Session Chair: Péter Körtesi

10:50 – 11:10	Tamás Kádek **Automatic Task Evaluation System for Operating System Configuration Tasks**
11:10 – 11:30	Mária Csernoch, Piroska Bíró, Domician Máté **Detecting the Eight Wastes in Digital Text Management**
11:30 – 11:50	Ildikó Papp **Application of Design Thinking in Engineering Education**
11:50 – 12:10	Marianna Zichar **Case Study on Using Inclusive Design Thinking in 3D Modeling**
12:10 – 12:30	Ádám Gulácsi **Removing the Burden of Syntax: Developing Computational Thinking and Algorithmic Skills of STEM Students**

10:50 – 11:10	**Tamás Kádek** - Senior Lecturer, University of Debrecen, Faculty of Informatics, Department of Computer Science

Automatic Task Evaluation System for Operating System Configuration Tasks

The University of Debrecen's Faculty of Informatics has a long experience of self-developed automatic task evaluation systems. In 2011, we started developing ProgCont software, which was specifically designed to automatically evaluate solutions to programming problems based on the experience of several previous experimental software projects.

Initially, the software supported the organisation of programming competitions, providing an objective, automatic, and almost immediate evaluation of submitted source code. It soon became evident that the solution used in the competitions could also be used in everyday teaching, either for practice exercises or for the assessment of exam questions. That is why we have added several features to the system over the last decade. The challenges of the last half-decade, primarily the pandemic, have significantly increased the importance of software such as ProgCont.

Using our experience in this field, we have also taken advantage of the benefits of automatic online evaluation software in another area of IT. Five years ago, in the rush to respond to the pandemic, a rudimentary solution for checking virtual machine configuration was born and deployed to check IT Security and Operating Systems in practice. Because the method worked in this case, it has stayed with us until today, after the pandemic. Almost five years on, the disadvantages/drawbacks of this new software have become clear, and several new needs have arisen to extend its usability.

In this article, we would like to present a redesigned second-generation software based on the experience gained, which can be used to control and practice a wide range of virtual machine configuration tasks in several subjects of the IT faculty. We believe that the system design developed can serve as a model for other areas and can be effectively applied in the field of engineering education.

11:10 – 11:30	**Mária Csernoch** – Associate Professor, University of Debrecen, Faculty of Informatics, Department of Computer Science **Domicián Máté** – Professor, University of Debrecen, Faculty of Engineering, Department of Engineering Management and Enterprise

Detecting the Eight Wastes in Digital Text Management

According to Wings (2006), computational thinking (CT) skills, should be essential alongside Reading, wRiting, and aRithmetic (3Rs), are to build students' knowledge effectively in informatics and engineering. Are these skills sufficient to develop their algorithmic and application abilities and connect different scientific fields? Are teachers prepared to help develop CT skills of their primary and secondary students in general? Which specific CT skills do incoming students often lack? How can teachers be better prepared to improve their students' CT skills? What strategies can teachers use to develop their level of CT skills? These questions highlight additional challenges in education.

To answer these questions, we conducted a series of tests. In the testing process four atomic text modification tasks were carried out and the participants' activities were logged in a text and a video file. In this paper, we present a case study that looks at how effectively a teacher can modify both incorrect and correct digital texts based on the eight types of waste in lean methodology. We compare this case to one (sample) where only Value Added (VA) and Requested Non-Value Added (RNVA) processes were used. Most RNVA activities came from errors in the chosen digital document that could have been avoided with proper editing.

The comparison showed that for the first task, the selected case took seven times longer to complete than the sample. Notably, the sample task was completed without any errors, while the assessed case often struggled to do the same, showing significant defects, without completing the task. Overall, we found that the eight wastes of lean were present in the first task. The process also indicated that learning from mistakes reduces the number of non-value-added activities and the time needed.

We also found that well-edited texts are easier to work with and less likely to have errors. Our case study indicates that not all teachers have the necessary level of computational thinking skills to develop their students' CT skills effectively. As a result, students may start college in informatics and engineering without the required CT skills, which should be considered and taken into account when producing their curricula.

11:30 – 11:50	**Ildikó Papp** – Associate Professor, University of Debrecen, Faculty of Informatics, Department of Data Science and Visualization

Application of Design Thinking in Engineering Education

Design thinking is a human-centered approach in business that helps to understand users, their needs, frustrations and motivations. This design and problem-solving method focuses on customer-centricity and creativity, redefining problems and aiming to create innovative and effective solutions to problems and situations that arise. The roots of the methodology date back to the 1950s, but since the 2000s it has been used significantly more frequently to develop better and more successful solutions, new products and services than competitors.

Design thinking is a non-linear, iterative process that combines different stages of the design process in corporate practice, including empathy, problem-solving, prototyping, and testing. Certain elements of the methodology can be transferred to different areas and levels of education and the environment should be nurturing the change and innovative way of thinking, taking into consideration inclusivity. Inclusive design thinking is an extended version of the original methodology to solving problems of disadvantaged or underrepresented groups in our society. We have dealt this topic in the framework of some previous Erasmus projects, but the actual project called eduIDT is definitely dedicated to application of these methods into higher education. We also prepared a practical guide (book, templates and tutorial videos) to provide a tool to European university teachers of technically oriented subjects.

In this talk, I will present a short overview of Design thinking and Inclusive design thinking methods and how we can take steps to achieve an inclusive society by integrating the above methods into engineering and technical education.

11:50 – 12:10	**Marianna Zichar** – Associate Professor, University of Debrecen, Faculty of Informatics, Department of Data Science and Visualization

Case Study on Using Inclusive Design Thinking in 3D Modeling

Inclusive Design Thinking (IDT) is an enhanced methodology derived from design thinking. It was developed by a collaboration of seven European universities, a non-profit organization, and a company as part of a 33-month-long ERASMUS+ project called eduIDT. The main aim of this project was to incorporate the IDT methodology into technology-oriented higher education, focusing on problem-solving for underrepresented groups. This approach seeks to create inclusive and innovative solutions for society.
Inclusion, as a new aspect of design thinking, emphasizes understanding and respecting everyone, not just focusing on the development of products, services, processes, and culture. It ensures that even those who have been historically excluded can effectively use the new products and services designed by engineers.
By the mid-point of the project, a practical guide on IDT was developed, and several professors from the partner universities became acquainted with its principles. The consortium implemented the method through a workshop where the already trained educators collaborated with a group of students from partner universities over the course of a week to apply the methodology.
IDT is a complex process consisting of five phases, with various tools available for use in each phase to facilitate progression to the next. If time constraints prevent the application of the entire methodology, the Mini Sprint Exercise serves as an excellent introduction to the fundamentals of the method. I utilized this approach while teaching students 3D modeling, and now I would like to share my experience.

12:10 – 12:30	**Ádám Gulácsi** – PhD Student, University of Debrecen, Faculty of Informatics

Removing the Burden of Syntax: Developing Computational Thinking and Algorithmic Skills of STEM Students

In higher education, solving programming exercises with a high-level programming language is a commonly used approach for developing computational thinking and algorithmic skills. However, this method has its limitations: learning the syntax of a high-level programming language puts an extra cognitive load on students, preventing them from focusing on problem-solving. Furthermore, computational thinking is not limited to programming: STEM students can benefit much more from solving problems within their own discipline, in different environments. This practical article proposes a collection of unplugged, semi-unplugged and plugged-in alternatives which can be used to develop the computational thinking and algorithmic skills of students.