DEVELOPING PEDAGOGICAL DISCOURSE FOR EFFECTIVE SIMULATION-BASED LESSONS IN PROGRAMMING

Sakirulai Olufemi Isiaq and Md Golam Jamil
Southampton Solent University
United Kingdom

Abstract

In this study, we explored pedagogical aspects of simulation-based programming lessons using the lens of students' and instructors' perceptions. We followed a qualitative approach using focus groups and semi-structured interviews with the students and instructors of a British university. Findings suggest a number of pedagogical guidelines including measuring prior knowledge of learners, varying instructional approaches, and addressing student learning preferences in pedagogical designs, amongst others. A key contribution of this study has been a rich pedagogical discourse containing effective simulation-based practices transferable among disciplines that can use simulation for learning and teaching.

Key words: simulation, pedagogy, meaningful learning, teaching, programming

The Need for a Simulation-based Pedagogical Discourse

The term simulation refers to the use of a technological device or model that facilitates elements of reality for the purpose of practical experience and learning enhancement (McGaghie, Issenberg, Petrusa, & Scalese, 2010). The utilisation of simulation for formal education has been in existence for more than two hundred years and the approach has been widely applied in medical, aviation and maritime education (Woolley, 2009; Wyatt, Archer, & Fallows, 2015). Presently, simulation is used in various disciplines including engineering, business, computing and education (Chini, Straub, & Thomas, 2016; Isiaq & Jamil, 2018). Consequently, the knowledge of simulation has extended from technological characteristics to a medium of learning and teaching (Harder, 2009). As a result, the need for understanding the pedagogical perspectives and best practices in simulation-based educational activities has been established (Rystedt & Sjoblom, 2012).

Simulation links real actions of future academic and professional work with similar learning environments and actions at formal educational institutions (Kelly, Forber, Conlon, Roche, & Stasa, 2014). In some cases, simulation can be an alternative for industry placement (Rochester et al., 2012). In simulation-based teaching, aspects of curricular content are integrated for providing comprehensive and standardised practical learning experiences to students (Gonczi, 2013). The approach offers a collaborative and supportive learning environment for imitating risky actions in a safe and corrective learning environment (Jeffries, 2012). However, simulation itself cannot lead to effective learning if the design and facilitation are not properly conducted (Dieckmann, 2009).

There are barriers to using simulation for learning and teaching, for example, lack of study resources, inadequate teacher preparation time and professional development, and teachers' lack of simulation experiences (Hayden, 2010). There is also a gap of theoretical understanding about how simulation contributes to learning (Bland, Topping, & Wood, 2011). Therefore, researchers and practitioners consider the need for a broader philosophical understanding of simulation concepts with transparent, systematic application and procedural use of the tool or model in education (Tun, Alinier, Tang, & Kneebone, 2015). In other words, it is important to discuss the theoretical aspects of simulation in relation to associated educational principles. In this regard, a balance between teaching-focused and learning-focused theoretical explanations needs to be explored (Kaakinen & Arwood, 2009). Hence, a rigorous discourse of simulation-based pedagogy, particularly the relevance, challenges and solutions need to be adequately constructed.

The paper explores various pedagogical aspects of simulation-based programming lessons including teaching preparation, content delivery approaches, learning-related challenges and assessment techniques. The findings build a pedagogical discourse by amalgamating diverse experiences and perceptions of students and instructors of a British university.

Aspects of Simulation-based Programming Pedagogy

The teaching and learning of programming are considered difficult because they require concrete understanding of the operational procedures of computational devices and models as well as enhanced code manipulation competence (Ma, Ferguson, Roper, & Wood, 2011). Additionally, a programmer is expected to be creative, problem solver and critical thinker (Bergin, Reilly, &Traynor, 2005). Moreover, high student engagement and meaningful learning experiences are essential for achieving programming-related competencies (Kujansuu & Tapio, 2004). These conditions may exacerbate the difficulty of designing and delivering programming lessons when simulation is involved.

According to learning theories such as behaviourism, cognitivism and constructivism, simulation is effective for acquiring knowledge and skills. For example, it involves 'learning by doing', the approach that follows the experiential learning model (Kolb, 1984). Simulation also allows conscious and repetitive practice that supports gaining mastery of certain skills (Sawyer et al., 2011). Moreover, simulation allows the creation of a self-directed learning environment, which presents relevance of the learning contents and their applications (Bryan, Kreuter, & Brownson, 2009). Although simulation-based programming lessons involve distinctive subject matters and unique educational approaches; a detailed discussion, particularly the pedagogical discourses of this field, has not been significantly developed in the literature. Yet, several educational issues including student engagement and teacher roles have been studied with a reference to programming lessons (White, 2017). For example, creativity and the applied features of programming subjects are emphasised for meaningful learning outcomes (Kujansuu & Tapio, 2004). It is also encouraged that simulation-based lessons combine activities that can stimulate students' behavioural, cognitive and emotional engagement (Isiaq & Jamil, 2018).

The Study

This study is conducted in two stages. Stage-1 built the ground for understanding programming lessons from the viewpoint of student engagement and meaningful learning. Stage-2 extended this understanding of learning to pedagogical practices with a specific attention to teaching aspects. The findings gained from these stages together have generated a rich pedagogical discourse that helps with the understanding of the educational dynamics in relation to simulation-based practices and its effectiveness in programming education.

Stage-1: Earlier Study

In this stage, we followed a mixed-method case study approach to explore the dimensions of student engagement in traditional and simulation-based programming sessions, and their impact on programming pedagogy. By using a self-report survey and a set of Critical Incident Questions (CIQ), our research provided useful findings on the dimensions of student engagement in simulation-based programming lessons (Isiaq & Jamil, 2018).

At this stage, we identified a strong interweaving relationship between three engagement dimensions, namely behavioural, emotional and cognitive in the simulation-based programming lessons. We found that simulation is able to facilitate personalised learning, higher engagement and a strong link between learning content and students' future work and profession. According to students' perceptions, simulation-based programming sessions were more collaborative and focused on specific learning goals. However, we identified the need for cognitively challenging tasks in such lessons for greater gains of meaningful learning. A key lesson learned at this stage was that the use of simulation for the delivery of programming lessons becomes effective when the pedagogical activities involve a balanced intervention of behavioural, emotional and cognitive exercises. These findings led us to study feasible approaches to designing and implementing suitable pedagogical activities that can ensure this balance of engagement dimensions. Therefore, we conducted the following stage of our research to understand the pedagogical aspects of simulation-based programming lessons through the lens of student perceptions and teacher reflections.

Stage-2: Present Study

With a continuation of the learning from Stage-1, Stage-2 focused on the exploration of effective approaches to designing and implementing simulation-based programming lessons. In this regard, we captured the experiences and perceptions of the instructors and students to evaluate the following aspects of teaching and learning:

- The pedagogical benefits for instructors and students in simulation-based programming modules
- The pedagogical challenges they face

- The pedagogical preparations required for effective participation in such academic programmes

Methodology

In Stage-2, we adopted a qualitative approach using focus group and semi-structured interview methods. The key reason for adopting a qualitative research approach was due to its provision of in-depth explanations of unexplored fields of this study (Creswell, 2007). Particularly, we used focus groups and interviews to collect a rich amount of opinion and experience of research participants through reinforcing and challenging their information (Stewart, Shamdasani, & Rook, 2007). Thirty-five students and four instructors of a second year computing module of a teaching-focused British university participated in the study. We had obtained ethical approval from the respective university before commencing our research.

First, the students attended five focus group sessions, each lasting about thirty minutes, and discussed the pedagogical activities that helped them to be engaged and achieve meaningful learning of the programming topics. Examples of the questions include: "When do you generally feel more engaged in the class?" "What is the role of your classmates in the lessons?" and "What makes you become inattentive in class and less engaged?"

Second, we interviewed four instructors of the programming module (simulation-based and traditional) and asked them questions about lesson planning and delivery techniques. Examples of the interview questions include: "How do you prepare the delivery of a simulation-based programming lesson?" "What challenges do your students face while participating in simulation-based programming lessons? Give some examples," and "how do you know that your students have achieved the required knowledge and skills of programming?'. Each interview lasted about twenty-five minutes and its semi-structured nature allowed us to improvise questions for a better clarification of the responses.

All of the focus group and interview participants had attended a series of specially designed simulation-based and traditional programming lessons. For this reason, the academic experiences of the participants were practical and relatable. Therefore, the perceptions and reflections of these research participants on the teaching and learning aspects related to simulation-based programming lessons were valid and reliable.

Both the focus group and interview sessions were audio-recorded and transcribed by a professional transcriber. Then, we processed the datasets using NVivo software. We followed a thematic analysis procedure while describing the findings. In reporting, we triangulated the focus group and interview data along with the findings from Stage-1, which provided richer perspectives and comprehensive understanding of the impact of simulation-based programming pedagogy (Johnson, Onwuegbuzie, & Turner, 2007; Teddlie & Tashakkori, 2009).

Findings and Analysis

The focus group and interview data individually and together constructed a useful discourse on simulation-based programming pedagogy. The findings have been categorised in four broad areas (see Figure 1 below).

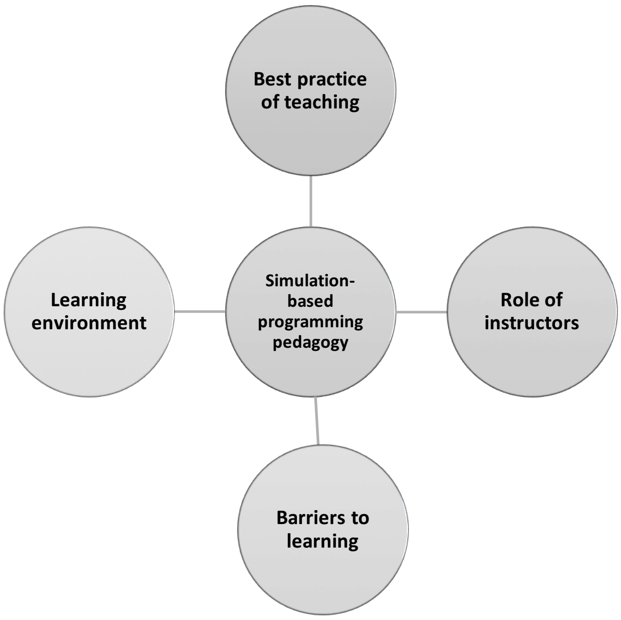

Figure 1. Discourse areas of simulation-based programming pedagogy.

Learning Environment

There was consensus among the instructors that the students are generally engaged in the simulation-based programming sessions. An instructor indicated that the attention and understanding of the students were greater compared to non-simulation-based sessions. The instructors also found the teaching of programming topics quicker and simpler in simulation-based sessions as students received real-time guided instructions in a step-by-step manner. A common challenge was a case of technological glitches, particularly when there was a network connection problem.

During the focus groups, students indicated the need for a strong link between lecturers and students when explaining or presenting new topics. They believed that the connection could be built through questioning and guided tasks. They also identified that preparing learning goals through 'working examples' could better facilitate the process. Some students suggested a number of essential conditions for making the simulation-based lessons more meaningful. First, prior to the commencement of any new topics they wanted structured guidelines and examples given to improve motivation and participation. They showed an interest in learning the process of programming, so they expected that their instructors would 'explain every detail' about how a program works. Second, students expected sufficient time for doing practical exercises. They also mentioned the need for a communication platform for sharing and consulting with instructors and classmates while completing tasks. Some students believed programming tasks could be better accomplished if there was a medium for prompt questioning and responses.

Barriers to Learning

Some students indicated a number of challenges they felt hindered their learning. For some, dis-engagement was the key barrier caused by lack of clarity of learning goals of some of the sessions. Less challenging tasks also caused dis-engagement:

> For me it was the xx lesson. I think it was so simple, I wasn't engaged at all … because there are different [proficiency] levels [of] … xx topic, and it was from beginning, and I was a little bit bored.

According to some of the students, they found the module quite exciting at the beginning, but gradually lost their enthusiasm. It is possible that they grasped the concepts and skills of programming very fast with the use of simulation, or the design of the module was not cognitively challenging enough in the latter part. Students also mentioned the need for challenging tasks, as they did not want 'to be spoon-fed.' Another barrier to learning was the different levels of prior knowledge of the students. According to a focus group participant;

> I know that some students don't have the solid knowledge, they get lost so easily, sometimes; students don't get everything all day.

The structure of teaching can also hinder learning in simulation-based sessions. Some students indicated that they felt their instructors were not fully supportive in some sessions. The finding suggests a need for greater teacher preparation and support for students in programming lessons.

Conversely, the instructors did not identify any major difficulties that their students had faced during the simulation-based programming lessons. However, they believed an over-reliance on the simulation-based activity caused some students' to give low attention to the tasks. Further application of other computing tools and techniques, such as artificial intelligence with a simulation tool could further enhance meaningful learning for students. The instructors mentioned that the lesson materials, such as worksheets and practice tasks that were prepared centrally, could be enriched by some modifications, for example through incorporating more problem-solving tasks.

Role of Instructors

There were varying preparation approaches by instructors for simulation-based programming sessions. One instructor mentioned that there was less preparation, as the person believed the simulation tool did most of the work. In addition, the instructor was less involved in the session design phase as the module leader completed this aspect mostly. Other instructors stated that they prepared themselves on the contents of the lessons mainly and, to some extent, the technological facilities in the classroom.

The students came up with some ideas about the effective roles of instructors. They found that programming concepts are complex, but could be learned easily through efficient teaching.

> There's couple of things with programing, every now and then where you think the task you're about to take is going to be incredibly daunting, and then when explained properly and well, it actually goes that it's really easy … and then applying it to other situations then again it also becomes easy…

The findings indicate an important role of the instructors in simulation-based programming sessions, which is generally perceived as highly technology-oriented. Some students stated that the instructors should organise collaborative learning tasks, as a few tasks were less inclusive:

> [In hands-on practice session] … you feel like: I'm gonna put my headphones on my head and ears and start to listen to music and don't worry about others; so you don't feel engaged [with others], you don't feel like a, yeah, feeling the lessons, it's just you, yourself and the computer.

The students also looked forward to demonstrations and detailed guidelines from the instructors. Additionally, they expected that the instructors would be more familiar with their learning needs and apply suitable pedagogical procedures for facilitating meaningful learning.

Best Practices of Teaching

By evaluating the engaging and non-engaging as well as enjoyable and non-enjoyable tasks through the lens of the research participants, the following instances of best teaching practices in simulation-based programming sessions have been revealed.

- Students did not like to be spoon-fed in the practice sessions, rather they wanted to experience challenging tasks, both independently and collaboratively. Prompt answers and explanations of students' queries helped them learn better.
- According to the instructors, although simulation-based programming activities were different from traditional programming lessons, still the assessment tasks were the same, such as timed examinations and written reports. Because of the practical nature of programming module, there is need for a rethink of assessment procedures in such academic programmes.
- The focus group participants mentioned that the instructors should be aware of the knowledgebase of their students. Otherwise, the teaching may become interesting for some students, but at the same time boring for others. The research participants advised the instructors to design differentiated simulation tasks for different levels of students.
- Demonstration of programming using simulation provided an enhanced learning experience for the students. The research participants believed programming could easily be learned when the instructor divides the tasks in steps and demonstrates the process as whole.

- Pre-session activities, such as reading materials and small tasks enhanced the productivity of practice sessions. Therefore, the instructors can design ‘flipped’ lessons where student can practise programming outside the class and share their learning experiences as well as demonstrate some steps in the class. However, it is important for the instructors to monitor the progress of the students as minor mistakes can lead them to failure and frustration about the programming tasks.
- The students and instructors emphasised the need for understanding the back-end processes of programming. They believed visual explanations of common errors could better explain the processes. They also deduced that the instructors should provide a rigorous support to students before they start practising programming, otherwise repeated mistakes can demotivate and dis-engage them from learning.
- The research participants prioritised the importance of psychological engagement for the effective learning of programming through simulation. They also suggested some pedagogical activities, such as workshops, collaborative tasks, process animation, book research and worksheets for engaging them to explore simulation and programming tasks more deeply.
- The students considered programming sessions to be demanding as they involved a high level of cognitive engagement and physical actions. Therefore, they recommended a flexible and comfortable learning environment, which allows refreshments and taking breaks in the middle of sessions.

Conclusion

The study identified some dissimilarity in the perceptions of students and instructors about simulation-based programming lessons. The issues included the nature of the learning environment, the roles of the instructors and pedagogical activities. Several learning barriers including less challenging tasks and ambiguity of pedagogical structures were identified as critical areas to address in the design and implementation of such academic programmes. The study provided several best practices of simulation pedagogy in programming sessions. However, we found the need for adequate pedagogical preparation as imperative for improving the quality of student learning experience.

References

Bland, A. J., Topping, A., & Wood, B. (2011). A concept analysis of simulation as a learning strategy in the education of undergraduate nursing students. *Nurse Education Today, 31*(7), 664-670.

Bergin, S., Reilly, R.& Traynor, D. (2005, October). Examining the role of self-regulated learning on introductory programming performance. *Proceedings of the First International Workshop on Computing Education Research* (pp. 81-86). New York, NY: ACM.

Bryan, R. L., Kreuter, M. W., & Brownson, R. C. (2009). Integrating adult learning principles into training for public health practice. *Health promotion practice*, *10*(4), 557-563.

Chini, J. J., Straub, C. L., & Thomas, K. H. (2016). Learning from avatars: Learning assistants practice physics pedagogy in a classroom simulator. *Physical Review Physics Education Research*, *12*(1), 010117-1-15.

Creswell, J. W. (2007). *Qualitative inquiry and research design: Choosing among five approaches* (2nd ed.). Thousand Oaks, CA: Sage..

Dieckmann, P. (2009). Simulation settings for learning in acute medical care. In P. Dieckmann (Ed.), *Using simulations for education, training and research* (pp. 40-138). Lengerich, Germany: Pabst.

Gonczi, A. (2013). Competency-based approaches: Linking theory and practice in professional education with particular reference to health education. *Educational Philosophy and Theory*, *45*(12), 1290-1306.

Harder, B. N. (2009). Evolution of simulation use in health care education. *Clinical Simulation in Nursing, 5*(5), 169-172.

Hayden, J. (2010). Use of simulation in nursing education: National survey results. *Journal of Nursing Regulation*, *1*(3), 52–57.

Isiaq, S. O., & Jamil, M. G. (2018). Enhancing student engagement through simulation in programming sessions. *The International Journal of Information and Learning Technology, 35*(2), 105-117.

Jeffries, P. R. (2012). *Simulation in nursing: From conceptualization to evaluation.* New York, NY: National League for Nurses.

Johnson, R. B., Onwuegbuzie, A. J., & Turner, L. A. (2007). Toward a definition of mixed methods research. *Journal of mixed methods research*, *1*(2), 112-133.

Kaakinen, J., & Arwood, E. (2009). Systematic review of nursing simulation literature for use of learning theory. *International Journal of Nursing Education Scholarship*, *6*(1).

Kelly, M. A., Forber, J., Conlon, L., Roche, M., & Stasa, H. (2014). Empowering the registered nurses of tomorrow: Students' perspectives of a simulation experience for recognising and managing a deteriorating patient. *Nurse Education Today*, *34*(5), 724-729.

Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development*. Upper Saddle River, NJ: Prentice-Hall.

Kujansuu, E., & Tapio, T. (2004). Codewitz–An international project for better programming skills. In *EdMedia: World Conference on Educational Media and Technology* (pp. 2237-2239). Waynesville, NC: Association for the Advancement of Computing in Education (AACE).

Ma, L., Ferguson, J., Roper, M., & Wood, M. (2011). Investigating and improving the models of programming concepts held by novice programmers. *Computer Science Education*, *21*(1), 57-80.

McGaghie, W. C., Issenberg, S. B., Petrusa, E. R., & Scalese, R. J. (2010). A critical review of simulation-based medical education research: 2003–2009. *Medical education*, *44*(1), 50-63.

Rochester, S., Kelly, M., Disler, R., White, H., Forber, J., & Matiuk, S. (2012). Providing simulation experiences for large cohorts of 1st year nursing students: Evaluating quality and impact. *Collegian*, *19*(3), 117-124.

Rystedt, H., & Sjoblom, B. (2012). Realism, authenticity, and learning in healthcare simulations: Rules of relevance and irrelevance as interactive achievements. *Instructional Science, 40*(5), 785-798.

Sawyer, T., Sierocka-Castaneda, A., Chan, D., Berg, B., Lustik, M., & Thompson, M. (2011). Deliberate practice using simulation improves neonatal resuscitation performance. *Simulation in Healthcare*, *6*(6), 327-336.

Stewart, D. W., Shamdasani, P. N., & Rook, D. W. (2007). *Focus groups: Theory and practice* (2 ed.). Thousand Oaks, CA: Sage.

Teddlie, C., & Tashakkori, A. (2009). *Foundations of mixed methods research: Integrating quantitative and qualitative approaches in the social and behavioral sciences*. Thousand Oaks, CA: Sage.

Tun, J. K., Alinier, G., Tang, J., & Kneebone, R. L. (2015). Redefining simulation fidelity for healthcare education. *Simulation & Gaming, 46*(2), 159-174.

White, M. (2017). Keep calm and simulate on: Faculty experiences and insights into implementing best practices in simulation. *Teaching and Learning in Nursing*, *12*(1), 43-49.

Woolley, M. (2009). Time for the Navy to get into the game! *US Naval Institute Proceedings, 135*(4) 1,274 34-39.

Wyatt, A., Archer, F., & Fallows, B. (2015). Use of simulators in teaching and learning: Paramedics' evaluation of a patient simulator? *Australasian Journal of Paramedicine*, *5*(2)

Author Details

Sakirulai Olufemi Isiaq
femi.isiaq@solent.ac.uk

Md Golam Jamil
md.jamil@solent.ac.uk