ISSN 2676-9425

CENTRAL-EUROPEAN JOURNAL

OF

NEW TECHNOLOGIES IN RESEARCH, EDUCATION AND PRACTICE

Eötvös Loránd University
Faculty of Informatics
Budapest, Hungary

Volume 4

Number 1

2022

Contents

Adopting Computer Science Pedagogical Patterns in Developing and Enhancing Computational Thinking Among Zero-Year Engineering Students

Loice Victorine ATIENO, TURCSÁNYI-SZABÓ Márta

Abstract Prospective undergraduate engineering students are assumed to possess Computational Thinking (CT) skills through basic programming. They are thus required to have considerable programming experience to enable them to execute tasks involving CT skills. Unfortunately, this is not the case as existing studies demonstrate lack of these skills, leading to poor academic performance. This was the case with first-year computer science international students at Eötvös Loránd University, faculty of Informatics, prompting the management to seek the causes of poor academic performance among the students and proper interventions. Two intervention courses were created to help develop and enhance CT among the students – *Introduction to Computational Thinking* and *Scratch Programming.* Apart from inculcating CT among learners, another underlying concern is that teachers are not keen to embrace CT approaches in the classroom. This is attributed to lack of time in utilising machine technology and lack of pedagogical skills. This was the case when the Scratch programming course was first offered. Like teaching any other subject, CT teachers require a collection of pedagogical experiences to enable effective CT teaching, including pedagogical patterns. Therefore, this paper demonstrates the adoption of computer science (CS) pedagogical patterns in the designing and teaching Scratch Programming course to zero-year students in the faculty to develop and enhance CT. The patterns were established through a literature review and tested in a course designed for zero-year computer science students for a period of 14 weeks. The patterns established through the literature review were deployed in the development and delivery of the course. The survey was conducted to establish the effectiveness of the tool. This formed part of preliminary results for the ongoing design research on integration of CT in creative learning.

Keywords: Computational Thinking, Pedagogical Patterns, Computer Science Pedagogical Patterns, Scratch Programming Course, Engineering Students

1. Introduction

Instituting computing and computational thinking (CT) in the school curriculum has triggered excitements and challenges related to new subjects [1]. This, therefore, necessitates the need for teachers to adopt new suitable subject delivery pedagogies especially, those related to algorithms, programming, and development of CT [2]. Despite substantial efforts to improve the understanding of CT, teaching CT for problem-solving is a new field for most teachers coupled with the complexity of problem-solving arising from executing new subject matter [3]. Teachers are also faced with challenges resulting from existing and emerging learning designs resulting from new ideas augmented by utilization of new technologies, with pedagogic theories and anecdotal explanations of other's practices being the only support available [4].

[1] Sue Sentance and Andrew Csizmadia, 'Computing in the Curriculum: Challenges and Strategies from a Teacher's Perspective', *Education and Information Technologies*, 22.2 (2017), 469–95 <https://doi.org/10.1007/s10639-016-9482-0>.

[2] Sentance and Csizmadia.

[3] Judith A. Cooper and Kristin L. Gunckel, *Teacher Perspectives of Teaching Computational Thinking* (Baltimore, Maryland, 2019).

[4] Diana Laurillard and Michael Derntl, 'Learner Centred Design - Overview', in *Practical Design Patterns for Teaching and Learning with Technology*, Yishay Mor (Rotterdam, Boston: Sense Publishers, 2014), pp. 13–16.

Implementing these theories and research can be challenging, hence the continuous research on practical methods of sharing successful and commendable practice amongst educators [5]. Numerous studies have been conducted on strategies and approaches for engaging learners in CT, focusing on creating new learning environments, tools, and activities that facilitate learning of CT [6]. Consequently, this has provoked the need for accessible, easy-to-use, and adaptable contextualised models and representations [7]. The representations and models (in form of case studies and learning design patterns) are effective tools and resources that facilitate educators' conceptualisation of innovative and alternative learning methods, particularly in complex tasks [8]. The critical question is, can CS pedagogical design patterns be employed in developing and enhancing CT?

Therefore, this paper seeks to demonstrate how CS pedagogical patterns were adopted in the development and implementation of a CT development course such as the Scratch programming Course.

2. Computational Thinking

CT supports capability advancement while lessening computing limitations as it involves thinking through problems and deploying the steps resulting to a solution [9]. CT is open to all and not just students of technology as it impacts learners' attitudes towards problem solving, ensuring success for the problem-solver. CT offers learners resources and aid for discovering new or distinctive problem-solving techniques hence boosting their confidence in problem solving. Teachers need to continually strive towards inculcating this sense of agency within their learners to shape their capacity to manage themselves in and out of the classroom and in the future. To facilitate the integration of CT, teachers' support is also essential. Teaching CT has since been perceived as mainly a constructionist idea [10].

According to constructionism, deep learning occurs when learners create their own purposeful projects together with other learners then meticulously reflect on the process while enabling freedom to explore their interests using technology [11]. Based on the constructionist approach,

---

[5] Robyn Philip, 'Finding Creative Processes in Learning Design Patterns', *Australasian Journal of Educational Technology*, 34.2 (2018), 78–94 <https://doi.org/10.14742/ajet.3787>.

[6] Mitchel Resnick and others, 'Scratch: Programming for All', *Communications of the ACM*, 52.11 (2009) <https://doi.org/10.1145/1592761.1592779>; Pratim Sengupta and others, 'Integrating Computational Thinking with K-12 Science Education Using Agent- Based Computation : A Theoretical Framework', *Educ Inf Technol*, 18 (2013), 351–80 <https://doi.org/10.1007/s10639-012-9240-x>.

[7] Sue Bennett and others, 'Learning Designs: Bridging the Gap between Theory and Practice', in *In ICT: Providing Choices for Learners and Learning. Proceedings Ascilite Singapore 2007.*, 2007, pp. 51–60; Matt Bower, 'Design Thinking and Learning Design', in *In Design of Technology-Enhanced Learning Integrating* (Bingley: Emerald Publishing., 2017), pp. 121–58; Peter Goodyear and Symeon Retalis, 'Learning, Technology and Design', in *Technology-Enhanced Learning: Design Patterns and Pattern Languages*, ed. by Peter Goodyear and Symeon Retalis, 2nd edn (Rotterdam, Boston: Sense Publishers, 2010), pp. 1–27.

[8] Philip.

[9] J. M. Wing, 'Computational Thinking', *Communications of the ACM*, 49(3).March 2006 (2006), 33–35 <https://doi.org/10.1109/vlhcc.2011.6070404>.

[10] Marina Umaschi Bers and others, 'Computational Thinking and Tinkering : Exploration of an Early Childhood Robotics Curriculum', *Computers & Education*, 72 (2014), 145–57 <https://doi.org/10.1016/j.compedu.2013.10.020>; Alan Buss and Ruben Gamboa, 'Teacher Transformations in Developing Computational Thinking : Gaming and Robotics Use in After-School Settings', in *Emerging Research, Practice, and Policy on Computational Thinking*, ed. by P. J. Rich and C. B. Hodges, 2017, pp. 189–203 <https://doi.org/10.1007/978-3-319-52691-1>.

[11] Bers and others.

various CT instructional approaches and techniques have been identified that can be used at different levels and across the curriculum [12]. This includes teachers using CT to model their own comprehension, learning, and progress (modelling) [13]; Collaboration among teachers to accomplish interdisciplinary projects (integration) [14]; teachers releasing responsibility gradually; teachers encouraging learners and giving tips and hints for problem-solving using probing questions rather than giving real solutions [15]; and using CT vocabulary across the curriculum [16]. Apart from the CT teaching approaches, teaching and learning in formal learning is structured. Various CT development models have been proposed that stipulate the CT to be developed and the process of CT development.

This is illustrated in "*A model for developing computational thinking skills*" [17] using a three-staged problem-solving cycle, dependent on CT to solve problems algorithmically: *problem definition* (problem formulation, abstraction, problem reformulation, and decomposition); *problem-solving* (data collection and analysis, algorithmic design, parallelization and iteration, and automation); and *analysing the solution* (generalization, testing, and evaluation) as shown in figure 1. The problem-solving process is a cycle starting from problem formulation and ending with evaluation of the whole process. CT is viewed as a problem-solving approach that requires creativity and critical thinking for the desired outcomes to be realised.

Despite the extensive research on the approaches, tools, and even activities for developing and enhancing CT in the classroom, an underlying concern is that teachers are not keen to embrace CT approaches in the classroom [18]. This is attributed to a lack of time in utilising machine technology, combined with the lack of pedagogical skills [19]. Like teaching any other subject, CT teachers require a collection of pedagogical experiences to enable effective CT teaching, including pedagogical patterns.

---

[12] Enoch Hunsaker, 'Computational Thinking', in *The K-12 Educational Technology Handbook*, ed. by A. Ottenbreit-Leftwich and R Kimmons (EdTech Books, 2020), pp. 1–16 <https://edtechbooks.org/k12handbook/computational_thinking>; Abeera P Rehmat, Hoda Ehsan, and Monica E Cardella, 'Instructional Strategies to Promote Computational Thinking for Young Learners', *Journal of Digital Learning in Teacher Education*, 36.1 (2020), 46–62 <https://doi.org/10.1080/21532974.2019.1693942>.

[13] Rehmat, Ehsan, and Cardella.

[14] Bers and others.

[15] Buss and Gamboa; Rehmat, Ehsan, and Cardella.

[16] Aman Yadav and others, 'Computational Thinking in Elementary and Secondary Teacher Education', *ACM Trans. Comput. Education*, 14.1 (2014), 16 <https://doi.org/http://dx.doi.org/10.1145/2576872 1.>; Hunsaker.

[17] Tauno Palts and Margus Pedaste, 'A Model for Developing Computational Thinking Skills', *Informatics in Education*, 19.1 (2020), 113–28 <https://doi.org/10.15388/INFEDU.2020.06>.

[18] Nur Hasheena Anuar, Fitri Suraya Mohamad, and Jacey-lynn Minoi, 'Contextualising Computational Thinking : A Case Study in Remote Rural Sarawak Borneo', *International Journal of LEarning, Teaching and Educational Research*, 19.8 (2020), 98–116.

[19] C. C. Selby, 'How Can the Teaching of Programming Be Used to Enhance Computational Thinking Skills? (Doctoral Dissertation, University of Southampton, England, United Kingdom). Https://Doi.Org/10.1016/j.Jsv.2010.04.020' (University of Southampton, England, United Kingdom), 2014) <https://doi.org/https://doi.org/10.1016/j.jsv.2010.04.020>.

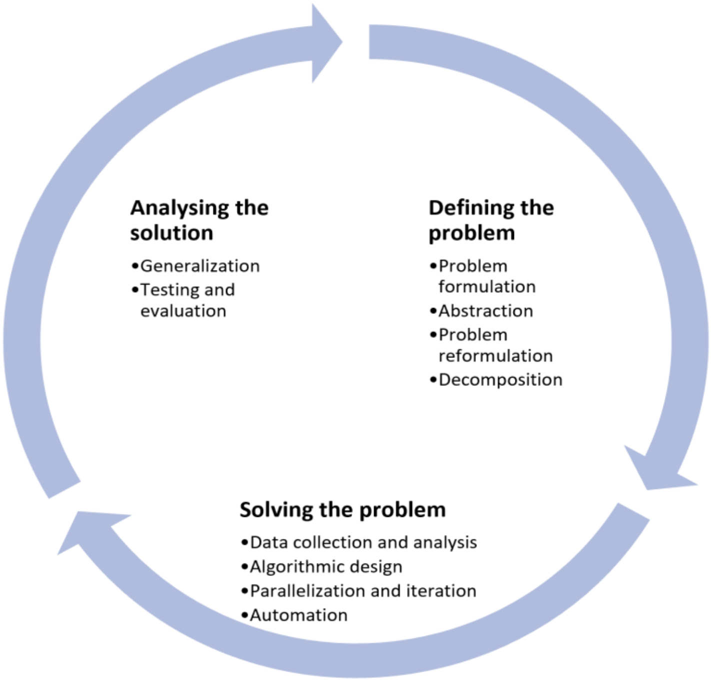


Figure 1: A model for developing computational thinking skills [20]

3. Pedagogical Patterns

Pedagogical Pattern (a term coined by the Pedagogical Patterns Project Team) endeavours to depict successful teaching practices to effectively share them with others as a means of learning from them [21]. Educational design has gained considerably from patterns especially in connection to: presenting the teacher-designer with an elaborate set of design initiatives; presenting the design ideas in a structured manner enabling understanding of relationships between design patterns; merging a clear expression of a design problem and solution, and recommending a justification which links pedagogical philosophy, research-based evidence and experiential knowledge of design; and presenting the knowledge in a way that it facilitates an iterative, fluid, process of design, extending over hours or days [22].

Pedagogical Patterns are not new ideas; rather, they are comprehensively tested and proven useful practices in various contexts by numerous people hence, they are not something *invented*, instead they are *discovered* [23]. The underlying assertion backing this effort is the tried and tested problem-

[20] Palts and Pedaste.

[21] Eva Magnusson, 'Pedagogical Patterns – a Method to Capture Best Practices in Teaching and Learning', in *Genombrottet Konferens 2006*, 2006 <https://www.lth.se/fileadmin/lth/genombrottet/konferens2006/PedPatterns.pdf>.

[22] Peter Goodyear, 'Educational Design and Networked Learning: Patterns, Pattern Languages and Design Practice', *Australasian Journal of Educational Technology*, 21.1 (2005), 82–101 <https://doi.org/10.14742/ajet.1344>.

[23] Magnusson.

solving approaches to solving repeated challenges or tackling ordinary requirements created by experienced educational practitioners [24].

A wide array of CS education pedagogical practices stem from teachers' expertise. Novice teachers find it hard and time-consuming to map these pedagogical practices to existing learning theories, hence having the teachers depending on their insight or pedagogies observed while students [25]. A pattern poses a problem and its solution together with the forces that must be employed enabling the solution suitable to the problem. The patterns seek to solve problems such as learner's motivation, choosing and sequencing teaching materials, evaluating students, among others. As the call for integration of CT into the curriculum grows wider, teachers need not only models, but also a collection of pedagogical patterns that can efficiently facilitate the development and enhancement of CT among learners. This paper looks at how CS pedagogical patterns can be utilised by teacher to develop and enhance CT in formal learning.

3.1 Computer Science Pedagogical Patterns

Patterns enable the design and delivery of a single course lasting one term or semester. Even though the emphasis is on CS, the patterns may be utilised in other fields as they aim to capture good practice in an organised manner, enabling their transfer to others, particularly to novice teachers [26]. The patterns include patterns for course preparation, course delivery, evaluation, feedback, among others.

Patterns Categories	**Brief Description**	**Sample Patterns**
Course Development	These patterns facilitate course preparation and choosing the material. They guide the comprehension of the course itself and not its delivery	New Pedagogy for New Paradigms, Need to Know, Abandon Systems, Early Bird, Spiral, Lazy Professor.
Course delivery (whole Course)	Guides the design on the organisation of the material and deciding on activities.	Early Bird, Spiral, Group Work, Lazy Professor, Active Student, Buddy System, Language Reinforces Paradigm, Write Over Read, General Concepts First, Study Groups, Reduce Risk, Stealth Instructor.
Course delivery (course or major topics introduction)	The idea of these patterns has to do with the first introduction of new material. How can you introduce new topics? What initial activities are appropriate?	Set the Stage, Lay of the Land, Visible Plan, Learn their Names, Fixer Upper, Occam, Read Before Write, Consistent Metaphor, High Leverage.

[24] Yishay Mor and Niall Winters, 'Design Approaches in Technology Enhanced Learning', *Interactive Learning Environments*, 15.1 (2007), 61–75 <https://doi.org/10.1080/10494820601044236>.

[25] Mary Lou Maher and others, 'Design Patterns for Active Learning', in *Faculty Experiences in Active Learning: A Collection of Strategies for Implementing Active Learning Across Disciplines* (University of North Carolina Press, 2020).

[26] Joseph Bergin, 'A Pattern Language for Course Development in Computer Science', 2002 <http://csis.pace.edu/~bergin/patterns/coursepatternlanguage.html>.

Course delivery (Evaluation & Feedback)	These patterns cover testing and student evaluation.	Fair Grading, Fair Project Grading, Fair Team Grading, Key Ideas Dominate Grading, Student Online Portfolios, Grade it Again Sam, Students Selected Activities, Trial Exam, Self-Test, Debrief After Activities, Peer Feedback. Communication: Rule of 1-Rule of 2, 24 by 7, Constant Feedback, Differentiated Feedback, Early Warning, Anonymous Mailbox, Ask Your Neighbour.
Course delivery (Dealing with problems)	Problems always occur in running any course. These patterns give some advice about being prepared for the inevitable.	Buffers, Prepare Equipment, Let the Plan Go, Debrief, Human Professor, Capture Everything.

Table 1: Sample Computer Science Pedagogical Patterns [27]

3.2 Pedagogical Patterns in Teaching Computational Thinking

Palts and Pedaste in "*A model for developing computational thinking skills*" [28]**,** outline three stages of CT development, with each stage outlining the key tasks to be carried out: *problem definition* (problem formulation, abstraction, problem reformulation, and decomposition); *problem solving* (data collection and analysis, algorithmic design, parallelization and iteration and automation); and *analysing the solution* (generalization, testing and evaluation).

Problem definition: constitute all the CT skills that are needed before starting to solve a problem. Problem-solving begins with *(a) problem formulation,* which is realised by researching in order to understand the problem to be solved. This is followed by *(b) abstraction* as definition of key ideas through identifying and extracting relevant information is vital in formulating the problem. The key aspects realised during abstraction are then modelled. There is then need to *(c) reformulate the problem* into a familiar and solvable one. Lastly, the problem is *(d) decomposed* into manageable tasks.

Problem solving: This is the second stage based on the model and constitute all CT skills implicated in crafting the problem solution. Algorithmic problem solving requires (e) *data collection and analysis* followed by *(f) algorithmic design* which is a series of ordered steps then deployment of *(g) parallelization and iteration* leading to the process *(h) automation*.

Analysing the solution: This involves *(j) generalization*, which is applying the problem-solving process to a broader array of problems. The last task is *(j) evaluation and testing*, involving processes' outcomes analysis (assessment and acknowledgment) based on efficiency and resource utilization. This also entail systematic testing and debugging, efficiency and performance constraints, error detection, among others. In formal learning, the above tasks can therefore be accomplished is a structured manner, as depicted in the flowchart in Figure 2.

To enable the learners to successfully accomplish these tasks, teaching CT requires a selection of various teaching approaches [29]. There are moments teacher-centred approaches are useful in introducing concepts and model capabilities; nevertheless, learner-centred pedagogies are paramount for learners of computing to fuse their understanding, knowledge transfer, creativity

[27] Bergin.

[28] Palts and Pedaste.

[29] Mark Guzdial, 'Education: Paving the Way for Computational Thinking', *Communications of the ACM*, 2008, 25–27 <https://doi.org/10.1145/1378704.1378713>.

development, and facilitate peer learning [30]. Continued professional development should enhance teachers' CT pedagogies and adaptation approaches based on student needs [31]. Unfortunately, there is no consensus on strategies for teaching and assessing the level of CT development in learners [32]. Complete integration of CT into the existing curricula unquestionably poses significant challenges, particularly for teachers, as they may not be able to determine suitable pedagogies for teaching CT [33]. This is evident in the concern relayed by experts regarding the shortage of qualified teachers for the new curriculum delivery, especially when introducing new ideas and concepts [34]. Therefore, this calls for the need for experts' documented experience to inform of pedagogical patterns in enhancing teachers' capability in teaching CT.

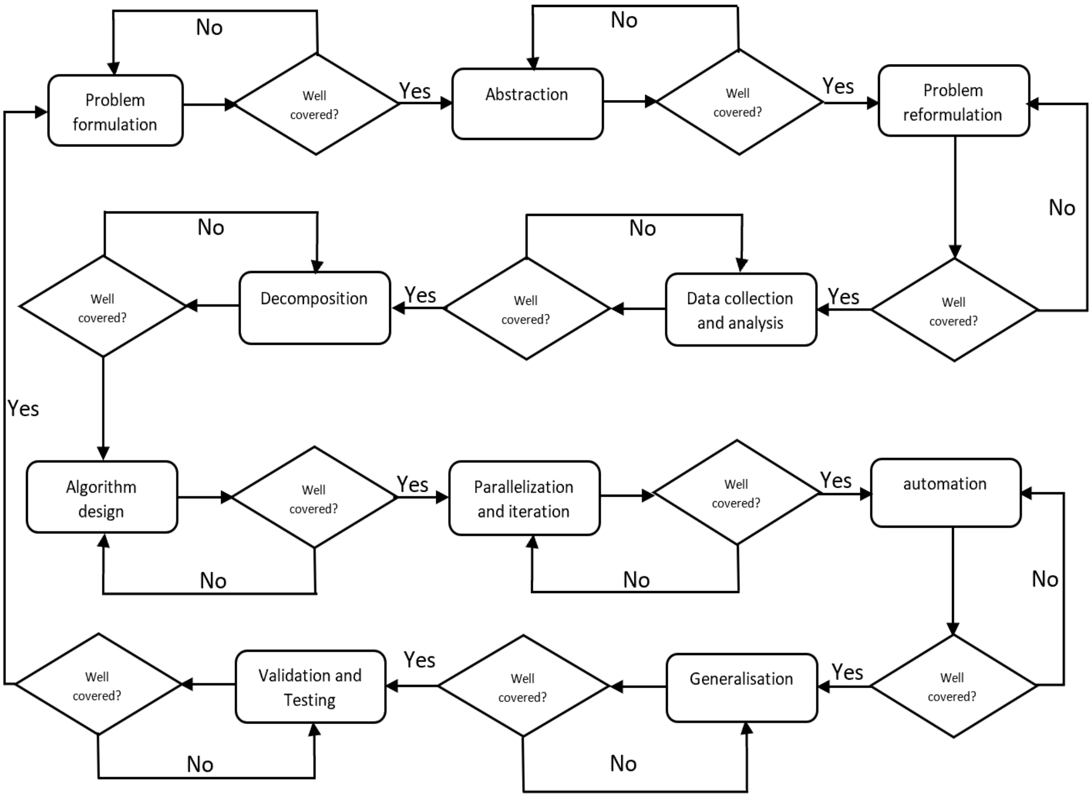


Figure 2: Flowchart Depicting the Process of Problem Solving with CT (Source: Author)

[30] Matt Bower, 'Redesigning a Web-Conferencing Environment to Scaffold Computing Students ' Creative Design Processes', *Journal of Educational Technology & Society*, 14.1 (2011), 27–42; Matt Bower and John G. Hedberg, 'A Quantitative Multimodal Discourse Analysis of Teaching and Learning in a Web-Conferencing Environment - The Efficacy of Student-Centred Learning Designs', *Computers and Education*, 54.2 (2010), 462–78 <https://doi.org/10.1016/j.compedu.2009.08.030>.

[31] Tara Stevens and others, 'Middle Level Mathematics Teachers' Self-Efficacy Growth through Professional Development: Differences Based on Mathematical Background', *Australian Journal of Teacher Education*, 38.4 (2013) <https://doi.org/10.14221/ajte.2013v38n4.3>.

[32] Karen Brennan and Mitchel Resnick, 'New Frameworks for Studying and Assessing the Development of Computational Thinking', *AERA*, 2012, 1–25 <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf> [accessed 10 May 2019].

[33] Matt Bower and others, 'Improving the Computational Thinking Pedagogical Capabilities of School Teachers', *Australian Journal of Teacher Education*, 42.3 (2017), 53–72 <hhttp://files.eric.ed.gov/fulltext/EJ1137876.pdf>.

[34] Wen J. Peng and others, 'Emerging Perceptions of Teacher Quality and Teacher Development in China', *International Journal of Educational Development*, 34.1 (2014), 77–89 <https://doi.org/10.1016/j.ijedudev.2013.04.005>.

The patterns comprise of the context (where the pattern is applied), problem (problems to be solved), constraints and forces (what necessitate the application of the problem), and solutions (solution to the problem). The overall pattern here is the development and teaching of Scratch programming course

Context: Pedagogical patterns are applicable to courses or programs with complex tasks and ambitious goals and are beneficial to novice teachers [35]. In this case, Scratch programming course was offered to students with a vast background, had basic or no programming knowledge, and required CT skills to advance their studies in computer science engineering.

Problem: There is an assumption that students joining engineering courses at the undergraduate level possess CT skills through basic programming skills. Thus, they are expected to have considerable programming knowledge that would enable them execute projects that require CT skills. Unfortunately, existing studies conducted among undergraduate students exhibit lack of these skills [36], leading to poor academic performance. This has made entry-level courses complex for these students as they lack adequate programming knowledge (or background) as per the course prerequisites. The Scratch programming course students were international students from different countries with varied education systems; hence, their background, experience, and motivation are widely divergent. This made the harmonisation of student learning progress difficult, resulting to the increase in the complexity and difficulty of teaching the course. To attain the eventual goal of the course, there is need for a pedagogical method that is not only able to draw students' interest but also to keep this interest during the entire period of the course hence the adoption of the existing pedagogical patterns.

Forces: Teaching CT calls for the balance between teacher-centred and student-centred approaches to learning. When designing the course content and the activities for the course, the teacher has to have in mind how the two approaches can be balanced for effective learning to occur. The Scratch programming Course consisted of several activities to be carried out by learners. In case of complex activities scaffolding was used to achieve the desired goal.

Solutions: Developing CT in formal learning environment like a university setup is structured and this is best elaborated by "*A model for developing computational thinking skills*" [37]**.** CT development in such an environment is teacher guided; nevertheless, the utilization of learner-centred pedagogies are vital for learners to fuse their understanding, knowledge transfer, creativity development and facilitate peer learning [38]. To enhance the acquisition of CT in such an environment, various pedagogical patterns are adopted during CT development. The successful completion of each activity in CT development is evaluated, leading to the ultimate learning goal.

[35] Zhen Jiang, Eduardo B. Fernandez, and Liang Cheng, 'P2N : A Pedagogical Pattern for Teaching Computer Programming to Non-CS Majors', in *PLoP '11 Proceedings of the 18th Conference on Pattern Languages of Programs Retrieved From*, 2011.

[36] Gábor Csapó, 'Placing Event-Action-Based Visual Programming in the Process of Computer Science Education', *Acta Polytechnica Hungarica*, 16.2 (2019), 35–57; Mária Csernoch and others, 'Testing Algorithmic Skills in Traditional and Non-Traditional Programming Environments', *Informatics in Education*, 14.2 (2015), 175–97 <https://doi.org/10.15388/infedu.2015.11>.

[37] Palts and Pedaste.

[38] Bower, 'Redesigning a Web-Conferencing Environment to Scaffold Computing Students ' Creative Design Processes'; Bower and Hedberg.

In the Scratch Programming course, the development of CT was done using different learning themes. Various activities were carried out, which included lecturing, group discussion, teacher-led class discussions, project development, peer reviewing, reflections and evaluation. Our solution focused on the achievement of the overall goal which is to develop and enhance CT among learners.

3.3 Sample Implementation of the Patterns in Teaching Scratch Programming Course

Course Development: The course content was developed in themes, as shown in Table 2 adopted from creative computing with scratch curriculum [39], scaled down to meet our requirements. The learners were expected to come up with their own projects based on the themes provided.

Topic	**Description**	**Required Sessions**
Introduction to Scratch	Introducing creativity in computing and Scratch using sample projects and hands-on experiences.	4
Exploring Arts	Exploring arts by creating projects that include elements of music, drawing and dance.	4
Digital Stories Telling	Exploring storytelling by creating projects that include characters, scenes and narrative.	4
Developing Games	Exploring games by creating projects that define goals, levels and rules	6
Final project	Developing independent projects by first identifying the suitable project, utilising problem solving with CT, collaborating with others to improve the project and presenting the project and its development process.	8

Table 2: Course Themes

Various CS pedagogical patterns were used as a guide in developing the Scratch programming course with some of the sample patterns described below. According to Bergin [40], there is need for patterns that can facilitate course preparation and choosing of the materials to be taught. This is basically about the course and not its delivery. While developing the CT course, there was **Need to Know** the content and the emphasis given to each topic. This is important as some topics are more challenging, longer or even more interesting than others. The Scratch programming course was structured in a thematic manner that gave the learners a range of projects to work on. This also calls for structuring the content to enable proper content completion. Essential ideas in the course were mined and placed first in the course structure hence the **Early Bird.** **Spiral** was necessary during the course development to attain the necessary information on the important topics. This was a practical course with several projects to be accomplished with limited time available. Therefore, it was important to plan the activities aiming at what the learners were to do and not what the teacher would do, hence **Lazy Professor**. The **Abandoned Systems** approach was adopted that allowed materials selection and methodologies that enable learners' improvement on various fronts, resulting in solving new and meaningful problems. For novice teachers, teaching

[39] Karen Brennan, Michelle Chung, and Jeff Hawson, 'Creative Computing: A Design-Based Introduction to Computational Thinking', *Nature*, 2011, 73 <http://scratched.gse.harvard.edu/sites/default/files/curriculumguide-v20110923.pdf>.

[40] Bergin.

something new can be challenging hence need to adopt **New Pedagogy for New Paradigms** that advocates for use of different pedagogy, not just different examples, to teach the new concept.

Course delivery: Design-based learning approach was used to deliver the course content. The design-based learning approach is a method known for involving learners in solving real-life design problems as they reflect on the process of learning [41]. It is characterized with: the creation of artifacts, personalization of the creations, collaboration and reflection [42] which form the core of our learning process. For each theme, the students were introduced to the new concepts, and they were expected to create projects that would help develop and enhance CT among them.

Organising the course materials and activities: In most courses there is more to be taught and less time available and this was the case with the Scratch programming course that expected students to master various CT skills and dispositions within a period of 12 weeks. Teachers are therefore advised to use examples and exercises that cover more than one idea or topic simultaneously, hence **Multi Pronged Attack**. The course used design-based approach to learning where students were expected to develop projects. The approach was also supported by the **Active Student** pattern as each project developed covered various ideas and concepts that needed to be mastered. The development of projects kept the students active both in and out of class. The course was structured into themes and the learners' given opportunities to come up with their own projects within the themes (**Students Selected Activities**).

Introducing the course or new topic: **Set the Stage** when introducing new topics and at the beginning of the course to get the learners' attention and make sure all are ready to learn. This was done by naming the topic for emphasis and reinforcement; reviewing prerequisites; and showing the target using the learning objectives. A **Lay of the Land** provided a dramatic picture of the target. A **Visible Plan** about the course was created and shared with the students at the beginning of the course.

Evaluation and feedback: Students were provided with constant **Feedback** to enable the students establish their understanding of what was taught. **Positive Feedback First** was used as criticisms help the students enhance the criticized facets and should be given in time as it can significantly increase motivation [43]. To ensure students have understood the topic, they were given a chance to try what had been taught on their own (**Try it Yourself**). The students were also encouraged to be less dependent on the teacher through the use of **Student Portfolio** and **Peer Grading**. To ensure fair grading **Key Ideas Dominate Grading** was used and this was made possible using rubrics. Lastly, **Anonymous Feedback** was solicited using questionnaires on the effectiveness of the teaching style.

Dealing with problems: when students breeched the classroom code of conduct instead of punishing them, the vice was turned into a learning activity (**Human Professor**) for example if they missed lessons, they were given activities to accomplish that will cover the lesson taught in their absence.

[41] Mehalik and Schunn, 'What Constitutes Good Design? A Review of Empirical Studies of Design Processes', *International Journal of Engineering Education*, 22.3 (2006), 519–32.

[42] Brennan, Chung, and Hawson.

[43] Bergin.

Implication: By adopting CS pedagogical patterns in teaching Scratch programming course, the major conclusion of the work is that the existing pedagogical patterns can be used in teaching new courses. This was evident in how the learners improved in various areas, such as their motivation towards the course, persistence in dealing with complex tasks, managing individual and group learning, and embracing feedback.

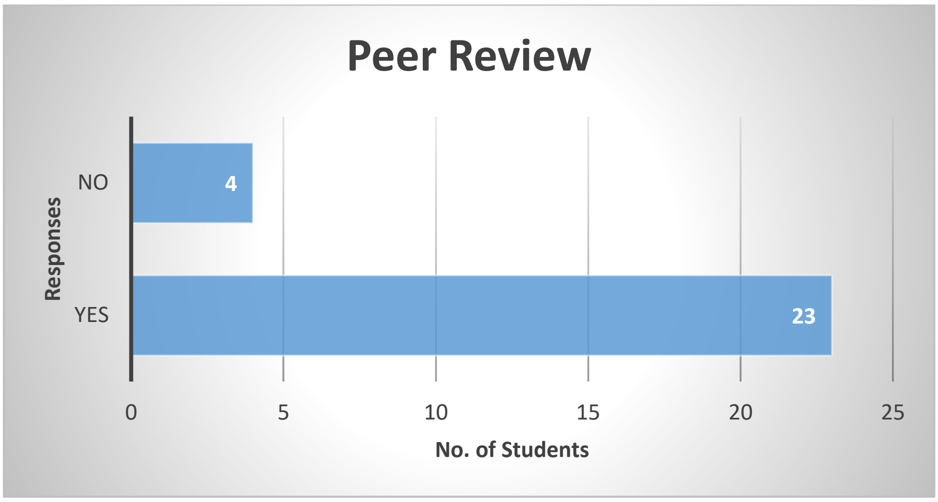


Figure3: Effectiveness of Peer Review

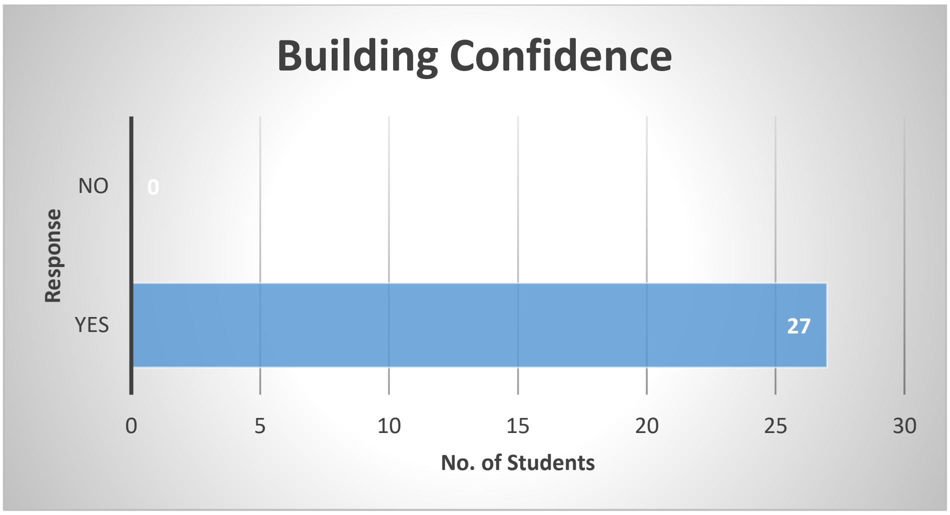


Figure 4: Building Student Confidence (Author)

Figures 3 and 4 shown the effectiveness of the design patterns on the learning outcome that has led to the development of CT. The patterns also helped in the selection of various learning activities, giving the learners a wide range of activities to select from (Figure 5), with most students preferring group activities. According to constructionism, deep learning occurs when learners create their own purposeful projects together with other learners then meticulously reflect on the process while enabling freedom to explore their own interests using technology [44]. The course was rated highly by the students (Figure 6). The sample results demonstrate the successful use of the patterns in the selected course and this can be applied in any other course.

[44] Bers and others.

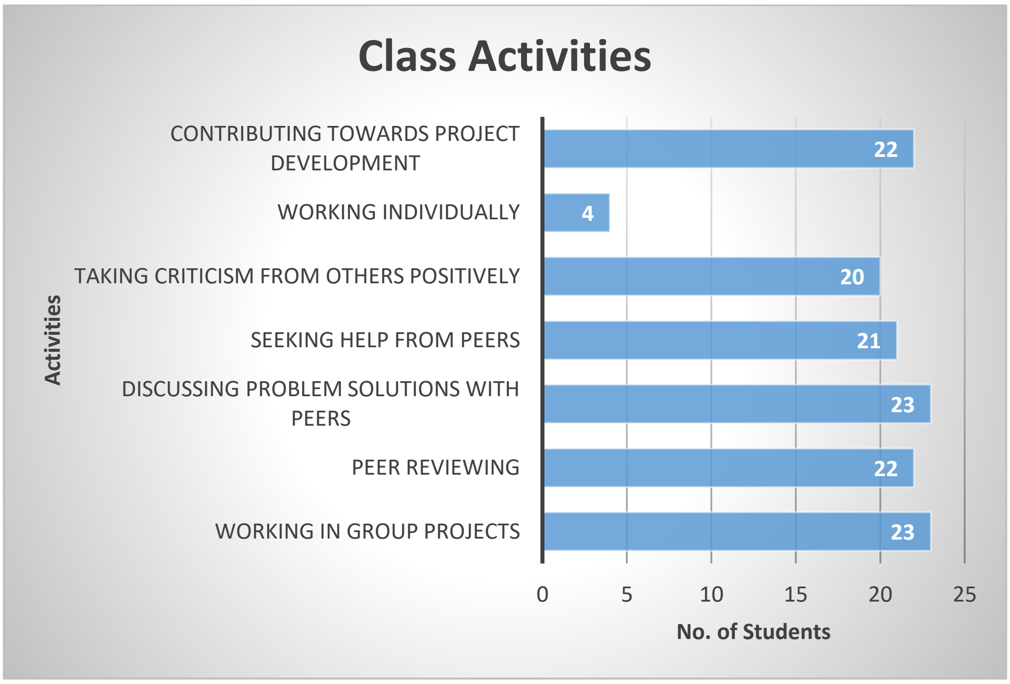


Figure 5: Learners' preference of activities (Author)

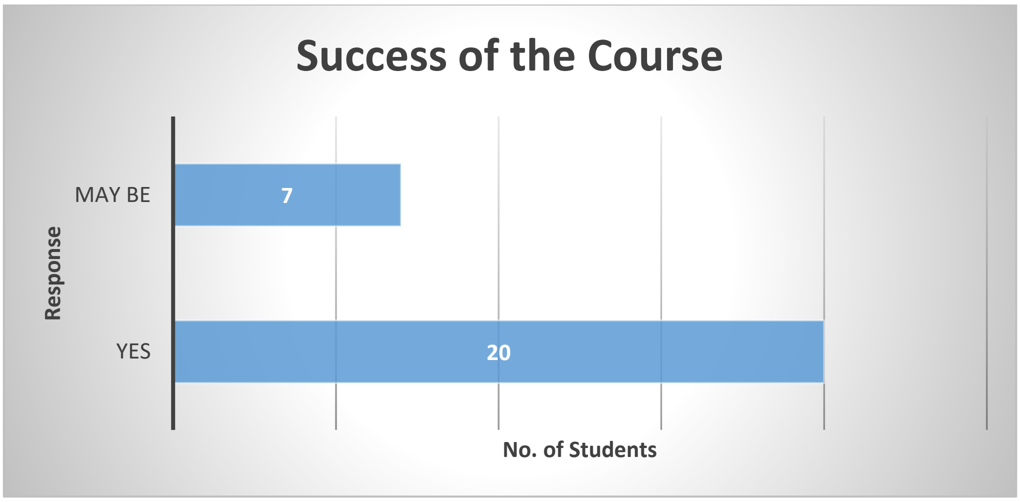


Figure 6: Overall effectiveness of the course (Author)

5. Conclusion and Recommendations

The goal of this paper was to share how various CS pedagogical patterns were used in the implementation of Scratching programming course to effectively develop and enhance CT among the learners in formal learning. Designing the course and eventually delivering it was guided by some of the sample pedagogical patterns outlined in section 2.3. Adopting the pedagogical patterns has led to the successful development of the course and its delivery. The research therefore advocates for the adoption of pedagogical patterns more so by novice teachers and enhancement of the patterns to enable development of CT in informal learning too. The patterns are also crucial in the in effective teaching of new courses should therefore be part of the teacher's tools for teaching.

References

1. N. H. Anuar, F. S. Mohamad, and J. Minoi: *Contextualising Computational Thinking : A Case Study in Remote Rural Sarawak Borneo*, International Journal of LEarning, Teaching and Educational Research, 19.8 (2020), 98–116. https://doi.org/10.26803/ijlter.19.8.6
2. S.Bennett, S. Agostinho, L. Lockyer, L. Kosta, J. Jones, R. Koper, and B. Harper: *Learning Designs: Bridging the Gap between Theory and Practice*, in In ICT: Providing Choices for Learners and Learning. Proceedings Ascilite Singapore (2007), pp. 51–60
3. J. Bergin: *A Pattern Language for Course Development in Computer Science*, (2002) http://csis.pace.edu/~bergin/patterns/coursepatternlanguage.html [Accessed on 15th March 2021]
4. M. U. Bers, , L. Flannery, E. R. Kazakoff, and A. Sullivan: *Computational Thinking and Tinkering : Exploration of an Early Childhood Robotics Curriculum*, Computers & Education, 72 (2014), 145–57 https://doi.org/10.1016/j.compedu.2013.10.020
5. M. Bower: *Design Thinking and Learning Design*, in In Design of Technology-Enhanced Learning Integrating, Bingley, Emerald Publishing, (2017), pp. 121–58 https://doi.org/10.1108/978-1-78714-182-720171008
6. M. Bower: *Redesigning a Web-Conferencing Environment to Scaffold Computing Students ' Creative Design Processes*, Journal of Educational Technology & Society, 14.1 (2011), 27–42
7. M. Bower and J. G. Hedberg: *A Quantitative Multimodal Discourse Analysis of Teaching and Learning in a Web-Conferencing Environment - The Efficacy of Student-Centred Learning Designs*, Computers and Education, 54.2 (2010), 462–78 https://doi.org/10.1016/j.compedu.2009.08.030
8. M. Bower, L. N. Wood, J. W. M. Lai, C. Howe, R. Lister, R. Mason, K. Highfield and J. Veal: *Improving the Computational Thinking Pedagogical Capabilities of School Teachers*, Australian Journal of Teacher Education, 42.3 (2017), 53–72 http://files.eric.ed.gov/fulltext/EJ1137876.pdf [Accessed on 2nd February 2021] https://doi.org/10.14221/ajte.2017v42n3.4
9. K. Brennan, , M. Chung, and J. Hawson, 'Creative Computing: *A Design-Based Introduction to Computational Thinking*, Nature, (2011) http://scratched.gse.harvard.edu/sites/default/files/curriculumguide-v20110923.pdf
10. K. Brennan, , and M. Resnick: *New Frameworks for Studying and Assessing the Development of Computational Thinking*, AERA, (2012), 1–25 http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf [Accessed on 19th March 2021]
11. A. Buss and R. Gamboa: *Teacher Transformations in Developing Computational Thinking : Gaming and Robotics Use in After-School Settings*, in Emerging Research, Practice, and Policy on Computational Thinking, ed. by P. J. Rich and C. B. Hodges, (2017), pp. 189–203 https://doi.org/10.1007/978-3-319-52691-1
12. J. A. Cooper and K. L. Gunckel: *Teacher Perspectives of Teaching Computational Thinking,* Baltimore, Maryland, (2019)
13. G. Csapó: *Placing Event-Action-Based Visual Programming in the Process of Computer Science Education*, Acta Polytechnica Hungarica, 16.2 (2019), 35–57 https://doi.org/10.12700/APH.16.2.2019.2.3
14. M. Csernoch, , P. Biró, J. Máth, and K. Abari: *Testing Algorithmic Skills in Traditional and Non-Traditional Programming Environments*, Informatics in Education, 14.2 (2015), 175–97, https://doi.org/10.15388/infedu.2015.11
15. P. Goodyear: *Educational Design and Networked Learning: Patterns, Pattern Languages and Design Practice*, Australasian Journal of Educational Technology, 21.1 (2005), 82–101,

https://doi.org/10.14742/ajet.1344
16. P. Goodyear and S. Retalis: *Learning, Technology and Design*, in Technology-Enhanced Learning: Design Patterns and Pattern Languages, ed. by Peter Goodyear and Symeon Retalis, 2nd edn, Rotterdam, Boston, Sense Publishers, (2010), pp. 1–27
17. M. Guzdial: *Education: Paving the Way for Computational Thinking*, Communications of the ACM, (2008), 25–27, https://doi.org/10.1145/1378704.1378713
18. E. Hunsaker: *Computational Thinking*, in The K-12 Educational Technology Handbook, ed. by A. Ottenbreit-Leftwich and R Kimmons, EdTech Books, (2020), pp. 1–16, https://edtechbooks.org/k12handbook/computational_thinking [Accessed on 20th July 2021]
19. Z. Jiang, E. B. Fernandez, and L. Cheng: *P2N : A Pedagogical Pattern for Teaching Computer Programming to Non-CS Majors*, in PLoP '11 Proceedings of the 18th Conference on Pattern Languages of Programs, (2011) https://doi.org/10.1145/2578903.2579163
20. D. Laurillard and M. Derntl: *Learner Centred Design - Overview*, in Practical Design Patterns for Teaching and Learning with Technology, Yishay Mor, Rotterdam, Boston, Sense Publishers, (2014), pp. 13–16 https://doi.org/10.1007/978-94-6209-530-4_2
21. E. Magnusson: *Pedagogical Patterns – a Method to Capture Best Practices in Teaching and Learning*, in Genombrottet Konferens, (2006) https://www.lth.se/fileadmin/lth/genombrottet/konferens2006/PedPatterns.pdf [Accessed on 18th July 2021]
22. M. L. Maher, , N. Dehbozorgi, M. Dorodchi, S. Macneil, and S. Diego: *Design Patterns for Active Learning*, in Faculty Experiences in Active Learning: A Collection of Strategies for Implementing Active Learning Across Disciplines, University of North Carolina Press, (2020)
23. Mehalik, and Schunn: *What Constitutes Good Design? A Review of Empirical Studies of Design Processes*, International Journal of Engineering Education, 22.3 (2006), 519–32
24. Y. Mor and N. Winters: *Design Approaches in Technology Enhanced Learning*, Interactive Learning Environments, 15.1 (2007), 61–75 https://doi.org/10.1080/10494820601044236
25. T. Palts, , and M. Pedaste: *A Model for Developing Computational Thinking Skills*, Informatics in Education, 19.1 (2020), 113–28 https://doi.org/10.15388/INFEDU.2020.06
26. W. J. Peng, E. McNess, S. Thomas, X. R. Wu, C. Zhang, J. Z. Li, and H. S. Tian: *Emerging Perceptions of Teacher Quality and Teacher Development in China*, International Journal of Educational Development, 34.1 (2014), 77–89 https://doi.org/10.1016/j.ijedudev.2013.04.005
27. R. Philip: *Finding Creative Processes in Learning Design Patterns*, Australasian Journal of Educational Technology, 34.2 (2018), 78–94 https://doi.org/10.14742/ajet.3787
28. A. P. Rehmat, E. Hoda and M. E. Cardella: *Instructional Strategies to Promote Computational Thinking for Young Learners*, Journal of Digital Learning in Teacher Education, 36.1 (2020), 46–62 https://doi.org/10.1080/21532974.2019.1693942
29. M. Resnick, , J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, E. Rosenbaum, J. Silver, B. Silverman and Y. Kafai: *Scratch: Programming for All*, Communications of the ACM, 52.11 (2009) https://doi.org/10.1145/1592761.1592779
30. C. C. Selby: *How Can the Teaching of Programming Be Used to Enhance Computational Thinking Skills?*, Doctoral Dissertation, University of Southampton, England, United Kingdom, (2014) https://www.researchgate.net/publication/299464834_How_can_the_teaching_of_programming_be_used_to_enhance_computational_thinking_skills [Accessed on 8th August 2021]

31. P. Sengupta, , J. S. Kinnebrew, S. Basu, G. Biswas, and D. Clark: *Integrating Computational Thinking with K-12 Science Education Using Agent- Based Computation : A Theoretical Framework*, Educ Inf Technol, 18 (2013), 351–80 https://doi.org/10.1007/s10639-012-9240-x
32. S. Sentance, and A. Csizmadia: *Computing in the Curriculum: Challenges and Strategies from a Teacher's Perspective*, Education and Information Technologies, 22.2 (2017), 469–95, https://doi.org/10.1007/s10639-016-9482-0
33. T. Stevens, Z. Aguirre-Munoz, G. Harris, R. Higgins, and X. Liu: *Middle Level Mathematics Teachers' Self-Efficacy Growth through Professional Development: Differences Based on Mathematical Background*, Australian Journal of Teacher Education, 38.4 (2013) https://doi.org/10.14221/ajte.2013v38n4.3
34. J. M. Wing: *Computational Thinking*, Communications of the ACM, 49(3), (2006), 33–35 https://doi.org/10.1109/vlhcc.2011.6070404
35. A. Yadav, C. Mayfield, N. Zhou, S. Hambrusch, and T. J. Korb: *Computational Thinking in Elementary and Secondary Teacher Education*, ACM Trans. Comput. Education, 14.1 (2014), 16 https://dl.acm.org/doi/10.1145/2576872. https://doi.org/10.1145/2576872

Authors

Loice Victorine ATIENO
Eötvös Loránd University
e-mail: atienomunira04@gmail.com

TURCSÁNYI-SZABÓ Márta
Eötvös Loránd University
e-mail: tszmarta@inf.elte.hu

About this document

Published in:

CENTRAL-EUROPEAN JOURNAL OF NEW TECHNOLOGIES IN RESEARCH, EDUCATION AND PRACTICE

Volume 4, Number 1. 2022.

ISSN: 2676-9425 (online)

DOI:

10.36427/CEJNTREP.4.1.2963

License

Learning SQL by practising on popular movie databases through an online platform

KIRÁLY Sándor, BALLA Tamás, KIRÁLY Roland

Abstract. The new Hungarian *National Base Curriculum* (NAT2020), unlike the previous NAT2012, makes knowledge of the SQL language mandatory for students in the advanced final examination, which poses new challenges for both students and their teachers. An educational portal, sqlsuli.hu, has been developed and launched in order to help students to learn SQL (*Structure Query Language*). The framework of this online learning platform has an extensive grader tool that helps students test their SQL commands without intervention from a teacher, thus providing a flexible learning experience. The created sample databases are based on *Harry Potter* and *Star Wars* data, which is expected to increase students' engagement and motivation for learning. The developed LMS (*Learning Management System*) offers the authors the ability to improve the courses based on the solved problems and the comments on the forums. The course is available in Hungarian language.

Keywords: SQL, Learning Management System, final examination, online platform

1. Introduction

SQL is a standard language for accessing and manipulating relational databases. For a software developer, the knowledge of SQL is as important as a high level programming language. According to the previous NAT (NAT2012) guidelines, graduate (final examination) students in computer sciences were not required to know SQL, since they were allowed to use Microsoft Access at both intermediate and advanced levels and its tool of QBE (*Query-By-Example*) grid. According to the new NAT (NAT2020), at the intermediate level, there is no database management in the practical exam, but at the advanced level, the QBE grid cannot be used, the final exam measures the practical use of the SQL language (especially the *select* statement).

Learning a programming language may be more difficult than learning SQL statements, meanwhile acquiring knowledge of SQL for students under the age of 18 is also time-consuming and not as easy as one might think. This is partly due to the nature of SQL, and the fact that it is fundamentally different from the other skills secondary school students learn during their course of study [15]. Several researchers have been involved in teaching SQL: Al-Shuaily and Renaud proposed applying SQL patterns [2], Mitrovic developed a *Knowledge-Based Teaching System* for SQL [9], Quer et. al developed a software tool, LearnSQL (*Learning Environment for Automatic Rating of Notions of SQL*), that allows the automatic and efficient e-learning and e-assessment of relational database skills [14]. Garner and Mariani developed a graphical user interface centred around the textual translation of a query which has the potential to improve the way in which users gain an understanding of SQL[6].

Although books and notes in Hungarian and online courses in English are available, it seems insufficient for most secondary school students. Educational games such as *SQL Island* or SQL *Murder Mystery* are also available [17],[18]. Portals such as Udemy.com, CodeAcademy, or tutorials on W3Schools or Tutorialspoint that teach SQL in an interactive way in English might be useful. In Hungary, this type of portal is not available, apart from our **sqlsuli.hu** portal, which was developed for people under the age of 18.

On our website, students can learn how to apply different SQL statements such as *Select*, *Update*, *Delete*, etc. The curriculum focuses on the *Select* statement since students primarily have to know how to apply this one in the final examination. Through 24 chapters, students can learn to use *Select* statement, beyond basic use, through *Join* statements to the nested *Select* statements. The statements of DDL (*Data Definition Language*) and DML (*Data Manipulation Language*) are also introduced to the users of the portal. DML statements (*Update*, *Delete from*, *Insert into* and *Insert into* with *Select*) are presented through four chapters with practice exercises. In a chapter, DDL statements (*Create table* and *Alter table*) are also presented with a practical exercise.

Although data modelling is not one of the requirements of the final examination in computer science, we explain to our students how the *Harry Potter* and *Star Wars* databases have been developed. Deletion and modification anomalies, as well as normal forms, are presented in the curriculum.

2. Learning and practising on this site

There are three attractive videos on the main page to make our site more engaging and one of them always appears randomly. After login, students can read a short description of SQL then they need to choose whether they wish to learn database planning or SQL. The material to be learned can be found on the left side of the screen. The exercise linked to the current topic is on the right side.

Students are asked to write the correct SQL statement in the panel on the right side. After the coding is completed, clicking on the *Send* (*Elküld*) button uploads the solution. If it is correct the next exercise that belongs to this unit appears, but if it is the last exercise, the next unit will be displayed. If the statement is not correct a warning message will be displayed. By clicking on *Help* (*Segítség*) button, the program displays a part of the solution for the student to be able to correct the mistake. If the student fails an exercise they can move on to another unit and come back later to try again. Students' attempts are stored in a database so teachers can help them with the solution.

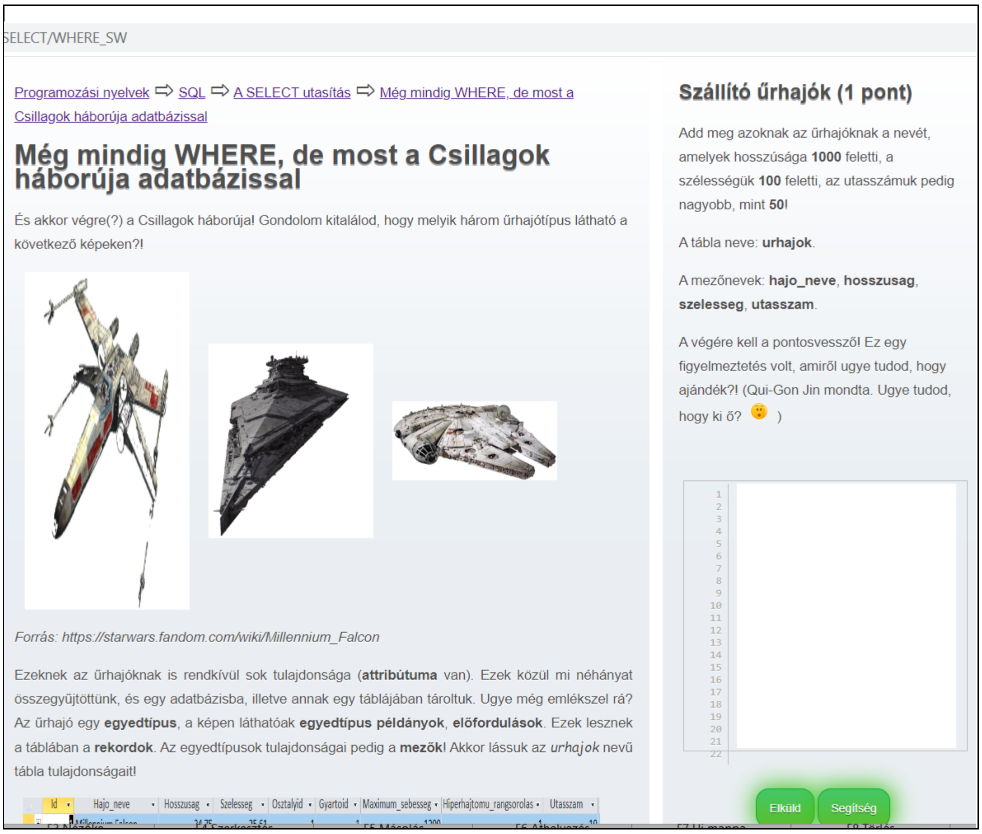


Figure 1: Learning and practicing on this site

The goal is to learn SQL statements (commands) first and then apply them in order to be able to solve database exercises. The curriculum mainly focuses on the *select* statement thus it explains its functions through twenty-five units. It is the most frequently used SQL statement and at the final exam in computer science, students need to apply mainly this one for solving different database tasks.

A consecutive sequence of lessons forms a chapter. We distinguish three types of lessons: **curriculum with assignments, independent assignments, and independent curriculum**. Independent teaching materials may contain blocks of text, audio-visual elements to support the learning process, and games may also be included here. Most of the lessons contain short tasks, in which case, in addition to the curriculum, a set of tasks appears on the right side to help students master the specific part of the lesson (see Figure 1). Most lessons contain more than two exercises. In these lessons, the structure and the content of the used tables are presented in the text.

After completing the required 24 lessons on *select* statement, students can get many interesting facts about *Hogwarts* and the *Galaxy*. After completing the required 24 lessons on select statement, students can get many interesting facts about Hogwarts and the Galaxy by solving exercises. For example, after submitting the correct statement, the name of the subject for which Hogwarts students were punished most frequently is displayed (see Figure 2). In these tasks, the tables that were used in the previous lessons are applied and all the information (the name of the tables, the exact name of the fields, the field names for setting the relationships between tables) for completing the exercises are available in the text of the task.

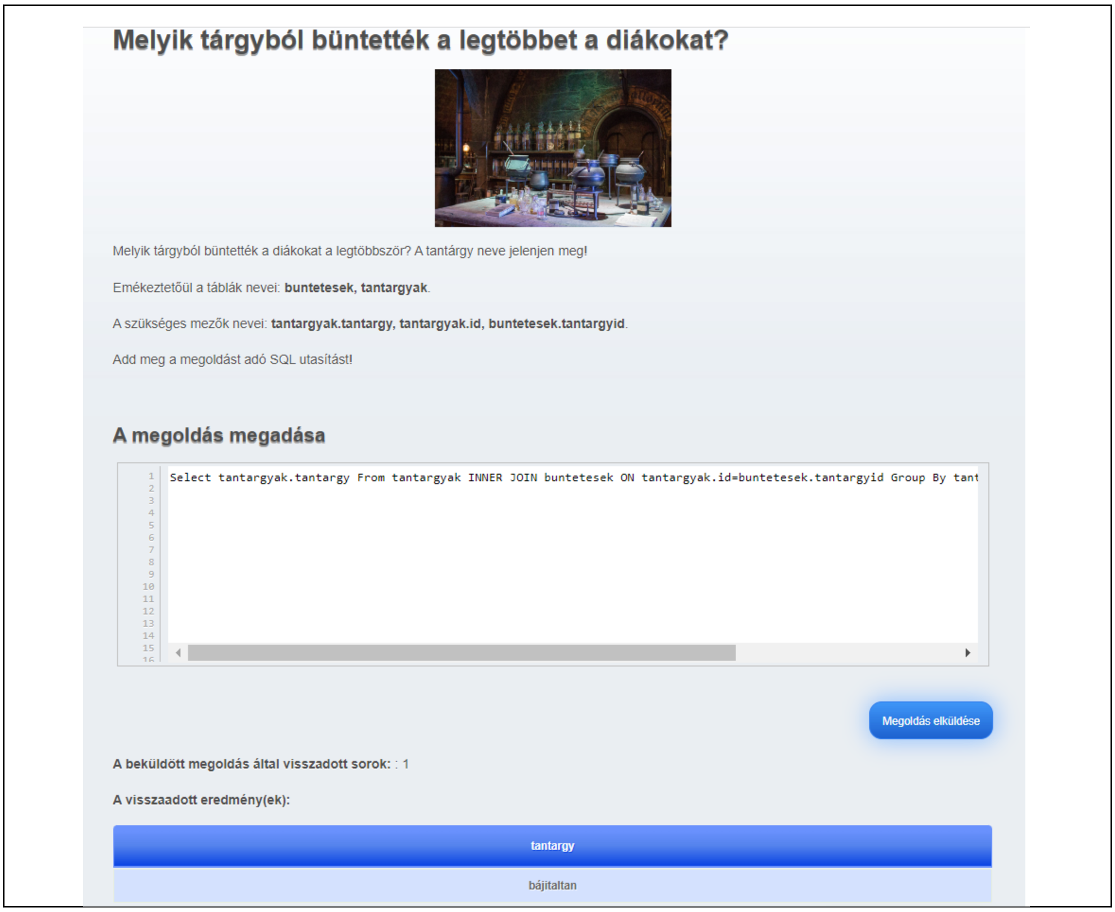


Figure 2: Independent assignment

To transfer the new knowledge the Harry Potter database is used, the tasks that check understanding also use this database. The Star War database is used to deepen the new knowledge students have learned.

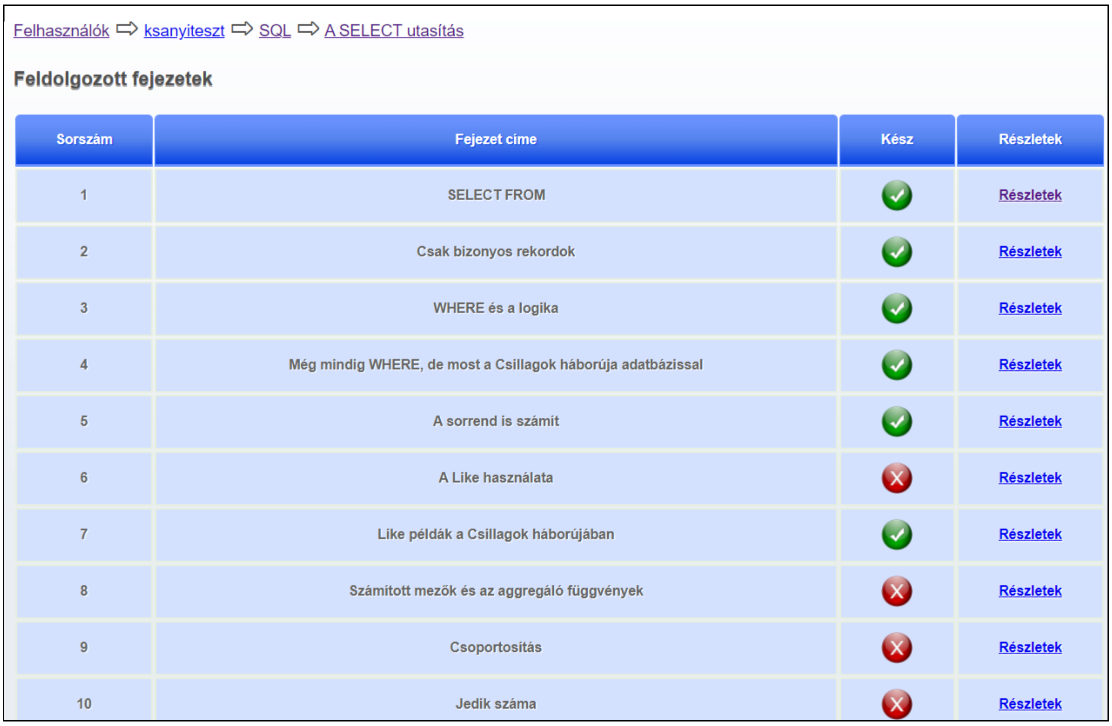
The portal contains a Forum for users to communicate with each other on a variety of subjects and to share their experience gained during coding, or they can collaborate with each other [1],[8]. By reading this forum, teachers can help students and improve the system.

3. The developed Learning Management System

The website has been written in PHP5 according to the MVC pattern. The curriculum, the SQL statements of the solutions of the exercises, the users' data and their actions are stored in a MySQL database. Currently, there are two types of users who can access the system; students and the administrator. Students can reach only the curriculum while the administrator has full access to both the materials and the users' data. Basically, after some modification, the LMS of kodolosuli.hu portal is used [12].

3.1. The admin interface

Users logged in as administrators can create new topics, new curriculums and new exercises or edit the previous ones and change the order of the units presented to students. It is possible to see the students' solutions, and this way the difficulty level of the exercises and the mistakes can be analysed as well. To check the students' progress, administrators have to choose the user's name then the name of the chapter (see Figure 3).



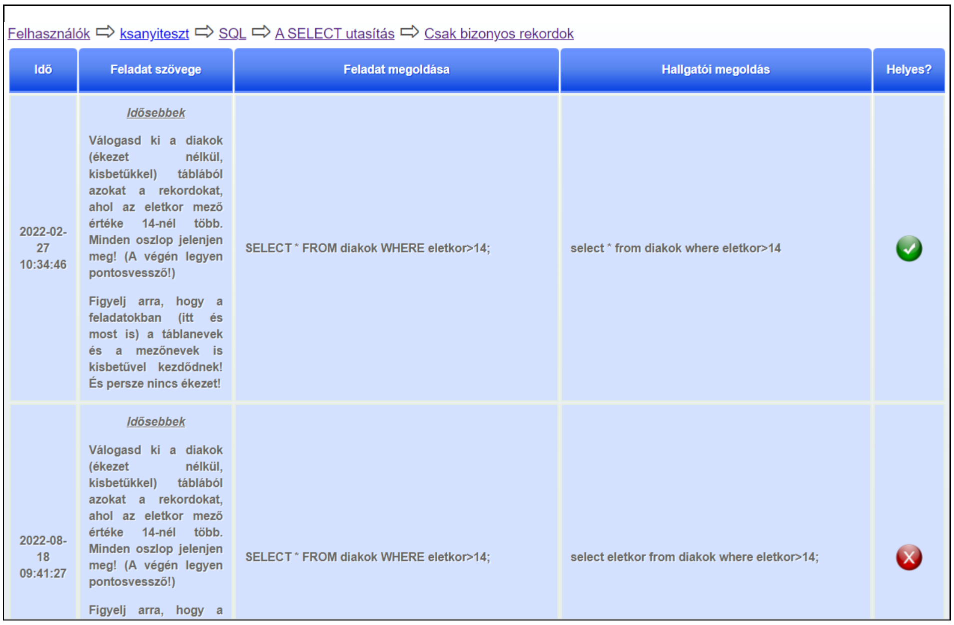
Felhasználók ⇨ ksanyiteszt ⇨ SQL ⇨ A SELECT utasítás

Feldolgozott fejezetek

Sorszám	Fejezet cime	Kész	Részletek
1	SELECT FROM	✓	Részletek
2	Csak bizonyos rekordok	✓	Részletek
3	WHERE és a logika	✓	Részletek
4	Még mindig WHERE, de most a Csillagok háborúja adatbázissal	✓	Részletek
5	A sorrend is számít	✓	Részletek
6	A Like használata	✗	Részletek
7	Like példák a Csillagok háborújában	✓	Részletek
8	Számított mezők és az aggregáló függvények	✗	Részletek
9	Csoportosítás	✗	Részletek
10	Jedik száma	✗	Részletek

Figure 3: Keeping track of users' progress

Our system also records how many times a unit was opened and how often a student tried to solve an exercise. It logs all the actions users perform so the publication of the achievements becomes easier, and it helps the developers improve the curriculum (see Figure 4).



Felhasználók ⇨ ksanyiteszt ⇨ SQL ⇨ A SELECT utasítás ⇨ Csak bizonyos rekordok

Idő	Feladat szövege	Feladat megoldása	Hallgatói megoldás	Helyes?
2022-02-27 10:34:46	Idősebbek Válogasd ki a diakok (ékezet nélkül, kisbetükkel) táblából azokat a rekordokat, ahol az eletkor mező értéke 14-nél több. Minden oszlop jelenjen meg! (A végén legyen pontosvessző!) Figyelj arra, hogy a feladatokban (itt és most is) a táblanevek és a mezőnevek is kisbetűvel kezdődnek! És persze nincs ékezet!	SELECT * FROM diakok WHERE eletkor>14;	select * from diakok where eletkor>14	✓
2022-08-18 09:41:27	Idősebbek Válogasd ki a diakok (ékezet nélkül, kisbetükkel) táblából azokat a rekordokat, ahol az eletkor mező értéke 14-nél több. Minden oszlop jelenjen meg! (A végén legyen pontosvessző!) Figyelj arra, hogy a	SELECT * FROM diakok WHERE eletkor>14;	select eletkor from diakok where eletkor>14;	✗

Figure 4: The log of a user's actions in a unit

3.2. The grader

The most important part of the system besides the curriculum is the grader. During the creation of a task, it is compulsory to define not only the text of the task, but the name of the database (*harry_potter* or *star_wars*) the task applies and the SQL statement representing the solution. After saving the task, the system checks the correctness of the SQL statement. If the MySQL engine fails to execute, the LMS returns the error message received from the engine.

When students upload their solutions by clicking on the *Send* (*Elküld*) button, first, the LMS passes the SQL statement to the MySQL engine to execute it. If it fails, the LMS sends back an error message to the user. If it is successful, the SQL statement representing the official solution will be executed as well. Then our LMS compares the results returned by the two SQL statements. If they are the same, the uploaded solution is considered correct. For these databases, users only have the right to *SELECT* which means they can read rows from the tables of the databases.

In the case of DML and DDL statements, a temporary database is used. When the user needs to create a table to solve an exercise, the CREATE TABLE statement will be replaced with CREATE TEMPORARY TABLE statement, thus, the new table is visible only within the current session, and is dropped automatically when the session is closed. In the case of INSERT, DELETE and UPDATE, our LMS makes the MySQL engine create the proper temporary table for the statement then the user's statement will be executed by the MySQL engine.

4. The curriculum

In an online environment, students' performance can be influenced by many factors that stem from students' individual characteristics and habits. These include the ability of students to maintain their intrinsic motivation to learn and their attention [3],[5],[10]. It is also feasible to influence these factors in an online environment since we have the opportunity to increase engagement, which can be defined as a student's cognitive process, active and emotional participation in the learning process [4],[13],[16]. All three factors of commitment (cognitive, behavioural and emotional) can be increased if the text of the curriculum is not only professionally correct, but also has a good style and personal tone [7]. The thought-provoking effect of the text can be facilitated by the inclusion of good metaphors, interesting examples, sometimes astonishing and provocative recordings, questions, contradictions, surprising and in many cases humorous twists [10],[11]. In *sqlsuli.hu*, directive e-learning is preferred since the material of this site is supposed to be new for the participants.

The text of our curriculum contains imaginary dialogues between the authors of the curriculum and the student, provocative questions and comments, and also interesting examples. The modified quotes from *Star Wars* and *Harry Potter*, such as "There are always two of the *Sith*, now the tasks too" or "Curiosity is not a sin but we should exercise caution with our curiosity but not in this portal" accompany the entire curriculum. We have inserted emoticons and pictures related to the movie in each unit. Videos have been created for a better understanding of *inner join* clause.

As it has been already mentioned, after students completed the units of the *select* statement chapter thus they are familiar with this statement they can gain interesting information about the *Wizard World* and the *Galaxy* by solving difficult exercises. These independent assignments specify the task and, in most cases, they contain the solution as an image. For example, "*Who are the students who shopped at Borgin & Burkes*". The solution is three Hogwarts students whose blurred faces can be seen in the assignment. If students enter the correct, statement they get the three names (see Figure 5).

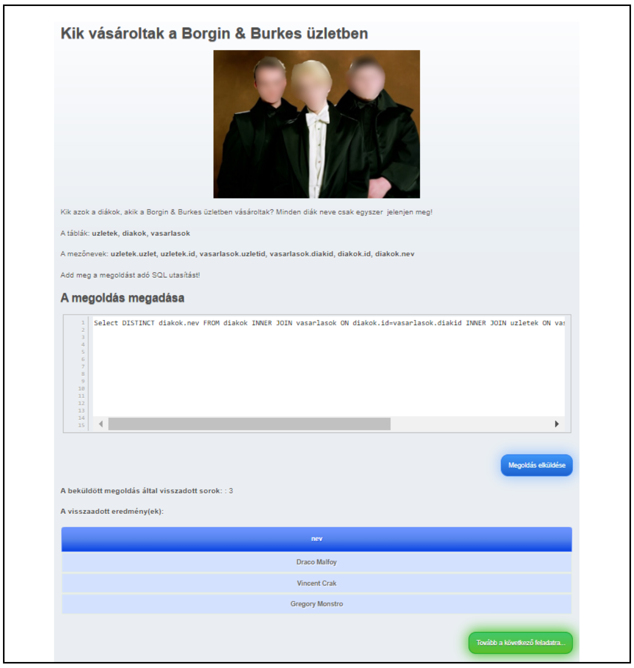


Figure 5: An independent task with blurred image and the solution

The registered users can also learn the DDL and DML SQL statements. To check the submitted SQL statements, the portal uses MySQL, but in the text, we draw the students' attention to the differences between the Oracle SQL and MS Access, such as TOP, LIMIT and ROWNUM, or the usage of apostrophes and quotation mark.

The portal describes the process of designing the *Harry Potter* and the *Star Wars* databases. Why the given tables are used, why we do not use only one table, and examples of how we eliminated the deletion and modification anomalies. Normal forms are also presented in these chapters.

To assess the knowledge of our students of database planning a very special test is used. They need to choose the correct answer from several answers by submitting a *select* statement. Each answer has a simple *select* statement. Our grader checks whether the *select* statement that belongs to the correct answer has been submitted. Thus, the completing these chapters is recommended for those who already know how to use the *select* command.

5. Conclusion and Future work

The aim of this study is to demonstrate a new website, *sqlsuli.hu,* an online interactive platform that offers a free SQL course. The goal is to reach secondary school students who wish to learn SQL and help them to be successful in the final exam in computer science. The website starts by presuming no prior knowledge at all, and lets students work through short exercises with gradually increased complexity. This website offers a SQL curriculum that uses *Harry Potter* and *Star Wars* databases. The developed LMS and Forum support keeping track of students` progress during lessons and foster the improvement of the material.

The site is open for anyone but used by the authors' students currently. We count on feedback from the authors' students, that makes it possible to correct any errors that may still exist in the portal, or make changes to lessons, wording and assignments. Then the portal will be promoted.

Development of educational games is also planned to provide a better understanding of the material.

Bibliography [1]

1. Ahmed Shamsul Arefin: *Pedagogy of Computer Programming: An Interactive and Collaborative Learning Approach*. Macquarie University Postgraduate Certificate of Higher Education(2015). EDCN 871 Final Project
2. H. Al-Shuaily, K. Renaud: *SQL Patterns: A New Approach For Teaching SQL*. 8th International Workshop. Teaching, LEarning and Assessment Of Databases. TLAD. 2010.
3. T. Balla, S. Király: *A discussion of developing a programming education portal*. Central-European Journal of New Technologies in Research, Education and Practice (2020): Volume 2, Number 2. DOI: https://doi.org/10.36427/CEJNTREP.2.2
4. R.C. Clark, R.E. Mayer: *E-learning and the science of instruction*. Pfeiffer, San Francisco, (2011) https://doi.org/10.1002/9781118255971

[1] The data as indicated in the source are to be followed. The main principle is to make the source reachable.

5. B. Faragó: *Tanulói aktivitás, aktív tanulás és tevékenység online környezetben*. In: Papp-Danka, Adrienn; Lévai, Dóra (szerk.) Interaktív oktatásinformatika, 2015.

6. Philip Garner and John A. Mariani: *Learning SQL in steps*. In: Journal on Systemics, Cybernetics and Informatics, Vol. 13, No. 4, 2015, p. 19-24.

7. K. Héjja-Nagy: *Tanulási stratégiák és a tanulói aktivitást befolyásoló egyéni feltételek online környezetben*. In: Papp-Danka, Adrienn; Lévai, Dóra (szerk.) Interaktív oktatásinformatika p. 33-49 (2015).

8. McDowell, C., Werner, L., Bullock, H., & Fernald, J: *The effects of pair-programming on performance in an introductory programming course*. SIGCSE '02 Proceedings of the 33rd SIGCSE technical symposium on Computer science education (2002): Pages 38-42. https://doi.org/10.1145/563340.563353

9. Antonija Mitrovic: *A Knowledge-Based Teaching System for SQL*. https://www.researchgate.net/publication/2425011_A_Knowledge-Based_Teaching_System_for_SQL. Accessed on 05 May 2022.

10. S. Király: *Tanulás támogatása digitális környezetben*. OKTATÁS-INFORMATIKA 2016: 1 pp. 29-40., 12 p. (2016)

11. S. Király: *How to Implement an E-learning Curriculum to Streamline Teaching Digital Image Processing*. ACTA DIDACTICA NAPOCENSIA 9: 2 pp. 13-22., 10 p. (2016)

12. S. Király, T. Balla, G. Kusper: *Code School: Learn to Code by Practicing*. MaCS 2016. Mathematics and Computer Science. Proceedings of the 11th Joint Conference on Mathematics and Computer Science. http://ceur-ws.org/Vol-2046/

13. N. Pellas: *The influence of computer self-efficacy, metacognitive self-regulation and self-esteem on student engagement in online learning programs*. Evidence from the virtual world of Second Life Computers in Human Behaviour, 35, 157-170, (2014). https://doi.org/10.1016/j.chb.2014.02.048

14. C. Quer, A. Abelló, X. Burgués, M.J. Casany, C. Martín, M.E. Rodríguez, O. Romero, T. Urpí: *E-Assessment of Relational Database Skills by Means of Learn SQL*. 9th International Conference on Education and New Learning Technologies, 2017. https://library.iated.org/view/QUER2017EAS https://doi.org/10.21125/edulearn.2017.0779

15. K. Renaud, J. van Biljon: *Teaching SQL - Which Pedagogical Horse for This Course?* Lecture Notes in Computer Science, 2004. DOI: 10.1007/978-3-540-27811-5_22

16. M. Wolf: *Learning to Think in a Digital World*. In: Bauerlein, M. (ed.): The digital divide: arguments for and against Facebook, Google, texting, and the ages of social network. Jeremy P. Tarcher/Penguin, New York. 34-37., (2007)

17. *SQL Island*. Game was Developed by Johannes Schildgen at the University of Technology, Kaiserslautern in the Heterogenious Information System Group. Available online: https://sql-island.informatik.uni-kl.de/ (accessed on 5 June 2022).

18. Knight Lab. *SQL Murder Mystery*. The Northwestern University. Available online: https://mystery.knightlab.com/ (accessed on 5 June 2022).

Authors

KIRÁLY Sándor
Eszterházy Károly Catholic University,
Institute of Mathematics and Informatics,
Hungary,
e-mail: kiraly.sandor@uni-eszterhazy.hu

BALLA Tamás
Eszterházy Károly Catholic University,
Institute of Mathematics and Informatics,
Hungary,
e-mail: balla.tamas@uni-eszterhazy.hu

KIRÁLY Roland
Eszterházy Károly Catholic University,
Institute of Mathematics and Informatics,
Hungary,
e-mail: kiraly.roland@uni-eszterhazy.hu

About this document

Published in:

CENTRAL-EUROPEAN JOURNAL OF NEW TECHNOLOGIES IN RESEARCH, EDUCATION AND PRACTICE

Volume 4, Number 1. 2022.

ISSN: 2676-9425 (online)

DOI:

10.36427/CEJNTREP.4.1.4465

License

Offline Situational Game for Teaching IT Systems

KOROM Szilárd, ILLÉS Zoltán

Abstract. There are still many offline methodologies and playable games out there for teaching various curriculum elements. These games help the students in understanding difficult IT concepts. This approach is called "Computer Science Unplugged" or *CSUnplugged* in short. As an example, think about sorting algorithms, binary numbers, data structures, or databases. In the following article, we would like to introduce a new situational game that focuses on software systems, and network communications by simulating multiple programmable devices, that communicate with each other in real-time. We believe that the game makes the students activities and helps them to understand critical and important terms in an interesting and playful way. The game is a simulation of a smart home system in a classroom environment and does not need any digital devices.

Keywords: CSUnplugged, without computers, smart home, informatics systems, network communication, offline game

1. Introduction

The essence of teaching computer science without computers is to give students a playful real-world experience of the important IT concepts and terms. The most important requirements against this methodology are that they should not need any digital devices and they should have as least prerequisites as possible. The goal is to help the students figure out an intuitive, playable solution for an IT problem.

On the main CS Unplugged webpage[1], there are many examples, translated into more than 20 languages. The "*CSUnplugged*" term was introduced in 1990 by Tim Bell, Mike Fellows, and Ian Witten in the first book about the topic [1]. Since then, hundreds of papers, and articles have been created for various students ages from primary [2] to high school [3] [4]. They cover a wide variety of topics within IT education. Sorting algorithms, for example, are not easy at all, but there are several very simple practices that can be played in the classroom [5]. For example, if we ask the students, they will likely be able to line up very quickly in terms of height. The actual algorithm may not be realized, but if they observe their own activity, the used methodology can be formalized. Most of the sorting algorithms can be played in reality, so the teacher is able to suggest and guide the students to line up in other ways.

Nowadays, these offline games have extensive literature. The concept can be applied to different subjects [6], but it is worth mentioning that not everything can be replaced, but certain parts of the curriculum can be substituted by these unplugged games. By the way, it has already happened with a few textbooks [7] [8] and systems that support coding learning, such as code.org [9]. The approach is not just popular but has been proven to be effective [10].

In this paper, we would like to introduce an offline situational game, that can be useful in the teaching of informatics systems. The topic is general, but the game is flexible, so the tone can be easily moved by the teacher based on a concrete curriculum.

[1] https://www.csunplugged.org/en/ (last viewed: 06/06/2022)

2. Description of the situational game

The situation is that we are in a classroom with the students, and we would like to build a smart home project, where the IT devices, controllers, sensors, transmission devices, gateways, and displays are all real students. For example, if we would like to measure the temperate at different points in the classroom, a student has to stand there "measuring" the temperature (actually guessing a number that is approximately realistic), and another student controls the lamp by standing next to the switch, another student monitoring the solution by using the whiteboard (writing the values on it). The aim of the game is to find the most efficient division and forms of communication as possible to operate the system. The key is how to modify the various "components" by expanding and complicating the system. As the task deepens and students invent new and new solutions, they engage new players as the game transforms into software architecture. Of course, the IT analogy is not (necessarily) formulated in the student's minds, but continuous reflection and teacher orientation can help in that. For example, the number of the needed programs or hardware can easily be defined by analyzing the roles (played by the students).

3. Technology background

First, we would like to define what kind of IT tools and knowledge are necessary to implement a solution for the problem above. The description is not exhaustive and is not detailed, as it is outside the scope of the article, but we would like to give a comprehensive picture.

3.1 Sensors and controllers

Sensors and controllers are low-level IT devices that can send or receive signals/data in the real world. Examples can be light, distance sensors or a simple LED, a switch, or even a motor. These are non-programmable target devices, that require a direct controller.

3.2 Local computing

These are the controllers of the previous layer, which are mainly suitable for M2M communication, i.e., they either provide some value while running continuously, or they act on a directed request (eg.: they turn on a lamp, send back data).

At this level, we can also talk about circuit solutions since the controller has to be connected to the peripherals somehow. Of course, "ready-to-use" devices are also available with GPIO (General Purpose Input and Output Connections). The most common of these, which can also be used for educational purposes, are the Ardunio and Raspberry Pi computers. The former can be used to create embedded systems, the latter to produce products very similar to the operation of a desktop computer.

3.3 Network and internet, data transmission

The most important question in this layer is what "communication" actually means. As for the IoT systems, the network protocols are the answers. Perhaps the most common are:

- MQTT or AMQP
- HTTP or REST

3.3.1 The MQTT and AMQP protocols

These protocols (or this type in more general) are very important from our paper's perspective because real-life communication is perhaps the most similar to this. The AMQP and MQTT are publish-subscribe-based protocols. The idea is that whoever forwards the message does not care who will receive the message. It is like having a discussion in a crowd. We say something very loud, and those who are listening will hear and understand the message, the others will ignore the report. The operating principle is that there is a central software (called a broker) that receives the published data and sends it to the entities who have subscribed to that particular type of message. This is very different from the request-response structure, as nobody requested the sent message, and the sender does not even know who will receive it (and whether the clients who wanted it actually received it). In general, this is very common in the multi-device world, including in smart home implementations. For example, a temperature sensor does not necessarily need to know who is curious about the data. The sensor transmits it to a central application (the smart home controller) from which users request data through a web interface or a mobile application.

3.4 IoT architectures in general

The general architecture of a real IoT solution is shown in Figure 1:

[11]

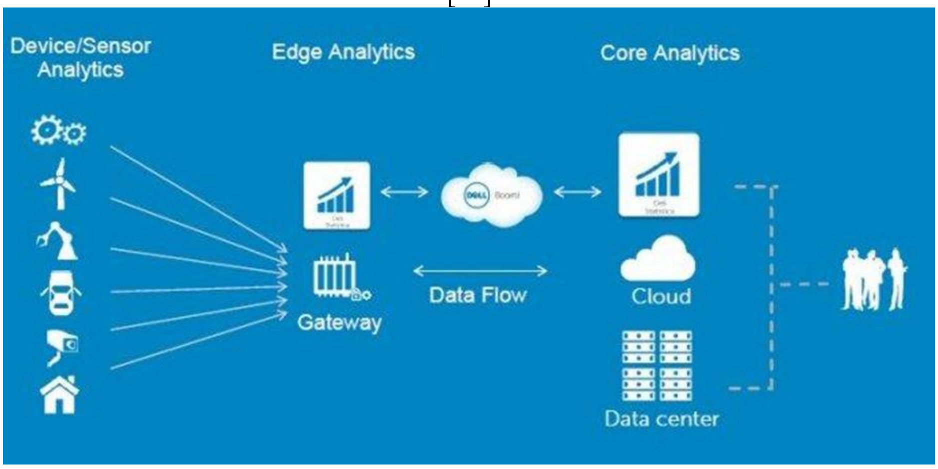


Figure 1: General concept of an IoT solution [11]

3.5 A concrete architecture

In the following, we would like to show what the architecture of our specific smart home solution might look like. We do not intend to fully detail them, as this is not the main focus of the article.

Of course, other solutions are conceivable. The examples are important because we actually want to play these with the students. And during the game, students are expected to design something like these, based on their own experiences.

3.5.1 Smart-home architecture without a server

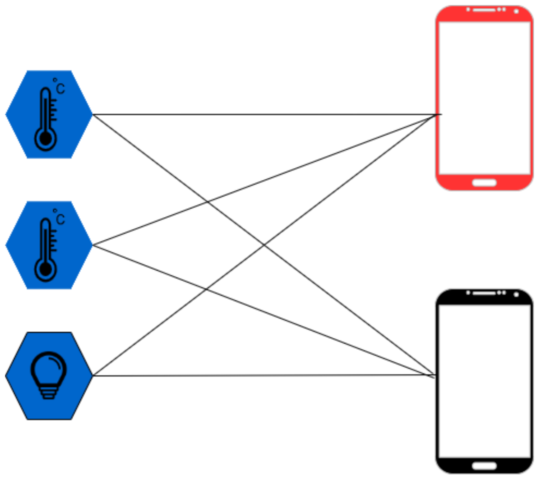


Figure 2: Serverless version of smart home architecture

In this case, we need to create a total of 3 applications:

- Temperature sensors are embedded devices, as they only measure and transmit data. A low-level program is running on them.
- Lighting is also an embedded device, as you only need to serve and handle a simple request. A low-level program is running on them.
- A mobile application that can map and communicate with devices available on the network. High-level programs are required. Graphical user interface needs (GUI) to be addressed here.

This is the most commonly used architecture in a normal user environment. The advantage is that the implementation is cheap and flexible, as it does not require the operation and maintenance of a server, and integration is easy. There are several tools on the market that are functional and usable just after downloading an application.

The biggest problem with the architecture is that the system is not integrated. It means we will need to download and use as many applications as many devices we have. Communication with devices is severely limited and we may run into various type of problems if we want to run many clients at the same time. If we are looking for a more complex solution, where the interoperability of devices and decision logic is necessary, the architecture is inapplicable (e.g.: we want an automatic temperature-based heating controller).

Another big disadvantage is that we do not have a central solution, that could care of the security issues, so all the individual components must handle their own security issues. One of the biggest problems in the IoT world is the lack of security handling [12].

3.5.3 Smart home architecture with a server

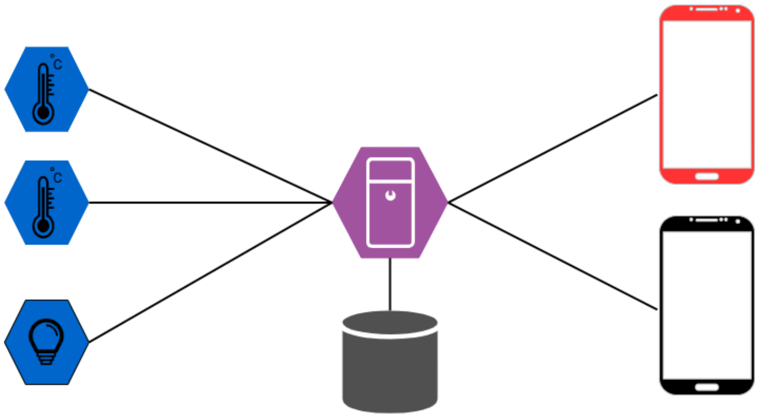


Figure 3: Server version of smart home architecture

In this case, we need to create 4 applications. The previous three as well plus a server. The server program is able to communicate with sensors and controllers, make decisions, serve and store data (for example, in a database), no appearance is required. The server program is by no means low-level, as it has to communicate across multiple threads and perform a variety of tasks.

The advantage is that with proper planning, it can be largely independent of who gets and gives data. The system is therefore scalable and easily expandable with new tools.

The system is more secure than the previous one since the sensors are completely hidden from the outside world, they are only communicating with the central program.

The downside is that it is both expensive and hard to maintain, as the tools are not independent of each other at all. Each component must be selected so that it is capable of being integrated with our own system. We also need a high available server computer.

4. Game description

The problem with playing the operation of IT systems is that a program, a real device, is very different from how we humans would actually act. For example, if we designate someone to measure the temperature at a certain point in a classroom, then write the result on the board, the student would probably solve this by guessing a number and then walking over to the whiteboard and writing it down. However, in the analogy of IT systems, there is a far simpler solution than creating a measurement software and an individual robot that can walk and draw. On the other hand, raising the issue is not useless, as it shows the difference between informatics and everyday thinking.

All in all, the main problem is that communication and perception are different from a machine's perspective. For example, we cannot instruct a student not to see or hear others but still be able to read particular messages.

The ultimate goal is to perform a system with students as actors that models the IT solution of a real smart home system.

4.1 Methodology

The planning must be done by the students and its complexity gradually deepens, layer by layer. The methodology is based on raising the problem and then guiding the students with questions. A possible set of questions is shown in Table 1.

Suggestions – guiding	Expected outcome	The game	IT analogy
What actors do we need if we want to measure the temperature in the 2 corners of the classroom, want to turn on the lamp, and want a user to be able to control this on the whiteboard as if the drawing were the controller application itself?	Someone is planning how to look at the application (draw it on the whiteboard). The thermometers are 1-1 students, someone is at the switch. There is a user who presses the drawn button on the board or yells if he/she needs data.	Everyone watches what the user on the whiteboard wants (presses a button, asks for something), and they respond to it individually.	Request-response protocols (e.g.: HTTP). There are several tools for different purposes. There is a network. GUI should also be designed. Peer-to-peer relationship.
The problem with the previous system is that the sensors/controllers can't actually see the button and can't actually talk.	A new student is assigned to control the application and forward the request to the controllers and draw when information is received.	When a user presses the button, only the controller student speaks and only with the selected "target device" and vice versa.	Server-client communication, but no database yet. Publish-subscribe-based communication.
We would like to monitor the temperature at real-time.	The sensor students keep saying out loud what they "measure".	The result is too chaotic, difficult to extract, and the drawing student cannot follow.	Transmission layer, gateway, queue. Continuous data transmission conforms to the MQTT / AMQP protocols.
Improve the system so that there is no noise or information clutter.	Data transfer students are assigned to transfer information between the sensors and the monitoring student.	The transmitters carry the information so there is no noise. If they get there at the same time, they have to line up.	Very strict architectural and communication rules for noise cancellation and efficient data transmission. Each component has a very well-defined task.
There should be more controllers and users (like for a real smart home project).	Students split the whiteboard in two and involve a new controller person and data transfer people.	Communication slows down because sensors and switches need to provide more data.	

Table 1: A possible course of the game

It is difficult to give specific guidance about the game because each of the suggestions depends on the solution the group gives to the previous one. For example, students may be given a fairly well-designed structure right from the start:

- There is a controller person
- There are data transmitters
- There is a drawing student
- Communication is disciplined and in accordance with the rules

Of course, even in this case, new components can be developed and added to the system. Another approach is to reverse the process to play a bad solution directly. Then the awareness and formalization of the problems can be the key

4.2 The tones

In its original form, the exercise focuses on network communication and software architectures. This can of course be modified. So, it's all about the questions the teacher asks, and the direction the teacher guides the students:

- **Algorithms**: what exactly is the "logic" (sensors, switches, controls, drawers) that each student must implement?
- **Hardware**: exactly what hardware would we need to actually do this? What are the requirements for the hardware, and what should they know?
- **Software system:** what software would we need to write to make this happen? How do we handle problems at the software level (data loss, corrupt data, backlogged requests)? What kind of implementations should be and used, what paradigms should be used (e.g.: console, web, mobile, windows)? Is there a need for an operating system on each hardware? If so, which one? If not, why not?
- **Data**: what data storage is needed? Who should do the storage and data management? What data packets should pass through the communication? What should these data packets look like?
- **Programming languages**: Which programming languages would be useful for the implementation? Which programming environments would be used to implement the different components? What programming paradigms could be used?

5. Real tests

5.1 Description of what happened

The game was tested with first-semester students of Eötvös Loránd University. The session was attended by 14 students and lasted 60 minutes. A questionnaire was also completed at the end of the session.

The game in this form is obviously not the most perfect, as the students already have a high school degree in computer science and are familiar with the concepts presented by the game. Nevertheless, the experience gained there was useful. Students were asked to participate both as "students" and as observing "teachers". Throughout the session, we constantly reflected on our own work and evaluated how useful the session was.

The game started with the presentation of the situation and the task. Without our intervention, the students roughly devised and implemented the following architecture (each circle corresponds to one person):

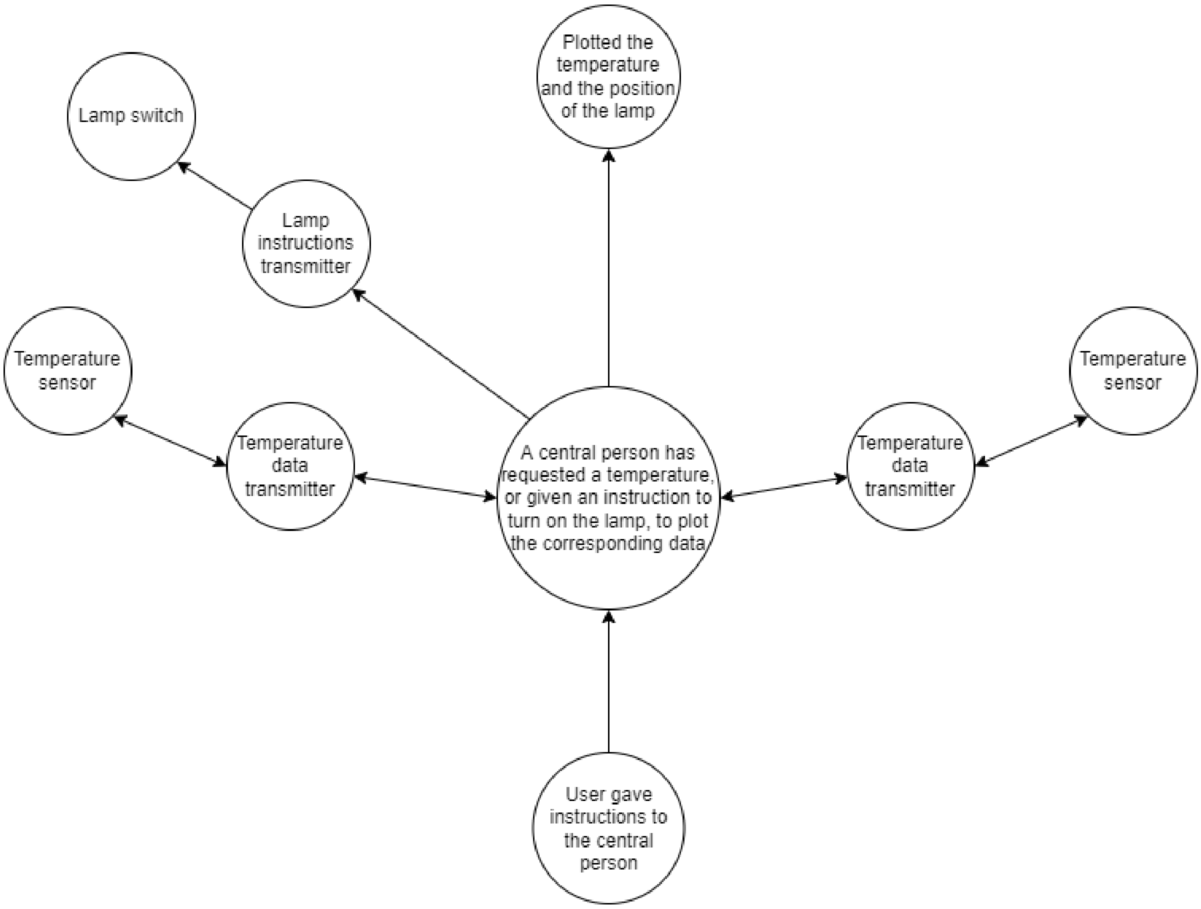


Figure 4: First architecture invented by students

From the planning to actually playing the project and considering it finished took about 20 minutes. During this time, there was a lot of brainstorming, discussion, and refinement, followed by the actual play. Obviously, because of the age and the situation, they were very disciplined and focused, which is not expected in a primary or secondary school, for example.

We had very few roles during the session. When they were ready with 1-1 phase, we gave them the next development opportunity or lesson to learn together (Was it effective this way? Could it be done differently? etc.) What happened in concrete terms is shown in Table 2.

Suggestions – guiding	Actual output	Lessons learned	IT analogy
Enter a new user in the game.	An interesting phenomenon compared to the previous ones was that when the two users asked for the light to be switched on at almost the same time, the central person had to wait until the corresponding data transmitting person returned.	During the game, they experienced this as a problem, because in reality, long seconds passed here, but we discussed that of course we are slower than the way the data actually goes, but this kind of race conditions happen with real software as well. The problem is needed to be solved.	HTTP (we compared it mainly to browsing); server-client connection; we would have to write 4 programs if we wanted to actually implement it.
Enter a second plotter/visualizer.	Here the central man was already overloaded, he couldn't do all his tasks because he would have had to communicate with too many people at the same time.	The two drawings cannot be done by one person, even if they have to draw the same thing. It was suggested among the students that the central man should be taken out of the system.	Queue, display completely separated from the server.
Let's solve it without a human transmitter (by speaking out loud).	Many people spoke at the same time, but the system worked through refinements.	Students perceived that the problem was communication. After 1-2 real-life tests, very strict rules were made about who could speak when, and everyone was given a different tone of voice to distinguish who was speaking. Afterward, we learned the lesson that even in real software you have to somehow strongly separate the messages, you have to send an identifier, so we know where the temperature came from.	Network protocols; data packet standards (data description languages); publish-subscribe protocol.
See the temperatures on the board continuously without being asked.	The two temperature-sensing men were saying the data out loud at the same time. A separate person had to be called in to record only the temperatures continuously.	Following the previous rules, the data can be separated.	Multithreading.
Let's solve it with two central people.	We couldn't actually play it, because we didn't have as many people as we needed.	It unnecessarily increases the number of communications and responsibilities.	-
Let's solve it without a central person.	It has been implemented, leaving many students without an assignment.	In this case, the publish-subscribe mode is more useful than the request-response model. The system has become more flexible.	-

Table 2: What happened during the live test

Overall, we think the session went well and delivered the expected results. The problems encountered were mainly architectural or network communication problems, and all of them can be analogised to an IT concept. The students (although not consciously) designed software architectures, faced critical and common problems and provided solutions.

Last but not least, the session was a lot of fun. The students had to work together, they had to find solutions together, and they could only do this by working well together. And the situations were often funny.

5.2 Answers to the questionnaire

In the followings, we would like to present the answers for the questionnaire.

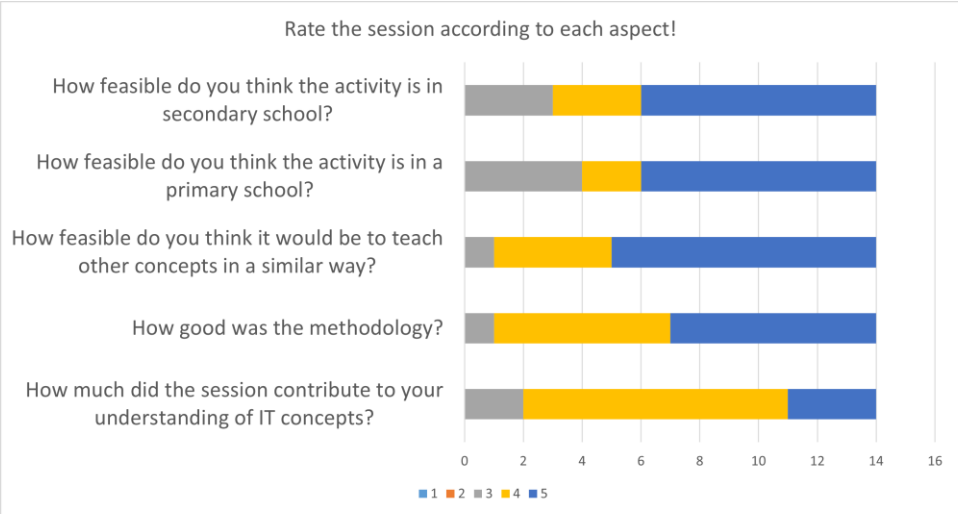


Figure 5: Rate the session according to each aspect!!

According to the graph in Figure 5, the students in the teaching profession rated the session as good or excellent. We welcome the confirmation that the method is worth trying in primary and secondary schools. Unfortunately, we missed this, but we should definitely do it at several levels and in several groups for the sake of completeness.

Although a 4 is good, you can't get past the results of the last question, where only three people answered with a 5. It could be because they were already familiar with most of the concepts, but of course, it could also be that they didn't think this was the most effective way of understanding. We can agree with that. We don't think the methodology is the most effective, but it is the most straightforward. Teaching software architectures or network communications in primary schools is far from trivial and is rarely implemented. The methodology offers a new alternative to this problem.

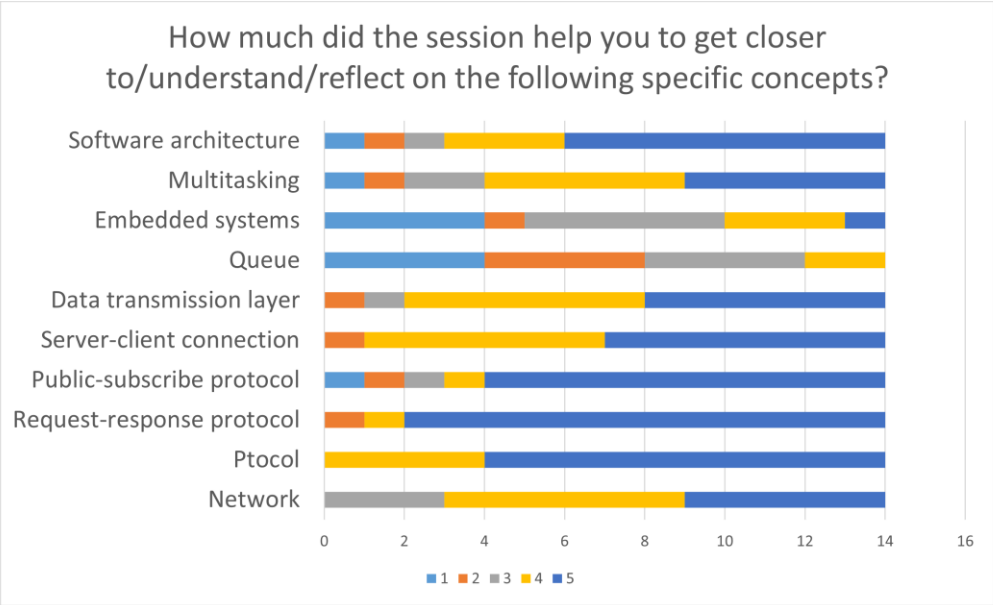


Figure 6: How much did the session help you to get closer to/understand/reflect on the following specific concepts?

Based on the results in Figure 6, it can be stated that students generally believe that the session contributed to their understanding of various IT concepts. Questions about network protocols were particularly well answered, which is important because this is what we wanted to focus on during the session. The notion of "queue" was left at the bottom, which is understandable because it was only just mentioned, and not particularly discussed.

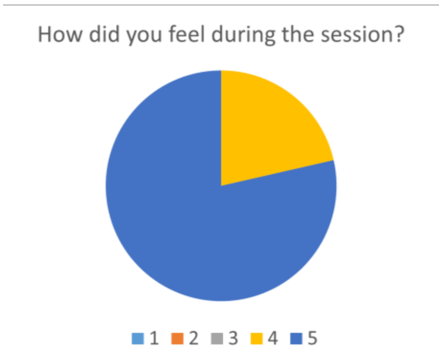


Figure 7: How did you feel during the session?

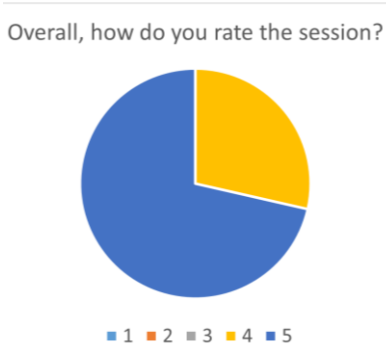


Figure 8: Overall, how do you rate the session?

Figures 7 and 8 show that, overall, students enjoyed the session and found it useful.

6. Conclusion

IT thinking can be taught and demonstrated without a computer. In this article, we present a situational game that can be used with primary and secondary school students, focusing on the design of software systems and network communication. The exercise can be valuable because these concepts are very difficult to demonstrate.

The method includes possibilities for modification (some of which we have presented), so that the teacher can adapt the game to a particular concept, age group and level of prior learning.

Bibliography

1. T. C. Bell, I. H. Witten, M. Fellow: *Computer Science Unplugged . . . off-line activities and games for all ages* (1998)
2. C. Duncan and T. Bell: *A Pilot Computer Science and Programming Course for Primary School Students*, in Proceedings of the Workshop in Primary and Secondary Computing Education, London, United Kingdom (2015) https://doi.org/10.1145/2818314.2818328
3. Y. Feaster, L. Segars, S. K. Wahba and J. O. Hallstrom: *Teaching CS Unplugged in the High School (with Limited Success),* in Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education, Darmstadt, Germany (2011) https://doi.org/10.1145/1999747.1999817
4. Á. N. Erdősné: *Teaching Graphs for Contestants in Lower-Secondary-School-Age*, OLYMPIADS IN INFORMATICS, vol. 11 (2017) 41-53 http://doi.org/10.15388/ioi.2017.04
5. P. Bernát: *The Methods and Goals of Teaching Sorting Algorithms in Public Education*, Acta Didactica Napocensia, vol. 7 (2014), 1-10
6. T. Bell, F. Rosamond and N. Casey: *Computer Science Unplugged and Related Projects in Math and Computer Science Popularization*, in The Multivariate Algorithmic Revolution and Beyond: Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday, H. L. Bodlaender, R. Downey, F. V. Fomin and D. Marx, Eds., Berlin, Springer Berlin Heidelberg (2012) 398-456 https://doi.org/10.1007/978-3-642-30891-8_18
7. B. Clarke: *Computer Science Teacher : Insight into the computing classroom*, Swindon, United Kingdom: BCS, The Chartered Institute for IT (2017)
8. O. Hazzan, T. Lapidot and N. Ragonis: *Guide to Teaching Computer Science*, London: Springer London (2014) https://doi.org/10.1007/978-1-4471-6630-6
9. *Lesson 4: Dance Party: Unplugged*, Code.org, https://curriculum.code.org/hoc/unplugged/4/ [Accessed 01/06/2022].
10. T. Bell and J. Vahrenhold: *CS Unplugged---How Is It Used, and Does It Work?*, in Adventures Between Lower Bounds and Higher Altitudes, H. Böckenhauer and W. Unger, Eds., Springer International Publishing (2018) 497-521 https://doi.org/10.1007/978-3-319-98355-4_29
11. V. Zhang: *TechTarget*, https://www.techtarget.com/iotagenda/blog/IoT-Agenda/The-pros-and-cons-of-IoT-edge-analytics . [Accessed 06 06 2022].
12. M. b. Mohamad Noor and W. H. Hassan: *Current research on Internet of Things (IoT) security: A survey,* Computer Networks, vol. 148 (2019) 283-294 https://doi.org/10.1016/j.comnet.2018.11.025

Authors

KOROM Szilárd
Eötvös Loránd University, Faculty of Informatics, Department of Media and Educational Informatics, Hungary,
e-mail: korom.szilard@inf.elte.hu

ILLÉS Zoltán
Eötvös Loránd University, Faculty of Informatics, Department of Media and Educational Informatics, Hungary,
e-mail: illes@inf.elte.hu

About this document

Published in:

CENTRAL-EUROPEAN JOURNAL OF NEW TECHNOLOGIES IN RESEARCH, EDUCATION AND PRACTICE

Volume 4, Number 1. 2022.

ISSN: 2676-9425 (online)

DOI:

10.36427/CEJNTREP.4.1.4452

License

Physical and Mental Stress in Work Activities

Ivana TUREKOVÁ, Alena HAŠKOVÁ

Abstract. Physical and mental stress are a part of each work activity and have, in some way, an impact on human organism. The impact, they have, can become evident either immediately or after a longer period time. In the paper the authors present results of a research which aim was to assess total physical load, caused by the muscular effort of upper and lower limbs, large muscle groups and local muscular effort, while working with burdens. To collect the necessary research data, the Holter analyzer was used on a research sample of two employees working with cleaning components in a production hall. Based on the achieved findings the author formulate conclusions, incorporation of which into the ergonomic and organizational measures can significantly reduce potential negative effects of the monitored stressors.

Keywords: physical stress, mental stress, local load, work with burdens, manual handling, ergonomic

1. Introduction

According to statistics of the National Health Information Center, a state-funded organization founded by the Ministry of Health of the Slovak Republic (http://www.nczisk.sk/en/Pages/default.aspx), upper limb diseases caused by long-term excessive and unilateral loading accounted for more than a half of the occupational diseases admitted. As regards the occupational diseases, mostly at risk were again workers aged between 50 to 59 years. At particular, in 2019 there were 347 newly diagnosed cases of occupational diseases and occupational poisonings (188 men and 159 women). Compared to the previous year, this was increase of 39 cases (1 man, 38 women). A long-term trend in the number of occupational diseases is presented in Figure 1.

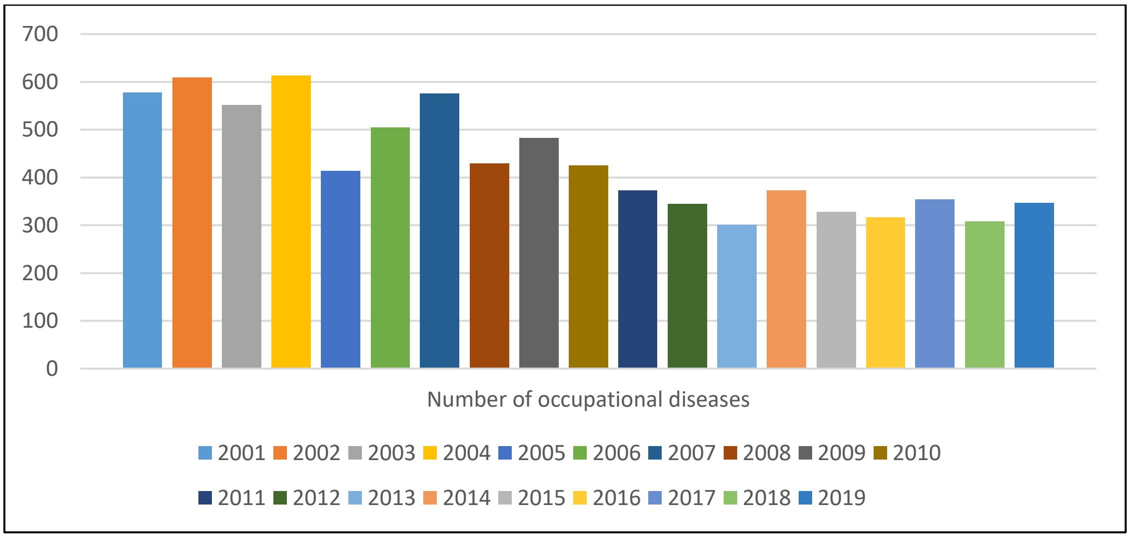


Figure 1: Development of the number of occupational diseases in the Slovak Republic from 2001 to 2019
Source: National Health Information Center

The most frequently reported occupational disease in 2019 was limb disease caused by a long-term, excessive, unilateral burden (181 employees, i.e. 52.2 % of all reported occupational diseases in the Slovak Republic). This was followed by diseases of bones, joints, muscles, blood vessels and limb nerves caused by working with vibrating instruments (62 employees, i.e. 17.9 %), infectious and parasitic diseases (42 employees, i.e. 12.1 %), hearing loss from noise (17 employees, i. e. 4.9 %) and skin diseases (12 employees, i.e. 3.5 %) [1].

Most occupational diseases have appeared in industry production, where people suffer mostly from pain in the upper limbs, as also infectious diseases. Mostly affected are workers aged between 50 - 59 year of old, with a slight increase in older age groups [2], [3].

2. Background of the issue

An individual worker's capacity to perform dynamic muscle work varies within very wide limits, and depends on his or her physical functional capacity, and particularly on cardiorespiratory capacity. Physical functional capacity in terms of cardiorespiratory (aerobic) capacity determined by maximal VO2, muscular performance (strength and endurance), and motor co-ordination (body control) is based on physiological mechanism: aerobic and anaerobic energy production, neuromuscular functions, and the regulation of body temperature. Anthropometric characteristics can be regarded as intervening factors in association with the output parameters of functional capacity. The utilization of different functional capacities is done by means of voluntary muscle contractions, which are impossible without an adequate level of motivation [4].

Effect of physical load on a human means to consider whether the physical activity during work (factors related to work facilities and workers` performance capacity) does not exceed physiological limits of workers and cannot induce health damage. While evaluating the worksite, it is necessary to focus on:

- space design and size of the worksite,
- used instruments and tools,
- working positions,
- manipulation with burdens and conditions for manipulation,
- position of controls,
- exerted force and frequency of use,
- total physical load,
- work and relax regime,
- shift rotation, etc. [5].

Space design and size of the worksite need to respect:

- anthropometric principles,
- physiological principles,
- psychophysiological principles,
- dimensions need to respect physiological dimension of the particular population, number of people on site, as well as security indicators, especially:
  - height of a manipulation plane,
  - space for lower limbs,
  - placement of controls and working tools respecting grasp distances and frequency of their use,
  - placement of working seats,
  - rationalization of working procedures, etc.

Working position depends on:

- character of performed work,
- dimensions and placement of a working site,
- quantity exerting a force to work,
- work intensity and precision.

Requirements for working movements are as follows:

- they must respect natural tracks and stereotypes (possibility of mutual adaptation, amplitude, force, speed, and rhythm),
- energy requirement must be proportional to the number and size of active muscle groups,
- to apply alternatively various muscle groups with a possibility to change to a working position with a small amount of static work,
- during activities requiring coordination of both limbs, the movements must be equally spread out on both limbs and their tracks must be analogical,
- if demands for the accuracy of movements increase, they must not be demanding for force exertion.

Physical work load means work load of motoric, cardiovascular and respiratory systems which is reflected in metabolism and thermoregulation. If there is a disproportion between the worker´s constitution and total muscular capacity and between the demand for physical fitness arising from the work tasks, the excessive straining of motoric apparatus appears [5], [6].

As to the muscular work there can be distinguished two forms:

- static muscular effort (isometric muscular contraction in which a tension increases) – more demanding,
- dynamic muscular effort (alternative application of muscular groups and alternation of the tension and release of muscles).

Assessment depends on the fact whether the work is executed by large muscles (representing more than 50 % of muscular mass) or by small muscular groups.

Manipulation with burdens means any rising, shifting, pulling, bearing, or moving burdens, with risk of health damage depending on various factors:

- characteristics of the burden (weight, storability, stability),
- required physical load (excessive weight, unsuitable working position),
- characteristics of working environment (insufficient manipulation space, micro-climatic environment),
- unsuitable regime of work and relax.

Local muscular stress is long-lasting, excessive and unilateral load on the same muscular groups, which leads to various diseases of bones, joints, muscles, tendons, ligaments, nerves, where the risk of health damage results from:

- intensive muscular force or multiple times repeated movements, especially in edge or unusual positions,
- other factors, such as the duration of force effect, spreading out of the used force in time, duration and organization of breaks, time for relaxation, etc., and
- the effect of other secondary factors, such as effect of vibrations, inconvenient micro-climatic conditions, bad conditions to grip the working tools, unsuitable personal working habits, insufficient training) [6], [7], [8].

3. Methodology of the research

Within the research, we carried out, assessment of physical load was done at the worksite for mending and cleaning of small metal components (electronic components) on the sample of two employees (women) working in the positions - cleaning components. The worksite in a production hall was naturally and artificially illuminated and naturally ventilated.

The basic working position was the sitting one with the possibility to alternate with a standing working position. Mostly small muscles of hands and forearms were exposed during work. The character of muscle work is static-dynamic with the dominance of a dynamic component. While working, the upper limbs of employees are bent in elbows in 90° angle.

During the cleaning of components, the worker can stand or sit on an adjustable seat. The worker cleans the components manually with cleaning agents. The used cleaning agents have Security Data Cards. After cleaning, the employee dries the component by compressed air. Two female employees who were selected for the assessment were 30-39 year old and they have performed the working operation for more than 5 years. Their body weight and body height were measured.

Measurement of *local muscular effort* was carried out by the Holter analyzer with the deviation of ± 3 %. For the assessment of testing results it was necessary to know the detailed time track of a working shift to calculate the time-weighted average whole-shift value of used muscle force. In the assessment the number of movements carried out during a working operation was taken into account, using video recordings of standard working operations to get accurate data.

Assessment of *total physical load* was based on whole shift measuring of heart frequency and determining the working energy output.

Assessment of *work with burdens* was carried out by direct observations, determining the weight of burdens, and by detailed analysis of photo-documents, video-recordings and working time tracks.

4. Research results

Basic micro-climatic conditions of a worksite were the following ones:

v_a (velocity of an air stream):	0.01 – 0.015 m.s-1
r_h (relative humidity):	32.0 %
t_a (temperature):	20.4 °C

Results of local muscular effort measurement

Permissible values of local muscular effort are determined in relation to the muscle forces and frequency of working movements. The particular parameters for assessment of local muscular effort were the following: average whole-shift force output by extensors and flexors of upper limbs, average whole shift number of working movements of hands, and maximum number of working operations with force output more than 60 % F_{max} at mostly dynamic work. Results of these measurements are presented in Table 1.

Assessed parameter for upper limb	Limb	Employee 1	Employee 2
Average whole-shift output force of extensors	Right	8.08 % F_{max}	11.25 % F_{max}
	Left	5.77 % F_{max}	10.14 % F_{max}
Average whole-shift output force of flexors	Right	6.25 % F_{max}	7.45 % F_{max}
	Left	7.01 % F_{max}	6.71 % F_{max}
Average whole-shift number of working movements of hands	Right	16870	18466
	Left	15979	17501
Working operations with the used force over 60 % F_{max} at dominantly dynamic work appeared up to*		2 x/shift	35 x/shift

Table 1: Results of assessment of relevant parameters of the local muscular effort

***Note:** The permissible number of working operations with the used force over 60 % F_{max} at dominantly dynamic work is up to 600 x/shift.*

Results of total physical load assessment

Permissible values of physiological indicators of working load, energy output and heart frequency are determined by the Directive 542/2007 on health protection against physical load at work, psychic working load, and sensory load at work and they depend on sex and age of the employee. Measured and calculated results are presented in Table 2 [9], [10], [11].

Assessed parameter	Units	Employee 1	Employee 2
Average whole-shift heart rate	(pules/min)	93	81
Increase of heart rate	(pules/min)	+11	+11
Average energy output/minute	($kJ.min^{-1}$)	4.62	4.62
Average whole-shift energy output	(MJ)	2.22	2.22
Whole-shift permissible energy output of women in age of 30-39 years	(MJ)	5.8	

Table 2: Heart rate and energy expenditure of the monitored employees

Assessment criteria of the whole-shift heart rate during work performed mostly by large muscle groups are presented in Table 3.

Age category	Assessment criteria of the whole-shift heart rate during work performed mostly by large muscle groups			
	Absolute values		Increase in heart rate over the starting value	
	A Average values	B Edge values	C Average values	D Edge values
30 – 39 years	106	115	29	32

Table 3: Values of whole-shift heart rate acc. to legislation regulations [13], [14]

Burdens

The weight of individual manipulated components did not exceed 5 kg. Only men are allowed to carry heavy burdens (over 15 kg). Measuring of work parameters related to manual manipulation with burdens did not show any exceeding of the given weight values.

5. Discussion

Measurement of local muscular effort did not prove any exceeding of average muscle forces in relation to the number of working movements per shift done by the assessed employees.

No exceeding of average muscle force in relation to the number of working movements per shift was determined for any of the assessed working operations. No exceeding of permissible values of energy outputs was proved for any of the assessed working operations. Assessment of work related to the manual handling of burdens did not show any exceeding of defined weight values.

In spite of these results (when all limit values are observed), it does not mean that the employer should be satisfied with the existing state. There are many ways how to improve working conditions of employees. Since majority of work at the assessed worksite was carried out in a sitting position, the next step should be assessment of ergonomic appropriateness of seats for employees, better organization of work, and continual education and motivation activities. The workload is also lowered by observing the principles of manipulation with materials and burdens, as well as by the introduction of the rotation of workers at worksites with different amount stress.

Pursuant to another legislation norm [14] the employer who did not rule out manual handling of burdens is obliged to ensure that the manipulation is as safe as possible for the employees, with the lowest level of the possible health risks. In order to get rid of or lower the effects of manual handling of burdens on employees` health, before its beginning the employer must

- assess the risk in the case of every kind of the manual handling of burdens,
- carry out respective measures, especially as regards physical strain, qualities of work environment, and requirements valid for the performed activity,
- ensure medical supervision, which is the assessment of health fitness of employees for the manual handling of burdens, taking into account individual risk factors and guide mass values.

A significant element in the operational system of OHS (Occupational Health and Safety) is the training and practicing of employees as regards appropriate handling of burdens as well as health hazards resulting from the inappropriate handling of burdens, which are usually documented in the Review of Hazards in Handling Burdens.

6. Conclusions

The measurements outcomes showed that as for the categorization of works regarding health hazards both assessed employees are proposed to be included into the works of the second category. The nature of muscle work was static dynamic, with the prevalence of the dynamic element.

Measurement of local muscular effort showed that work operations using force exceeding 60 % F_{max} as regards predominantly dynamic work are permissible max. 600 times per a shift, which was observed (obtained employees data reached 2 times/shift and 35 times per shift).

Similarly, permissible values of energy output for the assessed workers were not exceeded, since for none of them the shift permissible value reached the limit of 5.8 MJ.

Assessment of the particular works from the aspect of the manual handling of burdens did not show any exceeding of guide mass values since the mass of handled burdens did not exceed 5 kg.

One of the important indicators of the occurrence of serious health damages resulting from work is the occurrence of the reported occupational diseases, professional intoxications, and other work health damages. The seriousness of the problem of the high proportion of the mentioned professional diseases affecting musculoskeletal, vascular, and nervous systems of employees exposed at work to an excessive strain of upper limbs is underlined by the fact that they are on the increase. Damages of musculoskeletal system are at the forefront also in work inability of working population; therefore it is necessary to pay close attention to the assessment of local muscular effort as well as overall muscular effort as measures preventing undesirable work health damages.

Bibliography

1. SŠÚ: *Hlásenie zdavotného stavu SR* (2019). http://www.nczisk.sk/Statisticke-zistovania/Hlasenia_zdravotneho_stavu_obyvatelstva/Pages/default.aspx
2. Šolc, M., Mikloš, V. *Risk assessment in small construction companies*. Bezpečnosť práce v praxi, 1(2012), Iura Edition, spol. s.r.o., Bratislava (2012), 11-16
3. Janoušek, V.: *Co říkají data o pracovních úrazech při ruční manipulaci*. In: VII. ročník mezinárodní konference Bezpečnost a ochrana zdraví při práci, SPBI, Ostrava (2007)
4. Wilson, J. R., Corlett, N.: *Evaluation of human work*. Third edition, Edt.CRC Press-Taylor & Francis Group (2005) 1015 pp. https://doi.org/10.1201/9781420055948
5. Markulik, Š., Nagyová, A.: *Quality management system implementation in health service field*. Physiological Research, 59 (2010) 5, p. 40
6. Očkajová, A.: *Work environment and ergonomics*. Edt. UMB Banská Bystrica (2013), 402 pp.
7. Kordošová, M.: P*ersonal Protective Equipment*. Edt. Wolters Kluwer (2014), 306 pp.
8. Pačaiová, H., Sonay, J., Glatz, J.: *Safety risks and technical systems*. TU, SjF, Košice (2009), 246 pp.
9. Kapustová, M., Tureková, I., Balog, K.: *The Analysis of Complex Load of Machine Production Workers*. In: Advances in Physical Ergonomics and Safety. CRC Press-Taylor & Francis Group (2013), 203-212
11. Meerding, W. J., Ijzelenberg, W., Koopmanschap, M. A., Severens, J. L., Burdorf, A.: *Health problems lead to considerable productivity loss at work among workers with high physical load jobs*. Journal of clinical epidemiology, 58 (2005) 5, 517-523, https://doi.org/10.1016/j.jclinepi.2004.06.016
12. Makovická Osvaldová, L.: *Process safety managemen*t. Bezpečná práca, 1 (2014), 36-38
13. Vyhláška MZ SR č. 542/2007 Z.z. o podrobnostiach o ochrane zdravia pred fyzickou záťažou pri práci, pred psychickou pracovnou záťažou a senzorickou záťažou pri práci

14 NV SR č. 281/2006 Z.z. o minimálnych bezpečnostných a zdravotných požiadavkách pri ručnej manipulácii s bremenami

Authors

Ivana TUREKOVÁ
Katedra techniky a informačných technológií, Pedagogická fakulta, Univerzita Konštantína Filozofa v Nitre, Slovakia
e-mail: iturekova@ukf.sk

Alena HAŠKOVÁ
Katedra techniky a informačných technológií, Pedagogická fakulta, Univerzita Konštantína Filozofa v Nitre, Slovakia
e-mail: ahaskova@ukf.sk

About this document

Published in:

CENTRAL-EUROPEAN JOURNAL OF NEW TECHNOLOGIES IN RESEARCH, EDUCATION AND PRACTICE

Volume 4, Number 1. 2022.

ISSN: 2676-9425 (online)

DOI:

10.36427/CEJNTREP.4.1.2490

License