration of a design-based research project. According to Bakker and van Eerde (2015), design based research aims to formulate research questions and hypotheses that, when answered, will delineate a) what properties a product or intervention that has been designed for a specific purpose should carry or, from a broader perspective, b) how a desired effect may be achieved under specific circumstances.

### Developing the Platform

A Web-based software application was developed for use in the Operating Systems course as an educational platform that facilitates learning to use the GNU/Linux operating system command line as a problem solving tool and transferring these skills to real life scenarios. The following observations of the course instructor based on classroom experiences have been the starting point for designing such a tool:

- Regarding the use of the Linux command line, Turkish students are having hardships following educational material in the English language and having trouble finding quality material prepared in Turkish.
- Students are reluctant to install the Linux operating system on their personal computers with fear that they may break their computer and/or lose sensitive data.
- Students harbor prejudice against command line emulators, which they refer to as “the black screen”, and are unaware that most tasks they carry out using a GUI can be carried out just as efficiently, if not more, using the command line.

In the design phase, two very important resources regarding the training of complex cognitive skills have been referred to. One of these is the 4C/ID instructional design model developed by van Merriënboer (1997) and the other is the instructional design model offered by Jonassen (2001). The learning outcomes on problem-solving that this design seeks to enhance may be considered under the design-problems category of Jonassen's problem typology and the researcher's advice in this particular context has been heeded. Jonassen (2011) especially mentions the advantages of using numerous cases as problems to solve in the way of providing providing prerequisite information, sources of analogies and means of exercise. As for the 4C/ID model, problem cases are utilized for providing students with items for whole task practice. With this rationale, the goal has been set to create a learning environment that houses a repository of problem cases that can be solved mainly by using the Linux command line, along with their detailed solution steps. In this proposed model, however, the difficult task that is the generation of cases posing high quality educational problems lies on the shoulders of the lecturer, who would most likely base all the problems on her own experiences.

As stated above, one of the main purposes teaching the fundamentals of the GNU/Linux should be to prevent students from passing the course without really realizing that the operating system, without which the computer would be reduced to a mere heap of semiconductor, is actually a problem solving tool. In order to address this issue and to (a) help students enjoy the aforementioned benefits pertaining to cognitive processes as promised by Silver's problem-posing technique; (b) let students develop a critical perspective by letting them bring real world problems into the classroom and giving them a chance at praxis, as proposed by the Freirean approach to problem-posing; (c) enable the creation by students of multiple problem cases defined by Jonassen as sources of practice and analogies, which would be students to be solved by their peers, and; (d) enable the creation by students of whole task practice items as proposed by the 4C/ID model; the decision of including problem-posing method has been taken. This way, students may work as generators of questions/problems that may initially be used in part-task and later in whole-task practice sessions. The quality and quantity of student generated problems is expected to increase over time while students benefit from the pedagogical affordances of problem-posing technique that enables fostering of several cognitive and meta-cognitive skills.

In the case of this study, problem-posing activities are also expected to increase the awareness of students towards problems that may be solved using the Linux command-line as a tool while they try to pose educational problems that reflect real life scenarios. It should not be forgotten that although students tend to display negative prejudice towards the "black screen" in the Desktop Era of computing, the computer is a problem solving tool at heart and even if it may sound like a high-brow remark, the command line is the computer itself. According Stephenson (1999), interfaces such as GUI tend to set a distance between people and the computer, making it harder for them to understand the core mechanics of the device and act as layers of abstraction that may lead to alienation. There are cases where computing problems that may be solved in mere seconds using programmatic commands issued over the command line may take hours using the GUI, or may not even be solvable. The purpose of using problem-based pedagogies in Linux command line training is to help students be able to solve by themselves or at least identify and articulate these kind of problems, which may come in a wide array of contexts.

Eventually, the Platform was planned to be an open and collaborative learning environment where students may pose their educational problems related to the basics of Linux command line use and solve problems posed by their peers.

**Structural and Technical Features**

Following the previously mentioned general principles, a web-based software application prototype, where students may author two kinds of educational materials, namely "guides" and "problems", has been developed and served over the Web.

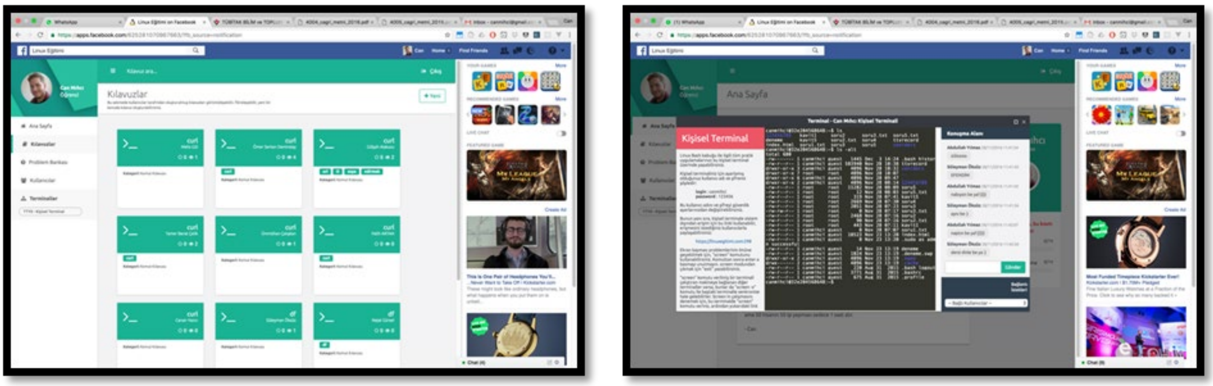
As the name suggests, "Guides" refer to student-created tutorial pages on various concepts regarding UNIX-based systems; as well as commands, utilities and programs that are specific to the Linux command line. The inspiration for this type of educational material has been information on most commands found in the man and --help pages generally accessible over the Linux command line. The initial idea has been to randomly assign each student 5 entries from a list of 50 basic commands/utilities/programs and let them create educational guides in Turkish language on the usage of these. Later, each student was to be assigned 10 randomly selected guides from a pool created in this manner and edit the content generated by their peers (whose names have been hidden for the purpose of anonymity) to perfection. This process involved the students in (a) making use of the man and –help pages, as well as resources on the Internet to come up with a general understanding (b) testing out their findings on a command line emulator; and (c) explaining to their peers the knowledge they constructed in this manner. A screenshot from the gallery of student generated guides uploaded onto the Platform has been given in Figure 1a.

The educational material "Problem" refers to educational problems posed by students on the platform. These represent the output of problem-posing activities. For the creation of this type of material, students have been asked to initially create a text that reflects a problem case, the scenario of which can be traced back to a real-life context, preferably inspired by their own experiences, needs or demands. After this, they were asked to configure a sandboxed environment powered by a virtual machine running a Linux distribution to reflect the structural elements of the problem case they created, so that the end result is a "problem space" where peers may attempt at solving the problem posed. A screenshot showing a sandboxed problem space that features (a) the problem text, (b) the command line for the virtual machine configured to reflect the problem and (c) messaging area has been shown in Figure 1b.

In order to enable users to create such problem spaces, the Platform was built in a way that lets users instantly launch remote virtual machines running a Linux distribution and access these from over a browser-based command line console. For this purpose, the Docker system based on the LXC architecture has been employed to launch containers running an Ubuntu system image. As for accessing these containers, an open source terminal emulator that runs in the web browser named shellinabox (https://github.com/shellinabox/shellinabox) has been used, as proposed by Morell and Jiang (2015). This way, users may instantly create Linux virtual machine instances that they can access and configure from over a web browser, associate these with both types of instructional materials (guides and problems) and even work collaboratively with terminal screen sharing and recording tools packages such as ttyrec and GNU Screen.

The Platform itself is a web application prototype developed using the isomorphic JavaScript application development framework Meteor, used commonly in rapid-prototyping Web projects (Fiala, Yee-King & Grierson, 2016) and which makes use of a document-based MongoDB database, as well as the Websockets protocol and the



(a) (b)

Figure 1. Screenshots of the Educational Platform (accessed in this instance over the Facebook Social Network as an integral application)

DDP technology to enable real-time interaction between users. The Platform has also been integrated with popular social networking services (SNS) of Facebook and Google+, enabling OAUTH based user login. Another reason for this design decision has been to provide users with timely notifications over SNS in order to engage them more closely. A screenshot showing the login screen of the application running within a frame as a Facebook app has been shown in Figure 1a. A profile is created for all users registering in the system and SNS integration allows for more complete profile pages, with avatar pictures.

**Implementation**

The Platform has been implemented in the last 7-week half of a 14-week long operating systems course in order to find answers to the research questions and to gather feedback for guiding further design and development efforts. The implementation took place with 45 students (K12 Computer Science Teacher Candidates) enrolled at the 3rd year Operating Systems course during the autumn semester of the 2016-2017 academic year of the undergraduate program of Computer Education and Instructional Technology at a Turkish University Faculty of Education. Qualitative data have been gathered from these students at the beginning and end of the implementation period.

In the first 7-weeks of the course that passed before the implementation stage started, the goal has been to let students gain prerequisite knowledge and skills required for problem-solving / problem-posing activities pertaining to Linux command line use. This involved regular coursework, with hands-on practice at computers in running Linux in a school laboratory, as well as homework practice using the Codecademy "Learn the Command Line" module, which offers a MOOC that teaches Linux command line basics using a sandbox environment (Codecademy, 2019). This decision was made so that students may become familiar with working in a sandboxed virtual environment accessed over a browser during the implementation. Based on anecdotes from previous years suggesting that students are having trouble working with content in English language, the course instructor has developed and shared with the students a Google Chrome browser extension, which translates Codecademy module content into Turkish.

With the coming of the midterm, the Platform was implemented into classroom activities, as well as into homework assignments. A rough schedule of the course activities related to the use of the Platform has been given in Table 1.

**Data Collection Tools**

*Rubric for Evaluating the Quality of Educational Problems Related to the Linux Command Line*

Jonassen (2011) explains five external characteristics for educational problems as (a) structuredness, (b) context, (c) complexity, (d) dynamicity and (e) domain specificity. Inspired by explanations made by Jonassen in this context and also based on his previously referenced definition of what qualities a good educational problem should possess, the authors have developed a Rubric for Evaluating the Quality of Educational Problems Related to the Linux Command Line (the Rubric). The sub-categories of the Rubric, all of which can be scored in the range of 0 to 3, can be given as follows with their weight multiplier given in parentheses next to their names: (a) Problem Situatedness (x5) - the degree to which the specific situation in which the problem takes place is related to the problem (Lee, 2004), (b) Story Context (x3) - the degree to which a problem presents a convincing story most likely reflecting a real-life scenario, (c) Problem Complexity (x3) - the amount of cognitive effort it would take to solve the problem, and; (d) Structural Uniqueness (x6) - the measure of how unique a given problem is based on its structural elements. A product of scores in each category with their relevant modifiers are added together to represent a final measure of a problem's overall quality, which may range between 0 and 51. The final look of the Rubric has been given in Table 2.

Table 1. Weekly schedule of course activities related to the use of the Platform.

| Week # | Activity |
|---|---|
| **1-5** | (a) Introduction to Operating Systems related concepts and the Linux Command Line. (b) Theoretical and practical coursework, (c) homework assignments at Codecademy Learn the Command Line module. (d) Creation of tutorial "Guides" by students for command line concepts randomly assigned to them. |
| **6** | **Alternative Question Posing:** A classroom activity at the computer laboratory where students were handed question paper from last year's midterm exam and were told to create a new question based on each one shown on the paper, considering the cognitive skill level targeted by the question, as well as its structure, difficulty and score. |
| **7** | (a) Introduction of the platform at Linuxegitimi.com, (b) Uploading of alternative questions created during Week 6 on the Platform, (c) Uploading of tutorial guides created in weeks 1-5 to the Platform. |
| **8** | Gathering of initial student opinions on the Platform following one week of use, via the First Impressions Questionnaire administered online. |
| **9-11** | Use of the Platform throughout classroom sessions, instead of the native command line interfaces of school laboratory computers. |
| **12-13** | **Authentic Problem Posing:** Homework activity that requires students to work in pairs and pose an educational problem that conforms to certain criteria. |
| **14** | (a) Final Exam for the course administered over the Platform, student responses recorded with a terminal session recorder (ttyrec) and grading was carried out over the recordings. (b) Final Evaluation Interviews. |

*Alternative Question-Posing Activity*

An exam form applied as the actual midterm exam of the Operating Systems course in the previous year was shared with the students for this activity. This exam form included 6 questions that measured knowledge in topics of the course up until midterm. All the questions in this form, except for question 2, lacked a narrative and consisted of simple instructions or demands for an answer. The students were instructed to create a new exam form that is similar to the one handed to them and were notified that their performance would affect academic success in the course. The criteria for success in the activity has been explained as "creating an alternative question for each of the questions in previous year's exam form, considering each question's score as well as the level of cognitive skills involved". In other words, the question's in previous year's midterm exam have been given to students as starting points for posing similar questions/problems. Being prospective computer science teachers and undergraduate students in a faculty of education and having passed pedagogical courses in previous years, participant students were assumed to be familiar with taxonomy of cognitive skills. The activity took place at the 6th week of the course, lasted 90 minutes and students were free to use the computers to access resources. Brief explanation about questions in the form, as well as their target cognitive skill level were given in Table 3. Student performance in this activity was measured using the Rubric for Evaluating the Quality of Educational Problems Related to the Linux Command Line.

*Authentic Problem-Posing Assignment*

At the 12th week of the course, students were asked to work in pairs for 2 weeks and pose an educational problem pertaining to the use of the Linux command line. It was explained that performance in this activity would affect their final exam scores and the scoring criteria has been shared with them. The criteria required the students to pose a problem that (a) had a convincing narrative derived from real life, possibly their own; (b) was complex enough; (c) was unique; (d) involved solution steps and (e) was delivered before the deadline. It was also noted that failure to comply with any one criterion would merely lower their score and not be interpreted as total failure. In order to standardize the presentation of problems posed, an empty template has been shared with students. This was the same template that the instructor had used throughout the course to deliver her own worked examples to students.

Table 2. Rubric for Evaluating the Quality of Educational Problems

| Sub-Category (multiplier) / Score | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| **Situatedness (x5)** | The posed question / problem has nothing to do with the command line. | The question/problem seems to be related to the command line. But it is actually related only with peripheral subjects (such as keyboard shortcuts). Or, it represents a situation where using the GUI would be more beneficial than the command line. | The question/problem is directly related to command line use. However, it is not situated in the particular context of Linux or UNIX or does not require use of GNU tools. Example: Simple file system manipulations that could be handled in a similar manner under Windows command line. | The question is directly related to the use of Linux command line. It involves the use of tools such as package managers and/or concepts such as disk mounting that are rather specific to Linux-based systems. |
| **Story Context (x3)** | The problem does not pose a story. All its structural elements are bared. There are only instructions / givens. | Question/problem displays a weak attempt at forming a narrative. This narrative is probably just an excuse to hide structural elements, there is no rationality for an actual problem. | The problem/question is presented with a narrative that tries to explain and rationalize why the situation is an actual problem that causes distress.. | The problem has a convincing story and a strong narrative displaying an interesting and problematic situation that could have been encountered in real life. It may also convey the meaning of being personally significant to its poser. |
| **Complexity (x3)** | The posed question/problem is not actually answerable / solvable. | The problem can be solved in a single step. In the case of questions, it only requires display of lower level cognitive skills such as recalling rote knowledge.. | The solution of the problem requires knowledge application and organization. Such as, solving a problem related to command line use by issuing certain commands in the correct order. However, programmatic constructs are not employed in the process. In the case of questions, it requires higher order skills such as summarization, explanation or categorization. | The solution requires actions in multiple domains of knowledge (e.g. both file system operations and network configuration and user account management). Or, it may also be that the solution makes extensive and/or creative use of programmatic structures. As for questions, answers require higher order skills such as interpretations. |
| **Structural Uniqueness (x6)** | The question / problem has been entirely copied from another source. It may also be that a few of its parameters have been changed without altering structural features. | The question/problem is heavily inspired by an example with modifications / additions to its structure. | The question/problem posed is not structurally inspired by previous examples encountered during the course but is probably pertaining to a case / concept that may routinely be encountered during daily use of the Linux operating system or the command line. | The question/problem features a structure that is quite unlike previous examples and it is pertaining to a concept/case that is most likely out of the scope of daily use of the Linux OS / command line. |

Table 3. Information on Questions in the Previous Year's Midterm Exam

| Q# | Cognitive Skill Targeted by Question | Brief Explanation of the Question Content |
|---|---|---|
| 1 | Understand (conceptual) | Instructs the student to summarily explain a concept pertaining to Operating Systems in general. |
| 2 | Apply (Procedural) | Mentions an example case where a user intends to install the GNU/Linux operating system is given as a weak narrative and the question requires the student to explain the steps to carry out this task. |
| 3 | Remembering (factual – conceptual) | Instructs the student to give information on peripheral knowledge pertaining to the Linux Command Line, such as meaning of special symbols or keyboard shortcuts. |
| 4 | Apply (conceptual) | Instructs the student to correctly write the few commands that carry out the given simple tasks |

| | | |
|---|---|---|
| 5 | Apply (conceptual) or Create (conceptual) (based on students' existing knowledge) | Instructs the student to create a Bash script file that includes several lines of commands and serves to fulfill a particular goal when executed. |
| 6 | Analyze (conceptual) | Requires the student to overview and explain the purpose of a rather long and complex chain of commands, some of which were not previously taught during the course and which, unbeknownst to the student, has a real-life use context. |

Student performance in this activity was also measured using the Rubric for Evaluating the Quality of Educational Problems Related to the Linux Command Line

*First Impressions Questionnaire*
At the end of the 8th week of the course, students were asked to voluntarily and anonymously respond to a questionnaire with eight open ended questions delivered online over Google Forms and in doing so express their opinions about the Platform. Questions sought to uncover the experience students have had using the Platform, the perceived purpose of using the Platform in this course, their overall satisfaction with the Platform and with problem-posing activities as a whole. Qualitative data collected with this form has been analyzed through content coding.

*Final Exam*
At the 14th week, the Final Exam for the course, measuring academic achievement in learning how to operate the Linux Command Line, has been administered to students. The exam, which was scored over 100 points, took place in the school computer laboratory as a practical test that required the students to carry out specific tasks on the Command Line. However, students used the Command Line interface at the Platform to submit their answers in the form of screen recordings captured with the ttyrec command line tool. The instructor has graded student answers by watching the recordings and observing student performance during attempts at answering questions, instead of evaluating the final product as an answer.

*Final Evaluation Interviews*
After the semester has ended, one-on-one interviews have been conducted with students in order gather information on the ways they engaged with the Platform, as well as the processes they went through in problem-posing activities. Participation in the interviews was voluntary. Overall, 8 students were interviewed. Qualitative data collected in this manner has been analyzed through content coding.

## FINDINGS

### First Impressions

The First Impressions Questionnaire was answered by 29 students out of 45. Responses were treated as qualitative data and themes along with their categories were formed by content coding. Each category and their subsequent themes have been given in Table 4, with numbers in parentheses representing the number of times a theme has been coded.

### Quality of Problems Posed by Students

Throughout the course, each student was tasked with posing 7 types of questions / problems and sharing these with their peers over the Platform. Six of these were questions / problems that were posed as alternatives to each of the six questions in the previous year's midterm exam, whereas the seventh problem was the product of the Authentic Problem-Posing Assignment. All the questions / problems were scored using the Rubric for Evaluating the Quality of Educational Problems Related to the Linux Command Line. Descriptive statistics for the distribution of scores for each question type posed has been given in Table 5.

In order to compare differences between quality of problems posed by each student in the group, a repeated measures statistical analysis of variance was intended to be used. However, several assumptions for this test failed, due to non-normal distribution and missing data in a block design comprised of related samples.

A Skillings-Mack test (Hollander, Wolfe & Chicken, 2013; Chatfield & Mander, 2009; Skillings & Mack, 1981) has shown that average scores in at least one of the problem categories were significantly different than the rest $\chi2(6) = 82.51$, $p < 0.05$. Post-hoc analysis with Wilcoxon signed-rank tests was conducted by excluding cases with missing values in a test-by-test basis and it was found out that rubric scores for problems posed during the Authentic Problem-Posing Assignment were significantly higher than scores of questions posed based on Q1 ($Z = -4.704$, $p < 0.05$), Q2 ($Z = -4.664$, $p < 0.05$), Q3 ($Z = -3.979$, $p < 0.05$), Q4 ($Z = -2.862$, $p < 0.05$), Q5 ($Z = -2.974$, $p < 0.05$) and Q6 ($Z = -3.347$, $p < 0.05$) in the Alternative Question-Posing Activity.

In order to find out whether a correlation exists between student academic achievement and quality of authentic student-posed problems, a Spearman's Rank-Order Correlation test was carried out between Final Exam scores and Authentic Problem-Posing Assignment Rubric Scores for 31 students that both took the exam and finished the assignment, which yielded no significant results ($r_s(29) = .297$, $p = .10$)

Table 4. Categories and Subsequent Themes Discovered in the First Impressions Questionnaire.

**Frequency of Visits**

- **(10)** Frequently visited
- **(11)** Visited once or twice
- **(4)** Never used
  - **(2)** generic excuses
  - **(2)** lack of time
- **(4)** Visited mostly during the weekend before the course

**Experience Using the Platform**

- **(7)** Easy to use, no problems to report
- **(11)** Reporting technical problems
  - **(5)** Slowness
- **(7)** Reporting structural problems
  - **(4)** Difficulty finding own content in the content pool

**Content-related**

- **(9)** Finds peer-generated content low quality
  - **(2)** Needless use of sandbox feature
  - **(2)** Plagiarized content ("copy-paste")
- **(4)** Found high-quality peer generated content
- **(3)** Viewed peer-generated content with curiosity on "how others have performed"
  - **(2)** Peer-generated content too similar to one another

**Interaction with Peer Generated Content**

- **(7)** Did not examine peer-generated content.
- **(9)** Examined peer-generated content, however partially.
- **(4)** Meticulously examined peer-generated content.

**Perceived Purpose of Use**

- **(16)** To practice using commands "without installing Linux"
- **(17)** A source of information on course-related topics
- **(10)** A repository for storing and sharing educational content created by students through homework activities
- **(17)** A social learning environment…
  - **(6)** … where future users may benefit from currently generated content
  - **(7)** … where personal problems may be shared and help may be received from peers in solving these
  - **(4)** … where peer-generated questions may be used as practice material
- **(11)** A communication tool between students for course-related topics
- **(7)** Mentions lack of course related resources in Turkish language
- **(3)** Mentions a likeness with "Codecademy"

**Recommendations**

- **(4)** Review of all content by an expert to prevent generation of useless or harmful content
- (4) Feedback from an expert
- **(2)** Grouped / ordered presentation of content
- **(2)** Tutorial videos should be added
- **(4)** Ability to search/filter content
- **(3)** Detailed user profiles should be added
- **(2)** Collaborative content-editing (Wiki style)
- **(2)** Instructor should also upload content

Table 5. Descriptive Statistics showing Problem Quality Rubric Scores for Each Category of Problems Posed

| **Posed Problem Based on** | N | $\bar{X}$ | S |
|---|---|---|---|
| Alternative Question Posing Q1 | 38 | 11.44 | 7.01 |
| Alternative Question Posing Q2 | 38 | 13.15 | 8.60 |
| Alternative Question Posing Q3 | 38 | 20.84 | 8.09 |
| Alternative Question Posing Q4 | 37 | 22.45 | 7.62 |
| Alternative Question Posing Q5 | 28 | 22.89 | 8.92 |
| Alternative Question Posing Q6 | 33 | 23.84 | 5.20 |
| None (Authentic Problem-Posing Assignment) | 31 | 32,09 | 8.83 |

## Final Evaluation

Eight students participated in semi-structured interviews that lasted half an hour on average. Student responses have been treated as qualitative data and themes along with their categories were formed by content coding of student responses. Table 6 shows categories and their relevant themes, with numbers in parentheses representing the number of times a theme has been coded.

## DISCUSSION

### Quality of Problems Posed by Students

Students have been given a chance at posing their own problems at two main occasions throughout the implementation. In the first occasion named Alternative Question-Posing Activity that took place before the implementation began, all students were given the same set of six questions and were asked to create new questions that are similar to the ones given to them. Similarity in this context has been explained to the students in pedagogical terms that they were assumed to be familiar with, such as Bloom's taxonomy of cognitive tasks. It was told that the new question they generated should be of similar complexity with the sample question they based it on. However, students were not explicitly instructed to include a story context in the students they generated in this manner, nor were they asked to try and generate a rather unique question. In contrast, the second occasion at the end of the implementation named Authentic Problem-Posing Assignment, involved the students work in pairs and attempt at posing a single problem that (a) is complex (b) has a story context – preferably based on real-life experiences. Students were also informed that their question should be unique.

Statistical tests were carried out and it was shown that the quality of educational problems posed in the Authentic Problem-Posing assignment by students, as measured by the Rubric, were significantly higher than all six types of questions created during the Alternative Question-Posing Activity. This could be due to the fact that the Rubric favored problems that had both a story context and unique a structure, both of which might have been hampered by the nature of the Alternative Question-Posing Activity. In contrast, these qualities were strictly required of students in the Authentic Problem Posing Assignment.

Lavy and Shriki (2007) display a study on the usage of “What If Not” scaffolding strategy for supporting the problem posing process by inspiring prospective mathematics teachers with sample problems, asking them to evaluate the schema and formulate a new problem by changing only certain parameters. These researchers conclude their research saying that the use of such a strategy may result in being attached to certain objects and forms in question generation and limiting creativity. It is important to mention at this point a study by Bonotto and Dal Santo (2014) that defines problem-posing as a form of creativity, which is by itself a highly-valuable yet elusive cognitive skill the mechanics of which is not fully understood. Therefore, although sample-problem scaffolding techniques such as the Alternative Question-Posing Activity shown in this study may initially help students pose problems and in doing so develop an awareness into the internal realm of problem-solving learning and an ability to tear down an educational problem into its components, it is natural for the problems generated in this manner to be of subpar educational value.

It is also worth noting that as far as posing new problems based on manipulating existing samples is concerned, although a statistical relationship has not been investigated, there seems to be a linear relationship between the cognitive requirement of the sample question (climbing up in Bloom’s taxonomy from remembering/understanding to applying/analyzing) and the educational quality of the question posed by the student.

**Student Satisfaction**
Based on data from the initial questionnaire and the final interviews, it has been found that, most students have used the Platform with varying frequencies of visit and somehow interacted with peer-generated content.
Aside from reporting a few technical bugs, which have been alleviated throughout the implementation, students were generally pleased with the Platform and the problem-posing activities it involved. To quote a few such remarks:

> *“… I think that (the website) is very nice. In fact, this website is frequently used in our current instructional technology oriented courses, as an example. All my classmates refer to this website as a good example, including it in their (presentations) as a best practice…”*
>
> *“… I thought that, in the future, when I start working as a computer science teacher (at K12), I can introduce this a website to children. It is not menacing at all, rather, it is quite friendly, highly usable…”*
>
> *“… I’d like to use this website for teaching children how to use Linux in the future. The design is great, the idea is great, the content is great. All that one has to do is design creative ways to integrate it into the course… ”*

In addition to reporting overall satisfaction, students adopted a utilitarian approach and expressed the ways they benefitted from using the Platform, indicating that it was useful as a reference material throughout the course and that the ability to launch Linux virtual machines on the fly was particularly handy for practice. However, one student reported that relying on this Platform throughout the course may prevent peers from ever installing an actual copy of the Linux operating system on their devices for personal use, which was indicated by several students to be a necessary experience to pose higher quality problems. Another way students found the Platform useful was in monitoring how their peers are doing in the course, by examining peer generated content.

**Student Opinions on Problem-Posing Activities**
Overall, students find problem-posing to be a valuable educational approach, referring to the way it fosters meta-cognitive skill development in their own words such as:

> *“… I think that problem-posing is one of the most effective instructional methods. Because, especially if the student can associate the problem with his own life, this may turn into a great educational motivator that facilitates permanency in learning. That is when he discovers things: he discovers a problem and then discovers a solution for it. The instructor, then, only serves as a guiding counselor in solving the problem”*

As such, there have also been numerous occasions of students reporting that they referred to real-life experiences in the Authentic Problem-Posing Assignment. An example could be:

Table 6. Categories and Subsequent Themes Discovered in the Semi-Structured Final Evaluation Interviews.

**Student Satisfaction**

- **(16)** I like the Platform…
  - **(3)** … and I'd like to use it even after the course has finished
  - **(5)** … and I find it visually pleasant
  - **(4)** … and I enjoyed using the personal Virtual Machine (VM) provided.

**Perceived Benefits**

- **(15)** I benefitted from using the Platform …
  - **(11)** … as a reference material for the course
  - **(10)** … especially by using the VM
    - **(5)** … since it's a better alternative to locally installing virtual machines myself.
  - **(5)** … by observing how my peers are doing.

**A Need for Sample Problems**

- **(4)** I based my problem on another problem posed by the instructor during the course
- **(3)** It is beneficial for the course instructor to provide us with sample problems
- **(3)** I felt I needed more sample problems during the course
- **(2)** Future users of the Platform may make use the content we generated as samples to base their problems on.
- **(5)** Structuring my posed problem based on an existing "problem posing template" was useful.

**References to Real Life Problems**

- **(12)** I posed a problem based on my personal, real-life experiences …
  - **(8)** … and I felt a personal involvement with the problem I posed
- **(4)** In order to pose problems about it, students need to use Linux in their daily lives.

**A Need for Social Interaction**

- **(5)** I checked content due to curiosity on how others have performed in tasks similar to mine
- **(5)** A peer-rating system may be considered for evaluating the quality of user generated educational content.
- **(2)** A space for public discussions outside of problem-posing activities, such as a forum section, may be needed.

**A Difference between Problem Solving and Posing**

- **(15)** Problem posing is beneficial as an educational activity…
  - **(8)** … because it leads to deep thought, fostering deeper/meaningful/permanent learning
  - **(4)** … but students need to know the subject matter very well.
- **(6)** I have never done this sort of activity before.
- **(6)** Problem posing is a difficult task…
  - (4) … because finding problems is hard.
  - (3) … because it's easy to find a problem but not so easy to find its solution
- **(3)** Problem-posing is an easy task…
  - **(5)** … not having to find solutions to posed problems makes problem posing an easier task.
    - **(2)** … but this reduces problem quality.

**Qualities an Educational Problem should Possess**

- **(3)** Solving it should benefit others…
  - **(2)** … educationally.
- **(5)** A high quality educational problem is one that is difficult to solve

**Miscellaneous Recommendations**

- **(5)** Weekly problem-posing assignments may be needed.
- **(6)** Content may be reviewed by "editors" before being published
- **(9)** The instructor should also generate content in the same way as students do.
- **(5)** Site content should be sorted and/or grouped based on a sensible algorithm (based on alphabetical order, problem type, problem difficulty or rating score)
- **(3)** The platform may be used by subject matter experts from the Industry, in addition to students.

> *"… Back then, I was taking the Web-programming course too. We often had to transfer our program code to the hos device. That's when I encountered the problem (I posed). The other one I considered posing, was also about something I commonly experience: So I thought, sometimes I forget to bring my USB pen drive to the school. That's a huge problem for me, especially if I got a homework file in it."*

At the beginning of the implementation and upon initial interactions with peer-generated content, some students have initially reported that they found some content to be low-quality, mostly based on peripheral features such as plagiarism and presentation format. At the end of the implementation however, student opinions on this matter had matured and they had begun expressing additional thoughts on what makes an educational problem worthwhile, reflecting the development of a meta-cognitive perspective on quality of educational problems. Examples include:

> *"… I think; a good educational problem is one that I can ponder even after a long time has passed (upon hearing or solving it)."*

> *"… (In posing problems) I felt I had to somehow express myself clearly. Thus, I had to take special care in choosing the right language and remain understandable. I had to do this both in the problem text and the solution text..."*

> *"… I thought, (my problem) should be something that can challenge students and lead them to conduct research. If the solution of my problem requires using 10 different commands, I thought, 8 of these should be things that the students know about while 2 should be things that he has to research on and discover by himself"*

Considering the references to problem complexity and structuredness in these responses, it is worth mentioning that these students had not undergone training on problem-solving learning and all the conclusions they drew were based on their own experience. While it is valuable for students to have developed awareness by themselves towards some of the elements that constitute a valuable educational problem, as put forth by Jonassen (2011), the question remains as to whether it would have bolstered this awareness to give the students an explicit yet short introduction to Jonassen's theories on problem solving learning.

The process that yielded this sort of change may have also led to students developing an awareness towards a clear distinction between problem-posing and problem-solving as cognitive activities. Students have reported many times that not necessarily having to come up with a solution for the problems they posed was making the task of problem posing much easier. Some, though, have reported that not having to formulate a solution when posing a problem may lead students to produce problems haphazardly, thus resulting in problems of lower educational value.

> *"… (As long as solution-posing is not mandatory) I think that, for most people, problem-posing activities may only mean 'fire-up a random question, then forget about who needs to solve it'"*

Others have expressed that they never undertook this sort of activity in their life and that finding problems in itself is a very difficult task, saying:

> *"… Other instructors give us problems and ask us to solve them. (In this course) We've had to create our own problem from scratch, sometimes even without needing to come up with a solution. In the beginning, I found this hard to get used to. It seemed quite different from what I've seen so far."*

> *"… It's hard. Finding a problem is hard. Especially one that is challenging enough for the student, one that is facilitating the permanent acquisition of knowledge. But yes, I think that it (problem solving) facilitates permanency in learning."*

> *"… I discovered how hard it is to be a teacher, to create exams for students to take. I've had considerable difficulty in finding a problem during the problem-posing assignment."*

Such remarks from participants were also encountered in a study by Shakurnia, Aslami and Bijanzadeh (2018), where a problem-posing group of students have been shown to have greater academic achievement in a course; albeit expressing an overall feeling of disregard and alienation towards the task.

Students also emphasized in different ways that in order to pose higher quality educational problems, one needs to be more knowledgeable about the subject matter at hand. This finding justifies the decision to make use of the Codecademy module as a homework in order to increase exposure and accelerate learning in the first two weeks of the program. However, ways to extend such an accelerated training to support problem-posing performance should perhaps be explored.

**Student Recommendations on Design and Implementation**

Students recommendations that can support the design and development efforts of the Platform for more efficiently facilitating the problem-posing process were broken down into a few major categories. The first category of recommendations indicated a demand for a larger quantity of high quality, authentic educational problem cases by subject-matter experts, such as the course instructor, for the students to take inspiration from or use as samples. It should be kept in mind that the instructor has shared such problem cases with students, as it can be inferred from their interview responses suggesting that they used these instructor-posed problems as samples. However, there were only three such problem cases shared with students (complete with their solutions) and the number may not have been enough. Also, the problem cases were shared with students over a file-sharing service and not over the Platform. Thus, students report that they need the instructor to actively partake in the content-generation efforts along with them.

Second category of recommendations reflects issues in content handling and feedback. Students reported that a smart sorting and grouping system (as opposed to the current alphabetical order) for content needs to be implemented. Whereas some students recommended a peer-review and rating system for content to be in place, so that problems can be sorted based on their structural features, contexts or complexity; others have advised that content needs to be supervised by an expert editor, such as the course instructor, before it is published. Needless to say, both systems would serve to fulfill a vital role of providing feedback for student generated content, which was another feature deemed necessary by many students.

The third category of recommendations is based on the aspect of social interaction. Some students have frequently mentioned that it was important for them to monitor the progress of others using the Platform, whereas some students wished that there would be other people using the platform, specifically subject matter experts. In the end, although the Platform was designed to support real-time interaction between students with screen sharing and instant-messaging capabilities, such interaction between peers may not have taken place since a student was quoted saying:

> *"… The entire time, I felt like there is nobody else but me using the platform in a given moment…"*

A mechanism for facilitating real-time interaction between students, which might help them with the impression that they're not alone, could be designed and implemented to alleviate this effect.

## CONCLUSION

This research is based on a design and development work in progress and showcases initial findings on a case where student problem-posing activities are observed over the use of a minimum viable product. Findings on student satisfaction levels hint that the instructional design and technical development progresses for the online learning environment are probably on the right track, albeit with many student recommendations to consider for improving the learning experience. An attempt at developing a scoring rubric for evaluating the educational value of student-posed problems within the context of a complex cognitive skill was made and its draft version has been shared with readers. Examining the findings using this tool, it can be concluded that sample-question scaffolding techniques are adequate for helping students come up with problems in the beginning, but student posed problems under explicit instructions to portray authentic, unique and personally relevant cases yield a significantly higher educational value. Students seem to develop a meta-cognitive understanding towards the nature of educational problems during the course of problem-posing activities. Nevertheless, there does not seem to be a correlation between academic achievement and quality of authentic educational problems posed by students, perhaps due to the claim that educational problem-posing may, above all, be heavily reliant on the mystic and elusive high-order cognitive skill of creativity.

## REFERENCES

Adams, D. R., & Erickson, C. (2001). Linux in education: Teaching system administration with linux. *Linux Journal*, *2001*(82es), 20.

Bakker, A., & van Eerde, D. (2015). An introduction to design-based research with an example from statistics education. *In Approaches to qualitative research in mathematics education* (pp. 429-466). Springer, Dordrecht.

Bonotto, C., & Dal Santo, L. (2014). How to foster creativity in problem posing and problem solving activities. *Technology, creativity and affect in mathematical problem solving*, 120.

Chatfield, M., & Mander, A. (2009). The Skillings–Mack test (Friedman test when there are missing data). *The Stata Journal*, *9*(2), 299.

Chong, P. (2008). On the Linux Teaching of Computer Specialty in Universities [J]. *Journal of Lishui University*, *5*, 017.

Codecademy, (2019). Learn the Command Line. Accessed Online at https://www.codecademy.com/learn/learn-the-command-line on 04.08.2019.

Driscoll, M. P. (1994). *Psychology of learning for instruction*. Boston, MA: Allyn and Bacon.

Frerejean, J., van Strien, J. L., Kirschner, P. A., & Brand-Gruwel, S. (2016). Completion strategy or emphasis manipulation? Task support for teaching information problem solving. *Computers in Human Behavior*, *62*, 90-104.

Fiala, J., Yee-King, M., & Grierson, M. (2016, June). Collaborative coding interfaces on the Web. *Proceedings of the 2016 International Conference on Live Interfaces* (pp. 49-58).

Gagné, R. M., Briggs, L. J., & Wager, W. W. (1992). Principles of instructional design (4th ed.). Fort Worth: Harcourt Brace Jovanovich College Publishers.

Goldweber, M. (2015). Programming should not be part of a CS course for non-majors. *ACM Inroads, 6*(1), 55-57.
Hsiao, J. Y., Hung, C. L., Lan, Y. F., & Jeng, Y. C. (2013). Integrating Worked Examples into Problem Posing in a Web-Based Learning Environment. *Turkish Online Journal of Educational Technology-TOJET*, *12*(2), 166-176.
Hollander, M., Wolfe, D. A., & Chicken, E. (2013). *Nonparametric statistical methods*. John Wiley & Sons.
Jonassen, D. H. (2011). Learning to solve problems: a handbook for designing problem-solving learning environments. New York: Routledge.
Kalyuga, S. (2007). Expertise reversal effect and its implications for learner-tailored instruction. Educational Psychology Review, 19(4), 509-539.
Khansir, A. A., & Dashti, J. G. (2014). The Effect of Question-Generation Strategy on Iranian EFL Learners' Reading Comprehension Development. English Language Teaching, 7(4), 38-45.
Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into practice*, *41*(4), 212-218.
Lai, C. H., Tho, P. D., & Liang, J. S. (2017). Design and evaluation of question-generated programming learning system. 2017 *6th IIAI International Congress on Advanced Applied Informatics* (IIAI-AAI) (pp. 573-578). IEEE.
Lan, Y.-F., & Lin, P.-C. (2011). Evaluation and improvement of student's question posing ability in a web-based learning environment. *Australasian Journal of Educational Technology, 27*(4).
Lavy, I., & Shriki, A. (2007). Problem posing as a means for developing mathematical knowledge of prospective teachers. In *Proceedings of the 31st Conference of the International Group for the Psychology of Mathematics Education* (Vol. 3, pp. 129-136).
Lee, Y. (2004). *Student Perceptions of Problems' Structuredness, Complexity, Situatedness, and Information Richness and Their Effects on Problem-Solving Performance* (Doctoral Dissertation). Retrieved from DigiNole FSU Digital Repository http://purl.flvc.org/fsu/fd/FSU_migr_etd-3202
Mishra, S. (2014). Developing students' problem-posing skills. In Proceedings of the tenth annual conference on International computing education research (pp. 163-164). ACM.
Mishra, S., & Iyer, S. (2013). Problem Posing Exercises (PPE): An instructional strategy for learning of complex material in introductory programming courses. In Technology for education (T4E), 2013 IEEE fifth international conference on (pp. 151-158). IEEE.
Morell, L., & Jiang, C. (2015). Using ShellInABox to improve web interaction in computing courses. Journal of Computing Sciences in Colleges, 30(5), 61-66.
Nicol, C., & Krykorka, F. (2016). The Place of Problems in Problem Based Learning: A Case of Mathematics and Teacher Education *Problem-Based Learning in Teacher Education* (pp. 173-186): Springer.
Nievelstein, F., Van Gog, T., Van Dijck, G., & Boshuizen, H. P. (2013). The worked example and expertise reversal effect in less structured tasks: Learning to reason about legal cases. Contemporary Educational Psychology, 38(2), 118-125.
Paas, F. G., & Van Merriënboer, J. J. (1994). Variability of worked examples and transfer of geometrical problem-solving skills: A cognitive-load approach. *Journal of educational psychology*, *86*(1), 122.
Perkins, D. N., & Salomon, G. (1992). Transfer of learning. International encyclopedia of education, 2, 6452-6457.
Pintrich, P. R. (2002). The role of metacognitive knowledge in learning, teaching, and assessing. *Theory into practice*, *41*(4), 219-225.
Renkl, A., & Atkinson, R. K. (2003). Structuring the transition from example study to problem solving in cognitive skill acquisition: A cognitive load perspective. *Educational psychologist*, *38*(1), 15-22.
Rogers, M. P. (2000, December). Working Linux into the CS curriculum. In *Journal of Computing Sciences in Colleges* (Vol. 16, No. 1, pp. 85-91). Consortium for Computing Sciences in Colleges.
Shakurnia, A., Aslami, M., & Bijanzadeh, M. (2018). The effect of question generation activity on students' learning and perception. *Journal of Advances in Medical Education & Professionalism, 6*(2), 70.
Silver, E. A. (1994). On mathematical problem posing. *For the learning of mathematics, 14*(1), 19-28.
Skillings, J. H., & Mack, G. A. (1981). On the use of a Friedman-type statistic in balanced and unbalanced block designs. *Technometrics*, *23*(2), 171-177.
Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and instruction*, *2*(1), 59-89.

Thompson, E., Luxton-Reilly, A., Whalley, J. L., Hu, M., & Robbins, P. (2008). Bloom's taxonomy for CS assessment. Paper presented at the Proceedings of the tenth conference on Australasian computing education - Volume 78, Wollongong, NSW, Australia.

Van Merriënboer, J. J. (1997). *Training complex cognitive skills: A four-component instructional design model for technical training*. Educational Technology.

Van Merriënboer, J. J. (2013). Perspectives on problem solving and instruction. Computers & Education, 64, 153-160.

Xu, J., Fang, M., Li, Y. F., & Fang, X. M. (2014). Experimental Cases Design for Linux Operating System Course. *The World and Chongqing*, *1*, 021.

Yang, H. P., & Wei, W. (2007). Linux operating system teaching and experimental study [J]. *Journal of Jilin Teachers Institute of Engineering and Technology*, *9*, 021.

Yodaiken, V. (1996). Cheap Operating Systems Research. In *Published in the Proceedings of the First Conference on Freely Redistributable Systems, Cambridge MA*.

Yu, F. Y., & Chen, Y. J. (2014). Effects of student-generated questions as the source of online drill-and-practice activities on learning. *British Journal of Educational Technology*, *45*(2), 316-329.

Yu, F.-Y., Liu, Y.-H., & Chan, T.-W. (2003). A networked question-posing and peer assessment learning system: A cognitive enhancing tool. *Journal of Educational Technology Systems, 32*(2-3), 211-226.