Information technology applications

ITA

1/2022

PANEURÓPSKA VYSOKÁ ŠKOLA
FAKULTA INFORMATIKY

PAN-EUROPEAN UNIVERSITY
Faculty of Informatics

VZDELÁVANIE VEDA VÝSKUM
EDUCATION SCIENCE RESEARCH

SSKI
SLOVENSKÁ SPOLOČNOSŤ PRE KYBERNETIKU A INFORMATIKU PRI SAV
SLOVAK SOCIETY FOR CYBERNETICS AND INFORMATICS

International Journal of Information Technology Applications (ITA)

Volume 11, Number 1, July 2022

AIMS AND SCOPE OF ITA

The primary aim of the International Journal of Information Technology Applications (ITA) is to publish high-quality papers of new development and trends, novel techniques, approaches and innovative methodologies of information technology applications in the broad areas. The International Journal of ITA is published twice a year. Each paper is refereed by two international reviewers. Accepted papers will be available online with no publication fee for authors. The International Journal of ITA is being prepared for the bibliographic scientific database Scopus.

International Journal of Information Technology Applications (ITA)

Instructions for authors

The International Journal of Information Technology Applications is welcoming contributions related with the journal´s scope. Scientific articles in the range approximately 10 standard pages are reviewed by two international reviewers. Reports up to 5 standard pages and information notices in range approximately 1 standard page are accepted after the decision of editorial board. Contributions should be submitted via e-mail to the editorial office. The language of contributions is English. Text design should preserve the layout of the template file, which may be downloaded from the webpage of journal. Contributions submitted to this journal are under the author´s copyright responsibility and they are supposed not being published in the past.

Deadlines of two standard issues per year

paper submission deadline	– end of May/end of October
review deadline	– continuous process
camera ready deadline	– end of June/end of November
release date	– Summer/Winter

Editorial office address

Faculty of Informatics, Pan-European University, Tematínska 10, 851 05 Bratislava, Slovakia
juraj.stefanovic@paneurouni.com

Published by

Pan-European University, Slovakia, http://www.paneurouni.com
Paneurópska vysoká škola, n.o., Tomášikova 20, 821 02 Bratislava, IČO 36 077 429
Civil Association EDUCATION-SCIENCE-RESEARCH, Slovakia, http://www.e-s-r.org
OZ VZDELÁVANIE -VEDA-VÝSKUM, Andrusovova 5, 851 01 Bratislava,
IČO 42 255 180
Slovak Society for Cybernetics and Informatics (SSKI) at the Slovak Academy of Sciences,
Slovenská spoločnosť pre kybernetiku a informatiku pri SAV (SSKI),
Ústav automobilovej mechatroniky, Fakulta elektrotechniky a informatiky STU,
Ilkovičova 3, 812 19 Bratislava 1, IČO 00 178 730, http://www.sski.sk

Electronic online version of journal

http://www.paneurouni.com/ITA
http://www.e-s-r.org

visit Archive and Instructions for authors:

Print

Multigrafika s.r.o., Rajecká 13, 821 07 Bratislava

Subscription

Contact the editorial office for details.
Older print issues are available until they are in stock.

ISSN: 2453-7497 (online)
ISSN: 1338-6468 (print version)

Registration No.: EV 4528/12

Contents

This issue is devoted to the event

International Conference 2022 Cybernetics & Informatics (K&I)
Visegrád, Hungary, September 11th -14th, 2022,
organised by Slovak Society of Cybernetics and Informatics,
under the auspices of Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava.

Selected papers from this event are re-printed in this issue under agreement of their authors.

Editorial

Editorial

Dear authors, dear readers,

this issue 1/2022 is dedicated to the 31th international conference 2022 Cybernetics & Informatics (K&I) which was held in September in Visegrád, Hungary, in hotel Silvanus. Alltogether 58 full papers were selected and invited for presentations, including their 124 authors. Six papers are re-printed in this issue as examples.

The conference focuses on presentation of latest development in control engineering, information technologies and related multidisciplinary fields in line with current development trends in Industrial Internet of Things. The conference provides a general forum for researchers, university teachers and users dealing with practical problems of control and ICT. Conference papers may range from theoretical works to engineering applications. Keynote speakers of the conference will be leading personalities in control engineering, ICT, AI and IIoT.

Conference is held every two years, the next one in 2024. Papers may range from theoretical works to engineering applications. Keynote speakers of the conference will be leading personalities in control engineering, ICT, AI and IIoT. All submitted papers will be peer-reviewed based on the full versions of manuscripts. A full version of the paper must be submitted even if the work will be presented as a poster. Papers were invited within the following fields:

- Methods and algorithms for modelling and control
- New information and communication systems with IoT techniques
- Embedded, distributed and networked control systems
- Cyber-physical systems, cloud computing, Big Data and extended reality
- Artificial Intelligence

Website of the last event: http://ki2022.sski.sk/

International Conference Cybernetics & Informatics 2022
Visegrád, Hungary, September 11th - September 14th, 2022

Application for Python Programming Language Education Developed by Unity Engine

Michal Hlavatý, Alena Kozáková and Oto Haffner
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava
Bratislava, Slovakia
Email: michal_hlavaty@stuba.sk

***Abstract*— Programming is not easy to learn or teach because different students need different explanations to understand such a complex concept. Traditional teaching methods and learning approaches are no longer appropriate for many kids. According to studies, such approaches impair students' motivation and interest in studying programming. According to research, games can help learn Python programming. The aim of this paper is to present our educational game that we developed in the Unity 3D game engine. The game focuses on pure basics of programming in Python language. Before and after completing the game the player has to fill out a survey that helped us analyze the project.**

Keywords—programming, Python, game, education

I. Introduction

The fast acceptance of gamification and the usage of game features in education has a wide range of implications on student outcomes and engagement. Gamification is defined as the intentional use of game design features in nongame contexts to promote desired behavior or solve problems. The concept of gamification has inspired a plethora of insightful studies attempting to comprehend its applicability in education and all aspects of life [1].

The quick development of technology always results in novel and exciting ways to engage students in learning and meet the expanding demands of education. However, traditional teaching strategies or even current software programs are consistently shown to be either ineffective or insufficient [2]–[4]. According to the National Research Council [5], science education is widely regarded as one of the most essential components of modern education because it is in charge of preparing citizens who are scientifically literate and encouraging essential 21st-century skills like adaptability or problem-solving. As a result, there has been a lot of interest in tools and methods that encourage scientific thinking as well as the educational theories that are incorporated into them [6]–[9].

One of the most well-known definitions of a game comes from game designer and producer Sid Meier, who says „a game is a series of interesting and meaningful choices made by the player in pursuit of a clear and compelling goal". A game is typically characterized as an activity aimed at achieving a specified state of things using only methods authorized by specific rules, where the means specified by the rules are more limited in scope than they would be in the absence of the rules, and where the primary purpose for accepting such limitation is to enable such activity [10].

Traditional teaching methods and learning approaches are no longer appropriate for many kids. According to studies, such approaches impair students' motivation and interest in studying programming. Several serious games or educational games have been created in an attempt to address these issues and promote the learning of computer programming [4], [11]–[13].

II. Python in Education

"Python is an interpreted, object-oriented, high-level, open-source programming language with dynamic sematic" [14]. It is widely used both in science and in R&D by large companies. This programming language is designed to be easy to learn which makes it an extremely good choice for beginners or elementary/high schools.

Python has various qualities that make it a suitable language for programming learning. According to [15], [16] its syntax is straightforward and objective; its dynamic typing determines variables automatically on execution time; and its interpreter offers quick feedback due to potential errors.

Python syntax's compactness allows to dedicate more energy to application and problem solving, significantly reducing development complexity and enhancing development efficiency. Python, being a lightweight-grammar programming language, has well adapted to the modern technology period making it suited for computer programming courses for logical syntax, often similar to English language. It also forces the programmer to write code neatly which is an important habit to learn early on.

Another advantage is versatility of this programming language. Python can be used for a range of tasks from web design to machine learning and is therefore excellent for education. This is because many young people do not know in what area they will be programming. Programming in Python is a great skill even for people who will never work as programmers. This is because of the scripts they can write making their daily work easier.

Open-source projects have huge amount of resources and materials for students and Python is no different. Official sites are written with great care and are often enough to solve most of a basic problems. The size of the Python community is also one of the assets. If a student has a problem, there is always someone who has encountered a similar problem or someone who can give advice.

The aforementioned advantages of Python are far from being all of them. So why mostly other languages are taught in pre-university environment and not Python? According to [17] it is because of Python is an interpreted and not compiled language which affects performance. In interpreted languages the code is easier to test or change, but those languages perform slower compared to their compiled counterparts. But the hardware is getting cheaper and faster on a daily basis and performance, in a smaller, educational scale, ceases to be a problem[18].

The aim of this work was to develop a game to be used for teaching purposes to explain fundamentals of programming in

978-1-6654-8775-7/22/$31.00 ©2022 IEEE

Python to absolute beginners. Programming is a difficult process to understand and many students may struggle with the basic principles, that are being taught in school by theoretical explanations. The game-like environment of this project should help these students to learn those basics with hands-on approach.

Most pupils in the traditional educational system sit and listen to a teacher's oral presentation. However, increasing student enthusiasm in studying through this strategy is frequently tough. During the sessions, most pupils grow preoccupied. To improve their learning motivation, a collection of board games was devised and used in after-class study. Students conducted chats with team members and learned from each other while playing the activities. As a result, pupils learned and taught one another. According to a research, board games can help you learn Python programming [19], [20].

When student learns by playing, he has an additional motivation, not only learning something new but the player also wants to know what new mechanics or puzzles the game introduces next. While playing the game, the player will also receive an instant feedback which can prevent learning bad coding habits and keeps the player captivated.

III. Content of the Educational Game

The main goal of the game is to explain and teach the players fundamental basics of programming in Python. Player can interact from first person view, with the 3D environment where information panels are placed.

The game starts with a basic explanation of **Boolean** and **variables**. The player needs to understand those principles, because the concept of *True* and *False* will be used throughout the whole game and without understanding of variables it is not possible to write a proper code. The second and third obstacle/exercise is about the **conditions** and the **if-else** statements which are used for decision making and directly correlate with the previous challenge. The if-else statement is the basic *control structure* and the player can learn this first step of how to control the program flow.

The fourth exercise is like a summary of the previous exercises and introduction of the **print** command to print messages or variables to a console or any output device.

In fifth exercise, the player will encounter the **list** data structure. It is crucial to understand this part because lists are most common ways to store data multiple times, so players can create basic databases. Lists are also used in last two exercises.

The sixth and seventh exercises are focused on the **for** loop and the **range** command. The loops are one of the most useful tools in programming and it is important to understand how to control them properly so that the loop stops exactly when programmer wants it.

IV. Structure of the Educationl Game

The player can move in the game world freely, however there is only one way through the exercises which means the game is divided in seven parts:

1. Boolean and variables
2. Conditions
3. If-else statement

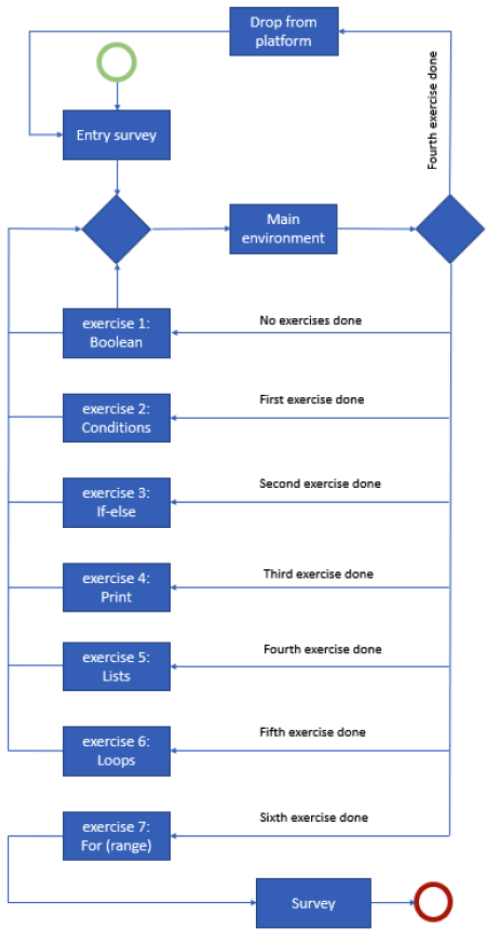


Fig. 1. Structure of the educational game

4. Print
5. Lists
6. Loops
7. For loop /with range

The game has only one level and at the beginning the player has only one little room to explore; however, the more exercises the player completes the bigger part of the level opens for him.

While designing the map of this game (Fig. 2) we tried to encourage the player to explore the level and reward his exploration by gaining access to more areas and to more abilities, e.g. jumping. After completing the fourth exercise the map shows a new area, but player cannot jump yet and it builds up expectations and desire to continue playing and learning.

To monitor the improvement of the players we created two surveys. The player takes one survey before the game and the other one after completing the game, so we can decide if this project serves the expected teaching purposes. The whole structure of the game can be seen in Fig. 1.

With every exercise, there is a table explaining the topic to the player. We tried to keep the explanation as brief and clear as possible. We want the player to understand a problem quickly, so that he needs not to read lot of text and can get to the entertaining part as soon as possible. An example of such an information table is in Fig.3 – it is one of the longest in the game.

Though, according to the feedback, some students prefer a more detailed explanation. We tried to cover these needs in the game too, by introducing the advanced help option while solving the exercise. Players can access the advanced help by clicking on the Python symbol (Fig. 4) during the exercise and they will be redirected to *w3school* documentation for the relevant command.

The exercises are in the form of puzzles (Fig. 4), where the player can drag and drop commands/numbers from the box to the code. After the check button is pressed, the code is evaluated by the Python interpreter. If the code works as intended, the exercise is considered complete and the level opens a new path for exploration. If the compilation of the exercise fails, the player has to try again.

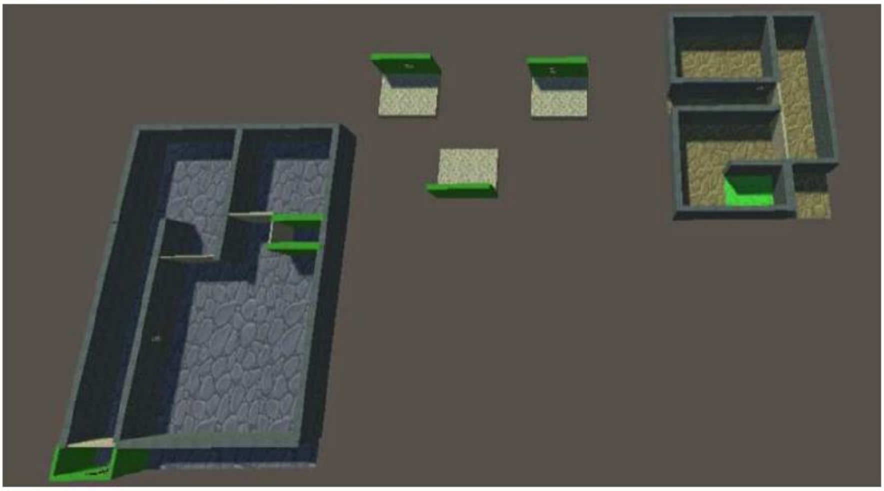


Fig. 2. Map of the level

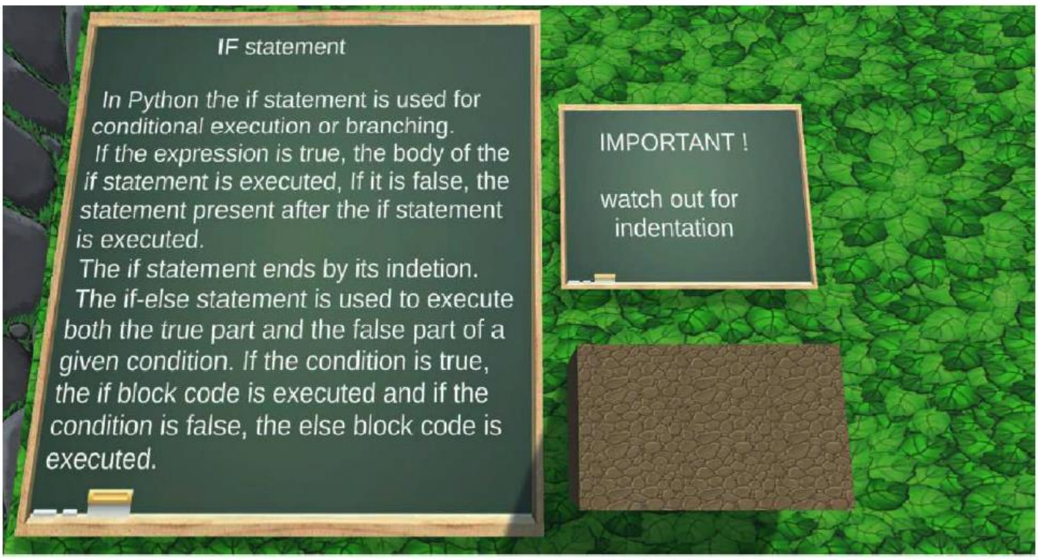


Fig. 3. Information table

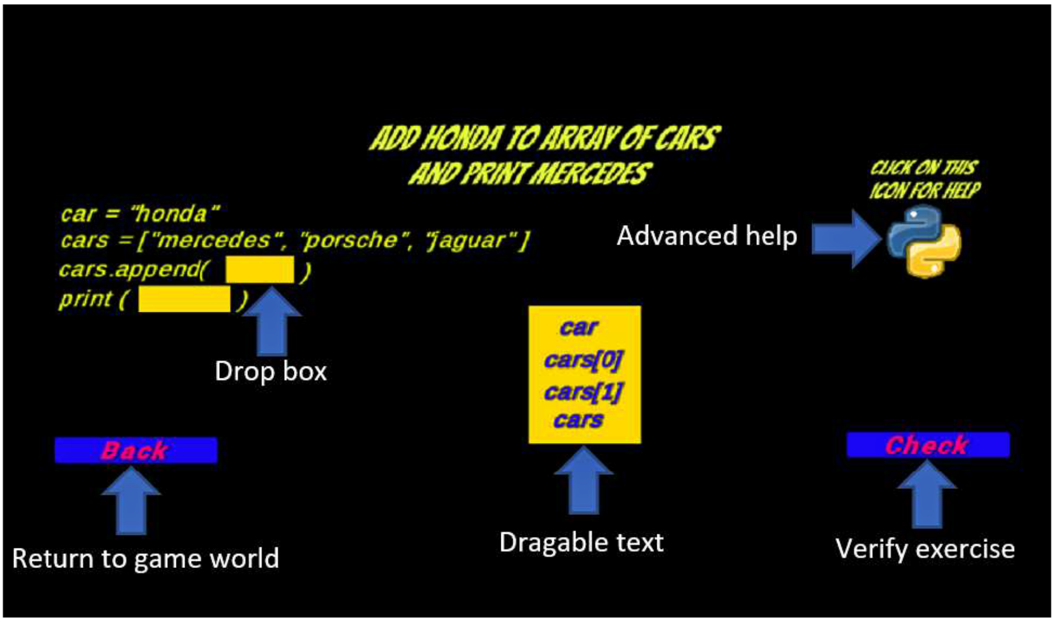


Fig. 4. Exercise example

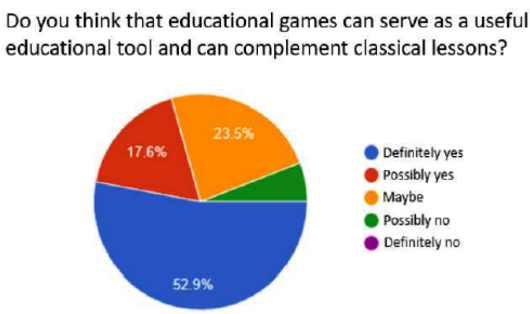


Fig. 5. Opinion of players on games in education before playing our game

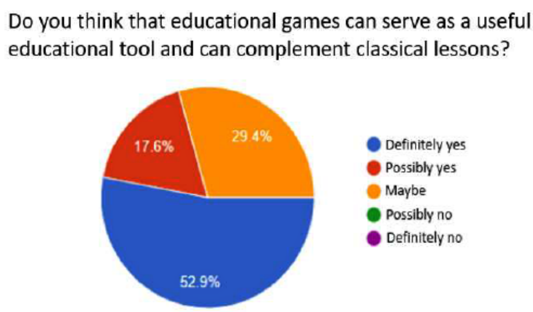


Fig. 6. Opinion of players on games in education after playing our game

V. FEEDBACK AND ANALYSIS

We tested our game directly in schools. Most of our participants were from high schools (70.06%), the others were from a university (29.04%). We were not able to fund any elementary school that would be willing to participate in the survey. The aim of the survey was to identify the skills of the participants before and after the playthrough by proposing some basic questions about programming in Python.

We also wanted to know how the players feel about educational games before and after their experience with our application. A comparison of users' opinion before and after they played the game is shown in Fig. 5 and Fig. 6. We managed to improve their opinion a little bit.

Another question was about programming skills (Fig.7 and Fig.8) and skills with Python (Fig. 9 and Fig. 10). Evidently, around a quarter of all players have evaluated that they improved their skills.

Next, we analyzed improvement of the participants in specific commands. In Fig. 11 and Fig 12 we can see assessment of improvements. Before the game, 35.3% of participants knew the right answer and after the game it was 68.8%.

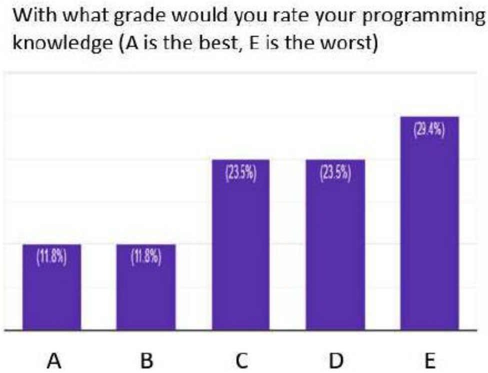


Fig. 7. Programming skill before the game

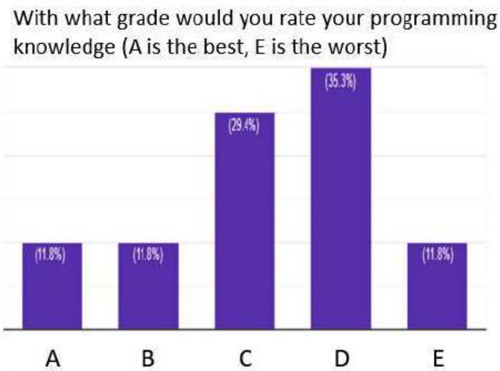


Fig. 8. Programming skill after the game

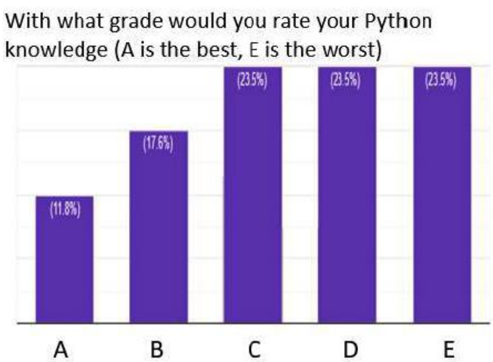


Fig. 9. Python knowledge before the game

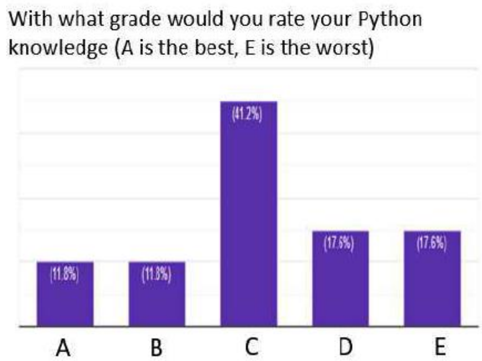


Fig. 10. Python knowledge after the game

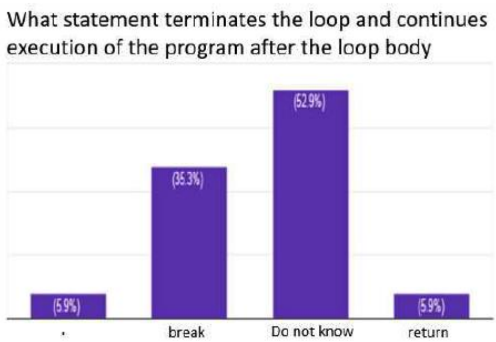


Fig. 11. Basic command knowledge before the game

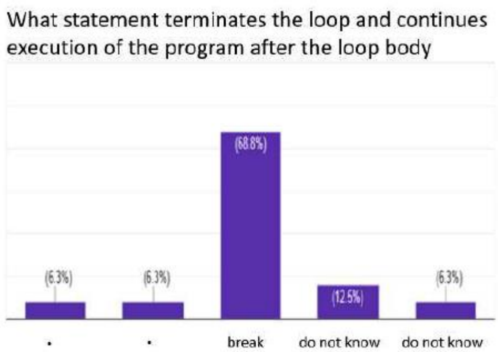


Fig. 12. Basic command knowledge after the game

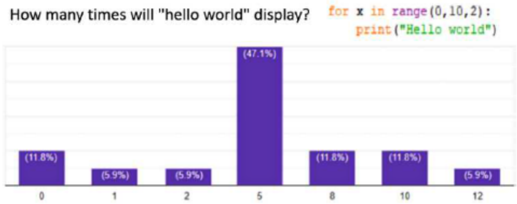


Fig. 13. Knowledge of range and for cycle

Fig. 13 shows the assessment of the answers to the last question, which was only presented in the exit survey, because we had many participants that never used any programming language, and we found this question too hard for such people. Only 47.1% answered correctly, it may be because the information needed to answer this question correctly was given as text table and students have not tried it on their own.

VI. Conclusion

The aim of this paper was to introduce interactive gaming environment for education to younger students and make it easier for them to learn Python. The Python is a great tool, and students are likely to encounter it in their professional or academic life. We created entry and exit surveys to help us analyze and evaluate our application. Based on our analysis, we can call our project a success. Players who played the game reported improvement in their programming skills and Python knowledge. Based on feedback we could improve graphics and make more use of special powers to make game more fun. Players also asked for mor content so in future, we can add more levels and explain another important subject related to programming like functions or object-oriented programming.

Acknowledgement

The paper was supported by the SRDA grant APVV-2021-0121 and by the Scientific Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic under the grant KEGA 016STU- /2020 . We would also like to thank to Cicka for helping with programming the implementation.

References

[1] K. Ofosu-Ampong, "The Shift to Gamification in Education: A Review on Dominant Issues," *J. Educ. Technol. Syst.*, vol. 49, no. 1, 2020, doi: 10.1177/0047239520917629.

[2] M. Baeten, K. Struyven, and F. Dochy, "Student-centred teaching methods: Can they optimise students' approaches to learning in professional higher education?," *Stud. Educ. Eval.*, vol. 39, no. 1, 2013, doi: 10.1016/j.stueduc.2012.11.001.

[3] E. Kucera, O. Haffner, and R. Leskovsky, "Interactive and virtual/mixed reality applications for mechatronics education developed in unity engine," in *Proceedings of the 29th International Conference on Cybernetics and Informatics, K and I 2018*, Apr. 2018, vol. 2018-January, pp. 1–5. doi: 10.1109/CYBERI.2018.8337533.

[4] E. Kučera, O. Haffner, R. Leskovský, J. Matišák, and P. Drahoš, "The Use of Modern Multimedia Trends for Popularization of Mechatronics by Presenting Its Present and Future Application," in *Communication Papers of the 2019 Federated Conference on Computer Science and Information Systems*, Sep. 2019, vol. 20, pp. 49–54. doi: 10.15439/2019f182.

[5] M. H. Rapporteur, *Exploring the Intersection of Science Education and 21st Century Skills: A Workshop Summary*. 2010.

[6] R. N. Landers, K. N. Bauer, R. C. Callan, and M. B. Armstrong, "Psychological theory and the gamification of learning," in *Gamification in Education and Business*, 2015. doi: 10.1007/978-3-319-10208-5_9.

[7] M. Kalogiannakis, S. Papadakis, and A. I. Zourmpakis, "Gamification in science education. A systematic review of the literature," *Educ. Sci.*, vol. 11, no. 1, 2021, doi: 10.3390/educsci11010022.

[8] M. Huba and Š. Kozák, "From e-Learning to Industry 4.0," in *ICETA 2016 - 14th IEEE International Conference on Emerging eLearning Technologies and Applications, Proceedings*, Dec. 2016, pp. 103–108. doi: 10.1109/ICETA.2016.7802083.

[9] S. Kozak, E. Ruzicky, J. Stefanovic, and F. Schindler, "Research and education for industry 4.0: Present development," in *Proceedings of the 29th International Conference on Cybernetics and Informatics, K and I 2018*, Apr. 2018, vol. 2018-January, pp. 1–8. doi: 10.1109/CYBERI.2018.8337556.

[10] S. Kim, K. Song, B. Lockee, and J. Burton, *Gamification in Learning and Education*. 2018. doi: 10.1007/978-3-319-47283-6.

[11] S. Papadakis and M. Kalogiannakis, "Using gamification for supporting an introductory programming course. The case of classcraft in a secondary education classroom," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 2018, vol. 229. doi: 10.1007/978-3-319-76908-0_35.

[12] M. A. Miljanovic and J. S. Bradbury, "A review of serious games for programming," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11243 LNCS. doi: 10.1007/978-3-030-02762-9_21.

[13] C. Kroustalli and S. Xinogalos, "Studying the effects of teaching programming to lower secondary school students with a serious game: a case study with Python and CodeCombat," *Educ. Inf. Technol.*, vol. 26, no. 5, 2021, doi: 10.1007/s10639-021-10596-y.

[14] Python Foundation, "About Python™ | Python.org," *About Python*, 2016.

[15] L. Grandell, M. Peltomäki, R. J. Back, and T. Salakoski, "Why complicate things? Introducing programming in high school using Python," *Conf. Res. Pract. Inf. Technol. Ser.*, vol. 52, 2006.

[16] Z. Xu, X. Shen, S. Lin, and F. Zhang, "Using Visualization to Teach an Introductory Programming Course with Python," in *Proceedings - 11th International Conference on Information Technology in Medicine and Education, ITME 2021*, 2021, pp. 514–518. doi: 10.1109/ITME53901.2021.00109.

[17] F. Georgatos, "How applicable is Python as first computer language for teaching programming in a pre-university educational environment, from a teacher's point of view?," Sep. 2008, [Online]. Available: http://arxiv.org/abs/0809.1437

[18] M. Pajpach, O. Haffner, E. Kučera, and P. Drahoš, "Low-Cost Education Kit for Teaching Basic Skills for Industry 4.0 Using Deep-Learning in Quality Control Tasks," *Electron.*, vol. 11, no. 2, Jan. 2022, doi: 10.3390/electronics11020230.

[19] L. W. Huang, L. W. Chen, and P. H. Cheng, "Board Game Design for Python Programming Education," 2019. doi: 10.1109/TALE48000.2019.9225988.

[20] O. Haffner, E. Kucera, D. Rosinova, A. Kozakova, P. Drahos, and M. Kocur, "Multimedia support for education of mechatronics," in *Proceedings of the 29th International Conference on Cybernetics and Informatics, K and I 2018*, Apr. 2018, vol. 2018-January, pp. 1–5. doi: 10.1109/CYBERI.2018.8337537.

Authors

Ing. Michal Hlavatý
Institute of Automotive Mechatronics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
michal.hlavatý@stuba.sk
He completed his master's degree in the field of automotive mechatronics in 2017. He is currently studying for the third year of doctoral studies dedicated to artificial intelligence and neural networks with a focus on reinforcement learning.

prof. Ing. Alena Kozáková, PhD.
Institute of Automotive Mechatronics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
alena.kozakova@stuba.sk
From 1985 to 2013 she was with the Department of Automatic Control Systems (later Institute of Control and Industrial Informatics). From 2018 she is a full professor at the Institute of Automotive Mechatronics, Faculty of Electrical Engineering and Information Technology, STU in Bratislava. Her research interests include robust, optimal and decentralized control.

doc. Ing. Oto Haffner, PhD.
Institute of Automotive Mechatronics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
oto.haffner@stuba.sk
Research activities in applications of computer and machine vision in mechatronic systems using modern ICT such as microcomputers, cloud, computational intelligence, etc. Current focus also on the use of deep learning in practical applications for Industry 4.0 such as visual quality control. He is also involved in modelling and simulation of production processes in manufacturing.

International Conference Cybernetics & Informatics 2022
Visegrád, Hungary, September 11th - September 14th, 2022

Implementation of Heterogeneous Multirobotic Cell Control Using Visualization Techniques

Martin Juhás
Institute of Applied Informatics, Automation and Mechatronics
Faculty of Materials Science and Technology in Trnava
Slovak University of Technology in Bratislava
Trnava, Slovakia
martin_juhas@stuba.sk

Bohuslava Juhásová
Institute of Applied Informatics, Automation and Mechatronics
Faculty of Materials Science and Technology in Trnava
Slovak University of Technology in Bratislava
Trnava, Slovakia
bohuslava.juhasova@stuba.sk

Pavel Važan
Institute of Applied Informatics, Automation and Mechatronics
Faculty of Materials Science and Technology in Trnava
Slovak University of Technology in Bratislava
Trnava, Slovakia
pavel.vazan@stuba.sk

***Abstract*—The article is focused on the use of progressive techniques in the process of design and implementation of a heterogeneous multirobotic cell control system. The task of visualization as a modern approach to the creation of applications for industrial automation is presented. The way of using the visual approach in the creation of the robotic cell workspace monitoring system and the identification of work parts based on machine vision is declared. Visual programming in the form of Stateflow diagrams is also used for the creation of a superior control system based on PLC. Visualization techniques are also used in the process of multirobotic cell operation control by a web-oriented application using augmented reality. The article provides a comprehensive view of the representation of visualization techniques in design process of applications for industrial automation on a specific implementation example.**

***Keywords*—heterogeneous multirobotic cell, control system, visualization, progressive design techniques**

I. Visualization in the Process of Creating Control Systems for Industrial Automation

Today's modern industrial technologies are based primarily on the combination of modern computer information processing technologies with industrial automation control technology. Their advantages complement each other and thus create forms of management and control technologies designed for the field of industrial production. Computer control mainly concerns the presetting and control of each step in the industrial production process through elements of unified algorithmization. In this way, it is possible to ensure that all types of data in the production process can be collected and processed efficiently in a timely manner and thus improve production efficiency and, as a result, achieve a comprehensive technology benefit for the company [1].

Applying the principles of the Industry 4.0 concept is possible with two different approaches. One of the options is a comprehensive or partial renewal of technological equipment, which means a significant financial and energy investment. However, it is possible to apply the principles of the Industry 4.0 concept even without the need for significant investments in the revitalization of complete technical equipment. By appropriate use of available progressive methods and (often freely available) software tools in conjunction with existing technical equipment, it is possible to achieve full or at least partial coverage of the requirements arising from this concept.

A. *Visualization as a Programming Technique for Industrial Logic Controllers*

The rise of wireless industrial technologies, sensors and sensor networks or cloud computing has supported the emergence of new requirements for industrial applications. These PLC IoT applications require the ability to send notifications based on sensor status through various information channels (e-mails, SMS, push notifications), easy access to data via PC and mobile devices and increasingly sophisticated data analysis. All this means that access to the content and logic of the control code needs to be changed. Due to the simpler, faster, and more efficient multi-domain implementation, a change in the actual code creation process is also required.

Graphic programming methods have been used in the areas of production and automation for quite some time. Software tools such as LabView from National Instruments or Simulink from MathWorks, including many patented PLC programming tools, are ultimately very similar to classic text languages such as Cobol or C. However, a major difference is their graphical programming interface, also referred to as visual programming. This element has significant advantages in terms of the convenience of creating the program and is currently quite commonly used in the creation of automation and production IT systems.

Visual programming uses a variety of graphically represented objects, based on which the programmer creates, configures, and modifies programming schemes, which represent various concepts and algorithms of programming. After completing the visual representation, this diagram can be translated or compiled into a standard executable program and then deployed to a specific hardware element, or it can be directly performed as a simulation replacement for that hardware. Graphic tools are ideal for the description and programming of complex automation systems, from the level of specific production machines, through the layer of a complex production line to the level of MES. This approach makes it possible to combine various complex parallel,

This publication was written with the financial support of the VEGA agency in the frame of the projects 1/0176/22 "Proactive control of hybrid production systems using simulation-based digital twin" and 1/0193/22 "Proposal of identification and monitoring of production equipment parameters for the needs of predictive maintenance in accordance with the concept of Industry 4.0 using Industrial IoT technologies"

978-1-6654-8775-7/22/$31.00 ©2022 IEEE

sequential, and iterative tasks, which would not be possible using classical text-sequence programming approach.

One, already relatively well-established method of visual programming is the use of Petri nets. Petri net as a method of visual modeling represents graphical and mathematical formalism of work procedures and processes. Petri nets have been used for quite some time to model and simulate the behavior of various complex systems, especially where it is necessary to capture parallel processes. The problem, however, is the implementation of the Petri net directly in the PLC. The possibility is, for example, the implementation of logic outside the PLC itself, or the currently developed methodology of application and platform independent implementation [2, 3]. Petri nets are also widely used in automated production and development of MES systems. For example, Simatic IT software from Siemens, which is used to manage production operations, is based on the concept of the Petri net.

Impossibility of exchange code or the programmer dependence are well-known drawbacks of traditional programming. A new multilevel graphical and modular approach for programming industrial PLCs is one of possible solutions. The proposed paradigm exploits several object-oriented features, improving the re-use of code, develops objects to be used in a plug and play way understandable by so many developers. Hence this framework reduces the PLC brand and programmer dependence [4].

For complex automation tasks, state diagrams have been proposed for PLC programming. However, state diagram method has some limitations. A new PLC design approach to eliminate these limitations was proposed. In this method, identification of the functions of datapath and controller and the inputs/outputs of both is needed. Next, the Algorithmic State Machine (ASM) charts are used to draw the state transitions for the controller. Finally, a ladder diagram is designed from the ASM chart [5].

A different method of visual programming is used in Matlab software with toolboxes Simulink, Stateflow and Simulink PLC Coder (Fig. 1). The graphic language Stateflow combines state diagrams, flow diagrams, or transition and truth tables with continuous and discrete systems via Simulink.

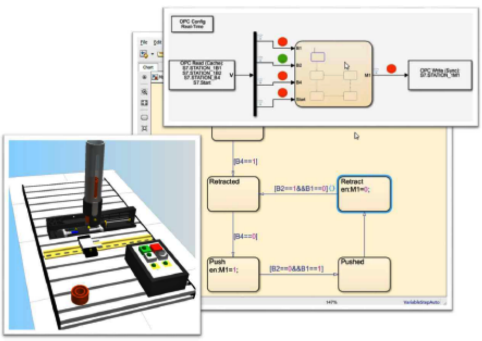


Fig. 1. Simulink Stateflow – Example of visual PLC programming.

The semantics of Stateflow are deterministic in this case because the user defines the order of system states processing. Stateflow enables the design and development of user interfaces, task scheduling, error management systems, control logic, communication protocols as well as hybrid systems. With the support of the Simulink PLC Coder toolbox, the result is a fully implementable code in the form of structured text according to the IEC 61131-3 standard for a wide range of PLC manufacturers [6-9].

B. Augmented Reality in Industrial Automation

Augmented Reality (AR) represents a direct or indirect view to a physically real scene, while certain elements can be enriched with additional digital information relevant to the object the user is looking at. So, it's a combination of the real world with a virtual environment. The virtual environment enriches the view of systems in the form of virtual models. The information presented through them is obtained from various information sources using off-line or on-line applications. Reality enrichment usually takes place in real time and in a semantic context with parts of the environment.

The engineering approach allows to superimpose virtual elements combined from CAD models and other additional data over views of real objects displayed on mobile devices, desktops, smart glasses, or specialized headsets. It is the superposition of the virtual environment over the real environment that distinguishes AR from Virtual Reality (VR), which gives the user access to a completely virtual environment created exclusively from digital elements.

With new advances in computer and manufacturing technology, there is a growing trend that allows users to directly access manufacturing information associated with manufacturing processes. Augmented and mixed reality has the ability to integrate responses to these requirements in real time into a real work environment, which is useful for specific production activities, especially assembly, as well as for training and maintenance.

There is a wide range of AR use in industrial environment, from intelligent predictive maintenance control using smart glasses [10], through control, monitoring and diagnostics of mechatronic devices using a mobile device [11] to modern alternative for control and monitor IoT devices using modern information and communication technologies, including virtual, augmented, and mixed reality [12].

Implementing augmented reality is usually more complex than implementing virtual reality because the AR system requires some mechanism to determine where the elements are in the real world so that the virtual elements can overlap as accurately as possible. Because many of these problems have already been addressed in the computer gaming industry, developers of AR applications, designed not only for the industrial sector, have adapted these principles and technologies to the gaming industry. For example, some applications use a headset equipped with a video camera with additional sensors, such as infrared sensors or radar. In addition to capturing regular video, these devices also measure depth and create perspective. The software, meanwhile, superimposes the virtual scene on the image of the real environment. Based on the evaluation of all information from the sensors, it is possible to draw a virtual model of the technical equipment in an adequate scale, position, and orientation with respect to the observer [13-17].

II. Heterogeneous Multirobotic Cell Configuration

The case study is focused on the implementation of a control system for a complex robotic workplace with a common shared dynamic workspace.

Fig. 2. Heterogeneous multirobotic cell.

The multirobotic cell, shown in Fig. 2, is part of the Laboratory of Control Systems of the University Science Park Campus Bottova (CAMBO) at the Faculty of Materials Technology STU in Trnava. The handling elements are represented by three autonomous robots of different manufacturers. The shared workspace is complemented by a conveyor belt and the entire space is equipped with a monitoring camera system. A programmable logic controller is used as a master control element. The complete list of technical elements of model heterogenous multirobotic cell is given in Tab. I.

TABLE I. Robotic Cell Hardware Configuration

Technical element	Specification
robot ABB IRB 120	control unit IRC5 Compact 6 axes payload 3 kg range 580 mm three-finger pneumatic gripper SMC MHSL3-16D solenoid valve SMC SY5120-5YO-C4F-Q rubber vacuum suction cup SMC B40 vacuum ejector SMC ZH13B solenoid valve VT307V
robot ABB IRB 140	control unit IRC5 Compact 6 axes payload 6 kg range 810 mm three-finger pneumatic gripper FESTO DHDS-16-A solenoid valve CPE10-M1BH-5L-M5 electromagnetic gripper E1AS 0111
robot Mitsubishi Melfa RV-2FB-D	control unit CR750-D 6 axes payload 3 kg range 504 mm two-finger parallel pneumatic gripper FESTO DHPS-16-A solenoid valve FESTO VUVG-L10-P53C-T-M5-1P3 optical sensor FESTO SOEG-L-Q30-P-A-S-2L
conveyor belt	width 300 mm length 1000 mm 3-phase asynchronous motor BONFIGLIOLI BN 63 A 4 3-phase frequency inverter SIEMENS SINAMICS 6SL3210-1KE11-8AB1 retroreflective photoelectric sensors TELEMECANIQUE SENSORS XUB1BPBNL2

Technical element	Specification
vision system Cognex In-Sight 7050-01	monochromatic resolution 800x600 pixels RS-232C and Ethernet
PLC SIMATIC S7-1200	Ethernet and Profinet HMI panel Siemens SIMATIC KTP400 Basic 6AV2123-2DB03-0AX0

III. Design and Implementation of a Control System

A. *Visualization in The Process of Control System Design - Identification of Components*

The control system of the model robotic cell includes a control subsystem of the dynamic component of the shared workspace - conveyor belt, implemented by a Siemens SIMATIC S7-1200 programmable logic controller. The shared workspace is monitored by the Cognex IS 7050-01 sensor subsystem in combination with photoelectric sensors.

In the model case, the Cognex IS 7050-01 machine vision system is designed to identify the type, position, and orientation of the objects to be processed.

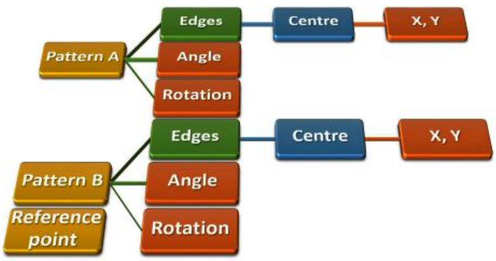


Fig. 3. Method of object parameter identification.

As shown in Fig. 3, the algorithm is based on the identification of the matching patterns representing the floor plan of the objects in the reference position. When the pattern is successfully identified, the object **type** is determined and the **angle** and direction of its **rotation** relative to the reference position is calculated. At the same time, shorter edges are identified in the areas, defined relative to the orientation of the object. Based on these edges, the **coordinates** of the center of the object with respect to the reference point are calculated.

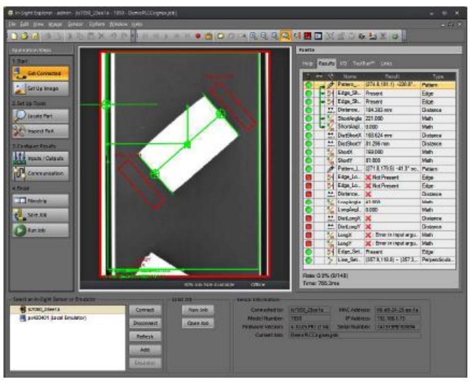


Fig. 4. Visual programming of the machine vision system using In-Sight Explorer.

The whole algorithm is created using the integrated software tool In-Sight Explorer utilizing a visual form of programming based on the creation and combination of algorithmic sequences of built-in functions (Fig. 4).

B. Visualization in The Process of Control System Design - Workspace Control

The PLC control subsystem, which ensures the movement of the conveyor belt based on information from the sensor subsystem as well as the control of the initialization of the identification algorithm in the machine vision system, is implemented with utilization of visual programming techniques - using the principles of state diagrams.

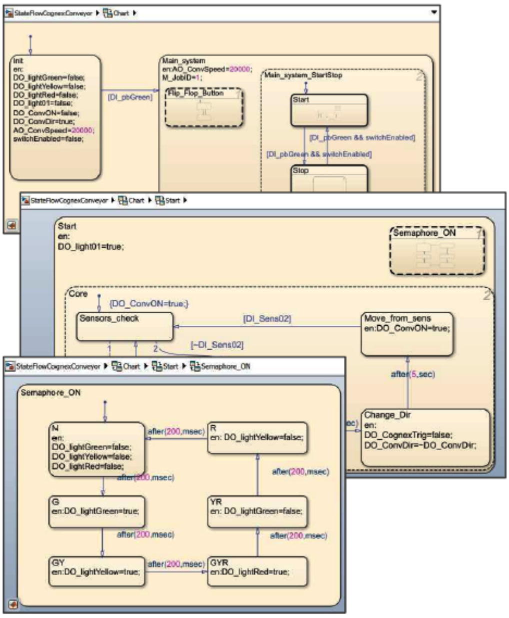


Fig. 5. Stateflow diagram of the PLC control subsystem.

The logic of the control system is designed in the form of states and transitions structured modularly and hierarchically. This method of logic controllers programming is supported by the Mathworks Matlab software with the Simulink and Stateflow toolboxes. The final implementation is realized with utilization of the Simulink PLC Coder toolbox, the result of which is a generated code in the SCL (Structured Control Language) format. This code can be used directly by uploading it to the SIMATIC S7-1200 PLC. The Stateflow diagram of the PLC control subsystem, used in model robotic cell is shown in Fig. 5.

C. Visualization in the Robotic Cell Control Process

The control system of the model robotic cell is complemented by a web-based interactive database application, which is used to visualize process and technical data using hybrid-marker-oriented augmented reality.

The implementation of the complementary system involves the use of several web-oriented technologies for sharing of process and technical information. The main purpose of the solution is to use the open-source technologies as much as possible, which means minimizing costs and at the same time maximizing the possibility of customizability of the system. A summary of used open-source technologies is in Tab. II.

TABLE II. OPEN-SOURCE TECHNOLOGIES OF THE COMPLEMENTARY CONTROL SYSTEM

Name	Purpose
aFrame	3D virtual scene model
AR.js	marker-based reorientation of the virtual scene
QRCode reader	subsystem identification by QR code
php	application kernel scripting language
mysql	database system of the application
JavaScript	scripting language providing application dynamics
jQuery	library of additional JavaScript functions
w3.css	framework supporting responsive web-app design

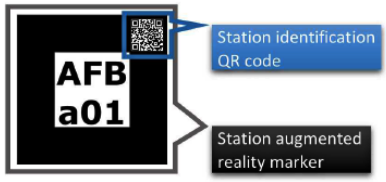


Fig. 6. Hybrid custom AR marker.

The sharing of data from the process level to the AR module is realized by the request of the application web server to the web server of the controlling PLC, as is shown on the communication diagram in Fig. 7. The supplementary part of the control system is designed universally for several possible control PLCs. A specific part of the cell, represented by a separate control PLC, can be identified by a QR code as a part of a hybrid AR marker (Fig. 6).

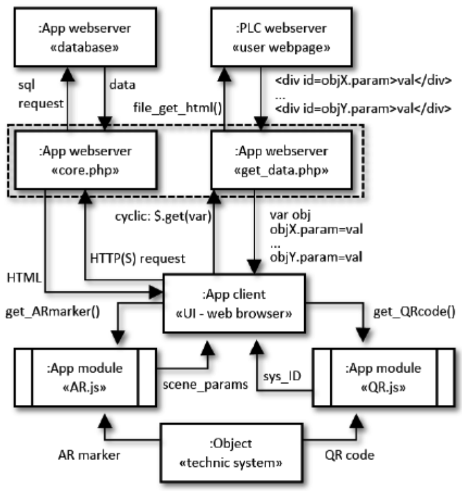


Fig. 7. Communication diagram of the visualization suplementary part of the cell control system.

After receiving the process information in basic form, it is processed on the application client's side (laptop, tablet, smartphone). This information is consequently displayed either on the information panel belonging to the object or in graphical form by changing the parameters of virtual objects (**position**, **color**, **scale**, **rotation**, **opacity**) directly representing a given element of industrial automation, as is shown in the application example in Fig. 8.

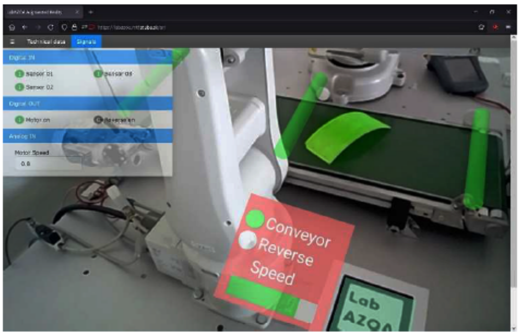


Fig. 8. Visualization suplementary part of the cell control system - visualization of process data.

Individual objects are implemented as interactive elements, where using the A-Frame object of the cursor type, the user is allowed to directly select the element of interest. The visualization part is supplemented by technical data of the selected element, which are read from the relational database located on the application server. The basic data are supplemented by references to the technical documentation of the object and subordinate elements in the hierarchical arrangement of the technical components of the robotic cell.

The modular concept of the application allows a very simple extension by any number of technological modules represented by individual elements:

- AR marker,
- identifier represented by QR code within the hybrid AR marker,
- virtual 3D model containing technical means of interest,
- list of input / output signals
- list of process variables,
- technical data / documentation stored in a relational database in a defined structure.

IV. Discussion and Evaluation of the Solution

The article focuses on the actual and currently much-discussed area of the fourth industrial revolution. From this issue, which includes a wide range of problems from various areas of industrial production, a view on the creation of control applications for industrial automation is selected. The main interest is focused on new, progressive approaches and methods of design and implementation of these applications.

The main purpose of the article is to point out to the fact, that the application of the principles of the Industry 4.0 concept is possible without requirement of significant investments in revitalization of complete technical equipment. By appropriate use of available progressive methods and (often freely accessible) software tools in conjunction with current technical equipment, it is possible to achieve full or at least partial coverage of the requirements arising from this concept.

In the case of the deployment of an industrial machine vision system as well as a superior control PLC, modern visual programming techniques are used. This approach eliminates the disadvantages of classical command programming such as parallel branches or multilayer nesting. An additional, but by no means less important part of the complex cell control system is a web-oriented application used to visualize the entire process, which in combination with a relational database and the use of augmented reality brings a completely different view on the production process.

The main qualitative indicators that prove the advantage of the described implementation method of the robotic workplace design, can be considered:

- independence from knowledge of a specific programming language,
- easy-to-applicable virtual commissioning,
- user-endpoint-hardware-independent no-installation marker-based augmented reality web-application,
- robot-brand independent communication framework,
- OPC UA ready.

A summary of techniques and technologies used in the process of design and implementation of the model solution is given in Tab. III.

TABLE III. Visualization Techniques and Technologies Used in the Control System Design

Technique / Technology	Purpose	Tool
Visual programming Edge computing	Identification of handling objects on the conveyor belt using Cognex IS 7050-01	In-Sight Explorer 6.1.3
Visual programming	Control algorithm of PLC Siemens SIMATIC S7-1200	Matlab 2019a Simulink 9.3 Stateflow 10.0 Simulink PLC Coder 3.0 TIA Portal V13
Visualization WebApp	Web application interface	Php7.2 MySQL 5.5 jQuery 3.2 w3.css 4.1
Visualization WebApp	Identification using a QR code via a web interface	Instascan QRCode reader
Visualization WebApp	Virtual 3D model of the cell for web browser	aFrame 1.0.4
Visualization WebApp	Augmented reality in the web interface	AR.js 3.3
Visualization Remote control	Communication between PLC Siemens SIMATIC S7-1200 and web application	TIA Portal V13 S7-1200 webserver Simplehtmldom 1.9.1 Portforwarding
Visualization Cloud solution	Display of the result of machine vision in the virtual control system Matlab Stateflow Offline image processing	In-Sight Explorer 6.1.3 ImageMagick-7.0.8 FileZilla Server 0.9 FTP

REFERENCES

[1] Z. Li and M. Jiang, "Application of Computer Information Processing Technology in Industrial Automation," in Journal of Physics: Conference Series, 2020, Issue 1678.

[2] I. Azkarate, M. Ayani, J.C. Mugarza and L. Eciolaza, "Petri Net-Based Semi-Compiled Code Generation for Programmable Logic Controllers," in Applied Sciences, 2021, 11(15), p. 7161.

[3] V. Năvrăpescu, I. Deaconu, A. Chirilă and A. Deaconu, "Petri net versus Ladder Diagram for controlling a process automation," 2013 8th International Symposium On Advanced Topics In Electrical Engineering (ATEE), Bucharest, Romania, 2013, pp. 1-4.

[4] A. Sapena-Bañó *et al*., "A new multilevel approach for Programmable Logical Controller (PLCs)," 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, 2015, pp. 1820-1825.

[5] G. Wang, "A New Approach for PLC Ladder Diagram Design," 2021 IEEE International Conference on Electro Information Technology (EIT), Mt. Pleasant, MI, USA, 2021, pp. 021-026.

[6] R. Hýl and R. Wagnerová, "Fast development of controllers with Simulink Coder," 2017 18th International Carpathian Control Conference (ICCC), Sinaia, Romania, 2017, pp. 406-411.

[7] C. Driver, "What You Need to Know About PLC Programming in the Age of IoT" [Online]. Available at: https://blog.temboo.com/plc-programming-for-iot-systems/ [Accessed: 4-2022].

[8] R. Robbins, "Visual Programming for Automation," [Online]. Available at: https://www.controleng.com/articles/visual-programming-for-automation/ [Accessed: 4-2022].

[9] M. Minas and G. Frey, "Visual PLC-programming using signal interpreted Petri nets," 2002 American Control Conference (ACC), Anchorage, AK, USA, 2002, pp. 5019-5024.

[10] M. Kostoláni, J. Murín and Š. Kozák, "Intelligent predictive maintenance control using augmented reality," 2019 22nd International Conference on Process Control (PC19), Strbske Pleso, Slovakia, 2019, pp. 131-135.

[11] E. Stark, E. Kučera, P. Drahoš and O. Haffner, "Proposal of Mechatronic Devices Control using Mixed Reality," 2019 Federated Conference on Computer Science and Information Systems (FedCSIS), Leipzig, Germany, 2019, pp. 925-928.

[12] S. Bucsai, E. Kučera, O. Haffner and P. Drahoš, "Control and Monitoring of IoT Devices Using Mixed Reality Developed by Unity Engine," 2020 Cybernetics & Informatics (K&I), Velke Karlovice, Czech Republic, 2020, pp. 1-8.

[13] M. Juhás and B. Juhásová, "Augmented Reality as an Instrument for Teaching Industrial Automation," 2018 Cybernetics & Informatics (K&I), Lazy pod Makytou, Slovakia, 2018, pp. 1-5.

[14] J. Etienne, "AR.js - Augmented Reality on the Web," [Online]. Available at: https://github.com/AR-js-org/AR.js [Accessed: 4-2022].

[15] M. Juhás, B. Juhásová and I. Halenár, "Augmented Reality in Education 4.0," 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 2018, pp. 231-236.

[16] J. R. Koelsch, "How Augmented Reality Became a Serious Tool for Manufacturing" [Online]. Available at: https://www.profoodworld.com/home/article/21259479/how-augmented-reality-became-a-serious-tool-for-manufacturing [Accessed: 4-2022].

[17] K. Kloberdanz, "Smart Specs: OK Glass, Fix This Jet Engine," [Online]. Available at: https://www.ge.com/news/reports/smart-specs-ok-glass-fix-jet-engine [Accessed: 4-2022].

Authors

Assoc. Prof. Martin Juhás, PhD.
Institute of Applied Informatics, Automation and Mechatronics
Faculty of Materials Science and Technology in Trnava
Slovak University of Technology in Bratislava
martin_juhas@stuba.sk
His research interests include identification, modeling and simulation of dynamic systems, analysis and synthesis of control systems with utilization of software tools like Matlab, Simulink or VRML. His next field of professional interest is oriented to design and implementation of information systems, including data modeling and functional modeling, together with their implementation by SQL, Oracle PL/SQL, php, html, css, javascript, Object Pascal or C #. In last years he deals with modeling and control of industrial robots (tool like ABB Robotstudio, CIROS Studio, RoboGuide or Mitsubishi MELFA Basic in combination with TIA Portal) with focus on robot-robot collaboration and communication in heterogeneous multirobotic workcells.

Assist. Prof. Bohuslava Juhásová, PhD.
Institute of Applied Informatics, Automation and Mechatronics
Faculty of Materials Science and Technology in Trnava
Slovak University of Technology in Bratislava
bohuslava.juhasova@stuba.sk
Her research and activities are focused in the areas of software engineering and control systems. In the field of software engineering she is specialized to analysis, identification, definition and design of information systems with focus to software architecture. In this area she has skill in using software tools like ErWin ERX, Enterprise Architect or Oracle SQL Developer – Data Modeler. Her next field of professional interest, control systems, is oriented to modeling and simulation of systems with utilization of software tools Matlab and Simulink.

Prof. Pavel Važan, PhD.
Institute of Applied Informatics, Automation and Mechatronics
Faculty of Materials Science and Technology in Trnava
Slovak University of Technology in Bratislava
pavel.vazan@stuba.sk
He is dealing with the problems of control of production systems and especially flexible production systems with focus on the design of various control strategies, or the optimization of control goals, mainly with the utilization of discrete event-driven simulation and simulation optimization, mostly from the point of view of its use; for example, in the proposal of an alternative procedure for determining the optimal size of the production batch, or the problem of the impact of priority rules in the scheduling of operations on production goals. Currently with proactive control of production systems using simulation-based digital twin.

International Conference Cybernetics & Informatics 2022
Visegrád, Hungary, September 11th - September 14th, 2022

Hybrid Intelligent MPC In Industry

Peter Karas
Faculty of Electrotechnics and Informatics
Slovak Technical University
Bratislava, Slovak Republic
petokaras@gmail.com

Štefan Kozák
Faculty of Informatics
Pan-European University
Bratislava, Slovak Republic
stefan.kozak@paneurouni.com

***Abstract*— The paper deals with current research challenges in modeling and constrained predictive control as applied to a nonlinear process used in chemical industry (chemical reactor for polypropylene production), and mainly focusing on AI - based or soft computing methods. Based on a combination of the classical approach to modeling of a complex nonlinear process with hybrid dynamics and SC control methods, a prediction model is obtained and subsequently used in the process of solving the optimization problem of constrained predictive control and applied to a simulation model of a reactor used in petrochemical industry.**

***Keywords*— Control engineering, mathematical models, MPC, Model predictive control, chemical plant, petrochemicals, hybrid MPC, neural networks**

I. Introduction

The aim of this work is to analyze the possibility of implementing modified advanced control in a petrochemical plant, and to develop an appropriate control method based on soft computing approaches (neural networks).

This work's primary target and purpose are to analyze the possible application of hybrid MPC methods based on the soft computing approach in the industry. The method's application has been simulated and analyzed using the real process data from an existing plant. The process selected for this task is common and essential in the petrochemical industry – polymerization in fluidized bed reactor. This process is nonlinear, affected by inputs and having several internal state variables [2]. The natural advantages of the optimal constrained control method provided by the MPC method may improve optimal plant operation [15]. Even a small improvement of operating condition may provide significant improvement in asset value. The polymerization reactor is online almost entire year, and it is primarily operated in optimal conditions developed by long term operation and experience. Through the comparison of the MPC with neural network observer and mathematical model, we identified the advantages of the optimal constrained control method provided by the MPC with neural observer.

A. Process description

The process selected for this work is a worldwide well-known and frequently used process - the Union Carbide gas-phase polypropylene process (UNIPOL). The UNIPOL process is a result of cooperation between Union Carbide Corporation and Shell chemical [13]. In the latest literature [14] can be found that the heat removal can be partially substituted by some amount of liquid evaporation in the fluidized bed. This principle is frequently called the condensation mode type of cooling. The process is designed for the gas-phase chemical reaction in the fluidized bed. Besides delivering the reactants, the cycle gas does two main functions: it ensures of the reactor bed fluidization and provides mixing to the reactor bed. The cycle gas circulates in the system, while in the cycle is filtered, pressurized by the compressor, cooled down in cycle Gas Cooler and then is returned to the reactor in the bottom section. The gas velocity is an important process parameter and defines optimal reactor bed conditions [6].

The reactor bed fluidization also depends on the particle size. The effect of the particle size on the polymerization reaction has been considered in [9]. The process selected is a polypropylene reactor in the polypropylene plant PP5 in a Slovak petrochemical company. This plant uses an Advanced Process Control system provided by the technology licensor, and the method behind this system is intellectual property of the APC company.

B. Production in the polymerization reactor

In the reactor are produced thirteen different products - thirteen different types of polypropylenes. For the research purposes we decided to select only the best representatives to prove the control system performance. The Impact co-polymers (ICP's) (or Random co-polymers (RCP's)) are produced with additional co-monomer, which means a significant problem for the control system. Due to involvement, the industrial control system must be scalable to provide flexibility and wide range of application. The products selected for the case study are

1. Product 1 - IM 1259
2. Product 2 - HT 306
3. Product 3 - HG 1007

IM 1259 will show the control system's ability to switch to a process with more reactants, while HT 306 and HG 1007 will present the control method scalability. Although the two HPP's (Homo polypropylenes) have the same number of reactants, different operation parameters are required.

II. Modeling and Control problem formulation

A. Polymerization reaction

The reaction occurs in the gas phase in the Ziegler – Natta catalyst. The gas stream provides monomer (C_3H_6), hydrogen (H_2), and Catalyst (Ziegler-Natta) to the reactor. The reaction occurs in the pores of Ziegler – Natta catalyst and the presence of co-catalyst together with selectivity control agent. The reactor is stirred ideally by the cycle gas flowing from the bottom to the top of the reactor, and the bed is at the minimum fluidization [8]. Homogenous reaction type has been considered in this study. The difference between homogenous and heterogenous polymerization is discussed by Harmon [16]. It is shown that the solution is scalable to the higher number of states while requiring more computational resources.

978-1-6654-8775-7/22/$31.00 ©2022 IEEE

B. Polymer production rate

The total polymer production rate is then expressed as the molecular weight of monomer and co-polymer multiplied by propagation rate [5].

$$R = M_{wC3}.R_{monomer} + M_{wC2}.R_{co-polymer}, \quad (1)$$

$$R_{co-polymer} = C_{C2}.Y_0(t).k_{pc}(i), \quad i = 1, \quad (2)$$

$$R_{monomer} = C_{C3}.Y_0(t).k_p(i), \quad i = 2 \text{ or } 3, \quad (3)$$

The purity of the reactant gives the stability of the molecular weight constant.

C. Material balance

The reactant material balances are based on the chemical reactions in the reactor [17]. The material balance for the reactants is defined as.

$$V_{FB}.\frac{dC_a(t)}{dt} = F_{in}^{Ca}(t).Cat(i) - k_a(t,i).\left(C_a(t) + \frac{v_{SGV}}{h_{FB}}\right), \ i=1,2 \text{ or } 3, C_a(0) = C_{a0} \quad (4)$$

$$V_{FB}.\frac{dC_{C3}(t)}{dt} = F_{in}^{C3}(t).C_{in}^{C3} - F_{out}^{CG}(t).C_{C3}(t) - C_{C3}(t).\left(k_{im}(t,i).P_0(t) + k_p(t,i).Y_0(t) + k_{fm}(t,i).Y_0(t)\right), \ i=1,2 \text{ or } 3, C_{C3}(0) = C_{C30} \quad (5)$$

$$V_{FB}.\frac{dC_{C2}(t)}{dt} = F_{in}^{C2}(t).C_{in}^{C2} - F_{out}^{CG}(t).C_{C2}(t) - C_{C2}(t).\left(k_{ic}(t,i).P_0(t) + k_{pc}(t,i).Y_0(t) + k_{fcm}(t,i).Y_0(t)\right), \ i=1,2 \text{ or } 3, C_{C2}(0) = C_{C20} \quad (6)$$

$$V_{FB}.\frac{dC_{H2}(t)}{dt} = F_{in}^{H2}(t).C_{in}^{H2} - F_{out}^{CG}(t).C_{H2}(t) - C_{H2}(t).(k_H(t,i).Y_0(t)), \ i=1,2 \text{ or } 3, C_{H2}(0)=C_{H20} \quad (7)$$

The volume flow rates F_{in}^{Ca}, F_{in}^{C3}, F_{in}^{C2} or F_{in}^{H2} are important for the control system. F_{in}^{Ca} is a manipulated variable and F_{in}^{C3}, F_{in}^{C2} or F_{in}^{H2} will be in the position of the measurable disturbance.

$$F_{in}^{Ca} = M_{in}^{Ca}(t).\frac{1}{3600}.\frac{1}{\rho_{Ca}}, \quad (8)$$

$$F_{in}^{C3} = M_{in}^{C3}(t).\frac{1}{3600}.\frac{1}{\rho_{C3}(t).\frac{p_{Rx1}}{p_{atm}}.\frac{T_{atm}}{T_{ref}}}, \quad (9)$$

$$F_{in}^{C2} = M_{in}^{C2}(t).\frac{1}{3600}.\frac{1}{\rho_{C2}(t).\frac{p_{Rx1}}{p_{atm}}.\frac{T_{atm}}{T_{ref}}}, \quad (10)$$

$$F_{in}^{H2} = M_{in}^{H2}(t).\frac{1}{3600}.\frac{1}{\rho_{H2}(t).\frac{p_{Rx1}}{p_{atm}}.\frac{T_{atm}}{T_{ref}}}, \quad (11)$$

The molar concentrations of the reactants on the reactor inlet are considered as measurable disturbance variables and are defined

$$C_{in}^{C3} = \rho_{C3}(t).\frac{1000}{1}.M_w^{C3}, \quad (12)$$

$$C_{in}^{C2} = \rho_{C2}(t).\frac{1000}{1}.M_w^{C2}, \quad (13)$$

$$C_{in}^{H2} = \rho_{H2}(t).\frac{1000}{1}.M_w^{H2}, \quad (14)$$

where the M_w^{C3}, M_w^{C2} and M_w^{H2} are molecular weights of the respective reactants. The concentration of the activated catalyst sites is then defined

$$V_{FB}.\frac{dP_0(t)}{dt} = k_a(t,i).C_a(t) - P_0(t).\left(k_{im}(t,i).C_{C3}(t) + k_{ic}(t,i).C_{C2}(t) + \frac{v_{SGV}}{h_{FB}}\right), \ i=1,2 \text{ or } 3, P_0(0)=P_{00} \quad (15)$$

The concentration of living polymer chains is defined according to several researches works [5 and 8].

$$V_{FB}.\frac{dP_l(t)}{dt} = P_{l-1}(t).\left(k_p(t,i).C_{C3}(t) + k_{pc}(t,i).C_{C2}(t)\right) - P_l(t).\left(k_p(t,i).C_{C3}(t) + k_{fm}(t,i).C_{C3}(t) + k_{pc}(t,i).C_{C2}(t) + k_{fcm}(t,i).C_{C2}(t) + k_H(t,i).C_{H2}(t) + \frac{v_{SGV}}{h_{FB}}\right), \ i=1,2 \text{ or } 3, P_l(0)=P_{l0} \quad (16)$$

According to Eq. (15) and (16), the zeroth moment of the distributions of the polymer chains is then defined

$$V_{FB}.\frac{dY_0(t)}{dt} = \frac{dP_l(t)}{dt} + \sum_{l=2}^{\infty}\frac{dP_l(t)}{dt} = P_0(t).\left(k_p(t,i).C_{C3}(t) + k_{pc}(t,i).C_{C2}(t)\right) - P_l(t).\left(k_p(t,i).C_{C3}(t) + k_{fm}(t,i).C_{C3}(t) + k_{pc}(t,i).C_{C2}(t) + k_{fcm}(t,i).C_{C2}(t) + k_H(t,i).C_{H2}(t) + \frac{v_{SGV}}{h_{FB}}\right) + \sum_{l=2}^{\infty}\left[P_{l-1}(t).\left(k_p(t,i).C_{C3}(t) + k_{pc}(t,i).C_{C2}(t)\right) - P_l(t).\left(k_p(t,i).C_{C3}(t) + k_{fm}(t,i).C_{C3}(t) + k_{pc}(t,i).C_{C2}(t) + k_{fcm}(t,i).C_{C2}(t) + k_H(t,i).C_{H2}(t) + \frac{v_{SGV}}{h_{FB}}\right)\right], \quad i = 1,2 \text{ or } 3 \quad (17)$$

We get the reduced ODE for zeroth moment distributions.

$$V_{FB}.\frac{dY_0(t)}{dt} = P_0(t).\left(k_p(t,i).C_{C3}(t) + k_{pc}(t,i).C_{C2}(t)\right) - Y_0(t).\left(k_{fm}(t,i).C_{C3}(t) + k_{fcm}(t,i).C_{C2}(t) + k_H(t,i).C_{H2}(t) + \frac{v_{SGV}}{h_{FB}}\right), \quad i = 1,2 \text{ or } 3, Y_0(0) = Y_0 \quad (18)$$

The gases are cycled through the system, continuously delivering the reactants to the reactor and removing the heat. The reactor's retention time must be considered. For the retention time, we can write

$$\frac{1}{T_{ret}} = \frac{v_{SGV}}{h_{FB}}, \quad (19)$$

where T_{ret} is the retention time, v_{SGV} is the superficial gas velocity in and h_{FB} is the height of the fluidized bed. The first moment of the distributions of the polymer chains are expressed as $Y_1(t)$.

$$V_{FB}.\frac{dY_1(t)}{dt} = P_0(t).\left(k_p(t,i).C_{C3}(t) + k_{pc}(t,i).C_{C2}(t)\right) + Y_0(t).\left(k_{fm}(t,i).C_{C3}(t) + k_{fcm}(t,i).C_{C2}(t)\right) + Y_0(t) - Y_1(t).\left(k_{fm}(t,i).C_{C3}(t) + k_{fcm}(t,i).C_{C2}(t) + k_H(t,i).C_{H2}(t) + \frac{v_{SGV}}{h_{FB}}\right), \ i=1,2 \text{ or } 3, Y_1(0)=Y_{10} \quad (20)$$

we used the definitions in Eq. (21) and Eq. (22)

$$\sum_{l=2}^{\infty} l.(P_{l-1}) = \sum_{l=1}^{\infty}(l+1).(P_l) = Y_1 + Y_0, \quad (21)$$

$$\sum_{l=2}^{\infty} l^2.(P_{l-1}) = \sum_{l=1}^{\infty}(l+1)^2.(P_l) = Y_2 + 2.Y_1 + Y_0, \quad (22)$$

The ODE for second-moment distributions $Y_2(t)$ is

$$V_{FB}.\frac{dY_2(t)}{dt} = P_0(t).\left(k_p(t,i).C_{C3}(t) + k_{pc}(t,i).C_{C2}(t)\right) + Y_0(t).\left(k_{fm}(t,i).C_{C3}(t) + k_{fcm}(t,i).C_{C2}(t)\right) + \left(2.Y_0(t) + Y_0(t)\right).\left(k_p(t,i).C_{C3}(t) + k_{pc}(t,i).C_{C2}(t)\right) - Y_2(t).\left(k_{fm}(t,i).C_{C3}(t) + k_{fcm}(t,i).C_{C2}(t) + k_H(t,i).C_{H2}(t) + \frac{v_{SGV}}{h_{FB}}\right), \quad i = 1,2 \text{ or } 3, Y_2(0) = Y_{20} \quad (23)$$

The dead polymer chain distributions is defined

$$X_n = \sum_{l=1}^{\infty} l^n . Q_l, \tag{24}$$

The zeroth, first, and second moments of dead polymer distributions are shown in Eq. (25, 26 and 27)

$$\begin{aligned} V_{FB}.\frac{dX_0(t)}{dt} &= Y_0(t).\Big(k_{fm}(t,i).C_{C3}(t) \\ &+ k_{fcm}(t,i).C_{C2}(t) \\ &+ k_H(t,i).C_{H2}(t)\Big) \\ &- X_0(t).\left(\frac{v_{SGV}}{h_{FB}}\right), \\ &i = 1,2 \; or \; 3,\; X_0(0) = X_{00} \end{aligned} \tag{25}$$

$$\begin{aligned} V_{FB}.\frac{dX_1(t)}{dt} &= Y_1(t).\Big(k_{fm}(t,i).C_{C3}(t) \\ &+ k_{fcm}(t,i).C_{C2}(t) \\ &+ k_H(t,i).C_{H2}(t)\Big) \\ &- X_1(t).\left(\frac{v_{SGV}}{h_{FB}}\right), \\ &i = 1,2 \; or \; 3, X_1(0) = X_{10} \end{aligned} \tag{26}$$

$$\begin{aligned} V_{FB}.\frac{dX_2(t)}{dt} &= Y_2(t).\Big(k_{fm}(t,i).C_{C3}(t) \\ &+ k_{fcm}(t,i).C_{C2}(t) \\ &+ k_H(t,i).C_{H2}(t)\Big) \\ &- X_2(t).\left(\frac{v_{SGV}}{h_{FB}}\right), \\ &i = 1,2 \; or \; 3, X_2(0) = X_{20} \end{aligned} \tag{27}$$

Y_m and X_m stands for an m-th moment of the polymer population balances. While Y_m means population balance of living polymer chain and X_m means population balances of inactive, dead polymer chains. The parameter $\frac{\mathbf{v_{SGV}}}{\mathbf{h_{FB}}}$ represents the retention time of the reactants in the reactor.

D. The energy balance

The major material inside is the fluidized bed, and the primary material is the produced pellets of polypropylene resin. $\mathbf{F_{out}^{CG}}.\tilde{\rho}_{CG}(i).c_p^{CG}.(T_{in} - T)$ is the amount of energy that enters and leaves to/from the reactor together with cycle gas.

The cycle gas density depends on the cycle gas composition and pressure, which is different for each product. The systems pressure is controlled by the input streams and cycle gas compressor power. The energy balance [17] of the fluidized bed is defined in Eq. (28)

$$\begin{aligned} &V_{FB}.c_p^{PP}.\rho_{PP}(t)\frac{dT}{dt} \\ &= F_{out}^{CG}.\tilde{\rho}_{CG}(t,i).c_p^{CG}.(T_{in} - T) \\ &+ V_{FB}.A_{FB}.(T_{in} - T) \\ &+ V_{FB}.\Big[k_d(t,i).I_d(t).P_0(t).(-H^d(i)) \\ &+ k_{im}(t,i).P_0(t).C_{C3}(t).\big(+H^{im}(i)\big) \\ &+ k_{icm}(t,i).P_0(t).C_{C2}(t).\big(+H^{icm}(i)\big) \\ &+ k_H(t,i).Y_0(t).C_{H2}(t).(+H^H(i)) \\ &+ k_{fm}(t,i).Y_0(t).C_{C3}(t).\big(+H^{fm}(i)\big) \\ &+ k_{fcm}(t,i).Y_0(t).C_{C2}(t).\big(+H^{fcm}(i)\big) \\ &+ k_d(t,i).Y_0(t).(-H^d(i)) \\ &+ k_{dI}(t,i).Y_0(t).I_d(t).(-H^{dI}(i)) \\ &+ k_p(t,i).Y_0(t).C_{C3}(t).(+H^p(i)) \\ &+ k_{pc}(t,i).Y_0(t).C_{C2}(t).(+H^{pc}(i))\Big], \\ &i = 1,2 \; or \; 3, T(0) = T_0 \end{aligned} \tag{28}$$

The total density is calculated from the gas composition.

$$\rho_{CG}(t,i) = \rho_{C3}(t), i = 1 \tag{29}$$

$$\rho_{CG}(t,i) = \frac{V_{C2}.\rho_{C2}(t) + V_{C3}.\rho_{C3(t)}(t)}{V_{C2} + V_{C3}}, i = 2 \; or \; 3 \tag{30}$$

The temperature and pressure inside the reactor have to be considered.

$$\tilde{\rho}_{CG}(t,i) = \rho_{CG}(t,i).\frac{p_{Rx1}}{p_{atm}}.\frac{T_{atm}}{T_{ref}}, \tag{31}$$

Eq. (31) shows the normalization of the cycle gas density to the reference pressure and temperature inside the reactor. The term $\mathbf{V_{FB}}.\mathbf{A_{FB}}.(\mathbf{T_{in}} - \mathbf{T})$ represents the energy removed from the system by the heat transfer from the resin to the cycle gas. The last term in the energy balance Eq. 28 is the heat generated by the polymerization reaction. The terms $\mathbf{H}^{\cdots}$ are the enthalpies of the chemical reaction and are specific to the reactants, product, and reaction type. The signing convention has been made while $+\mathbf{H}^{\cdots}$ sign means that the energy has been generated and the sign $-\mathbf{H}^{\cdots}$ means that the energy or heat has been consumed. The polymerization is an exothermic reaction, and therefore most of the heat is generated rather than consumed.

E. The product switching

As mentioned before, to demonstrate the product selectivity, this work considers three different polymers – products in the controller. Several model parameters are product dependent, and in the previous section, the Eq. are denoted as functional of $\boldsymbol{i}$. The simulations and model tuning has been done for parameters are taken from literature and listed in Tab.1.

TABLE I. PRODUCT PARAMETERS

<table>
<tr><th>Prod. 1 (IM 12 59)</th><th>Prod. 2 (HT 3 06)</th><th>Prod. 3 (HG 10 07)</th></tr>
<tr><td colspan="3">$k_{a0}(i) = 1$</td></tr>
<tr><td>$k_{im0}(i) = 22$ [5]</td><td>$k_{im0}(i) = 22$ [5]</td><td>$k_{im0}(i) = 22$ [5]</td></tr>
<tr><td>$k_{ic0}(i) = 2{,}2$ [11]</td><td>$k_{ic0}(i) = 2{,}2$ [11]</td><td>$k_{ic0}(i) = 2.2$ [11]</td></tr>
<tr><td colspan="3">$k_{H0}(i) = 100$</td></tr>
<tr><td>$k_{fm0}(i) = 100$</td><td>$k_{fm0}(i) = 100$</td><td>$k_{fm0}(i) = 100$</td></tr>
<tr><td>$k_{fcm0}(i) = 100$</td><td>$k_{fcm0}(i) = 0$</td><td>$k_{fcm0}(i) = 0$</td></tr>
<tr><td colspan="3">$k_{d0}(i) = 1000$ [12]</td></tr>
<tr><td colspan="3">$k_{dI0}(i) = 1000$ [12]</td></tr>
<tr><td>$k_{pc0}(i) = 5$</td><td>$k_{pc0}(i) = 8$</td><td>$k_{pc0}(i) = 30$</td></tr>
<tr><td>$k_{p0}(i) = 2$</td><td>$k_{p0}(i) = 0$</td><td>$k_{p0}(i) = 0$</td></tr>
<tr><td colspan="3">$H^a(i) = 7.64$ [11]</td></tr>
<tr><td>$H^{im}(i) = 7{,}2$ [10]</td><td>$H^{im}(i) = 7{,}2$ [10]</td><td>$H^{im}(i) = 7{,}2$ [10]</td></tr>
<tr><td>$H^{ic}(i) = 7{,}2$ [10]</td><td>$H^{ic}(i) = 7{,}2$ [10]</td><td>$H^{ic}(i) = 7{,}2$ [10]</td></tr>
<tr><td colspan="3">$H^H(i) = 10{,}7$ [10]</td></tr>
<tr><td>$H^{fm}(i) = 65{,}42$ [10]</td><td>$H^{fm}(i) = 65{,}42$ [10]</td><td>$H^{fm}(i) = 65{,}42$ [10]</td></tr>
<tr><td>$H^{fcm}(i) = 65{,}42$ [10]</td><td>$H^{fcm}(i) = 65{,}42$ [10]</td><td>$H^{fcm}(i) = 65{,}42$ [10]</td></tr>
<tr><td colspan="3">$H^d(i) = 0$</td></tr>
<tr><td colspan="3">$H^{dI}(i) = 0$</td></tr>
<tr><td>$H^{pc}(i) = 7{,}2$ [10]</td><td>$H^{pc}(i) = 7{,}2$ [10]</td><td>$H^{pc}(i) = 7{,}2$ [10]</td></tr>
<tr><td>$H^p(i) = 10{,}2$ [10]</td><td>$H^p(i) = 10{,}2$ [10]</td><td>$H^p(i) = 10{,}2$ [10]</td></tr>
<tr><td>$Cat(i) = 10000$</td><td>$Cat(i) = 10500$</td><td>$Cat(i) = 8000$</td></tr>
</table>

The switching between the products is discrete [18]. According to product type state variable, the product-specific rate constants and enthalpies are selected together with catalyst density [4].

F. Hybrid nonlinear MPC

We propose using a neural network for the feedback estimation of the state variables, which are virtual and therefore not measurable by any measuring device [18]. As shown in Fig. 3, the state feedback is separated into two groups of variables. The variables $\mathbf{x}_M(\mathbf{t}) = [\mathrm{C_a}, \mathrm{C_{C3}}, \mathrm{C_{C2}}, \mathrm{C_{H2}}, \mathrm{T}]^{\mathrm{T}}$ and $\mathbf{x}_V(\mathbf{t}) = [\mathrm{P_0}, \mathrm{Y_0}]^{\mathrm{T}}$ represent the feedback states. The state vector represent the molar concentrations of the reactants $\mathrm{C_a}, \mathrm{C_{C3}}, \mathrm{C_{C2}}, \mathrm{C_{H2}}$ and the temperature at which the reaction occurs T. The vector $\mathbf{x}_V(\mathbf{t})$ represent the unmeasurable state vector, which stands for virtual molar concentrations of the active catalyst sites and living polymer chains.

Since a nonlinear MPC controller is a discrete-time controller [18] and the state function is continuous-time, the model gets discretized by the controller using the implicit trapezoidal rule. This method approximates the integration over an interval by breaking the area down into trapezoids with more easily computable areas. The integration method is based on the summation of partial areas of the trapezoids.

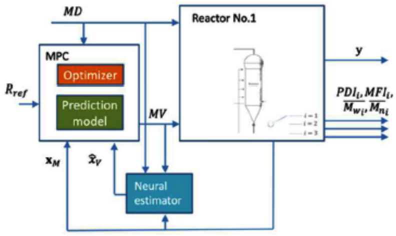


Fig. 1: Simulation model structure with MPC and neural estimator

In Fig. 1 is shown the control structure with a neural estimator for the unmeasurable state variables. The product type, which is the discrete state, determines the type of product and the qualitative properties of PDI and MFI.

G. Hybrid neural estimator

In Fig. 2 is the prediction model as a combination of multiple MLP's.

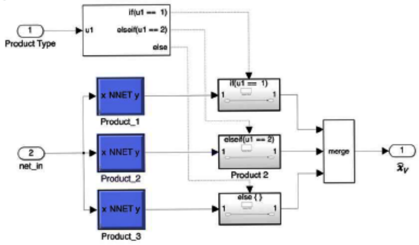


Fig. 2: Prediction model with different MLP for product types

The multiple MLP's shown in Fig. 2 are different neural networks trained with product specific data. Each MLP has its configuration with weights. A good insight into the neural observers and estimators provides Lakhal [20].

H. Control problem formulation

The prediction model is nonlinear. We are interested in controlling the production rate. The output function is defined in Eq. (1). The manipulated variable is the catalyst feed, represented by volume flow defined in Eq. (8). The disturbances are defined as

$$\boldsymbol{M_{DC}}(t) = [F_{in}^{C3}, F_{in}^{C2}, F_{in}^{H2}, T_{in}, v_{SGV}, h_{FB}, p_{Rx1}, \rho_{C3}, \rho_{C2}, \rho_{H2}, \rho_{PP}]^T \quad (32)$$

$$\boldsymbol{M_{DD}}(t) = [\mathrm{i}]^T \quad (33)$$

Eq. (32) shows the continuous signals, while Eq. (33) discrete disturbance – product switching.

The internal states of the prediction model are expressed in Eq. (34).

$$\boldsymbol{x}(t) = [C_a, C_{C3}, C_{C2}, C_{H2}, P_0, Y_0, T]^T \quad (34)$$

The control problem is formulated as minimization of a cost function designed to make the controller's minimum effort to track the reference signal with the output variable. The control problem can be written as defined in Eq. (35).

$$\min_{\boldsymbol{z}(\cdot)} \mathcal{J}\left(\boldsymbol{z}(t), T_c, T_p\right) \quad (35)$$

T_p and T_c are prediction and control horizon with the condition $T_c \leq T_p$ and $\boldsymbol{z}(.)$ is a solution to the optimization problem – a quadratic programming decision.

The objective function J, also called the cost function, is the sum of individual terms, which each focus on the specific aspect of the controller performance [15]. The objective function is formulated in Eq. (36, 37 and 38).

$$\mathcal{J}\left(\boldsymbol{z}(k)\right) = \mathcal{J}_y\left(\boldsymbol{z}(k)\right) + \mathcal{J}_{\Delta u}(\boldsymbol{z}(k)) \quad (36)$$

$$\mathcal{J}_y(\boldsymbol{z}(k)) = \sum_{i=1}^{T_p} \{w_i^y . [r(k+i|k) - y(k+i|k)]\}^2 \quad (37)$$

$$\mathcal{J}_{\Delta u}(\boldsymbol{z}(k)) = \sum_{i=0}^{T_p - 1} \{w_i^{\Delta u} . [u_j(k+i|k) - u_j(k+i-1|k)]\}^2 \quad (38)$$

The proposed nonlinear programming solver presented in this thesis is the active set algorithm [21]. The method is based on solving the Kuhn-Tucker equations using SQP principles [21]. The SQP algorithm executed each sample period finds a solution to the quadratic optimization problem defined in Eq. (35). The solution represents an optimal manipulated variable sequence.

The output of the prediction model and manipulated variable are constrained with the condition shown in Eq. (39).

$$R := \{y(t) \in \mathbb{R}^1 | 0\, t.h^{-1} \leq y \leq 50\, t.h^{-1}\}$$

$$U := \begin{Bmatrix} u(t) \in \mathbb{R}^1 | 0\, m^3.h^{-1} \leq u \leq 200\, m^3.h^{-1} \\ -2\, kg.h^{-1} \leq \Delta u \leq 2\, kg.h^{-1} \end{Bmatrix}$$

$$X := \{x(t) \in \mathbb{R}^{12} | -\infty \leq x \leq \infty\} \quad (39)$$

The first step of the sequence is applied to the process, and the process repeats each sampling period.

III. Case Study

The simulations have been done with the step change of a disturbance signal. In Tab.2 is listed an overview of what step changes have been conducted to check the mathematical model performance. The arrows represent the direction of signal change.

TABLE II. STEP CHANGES OF MEASURED DISTURBANCES

Variable/disturbance signal name […]	Time [s]
Superficial gas velocity	2250 ↑,3250 ↓
The actual level of fluidized bed	4250 ↑,5250 ↓
Reactor gas cap pressure	6250 ↑,7250 ↓
Cycle gas specific gravity	8250 ↑,9250 ↓
Polypropylene density	10250 ↑,11250 ↓
Hydrogen density	12250 ↑,13250 ↓
Ethylene density	14250 ↑,15250 ↓
Ethylene volume flow rate into the reactor	16250 ↑,17250 ↓
Hydrogen volume flow rate into the reactor	18250 ↓,19250 ↑
Propylene volume flow rate into the reactor	21250 ↓,22250 ↑
Cycle gas temperature at the reactor inlet	24250 ↑,25250 ↓

The results representing the testing and verification scenario summarized in Tab. 2 are shown in Fig. 3. To verify the inteligent control system design, we propose a set of verification criteria, which will result in the final statement and best candidate.

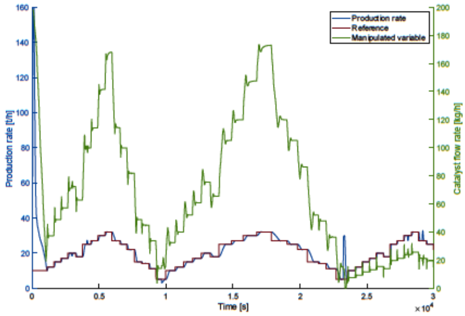


Fig. 3: Step response to the changes

IV. CONCLUSION

A. Verification and results

The controller parameters that characterize the best candidates are sampling period, prediction horizon, and control horizon. The following criteria are evaluated in this section:

1. Compensation of the measurement noise
2. Compensation of the measurement disturbance
3. Transient state stability
4. Manipulated variable stability and manipulator work required
5. Computational requirements and effect of prediction and control horizon
6. Reference tracking

The comparison between the mathematical and neural model with three dedicated neural networks is shown in Fig. 4.

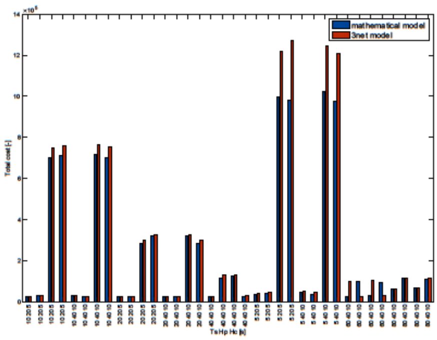


Fig. 4: Mathematical and neural network prediction performance comparison

In Fig. 4 shows better performance of the neural network version (3net) than the mathematical version for specific sample times and prediction/control horizon. The total cost is calculated as the sum of all qualitative parameters. Lower value means better performance, shorter time for execution, or less energy needed by the manipulator.

The critical quality criteria is the reference tracking ability and energy required to manipulate the regulator. The best alternatives are shown in Tab. 3.

TABLE III. BEST CONFIGURATION ACC. TO REF. TRACKING AND ENERGY REQUIREMENTS

Alternative (Cost)	Sampling Period [s]	Prediction Horizon [samples]	Control Horizon [samples]
1 (22101)	20	20	5
2 (22103)	20	40	10
3 (23846)	10	40	10
4 (23923)	10	20	5
5 (27307)	40	40	10

The alternatives 1 and 2 can be considered equally good in terms of reference tracking ability and energy requirements.

The computational time needed for the simulation is a function of neural network complexity and reflects the cost of effort needed for the required accuracy by MPC. This qualitative parameter has been included in the evaluation to present the effect of neural network size and the computational requirements. The best alternatives are shown in Tab. 4.

TABLE IV. ALTERNATIVES ACC. TO REF. TRACKING AND COMPUTATIONAL REQUIREMENTS

Alternative (Cost)	Sampling Period [s]	Prediction Horizon [samples]	Control Horizon [samples]
1 (22293)	20	20	5
2 (22982)	20	40	10
3 (25815)	10	40	10
4 (26898)	10	20	5
5 (27692)	40	40	10

The difference between alternative 1 and 2 has widened, and the best candidate became alternative 1.

Therefore, the quality evaluation and the best candidate selection were also compared according to energy and computational requirements. The best alternatives are shown in Tab. 5.

TABLE V. ALTERNATIVES ACC. TO ENERGY AND COMPUTATIONAL REQUIREMENTS

Alternative (Cost)	Sampling Period [s]	Prediction Horizon [samples]	Control Horizon [samples]
1 (310)	80	40	10
2 (465)	60	40	10
3 (614)	20	20	5
4 (719)	40	40	10
5 (778)	80	40	10

The difference between alternative 1 and 2 has widened, and the best candidate became alternative 1. According to evaluation criteria, the optimal configuration of the MPC for the specific system – polypropylene reactor is the predictive model controller with a sampling period of the 20s, prediction horizon of 20 samples and control horizon of 5 samples.

B. Selecting sampling period, prediction horizon and control horizon

The sampling period of the controller has a significant effect on the overall performance of the closed-loop response [19]. The sampling period effect has been checked with several MPC configurations and prediction and control horizons. The results are different for experiments with measurement noise applied to the feedback signals and situations without any noise. In Fig. 5 and Fig. 6 we show the effect of varying sampling periods on the qualitative parameters.

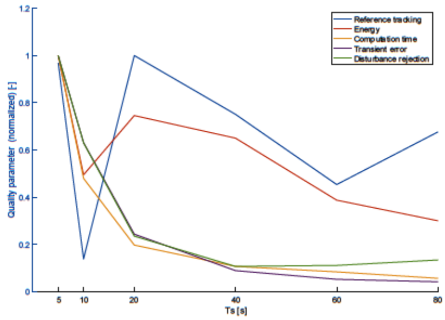


Fig. 5: Sampling period effect on MPC configuration without noise

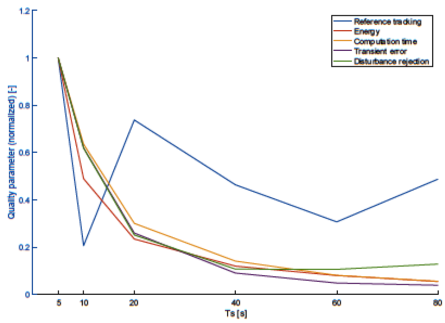


Fig. 6: Sampling period effect on MPC configuration with random noise

The article provides an insight into the kinetic modeling of the specific chemical reaction. It is shown that the approach to the mathematical modeling of the kinetic process during the polymerization reaction is generally applicable in the control theory [5]. An application of the kinetic model for MPC of a highly nonlinear process with discrete states is shown [5]. The article proposes a solution to the control problem using a hybrid controller, which uses neural networks to predict and model the non-measurable states and presents a solution to the discrete states in a hybrid prediction model with discrete sub-model switching. Another achievement to be considered is the scalability and flexibility of the proposed control system. An analysis of the controller performance for multiple system configurations has been conducted again. The numerous neural networks' advantage compared to a single-network solution has been proven and analyzed using objective criteria.

Implementation of the ANN for feedback estimation can eliminate the current heuristic approach, which is highly dependent on the operator. Manual input of the process control can be reduced and thus increase the effectivity and safety.

Acknowledgment

Research on Advanced Algorithms and Process Modelling in Domains of Applied Informatics (RAPDAI)
Grant Agency: Grant Agency Academia Aurea (GAAA)
Project number: GAAA/2022

References

[1] Karas, P., Kozák, Š., 2017, Artificial neural network approach to modeling of polypropylene reactor, In International Journal of Advances in Chemistry (IJAC), Vol.3, No. 3, p. 22 - 30

[2] Karas, P., Kozák,, Š., 2018, Highly nonlinear process model using optimal artificial neural network, In Cybernetics and Informatics, 29th International conference proceedings, IEEE catalog number: CFP18E30-PRT, ISBN: 978-1-5386-4419-5, Vol. 19, p. 14 – 22

[3] Shamiri, A. et al., 2015, Modified two-phase model with hybrid control for gas phase propylene copolymerization in fluidized bed reactor, In 2015 Chemical Engineering Journal, No. 264, p. 706 – 719

[4] Karas, P., Kozák,, Š., 2019, Nonlinear process prediction using neural network model, In 2019 ELITECH, 21-th Conference of Doctoral Students, 29 May 2019, FEI STU, Bratislava, Slovakia, p. 8 - 14

[5] Shamiri, A. et al., 2010, Kinetic modeling of propylene homo polymerization in gas-phase fluidized-bed reactor, In Chemical Engineering Journal, Vol. 161, p. 240-249

[6] Shamiri, A. et al., 2013, Dynamics and Predictive Control of Gas Phase Propylene Polymerization in Fluidized Bed Reactors, In Process Systems Engineering and Process Safety, Chemical Journal of Chemical Engineering, Vol. 21, 2013, pp. 1015-1029

[7] Shamiri, A. et al., 2011, Temperature Control of Industrial Gas Phase Propylene Polymerization in Fluidized Bed Reactors Using Model Predictive Control, In 3rd International Conference on Chemical, Biological and Environmental Engineering, IPCBEE, Vol. 20, p. 19 - 24

[8] Ibrehem, A.S. et al., 2008, Mathematical Model and Advanced Control for Gas-phase Olefin Polymerization in Fluidized-bed Catalytic Reactors, Chinese Joumal of Chemical Engineering, Vol. 16, p. 84 – 89

[9] Kim, J. Y., 1998, Particle size distribution and its effect on the olefin polymer reactor dynamics over olefin polymerization catalysts with multiple active sites, In Korea J. Chem. Eng., Vol. 15, No. 4, p. 451-457

[10] Luo, Z.-H. et al., 2009, Steady – state and dynamic modeling of commercial bulk polypropylene process of Hypol technology, In Chemical engineering journal, Vol. 149, p. 370 – 382

[11] Bai, F., et al., 2003, Physical Constants of Poly(propylene), The Wiley Database of Polymer Properties

[12] Alshaiban, A., 2008, Active site identification and modeling of polypropylene made with Ziegler – Natta catalysts

[13] Ibrehem, A.S. et al., 2009, Advanced Control of a Fluidized Bed Using a Model-predictive Controller, Australian Journal of Basic od Applied Sciences, Vol. 3, No. 4, p. 3954 - 3974

[14] Ibrehem, A.S. et al., 2008, Decentralized Advanced Model Predictive Controller of Fluidized-Bed for Polymerization Process, Iran. J. Chem. Chem. Eng., Vol. 31, No. 4, p. 91 – 117

[15] Mikleš J., Fikar M., 2004, Modelovanie, identifikácia a riadenie procesov II, Ed. 1

[16] Harmon Ray, W. 1972, On the Mathematical Modeling of Polymerization Reactors, In Journal of Macromolecular Science, Part C, Vol. 8:1, p. 1-56

[17] Lona, L. M. F., 2018, A step by step approach to the modelling of chemical engineering processes, Springer International Publishing AG

[18] Lunze, J., Lamnabhi-Lagarrigue, F., 2009, Handbook of Hybrid Systems Control Theory, Tools, Applications, Cambridge University Press, ISBN-13 978-0-511-64178-7

[19] Kozák, Š., 2003, Development of Advanced Control Methods and Their Successful Incorporation in the Industry, In Journal of Cybernetics and Informatics, ISSN 1336 - 4774, Vol. 1

[20] Lakhal, A.H. et al., 2010, Neural Network Observer for Nonlinear Systems Application to Induction Motors, International Journal of Control and Automation, Vol. 3, No. 1

[21] www.mathworks.com – Matlab SW package reference manual

TABLE VI: NOMENCLATURE AND USED SYMBOLS

Variable	Explanation
V_{FB}	Fluidized bed volume $V_{FB} = \pi . r_{Rx1}^2 . h_{FB} . (1 - \varepsilon_{mf})$ $[m^3]$
$C_a(t)$	The molar concentration of catalyst particles in reactor $[mol.m^{-3}]$
$F_{in}^{Ca}(t)$	Catalyst volume flow rate into the reactor $[m^3.h^{-1}]$
$Cat(i)$	The molar concentration of catalyst particles at reactor inlet $[mol.m^{-3}]$
$k_a(t,i)$	The activation reaction rate constant $[s^{-1}]$
$v_{SGV}(t)$	Superficial gas velocity $[m.s^{-1}]$

$h_{FB}(t)$	The actual level of fluidized-bed [m]
$C_{C3}(t)$	The molar concentration of propylene in reactor $[mol.m^{-3}]$
$F_{in}^{C3}(t)$	Propylene volume flow rate into the reactor $[m^3.h^{-1}]$
C_{in}^{C3}	The molar concentration of propylene at reactor inlet $[mol.m^{-3}]$
$F_{out}^{CG}(t)$	Cycle gas volume flow $[m^3.s^{-1}]$
$k_{im}(t,i)$	Initiation by propylene reaction rate constant $[m^3.mol^{-1}.s^{-1}]$
$P_0(t)$	Active site concentration $[mol.m^{-3}]$
$k_p(t,i)$	Propagation by monomer reaction rate constant $[m^3.mol^{-1}.s^{-1}]$
$Y_0(t)$	0-th moment of living polymer chains $[mol.m^{-3}]$
$k_{fm}(t,i)$	Transfer to propylene reaction rate constant $[m^3.mol^{-1}.s^{-1}]$
$C_{C2}(t)$	The molar concentration of ethylene in reactor $[mol.m^{-3}]$
$F_{in}^{C2}(t)$	Ethylene volume flow rate into the reactor $[m^3.h^{-1}]$
C_{in}^{C2}	The molar concentration of ethylene at reactor inlet $[mol.m^{-3}]$
$k_{ic}(t,i)$	Initiation by ethylene reaction rate constant $[m^3.mol^{-1}.s^{-1}]$
$k_{pc}(t,i)$	Propagation by ethylene reaction rate constant $[m^3.mol^{-1}.s^{-1}]$
$k_{fcm}(t,i)$	Transfer to ethylene reaction rate constant $[m^3.mol^{-1}.s^{-1}]$
$C_{H2}(t)$	The molar concentration of hydrogen in reactor $[mol.m^{-3}]$
$F_{in}^{H2}(t)$	Hydrogen volume flow rate into the reactor $[m^3.s^{-1}]$
C_{in}^{H2}	The molar concentration of hydrogen at reactor inlet $[mol.m^{-3}]$
$k_H(t,i)$	Transfer to hydrogen reaction rate constant $[m^3.mol^{-1}.s^{-1}]$
$M_{in}^{Ca}(t)$	Catalyst mass flow rate into the reactor $[kg.h^{-1}]$
ρ_{Ca}	Catalyst suspension density $[kg.m^{-3}]$
$M_{in}^{C3}(t)$	Propylene mass flow rate into the reactor $[kg.h^{-1}]$
$\rho_{C3}(t)$	Propylene density $[kg.m^{-3}]$
p_{Rx1}	Reactor gas cap pressure [bar]
p_{atm}	Atmospheric pressure [bar]
T_{atm}	Atmospheric temperature [K]
T_{ref}	Reference temperature [K]
$M_{in}^{C2}(t)$	Ethylene mass flow rate into the reactor $[kg.h^{-1}]$
$\rho_{C2}(t)$	Ethylene density $[kg.m^{-3}]$
$M_{in}^{H2}(t)$	Hydrogen mass flow rate into the reactor $[kg.h^{-1}]$
$\rho_{H2}(t)$	Hydrogen density $[kg.m^{-3}]$
M_w^{C3}	The molecular weight of propylene $[kg.kmol^1]$
M_w^{C2}	The molecular weight of ethylene $[kg.kmol^1]$
M_w^{H2}	The molecular weight of hydrogen $[kg.kmol^1]$
$P_1(t)$	Active site concentration with the length of 1 $[mol.m^{-3}]$
$P_l(t)$	Active site concentration with the length of l $[mol.m^{-3}]$
T_{ret}	Reactor retention time [s]
$Y_1(t)$	The first moment of living polymer chains $[mol.m^{-3}]$
$Y_2(t)$	The second moment of living polymer chains $[mol.m^{-3}]$
$X_0(t)$	0-th moment of dead polymer chains $[mol.m^{-3}]$
$X_1(t)$	The first moment of dead polymer chains $[mol.m^{-3}]$
$X_2(t)$	The second moment of dead polymer chains $[mol.m^{-3}]$
c_p^{PP}	Specific heat capacity of polypropylene at 342 K $[J.mol^{-1}.K^{-1}]$
$\rho_{PP}(t)$	Polypropylene density $[kg.m^{-3}]$
$T(t)$	The temperature in the reactor [K]
$\tilde{\rho}_{CG}(t,i)$	Cycle gas specific gravity $[kg.m^{-3}]$
T_{in}	Cycle gas temperature at the reactor inlet [K]
A_{FB}	Cycle gas to fluidized bed heat transfer coefficient
$-H^d(i)$	The heat of reaction – deactivation $[J.mol^{-1}]$
$+H^{im}(i)$	The heat of reaction – initiation by propylene $[J.mol^{-1}]$
$+H^{icm}(i)$	The heat of reaction – initiation by ethylene $[J.mol^{-1}]$
$+H^H(i)$	The heat of reaction – transfer to hydrogen $[J.mol^{-1}]$
$+H^{fm}(i)$	The heat of reaction – transfer to propylene $[J.mol^{-1}]$
$+H^{fcm}(i)$	The heat of reaction – transfer to ethylene $[J.mol^{-1}]$
$-H^{dl}($	The heat of reaction –reaction poisoning $[J.mol^{-1}]$
$+H^P(i)$	The heat of reaction –propagation by propylene block $[J.mol^{-1}]$
$+H^{pc}(i)$	The heat of reaction –propagation by ethylene block $[J.mol^{-1}]$
ε_{mf}	The void fraction at minimum fluidization [-]
$\rho_{H2}(t)$	Hydrogen density $[kg.m^{-3}]$

Authors

Ing. Peter Karas, PhD.
petokaras@gmail.com
He finished his studies at the Faculty of Electrical Engineering and Information Technology in the field of Automation and Control in 2005. In 2021, he completed his doctoral studies in the field of modeling, simulation and control of non-linear systems through artificial intelligence methods. Currently he works as a project engineering manager in Slovenské elektrárne a.s. (Slovak powerplants company). His main interest is the management and technical preparation of investment projects for conventional and nuclear power plants.

prof. Ing. Štefan Kozák, PhD.
Faculty of Informatics, Pan-European University, Bratislava, Slovakia
stefan.kozak@paneurouni.eu
Currently at the Institute of Applied Informatics at the Faculty of Informatics, Pan-European University in Bratislava. His research interests include system theory, linear and nonlinear control methods, numerical methods and software for modeling, control, signal processing, IoT, IIoT and embedded intelligent systems for digital factory in automotive industry.

International Conference Cybernetics & Informatics 2022
Visegrád, Hungary, September 11th - September 14th, 2022

Monitoring of Discrete-event System Controlled by Revolution Pi Using 3D Engine

Erik Kučera, Oto Haffner and Dominik Janecký
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava
Bratislava, Slovakia
Email: erik.kucera@stuba.sk

***Abstract*—In this paper, we will address the issue of digitalisation of various production processes, using modern information and communication technologies. This trend falls under the fourth industrial revolution, Industry 4.0. One of the challenges is the creation of advanced human-machine interfaces (HMIs) for cyber-physical systems. New forms of interfaces can be implemented in augmented reality. The paper therefore deals with the connection of a discrete-event system with 3D engine Unity. 3D engines are used to develop augmented reality applications.**

***Index Terms*—Unity engine, augmented reality, mixed reality, plc, mechatronics**

I. Introduction and Related Works

Augmented reality (AR) could be called one of the biggest technology trends of our time, and it continues to expand [1]. One reason for this is that smartphones and other devices that support augmented reality are much more affordable. The most commonly used applications nowadays make use of a smartphone's camera, which the user can use to view the real world on the screen of a smartphone, or other device that supports this feature, and then rely on an augmented reality application to enhance this real world in a number of ways through various digital overlays, such as [2]:

- Adding routes in real time
- Overlaying images, digital information or 3D models
- Inserting labels
- Changing the appearance with filters

In our perspective, we would focus on the use of augmented reality in industry and specifically in automation [3]. This technology is used by many companies in industry, where data from CAD systems is used to train workers or students and also standardizes workflows and enables collaboration [4].

The first example is Siemens. This company has particularly focused on using augmented reality to support many types of human-centric activities. An example is a maintenance application where technicians are guided through routine transmission maintenance by the Assist AR software product. Once the technician uses LIDAR to scan a realistic model of the transmission, the process begins. Assist AR then provides instructions by highlighting individual elements, such as bolts or faceplates, in the correct order, in a workflow on the computer screen. Also, this application animates the individual elements to show how the parts are removed, replaced or even reassembled (Fig. 1) [5].

Fig. 1. Demonstration of Assist AR when disassembling the gearbox [5]

In its simplest form, augmented reality is mainly used to identify a component, a machine or a piece of equipment or an entire factory and then offer information about that section. More sophisticated applications also include interactive 3D models or holographic animations depicting the various steps that need to be performed.

Augmented reality in industry is used in a number of ways. The first use would be to train new workers, or to guide already experienced workers through various workflow improvements. This style helps the workers to achieve faster and also better skill acquisition and also it adds to the confidence of the workers and users for field use.

As a second way, we could mention the identification of assets. An example might be an operator or technician who needs to quickly find individual assets from many products. Augmented reality can help in this regard and identify these assets quickly, easily and reliably at the same time.

During the Covid-19 pandemic and due to various travel restrictions, another usage of augmented reality developed, namely the transmission of important information or even knowledge at a distance. However, this part has not only expanded due to the pandemic but also due to the lack of experienced personnel. In fact, older and more experienced workers were not able to travel to the field to assist their less experienced colleagues. This is where augmented reality helped a lot. In fact, with this technology, the more experienced workers were able to see what their colleagues or local

978-1-6654-8775-7/22/$31.00 © 2022 IEEE

engineers were seeing and were able to assist them remotely.

Modern forms of visualisation are now realized on the basis of the development of new ICT technologies (e.g. interactive applications made in 3D engine [6], virtual reality or augmented/mixed reality) [7]. Visualisation of process modelling, identification and control of complex mechatronic systems, elements and drives using virtual and augmented/mixed reality allows students to get a much quicker and better understanding of the studied subject compared to conventional education methods.

The ability of AR to assist in remote communication and also to make collaboration more efficient is a step forward for process design or facility layout. Thus, we can talk about a fourth way in which augmented reality is being used in industry. The user himself can place a virtual device in a space in the real world in order to verify the layout of the system or whether the space allocated for this system is sufficient. This prevents miscalculations or major budget cuts before the actual installation of the desired system [8].

Implementing augmented reality is much more difficult than implementing virtual reality itself. This is because virtual reality does not require the exact dimensions of the real world, as is the case with augmented reality. Thus, it needs elements of the different parts of the real world for precise placement and overlapping. Thanks to the gaming industry, many of these problems have been solved, and many developers who have worked for the production sector have borrowed the technology from the gaming industry.

Computer networks are now used for more than just connecting personal computers. Even at the level of specific sensors and actuators, smaller devices can connect to them. Modern single-board computers and microcontrollers that can be quickly connected to the global Internet are responsible for this trend. The result is a new paradigm - Internet of Things (IoT) as an integral part of the Industry 4.0. Without it, the vision of the fourth industrial revolution would not be possible. In the field of digital factory is a successor of machine-to-machine (M2M) communication. Currently, industrial HMI (human-machine interface) panels, consoles, web or mobile applications are used to control and monitor mechatronic systems in Internet of Things (IoT) networks. Using these conventional control and monitoring methods of mechatronic or cyber-physical systems within IoT networks, this method may be fully satisfactory for smaller rooms. Since the list of devices fits on one screen, we can monitor the status and control these devices almost immediately. However, in the case of several rooms or buildings, which is the case of digital factories, ordinary ways of interacting with mechatronic or cyber-physical systems become troublesome. In such case, there is the possibility to apply advanced digital technologies such as augmented reality. Using these technologies, digital (computer-generated - CGI) objects can be inserted into the real world. The aim of article [9] was to describe design and implementation of a new method for control and monitoring of mechatronic systems connected to the IoT network using augmented reality to create an innovative form of HMI.

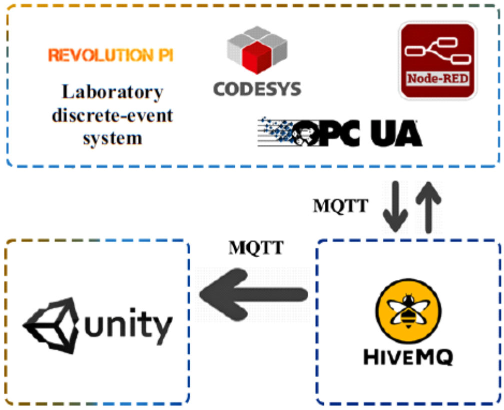


Fig. 2. Scheme of the application system

This article [9], however, was concerned with the creation of an augmented reality HMI for a system that was controlled by a simple microcontroller suitable for hobby projects and prototyping. However, there is a need to take this issue further towards Industry and Industrial IoT. Therefore, the present paper discusses the possibilities of creating an HMI in augmented reality for a system that is controlled by an industrial computer (PLC). As a first step, it was necessary to connect the 3D engine with the PLC and the discrete-event system. This is the subject of the present paper.

II. Monitoring of Discrete-event System using PLC and 3D Engine

Fig. 2 shows a scheme of the system that was implemented to demonstrate the monitoring of a discrete-event system using a graphical user interface (GUI) programmed in Unity 3D engine. The monitored variables are those of the Fischertechnik laboratory discrete-event system. This kit is controlled by an open-source PLC (or sometimes it is called industrial PC - IPC) of the Revolution Pi series. The control program (ladder diagram) is created using the CODESYS development environment. This environment also offers the creation of an OPC UA server, which has also been realized. In addition to the control program and the OPC UA server, the PLC also runs a program implemented using Node-RED. This program reads variables from the OPC UA server and sends them using the MQTT protocol to the HiveMQ broker, which is located in the cloud. The application in Unity 3D engine can use MQTT protocol to receive data from HiveMQ and display it to the user.

A. Connection of laboratory discrete-event system and PLC

In the beginning, it was necessary to connect the individual components of the laboratory system. These were PLC Revolution Pi, RevPi Connect + feat. CODESYS (Fig. 3). Next was the RevPi DIO, i.e. the digital input/output module, the

Fig. 3. Revolution Pi, RevPi connect + feat. CODESYS

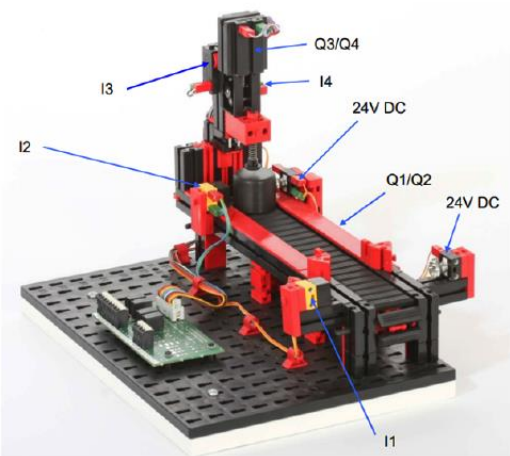


Fig. 4. Punching machine with conveyor Fischertechnik 96785 with sensors and actuators description

MDR-60-24 DIN rail power supply and the Fischertechnik 96785 (Fig. 4). This was a punching machine with a conveyor belt.

In Fig. 5 we can see the final wiring of the laboratory system.

During the solution, it was necessary to work with the operating system that resides on the Revolution Pi. This is a modified Raspbian, with which it was possible to verify the individual functions of our PLC. The Revolution Pi could be connected to a monitor using a micro-HDMI connector and as our PLC also had 2 USB ports, it was possible to plug in a mouse and keyboard for additional control.

After powering on the operating system, a login was required in the Raspbian command prompt. The login credentials were provided by the manufacturer. After a successful login,

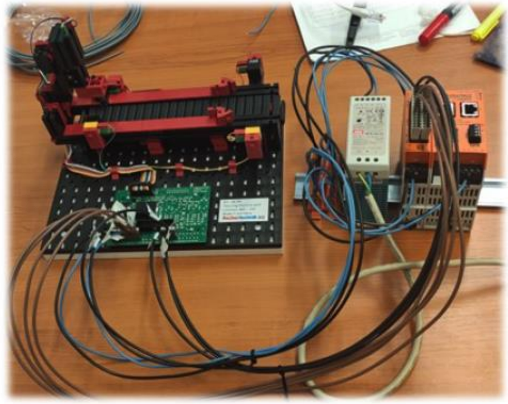


Fig. 5. Laboratory system wiring

the option was to continue in the command line style, or in the so-called Raspbian graphical interface. We chose the graphical interface option, which was entered using the *startx* command in the command line.

For further tasks, it was necessary to provide an internet connection to our operating system in the Revolution Pi. Since the PLC itself had two RJ45 Ethernet interfaces, we connected our PLC using an RJ45 wire - first to the router and then to the computer itself, which we used to program the PLC. For further work, we found out the IP address to which our system was connected, and proceeded with other tasks.

B. Communication and installation

After successful connection of the whole system and correct connection to the internet network, we were able to continue working with our system. Using the IP address we found in the previous step, we moved on to the PiCtory configuration (Fig. 6). PiCtory is a web browser application that aids in the communication and configuration of the PLC. After logging on to the localhost of our PLC, we also composed our system in virtual form using PiCtory. Thus, we have a PLC with a DIO module connected to it. Using this application, it was possible to set up and annotate the individual pins that we used in our project.

We used 4 input and 4 output pins. Output pins were under numbers 1-4, which we used to control the conveyor motor forward and backward and the punching machine motor up and down. Input pins controlled the two photosensors on the edges of the conveyor and the buttons on the punching machine. We can see the specific input/output PLC pins (left side of the list) controlling specific parts of the machine (right side of the list):

- `I_11` - Button I3 (bottom)
- `I_12` - Button I4 (top)
- `I_13` - Photosensor at the end of the belt (I1)

Fig. 6. PiCtory

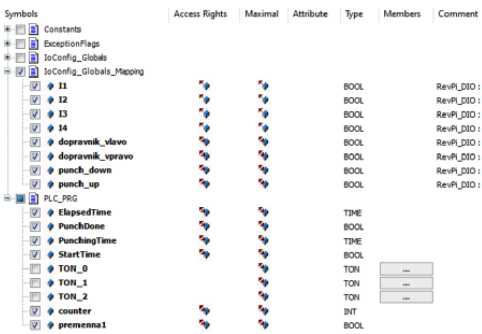


Fig. 7. Symbol configuration

- `I_14` - Photosensor under the punching machine (I2)
- `O_1` - Conveyor left
- `O_2` - Punching machine up
- `O_3` - Conveyor right
- `O_4` - Punching machine down

C. PLC programming

For programming the PLC system itself we used the CODESYS program, with which our Revolution Pi was compatible. This was CODESYS V3.5 SP17 Patch 2, which we used on a separate computer running Windows. It was then necessary to include our RevPi and RevPi DIO module in the program.

To program the control system, it was necessary to determine the variables. There are two ways of specifying variables, either globally or locally in the program. In our program we have specified variables in both ways. The variables controlling sensors *I1* to *I4* and the variables controlling the motors of the laboratory system *O_1* to *O_4* were determined as global variables, which we register directly at the inputs and outputs of the DIO module. Variables such as counter, time or auxiliary variables for controlling the system, we have determined as local variables directly in the program.

After determining all the variables, we wrote a short code that controls our laboratory system. The program is written in ladder logic style. We chose this language for its clarity as the system will be used for educational purposes.

The program waits for the product to be inserted on sensor *I1*, which is located at the beginning of our production line. When the object is inserted at the beginning of the line, the program starts a loop where the product goes under the punching system where sensor *I2* is located. Then, when the conveyor stops, the punching system starts moving downward until sensor *I3*, which is in the form of a button, is activated at the bottom of the rails on which the punching system moves. The system punches for five minutes and then returns to its original position at the top of the rails. Sensor *I4* is used to detect the upper limit of the punching system displacement. After this cycle, the product is moved by the conveyor to the right side of the laboratory system, where the program is terminated and the next cycle begins.

D. OPC UA server

To acquire and manipulate the information produced by our laboratory system, we needed to use the Node-RED programming tool, which is located directly in our Revolution Pi. To continue in this direction, we needed certain variables

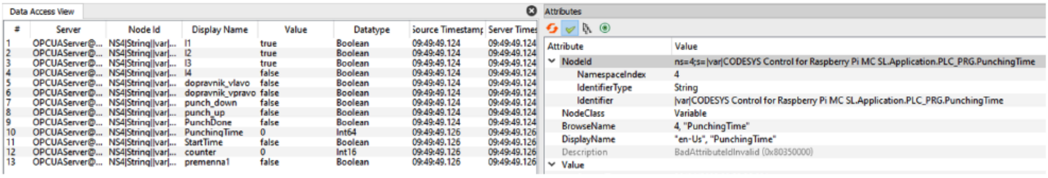


Fig. 8. Variables in OPC UA client - UAExpert

that we were interested in monitoring. These variables are needed to be published to the OPC UA server.

In CODESYS we have created a so-called Symbol configuration, where we have chosen the individual variables using a simple option, as we can see in Fig. 7.

To create the OPC UA server itself, it was necessary to set the certificate for OPC UA. In the Security screen section of CODESYS itself we searched for individual certificates that had not been created yet and created the necessary certificate.

Subsequently, it was necessary to verify the functionality of this server, but during various testing of the OPC UA server using the client in the UAExpert application, we found that there were problems with the Revolution Pi setup. We found an error in the security of the PLC itself in CODESYS. This was an additional password security on our system and also an anonymous login to the system. Once the bug was eliminated, we moved on to the actual tracking of the individual variables. For proper functionality of the entire system, it was also necessary to connect our PLC to the actual source of the internet connection and based on this fact, adjust the settings in CODESYS such as the IP address and so on. After verifying the functionality of the connection again, we connected to the OPC UA server of our PLC system and found all the variables that were needed to continue with the UAExpert program. In Fig. 8 we can see the interface of this program with the different variables.

On the right side of Fig. 8, we can see the value of our nodes that were needed to identify the observed variables in Node-RED. As an example, we can take the variable that we have shown in Fig. 8. We can see that it is the Punching time variable that we have set in CODESYS for a certain time during which our punching system creates a hole in the product. Since the PLC was not punching at this point, the value was zero. For us, the important data from this program was *NodeId* and *DataType*. *NodeId* in this case was *ns=4;s=|var|CODESYS Control for Raspberry Pi MC SL.Application.PLC_PRG.PunchingTime*, where we can see that for each variable. The majority of the ID value did not change and the last part of ID changed based on the name of the variable used. We used *DataType* for the data type of the variable so that we could use it in the Node-RED program afterwards.

E. Node-RED

Since all the necessary information was collected, we proceeded using Node-RED. The first task was to activate this service in the PLC system using the PiCtory interface, which then guided us to the correct IP address and port number to use our Node-RED. After successfully navigating to this IP address, it was necessary to test the connection to the OPC UA server located at our desired address. We did this by first installing all the necessary add-ons in Node-RED, specifically this was *node-red-contrib-opcua*. This add-on offered us many additional nodes to work with the OPC UA server and client. Then, using the OPC UA client node in Node-RED, we connected to the server.

Since our task was not only to get data from the PLC but also to send it to Unity, we chose the MQTT broker HiveMQ for this task. It is a freely available cloud broker for various uses and allows connecting IoT devices. We created functions for each variable, which we used to split the variables into different MQTT topics.

In Node-RED, to send data to the MQTT broker, we need an *mqtt out* node. This node is used to forward individual data to topics on our broker's server.

In Fig. 9 we can see the final version of the Node-RED we have set up, which contains all the necessary elements and nodes to continue with the project.

F. Creating an interactive application in Unity engine

Since all the previous steps have been completed, we started to look at the topic of the interactive application itself. For the interactive application, we decided to choose the Unity engine.

The first thing to do was to connect Unity to the MQTT broker.

The next step was the creation of the application itself. For functionality we created two scripts. The first one is called *mqttReceiver*. This script was then assigned to a newly created game object in Unity called *MQTT_Receiver* (Fig. 11). Here, the broker address and port needed to be set. It also needed to subscribe to a topic, here specifically *Diplomova_Praca_I1*.

The second script that needed to be created was *mqttController*. Again as before, we needed to insert this script into a new game object in Unity, which we called *ControlledObject*. We also inserted the previously created object into the Event Sender entry, for communication with each other.

After successfully testing the functionality of the data extraction from HiveMQ, we displayed all the observed variables in the application by duplicating all the game objects that were needed. Each variable was assigned a text describing which variable it is in that row, as well as the value of the variable

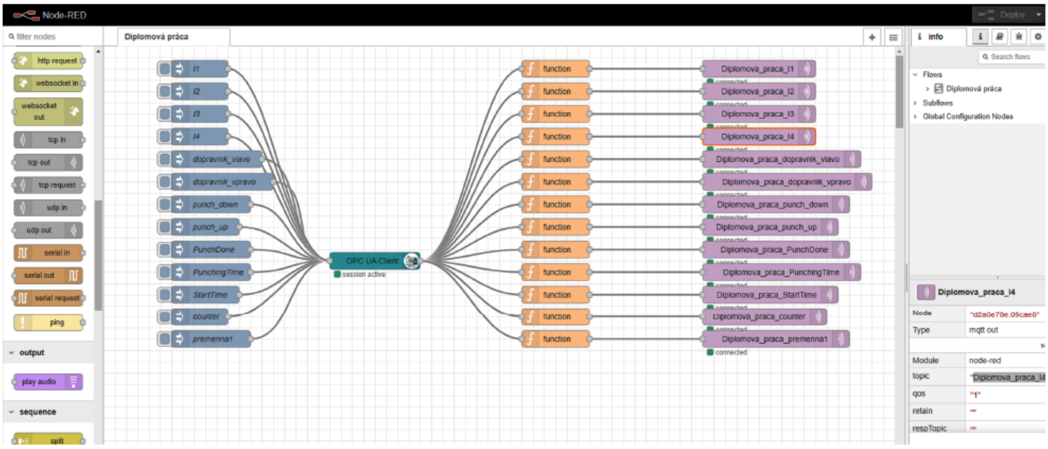


Fig. 9. Final version of the Node-RED program for transferring data to the MQTT broker

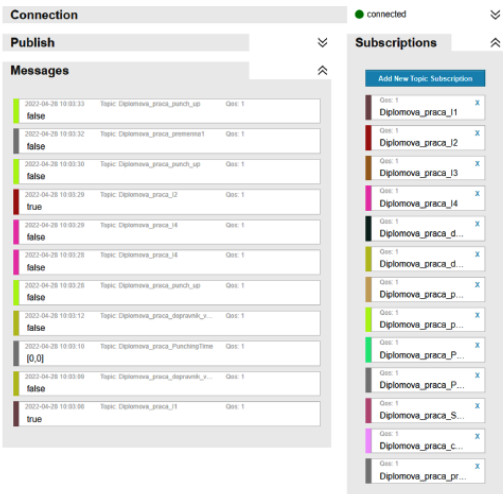


Fig. 10. Insight into broker and correctness of transmitted variable data

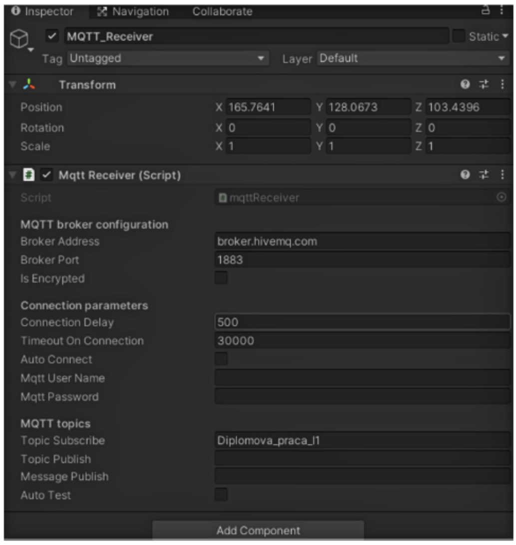


Fig. 11. Modification of MQTT_Receiver

itself. In Fig. 13, we can see the hierarchy that makes up our application.

It was then necessary to run the interactive application and make sure that the data from the PLC system was correctly sent via Node-RED and HiveMQ to Unity. In the following figures (Fig. 14-16) we can see the finished and functional application with the individual descriptions at the beginning of the whole process. We can notice that instead of specific values, a dash is placed to represent a process not yet started.

III. Conclusion

Currently, during the ongoing Industry 4.0 revolution, new forms of human-machine interfaces (HMIs) are being sought for monitoring and control of mechatronic (cyber-physical) systems. Such HMIs can be implemented using augmented reality. In order to realize such a concept, it is necessary to connect a 3D engine with the controlled system. Therefore, the paper dealt with the interfacing of open-source PLC (Revolu-

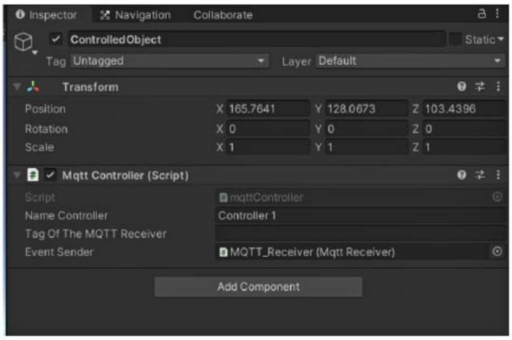


Fig. 12. ControlledObject settings

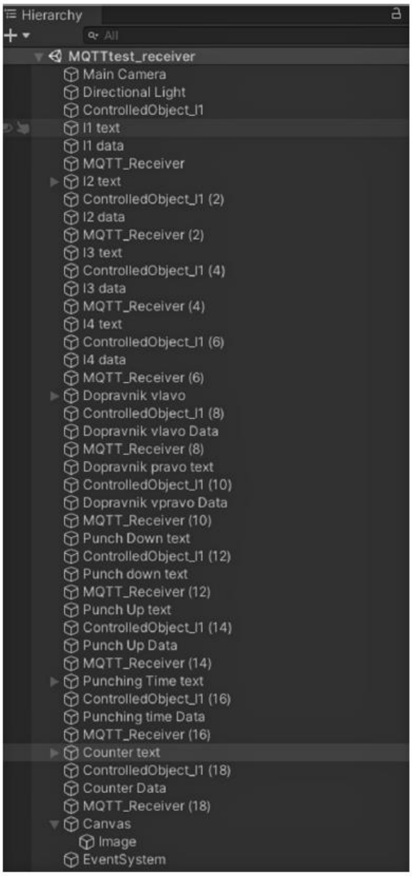


Fig. 13. Interactive application hierarchy in Unity

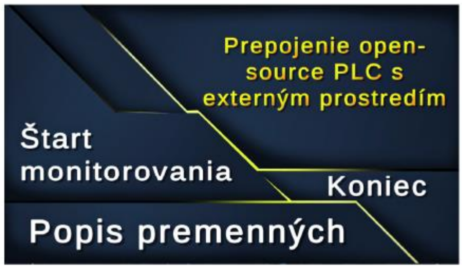


Fig. 14. Application menu

Fig. 15. Variable labels and descriptions

tion Pi series) with the application implemented in 3D engine. OPC UA server, Node-RED middleware and MQTT broker were used for their interconnection. The future challenge will be the implementation an augmented reality application in the 3D engine Unity. The current application only works on the computer screen. However, it is an important start for testing the possibilities of interfacing discrete-event PLC systems with Unity 3D engine.

ACKNOWLEDGMENT

This research was funded by the Slovak research and Development Agency under the contract no. APVV-21-0125, by the Cultural and Educational Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic

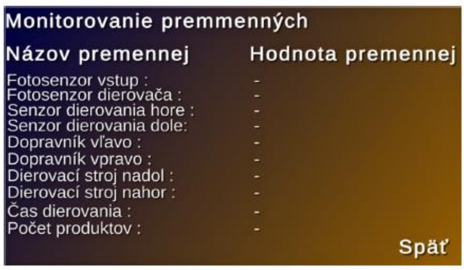


Fig. 16. Variable monitoring

KEGA 016STU-4/2020 and 039STU-4/2021, by the Scientific Grant Agency of the Ministry of Education, Research and Sport of the Slovak Republic No. 1/0107/22. Special thanks for the help with initial analysis of current applications and software implementation to Katarína Gajdošíková.

REFERENCES

[1] P. A. Rauschnabel, A. Brem, and Y. Ro, "Augmented reality smart glasses: definition, conceptual insights, and managerial importance," *Working paper, The University of Michigan-Dearborn, Tech. Rep.*, 2015.
[2] B. Houston. (2020) What is augmented reality (ar)? a practical overview. [Online]. Available: https://www.threekit.com/blog/what-is-augmented-reality
[3] K. Sloan. (2016) Rotor brings toyota showroom 360 to life with unreal engine. [Online]. Available: https://www.unrealengine.com/showcase/rotor-brings-toyota-showroom-360-to-life-with-unreal-engine
[4] D. Parvati, W. L. Heinrichs, and Y. Patricia, "Clinispace: a multiperson 3d online immersive training environment accessible through a browser," *Medicine Meets Virtual Reality 18: NextMed*, vol. 163, p. 173, 2011.
[5] Siemens. (2022) Manufacturing work instructions. [Online]. Available: https://www.plm.automation.siemens.com/global/en/products/manufacturing-planning/manufacturing-work-instructions.html
[6] Triseum. (2017) Variant: Limits. [Online]. Available: https://triseum.com/calculus/variant/
[7] J. Majernik, M. Madar, and J. Mojzisova, "Integration of virtual patients in education of veterinary medicine," in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2017. doi: 10.15439/2017F134
[8] J. R. Koelsch. (2021) How augmented reality became a serious tool for manufacturing. [Online]. Available: https://www.automationworld.com/process/iiot/article/21259479/how-augmented-reality-became-a-serious-tool-for-manufacturing
[9] E. Stark, E. Kučera, O. Haffner, P. Drahoš, and R. Leskovský, "Using augmented reality and internet of things for control and monitoring of mechatronic devices," *Electronics*, vol. 9, no. 8, 2020. doi: 10.3390/electronics9081272. [Online]. Available: https://www.mdpi.com/2079-9292/9/8/1272

Authors

doc. Ing. Erik Kučera, PhD.
Institute of Automotive Mechatronics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
erik.kucera@stuba.sk
His focus is mainly on modern information and communication technologies and their use in the context of fourth industrial revolution Industry 4.0. This icludes e.g. internet of things, virtual and mixed reality, cloud computing and new microcontrollers.

doc. Ing. Oto Haffner, PhD.
Institute of Automotive Mechatronics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
oto.haffner@stuba.sk
Research activities in applications of computer and machine vision in mechatronic systems using modern ICT such as microcomputers, cloud, computational intelligence, etc. Current focus also on the use of deep learning in practical applications for Industry 4.0 such as visual quality control. He is also involved in modelling and simulation of production processes in manufacturing.

Ing. Dominik Janecký
Institute of Automotive Mechatronics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
He studies PhD. progamme Mechatronic Systems. His focus is mainly on virtual and mixed reality and their use in the context of new human-machine interfaces.

Modular e-Commerce Data Warehouse using Microservices

1st Adam Žák
Faculty of Informatics and Information Technologies
Slovak University of Technology Bratislava
Bratislava, Slovakia
zak.adamx@gmail.com

2nd Martin Bobák
Institute of Informatics
Slovak Academy of Sciences
Bratislava, Slovakia
martin.bobak@savba.sk

***Abstract*—Data warehouses are systems that are used today by many companies for storing a large amount of structured data. Microservices allow splitting the whole software system into multiple parts, which can be deployed separately and work on their own. These services communicate and manage individual software functionalities.**

The paper describes a proposal of a data warehouse built with a microservice architecture. Also, the microservice data warehouse is compared with a classic monolithic data warehouse.

***Index Terms*—data warehouse, microservices, modularity, scalability**

I. Introduction

Cloud computing pawed a way to better resource utilization. It makes it possible to share infrastructure, platform, and software as a service with users who do not need to maintain the service. The next step was customization of platforms offered as a cloud service [1]. This approach is useful for many industrial as well as research communities [2] which are dealing with exascalability [3]. When lightweight resource sharing was more mature the new concept of software system design emerged. This new approach is microservices.

The paper is focused on microservices effectiveness within data warehouses which are subject-oriented, integrated, time variant, and non-volatile collection of data in support of management's decision making process [4]. Microservices as a concept are attracting a lot of attention these days and are also being implemented by many companies. Various other companies see this interest and want to also migrate to microservices. But this is not as straightforward as it can look. The microservice architecture brings many benefits to programming and development, but also many disadvantages. This study investigates parts of this problem, such as scaling, networking, comparison to the monolith data warehouse, and so on.

Corresponding author: M. Bobák (email: martin.bobak@savba.sk).
This work is supported by APVV grant no. APVV-20-0571 and VEGA grant no. 2/0125/20

978-1-6654-8775-7/22/$31.00 ©2022 IEEE

II. Microservices

Microservices can be understood as basic applications, with a simple task. These basic applications can run independently on different servers, can be scaled individually, according to the current server load, tested autonomously, and so on [5]. The precise definition is: "Microservices are a software development technique, a variant of the service-oriented architecture (SOA) architectural style that structures an application as a collection of loosely coupled services [6]."

Microservices are building on containerization. Splitting software into multiple separately running parts, was not possible in the past or was very costly and hard to maintain. But thanks to virtualization of operating system it is much easier and cheaper. It allows us to run multiple instances of the execution environment. These instances are behaving and looking like their servers, with their dynamic resource management [7].

A. Advantages and disadvantages of microservices

Thanks to a different approach that microservices bring, it is possible to achieve multiple advantages over monolithic architecture. The main advantages are the following [8], [9]: easier maintenance of source code, faster development, scaling of application, simplified testing, and easier bug fixing.

The different approach of microservices brings a lot of advantages, but it also brings disadvantages. A software architect has to take into account [10]–[12]: performance of the system, economical aspects of the solution, management of microservices, communication between services, hard Bug fixing (within the system as the whole), and test integration .

B. Micorservice and monolith approach comparision

Monolithic architecture is well-known approach, that is used regularly today. When developers build software with this monolithic approach, they think of it as a whole program in one piece. It contains everything, necessary for running, like frontend, backend, or management tools. Also in this architecture, developers usually have one database for the whole system [13].

This is one of many comparisons, that is needed to analyze before decision between monolithic and microservice architecture is made. Based on the study [14], it is possible to identify crucial points such as: maintenance, deployability, team organization, cost, scalability, reusability, size of project, size of team, and price of migration.

On the other hand, microservice architecture is still a very young and evolving approach, and as such, still has many potential problems. Some of these problems are: microservice granularity, language diversity, code management, and managing API changes [15].

III. Related work

There are two types of data warehouses, local and cloud. Normal warehouses are the classic way of using them. Cloud warehouses on other hand are using cloud architecture. They are stored in computers elsewhere, not under your control and you access them by internet. This work is more focused on cloud-based data warehouses, as they are more used today.

TABLE I
COMPARISON OF THE STATE OF THE ART APPROACHES.

Data warehouse	Pros	Cons
Amazon Redshift	Columnar storage Querying language Strong AWS integration	Uniqueness Not Enforced Structured data Not multi-cloud solution
Google BigQuery	Easier management of warehouse Pay per use Multi-cloud solution	Overkill for smaller data sets Flat pricing -
Snowflake	Easy implementation Auto-scaling Structured and semi-structured data	Customer is dependent on infrastructure Big data load No support for unstructured data.

IV. Microservice-oriented architecture for Data Warehouse.

The microservice system is divided into multiple parts, each working with its own related services on its own task. This is building some kind of tree structure. Each rectangle representations one microservice. Figure 1 is showing a simplified architecture diagram.

Proposed microservice flow is shown on invoices example in bigger detail. This example is part of the data warehouse that is working with invoices, loading data, and storing them. We can see a diagram of this flow in figure 2.

The proposed microservice-based data warehouse is composed of the following main parts:

Admin service - this is a microservice, which is responsible for the management of the whole data warehouse and data pipeline. In this case, it is also responsible for uploading of files.

Dispatcher - Microservice, which decides, what is the type of file by its name and type. For example, the GLS delivery service is sending .xls files, so when the .xls file arrives at the dispatcher, together with parameter type = 'gls', it will

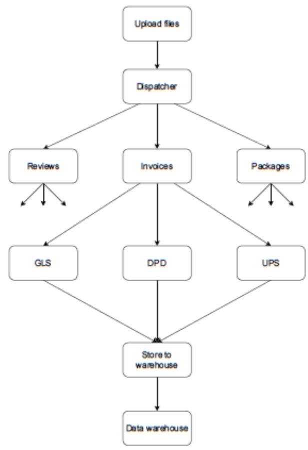


Fig. 1. Tree structure of microservice architecture.

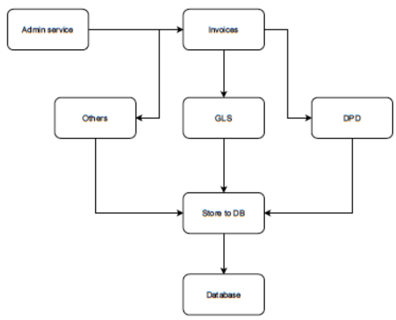


Fig. 2. Map showing services in data flow.

recognize, that it is an invoice from the GLS company. After this recognition, it chooses where to send this data.

GLS - This microservice is processing GLS invoices. It receives data, which are transformed to the requested format, and then they are forwarded. This microservice is the parser, and each type of file and company has individual parser like this.

Store to DB (DB is abbreviation for word Database) - The responsibility of this microservice is to receive data from parsers (like GLS service) and then store the parsed data correctly in the database. It must pick the right table and insert all additional important data.

Database - The purpose of this service is running a data warehouse database.

Concept sumarization The goal of the microservice concept is to divide each part of ETL into a microservice. Each part of stored data, like invoices, package information, user reviews, and so on, has its own services its own parser (like GLS for example). But they also share some services. They share storage in the data warehouse (Database), share extract (Admin service) and also share load (Store to DB).

V. Experiments

The proposed approach is evaluated by the following hypothesises: 1) Microservice-based data warehouse will be bit faster than a monolithic data warehouse, but the difference will be small, 2) Microservice-based data warehouse will be more capable to handle problems with overloaded modules, and 3) Microservices are more suitable for working with large data.

The evaluation environment has the following configuration: CPU: 1.35GHz Intel i5-4210U, RAM: 12GB GPU: Intel Haswell-ULT, and OS: Linux - Pop!_OS 20.04

A. Monolithic data warehouse comparison

The system chosen as the monolithic data warehouse is qFlow developed by SoftPoint [1]. qFlow is a workflow management application that offers multiple options, one of which is a data warehouse for your chosen files loaded from different places.

An invoice in qFlow takes an average of 1:38 to process. This is caused by many additional logs that are performed in qFlow. In the microservice program, the average time to process an invoice is 13 seconds. This is a major difference, but is not caused by a different architecture, but rather because of a different implementation (such as multiple logs on side of qFlow, but not on in microservices).

To obtain more precise values and a better understanding of the difference between these two options, the execution time has been divided by the average execution time of one invoice. This produces a number that shows how many invoices with average run time could be processed. The smaller number is better because it means that for a shorter period of time, a larger number of invoices could be processed.

This number can be compared with the numbers from experimenters with microservices-based solution. The graph 3 shows that the monolithic architecture performs better without scaling on 10 uploaded invoices. But further scaling of processor containers enables better utilization and higher speed, which reflects on smaller division number.

This graph shows that it performs even better than microservices when they are not scaled. With further scaling, the execution time is reduced and microservices start to perform better than the monolith.

The graph with 20 invoices shows that even in the default configuration of microservices, they perform better than a monolith. Scaling speeds the program, but runtime does not go down below 10,75 which is 2 minutes and 9 seconds.

The last test was performed on 30 invoices in the system. Again, it is visible that even in the default configuration, the microservices performed better. Scaling the reduced time even more to 15.6 division number, which is equal to 3 minutes and 8 seconds.

[1]qFlow: https://softpoint.tech/qFlow/

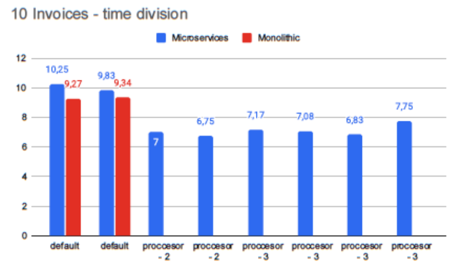


Fig. 3. Division number comparison of 10 invoices uploaded to system

Fig. 4. Division number comparison of 20 invoices uploaded to system.

B. Effectiveness tests

The first type of experiment aimed at a different speed in containers of different scales. With the use of a larger number of invoices or reviews, bottlenecks are more significant and have a larger impact on the program. This is demonstrated, for example, in figure 6. One of consequences is a slower

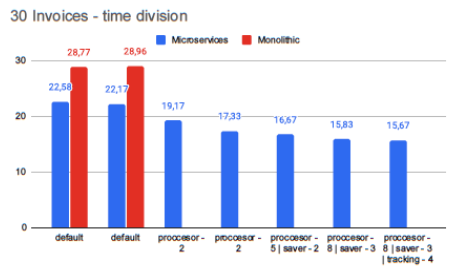


Fig. 5. Division number comparison of 30 invoices uploaded to system

execution time, as shown at the beginning of table II and also in table III.

1) Invoice effectiveness test: The initial test case was in the invoice part of the program (GLS invoice). An invoice is uploaded to the Admin Service, then processed in the processor, and saved in saver. In a case like this, this is enough. But in the case of multiple invoices uploaded at once, this is not enough. Five invoices uploaded at once create this kind of flow diagram as shown in figure 6.

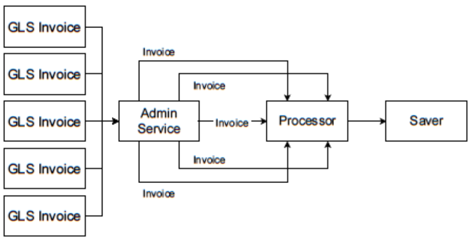


Fig. 6. Flow of 5 invoices in program.

Figure 6 clearly shows that in the case of containers scaled as this, the processor container creates a bottleneck in the program. While one invoice is being processed, the rest are waiting. This slows the program and can be easily solved by scaling the number of containers.

When there is only one container per application module and 10 invoices are being uploaded, their processing takes between 2:02 and 2:07 minutes. An additional processor container saved 36 seconds. This is a big advantage of microservices. A container that creates a bottleneck can be deployed multiple times, and the bottleneck can be removed.

In the case with 10 uploaded invoices, the additional horizontal scaling does not anymore shorten the execution time. This is because this amount of invoices is not enough to better utilize additional containerization.

2) Reviews effectiveness test: In this test scenario, the review processing speed is being tested. Reviews are downloaded once a day using a cron job in the background. To test this part of the program, this background job was not scheduled once a day but 10 times (or more in some cases). One file has around 6000 reviews, so 10×6000 reviews equal to 60000 reviews that need to be parsed and saved.

This part of the program has two options to run. One is classic, the same as invoices; processors receive one XML file, parse it, and then send parsed data to the saver, which stores those into the database. Another option is to go review, by review in the XML file, parse this one, send it to background job, which then calls saver, which then stores this one review into the database.

These measurements are taken in the second-mentioned version, where each line of the file has been stored separately.

In a default configuration, where each container is deployed only once, 10 review files were processed in 11 minutes and 21 seconds. In the version, with two processor containers, the execution time was 9:41 minutes. This means that 1:40 is obtained, thanks to the double processor container. The next experiment was with two single_review containers. With two of these containers, the run-time was 9:37 minutes. This is very much the same as for double processors. Another experiment was a combination of 2 single_reviews and 2 savers. This releases pressure from the saver, as it was one container that received data from 2 containers. This manifests itself also in time, which is 7:10 minutes. That means another 2:27 minutes saved.

The last 3 experiments were with 20 and 30 review files. In the case of 20 files and 20 single_reviews, 20 savers, the execution time was 7 minutes and 9 seconds. In the case of 30 files and 30 single_reviews, 30 savers, the execution time shortened to around 1 minute and 20 seconds. This is the best execution time achieved in review-based testing. 30 files, 30 savers, and 30 single_reviews are the best set-up of microservices in the reviews part of the program.

TABLE II
MEASURED TIMES OF EXECUTION IN DIFFERENT CONFIGURATIONS. TIMES ARE SLIGHTLY HIGHER BECAUSE COMPUTER WAS DIFFERENTLY LOADED (WHICH RESULTS IN 30 ADDITIONAL SECONDS AS IN DIFFERENT TESTS).

Test type	Execution time	Scale
10 GLS invoices	02:05	default
10 GLS invoices	02:03	default
10 GLS invoices	01:58	default
10 GLS invoices	01:38	processor - 2
10 GLS invoices	01:31	processor - 2
10 GLS invoices	01:24	processor - 2
10 GLS invoices	01:21	processor - 2
10 GLS invoices	01:26	processor - 3
10 GLS invoices	01:25	processor - 3
10 GLS invoices	01:22	processor - 3
10 GLS invoices	01:33	processor - 3
10 GLS invoices	02:09	saver - 2

TABLE III
MEASURED TIMES OF EXECUTION IN DIFFERENT CONFIGURATIONS.

Test type	Execution time	Scale
10 Heureka reviews	11:21	default
10 Heureka reviews	09:41	processor - 2
10 Heureka reviews	09:37	single_review - 2
10 Heureka reviews	07:10	single_review - 2 — saver - 2
10 Heureka reviews	05:29	single_review - 5 — saver - 5
10 Heureka reviews	03:31	single_review - 10 — saver - 10
10 Heureka reviews	03:20	single_review - 20 — saver - 20
10 Heureka reviews	03:17	single_review - 20 — saver - 20
30 Heureka reviews	02:05	single_review - 30 — saver - 30
30 Heureka reviews	01:20	single_review - 30 — saver - 30
30 Heureka reviews	01:19	single_review - 30 — saver - 30

C. Network tests

This section focuses on testing the network and its use. Almost all tests are executed on the review processing part of the solution. The measured numbers show directly the network load and how it grows with scaling.

The first test was done using Heureka reviews, also with two processor containers. The network load is roughly the same, around 162 - 165 KiB/s. After adding a saver container, the network load changes. With one saver container, the load is approximately 162 KiB/s, with two savers, the load is approximately 243 KiB/s.

This grows directly further; more containers equal more network load. The case with 20 savers and single_reviews is no exception. A double number of containers means that the network load also doubles. This network load exceeds 1 MiB/s.

Scaling also affects the network load at rest, without any request, as TCP keep-alive messages are sent to more containers. The average load was equal to 23.5 KiB/s.

D. Load size test

The graphs show that with the increasing number of invoices, the gap between microservices and the monolithic solution is increasing. In the case of 3, the difference between the fastest container configuration and the monolithic solution is approximately 3 division numbers; with 20 invoices, the difference increases to about 10 and in the case with 30 invoices, it increases to up to 13. But this also applies to comparison among microservices themselves. The biggest gap between fastest and slowest times in differently scaled containers is increasing.

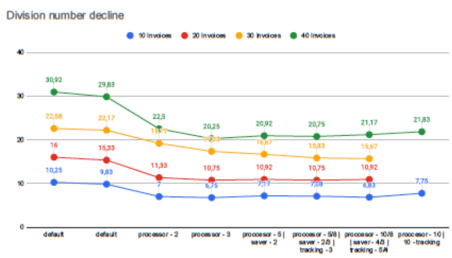


Fig. 7. Comparison of different load and execution times.

This is shown in the graph 7. With an increasing number of uploaded invoices, the execution time automatically increases. The first two points in the graph are always with the default number of containers, which means that there are no additional containers, only 6. The third point is where an additional processor container was added. In all four conditions tested with invoices, the second processor container helped. Except for 30 invoices, where growth was slightly slower, in each larger configuration, this difference was greater than in the previous test.

E. Discussion

In this section results of experiments and related hypotheses are discussed.

1) Hypothesis number 1: : "Microservice-based data warehouse will be bit faster than a monolithic data warehouse, but the difference will be small." was proved to be true in this case. With the right configuration, microservices work faster. This difference is becoming more and more visible with a larger and larger load.

The containers are performing better in the tests, they are faster than the monolith, thanks to scalability. However, these tests also proved that they can be slower in some scenarios.

The question for developers now is: is it better to build a whole program with microservices, or is it better to stay with the classic monolithic approach?

The experiments showed that they also need the right setup for it. Without this setup, they can cause more harm than benefit. This can be seen again in the graphs, which show that, with scaling, there was acceleration in execution time, but without it, for example, with 10 invoices, the program ran slower.

2) Hypothesis number 2: The original hypothesis is as follows: "It will be easier to solve problems with overloaded parts of the data pipeline." This assumption is addressed in V-B, which talks about execution time speed. The running time of the program depends on the load. In the event of overload, bottlenecks arise and slow down the program. The given section examines how the execution time changes with the change in the number of scaled containers.

The monolithic solution lacks options to address these unexpected cases of overload. If such an unexpected problem arises, it means that the program needs to be rewritten or better computer units can be bought to improve performance. Both of these options are costly and time-consuming.

On the other hand, microservices offer a great solution with scaling. It is directly visible in tables II and III. As the number of scaled containers grows, the execution time decreases. This is happening because scaling removes bottlenecks.

All experiments are tested on the same computer with the same configuration. And just one additional step in deployment shortened execution time from circa 11 minutes to somewhere in between 1:30 and 2:30 minutes.

Our experiments confirms that multiple containers make better use of hardware infrastructure and parallelization. The first image shows the utilization of cores without scaling any containers. The second image shows how cores are utilized with the processor container scaled three times. More CPU load is placed on the processor, but the execution time is much faster. All of this shows that the second hypothesis is true.

3) Hypothesis number 3: states: "Microservices are most efficient when working with large data.".

Experiments showed that with larger loads, microservices offer faster speed than monoliths. In cases with a smaller number of uploaded files, qFlow performance was very close to microservices, sometimes even better.

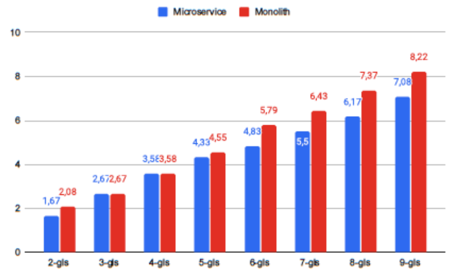


Fig. 8. Different division number from 1 uploaded invoice to 9 invoices.

The graph 8 shows how the gap between the monolithic and microservice solution is increasing with the increasing number of invoices. In cases with 3 and 4 invoices, the division number is even the same for both systems.

As the number of files increased, qFlow started to slow down compared to microservices. The graph 5 shows that in a good microservice configuration, the gap between monolithic and microservices is the 13 division number (28-15). This is a continuation of the trend from 8, where the largest difference is one division number. Note that the containers were not scaled in this configuration, so the difference could be larger. But also if we compare nonscaled version of microservices, in case with 30 invoices, the division number difference is around 6-7.

This hypothesis is also proved in the experiments with reviews. In this case, the program was not compared to qFlow, but to itself. Reviews, as mentioned above, offer two options to run. In one case, they are less containerized, and in another they are more containerized.

The detail is shown in graph 9, where the blue line represents the more divided and containerized part of the program, and the red line represents the simpler and less containerized part.

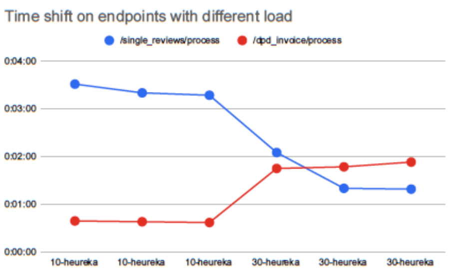


Fig. 9. Graph of time shift on endpoints with different load.

It proves Hypothesis 3 and also proves that microservices are not a good option for smaller systems. Review processing is designed to download one review per day and process it. There is no need for microservices and, in this scenario, they are creating more of an obstacle than helping.

VI. Conclusion

The monolithic qFlow data warehouse is compared to the microservice-based data warehouse. Invoice experiments have shown that microservices are performing faster in most cases, but they have also shown that in many cases this difference is not large. The difference increases with the larger number of invoices. However, to achieve this, the microservice data warehouse must be correctly configured. This is not an easy task and can be quite difficult to find the right setup of containers.

The research in this work proved that microservices are a better and easier option when it comes to scaling applications. Experiments showed that the option to scale different parts of the program as needed can dramatically change the execution time of processes in the system. Monolithic applications, on the other hand, need more work and time when they encounter a situation like this.

References

[1] M. Bobák, L. Hluchý, and V. Tran, "Tailored platforms as cloud service," in *2015 IEEE 13th International Symposium on Intelligent Systems and Informatics (SISY)*, 2015, pp. 43–48.

[2] M. Graziani, I. Eggel, F. Deligand, M. Bobák, V. Andrearczyk, and H. Müller, "Breast histopathology with high-performance computing and deep learning," *COMPUTING AND INFORMATICS*, vol. 39, no. 4, p. 780–807, Jan. 2021. [Online]. Available: https://www.cai.sk/ojs/index.php/cai/article/view/2020_4_780

[3] M. Bobák, A. S. Z. Belloum, P. Nowakowski, J. Meizner, M. Bubak, M. Heikkurinen, O. Habala, and L. Hluchý, "Exascale computing and data architectures for brownfield applications," in *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2018, pp. 450–457.

[4] W. H. Inmon, "What is a data warehouse," *Prism Tech Topic*, vol. 1, no. 1, pp. 1–5, 1995.

[5] X. Larrucea, I. Santamaria, R. Colomo-Palacios, and C. Ebert, "Microservices," *IEEE Software*, vol. 35, no. 3, pp. 96–100, 2018.

[6] B. Christudas, *Practical Microservices Architectural Patterns: Event-Based Java Microservices with Spring Boot and Spring Cloud.* Apress, 2019.

[7] K. Kolyshkin, "Virtualization in linux," *White paper, OpenVZ*, vol. 3, no. 39, p. 8, 2006.

[8] J. Thönes, "Microservices," *IEEE software*, vol. 32, no. 1, pp. 116–116, 2015.

[9] B. Mayer and R. Weinreich, "A dashboard for microservice monitoring and management," in *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*, 2017, pp. 66–69.

[10] P. Utomo and F. Falahah, "Conceptual model of a dashboard for monitoring microservices," *EAI Endorsed Transactions on Cloud Systems*, vol. 6, no. 18, 2020.

[11] D. Shadija, M. Rezai, and R. Hill, "Microservices: granularity vs. performance," in *Companion Proceedings of the10th International Conference on Utility and Cloud Computing*, 2017, pp. 215–220.

[12] D. Savchenko, G. Radchenko, and O. Taipale, "Microservices validation: Mjolnirr platform case study," in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015, pp. 235–240.

[13] M. Kalske, N. Mäkitalo, and T. Mikkonen, "Challenges when moving from monolith to microservice architecture," in *International Conference on Web Engineering.* Springer, 2017, pp. 32–47.

[14] F. Auer, V. Lenarduzzi, M. Felderer, and D. Taibi, "From monolithic systems to microservices: an assessment framework," *Information and Software Technology*, vol. 137, p. 106600, 2021.

[15] Y. Wang, H. Kadiyala, and J. Rubin, "Promises and challenges of microservices: an exploratory study," *Empirical Software Engineering*, vol. 26, no. 4, pp. 1–44, 2021.

Authors

Bc. Adam Žák
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava, Slovakia
zak.adamx@gmail.com
His study field covers computer science as a whole, from neural networks, and data engineering to computer graphics and games. His main professional activity is oriented toward microservices and their application within data warehouses. Currently, he is focused on the use of semantics in both these areas of computer science.

Mgr. Martin Bobák, PhD.
Institute of Informatics
Slovak Academy of Sciences in Bratislava, Slovakia
martin.bobak@savba.sk
His field of research is multi-cloud computing and architectures of distributed, cloud-based applications. He is the (co-)author of scientific publications and has participated in several European (e.g. H2020, Horizon Europe) and Slovak R&D projects. He is also a university lecturer at the Faculty of Informatics and Information Technologies, Slovak University of Technology.

International Conference Cybernetics & Informatics 2022
Visegrád, Hungary, September 11th - September 14th, 2022

Remote Data Acquisition Using Cloud Database

Filip Žemla and Ján Cigánek
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava
Bratislava, Slovakia
filip.zemla@stuba.sk, jan.ciganek@stuba.sk

***Abstract*— The presented paper deals with the design and implementation of a complex SCADA system for the control of a production line model and for the remote data acquisition using cloud database. The control of the educational model of the production line is based on a PLC device using the visualization application of the production process in real time and recording of process data in a cloud database. The main components of SCADA systems used in control processes, advanced software for creating SCADA applications and basic information about cloud databases are also defined. It is described the design and implementation of the control program in the TIA Portal environment and the design of the visualization application in the Promotic environment. The form of a cloud solution based on the Microsoft Azure platform for the needs of process data storage is presented.**

Keywords— SCADA system; HMI; Visualization system; PLC; Production line; Cloud database; Data acquisition;

I. Introduction

Digitization with automation is significantly changing our world and the world of industry. The volume of data in modern industrial plants is constantly growing, which brings new challenges for SCADA system solutions. In the long term, it is necessary to manage and archive huge volumes of data that are demanding on the capacity of storage devices. The task of modern SCADA systems is to monitor and control production processes as reliably, efficiently and productively as possible, while it is necessary to constantly optimize them. The means of automation have expanded from large automated lines to smaller ones, to individual machines, buildings or even smart homes [1].

The possibility of checking and obtaining data in a more accessible way, mutual communication between devices and their control or editing over local networks or via the Internet is increasingly required. These devices enable the remote collection of data into databases, based on which it is possible to evaluate both the devices themselves and the units to which they are connected [2, 9].

The advantage of the automation of production processes is almost continuous operation with minimal human intervention, which mainly takes care of inspection or service. It also enables production in life-threatening environments, such as explosive environments or chemical operations, where it is possible to limit the occurrence of threatening situations and close production to separate rooms or cells without the presence of people [3].

The main goal of this article is the design and implementation of a complex SCADA system for managing a reduced production line model and subsequent remote collection of process data using cloud database. This model will be controlled by a PLC device that turns on and off individual output devices based on the designed logic. The course of the automated production process is displayed in a visualization application on the main computer station, from which the line can also be controlled manually. In the event of a malfunction or error on any of the devices, an alarm is issued to notify the user of the situation. Acquired process data on the number of manufactured pieces, machine hours and generated alarms will be securely stored in a database created in the cloud space. The transition from data collection on physical database servers to cloud databases offers many advantages, but also difficult challenges. It is important to constantly optimize and modernize application solutions so that they keep up with developing technologies and, above all, to bring the highest possible efficiency of investments.

II. SCADA systems

SCADA (supervisory control and data acquisition) systems are used for real-time monitoring and remote control of processes in various industries. It derives information, transfers it to the system, displays it on the computer screen and alerts the supervising operator. It also performs the necessary analysis and checks, for example by determining the severity of the fault, and displays this information in a logical and organized manner.

SCADA systems can be relatively simple, such as one that monitors environmental conditions in a small office building, or incredibly complex, such as a system that monitors all activity in a nuclear power plant or municipal water supply.

A SCADA system allows an operator at a location central to a widely distributed process, such as an oil, gas, pipeline, or irrigation system, to make setpoint changes to remote process control units. This means opening or closing valves, operating switches, monitoring alarms and collecting required data. The benefits of SCADA systems are best appreciated when processes are very large - hundreds or even thousands of kilometers from one end to the other. In this way, operating costs are reduced, as routine checks are not required to monitor

978-1-6654-8775-7/22/$31.00 ©2022 IEEE

the operation of the equipment. The value of these benefits increases even more if the facilities are very far away and require extra effort to visit them [1, 2].

A. *Basic parts and functions of SCADA*

A SCADA system consists of several remote terminal units (RTU, PLC) that collect process data and send this data back to the master station via a communication system. The master station displays the acquired data and also allows the operator to perform supervisory control. Key features of SCADA software include user interfaces, graphic displays, alarms, trends, RTU (and PLC) interface, scalability and data access [3].

In a more complex SCADA system (Fig. 1), there are essentially five levels or hierarchies:

1. industrial instrumentation and control devices,
2. controllers and terminals (PLC, RTU),
3. communication system (infrastructure),
4. main station (HMI),
5. database system.

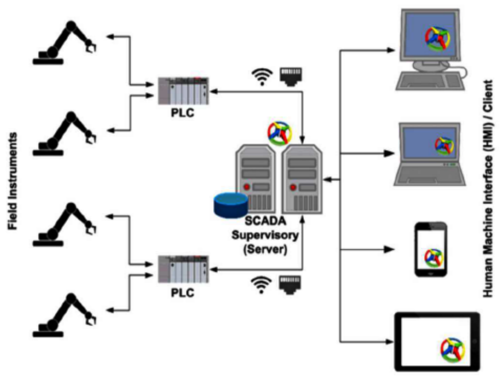


Fig. 1. Basic architecture of the SCADA system [12]

In general, SCADA systems can be described as a combination of telemetry and data acquisition techniques. SCADA collects data through a remote terminal unit (RTU/PLC) and transmits it back to the central workplace. It also performs the necessary analysis and control, and then displays this information on several screens or operator displays. The required control activities, i.e. action interventions either automatically or by the operator, are then transferred back to the process.

Accurate and timely data enable process and equipment operation to be optimized. The advantages include more efficient, more reliable and, above all, safer operation [3].

B. *PLC devices*

A programmable logic controller (PLC) is a special form of microprocessor controller that uses programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic for machine and process control purposes. It is designed to be operated even by engineers with limited knowledge of computers and programming languages. PLC designers have pre-programmed it so that the control program can be entered using a simpler and more intuitive form of language. The term logic is used because programming primarily concerns the implementation of logical and switching operations. Input devices (sensors, switches) and output devices (motors, valves, etc.) are connected to the PLC in the controlled system. The programmer enters a sequence of instructions (program) into the memory of the PLC, which then monitors the inputs and outputs accordingly and sends the control signals for which it was programmed.

PLCs have the great advantage that the same basic controller can be used in a wide range of control systems without the need for new wiring. All that is needed to change the control system and its rules is to upload a new program (instructions) to the device. The result is a flexible and cost-effective system that can be used with control systems that vary widely in nature and complexity.

The input-output interface used in a PLC can take two forms: compact and modular. The fixed type is connected to a small or micro PLC system, where all functions are integrated into one unit. The number of I/O ports is fixed within each such model and cannot be changed. Modular types use an expandable panel to hold I/O modules, allowing the number and type of modules to be changed. Input interfaces receive signals from machines or process equipment and convert them into signals that the control unit can work with. Output interfaces are used to control actuators that control some process [6, 7].

Programmable logic machines were designed so that the developer does not require a high level of knowledge of programming languages.

The entire program is executed in one cycle. It can be interrupted and then returned to continue or restarted. The IEC 61131 standard for PLC standardization, standardizes and defines the methods used for PLC programming. The most frequently used PLC programming method is based on the use of ladder diagrams (Ladder Diagram - LD), also known as the language of contact (relay) diagrams. So writing a program is equivalent to drawing switching circuits. A ladder diagram consists of two vertical lines representing electrical wires. The circuits are connected by horizontal lines, which represent quasi rungs of the ladder between the two verticals.

Another commonly used standardized programming language is the Function Block Diagram (FBD), which is used for PLC programs described in terms of graphic blocks. It is a graphical language for displaying signal and data flow through blocks, which are reusable software elements. The function block is shown as a rectangular block with inputs entering from the left and outputs exiting from the right.

A programming method that can be considered as inputting a ladder diagram using text is the instruction list (IL). Programming is done using a series of instructions, with each instruction on a new line. Each instruction consists of an operator followed by one or more operands.

The method of programming using a sequential function chart (sequential function chart - SFC) is used to visually represent the operation of the system and display the sequence of events involved in its operation. An operation is described by a number of separate sequentially connected states or steps, which are represented by rectangular boxes, each of which represents a specific state of the controlled system.

The last standardized type is the method of programming using structured text (Structured text - ST), when programs are written as a series of commands separated by semicolons. Predefined commands and subroutines are used to change variables that are defined by values, internally stored values, or inputs and outputs [6].

C. Communication system

A communications network is a collection of computer hardware and software, along with cabling, networking devices such as switches, routers, and the like, that allow computers to communicate with each other. A network provides a means of sharing information between industrial computers, PLCs, or input/output devices.

A network consists of two or more devices connected via a shared media to share files or peripherals such as modems, printers and others. It serves to connect the main central station with control units (RTU, PLC), HMI and can use industry standard or manufacturer-protected proprietary protocols. Both the RTU and PLC work autonomously in real time to control the process using the last command provided from the control system. A failure of the communication network does not necessarily mean that the process control of the device stops, but when communication is restored, the operator can continue to monitor and control [4].

Network topology is the first point to consider when designing how devices in a SCADA system are to be interconnected. Topology refers to the pattern of interconnection of various elements of computers. There are two types of network topologies, physical and logical.

Physical topology refers to the layout of network cables, their connection to computers or other network devices. A logical topology defines how data is transferred from one device to another regardless of the network hardware link. In general, the logical topology is defined by the network protocol, while the physical topology is mainly defined by the physical arrangement of network cables and devices [9].

D. Human-machine interface

The master station (MTU) is essentially a supervisory computer system. This station serves as a central processor and has two main functions that are the core of the SCADA system. The first is the regular acquisition of process data from the RTU (PLC) and secondary stations, and the second is the control of remote devices through the operator's station. MTU is a device that issues commands to RTUs located at remote locations from the control element, collects required data, processes, stores and displays information in the form of images, curves, tables for operators and thus helps to make control decisions. Communication between MTU and RTU is two-way, however, the main difference is that the RTU cannot initiate the conversation. Communication between the MTU and the RTU is initiated by programs within the MTU unit that are run either by operator instruction or automatically. In 99% of cases, instructions and messages for the RTU sent from the MTU are triggered automatically. When the master station (MTU) requests the necessary information, the RTU sends it. This principle of communication is referred to as master-slave. After receiving the required data, the MTU communicates with other devices and HMI screens for operators with which it is connected via a LAN/WAN network [4].

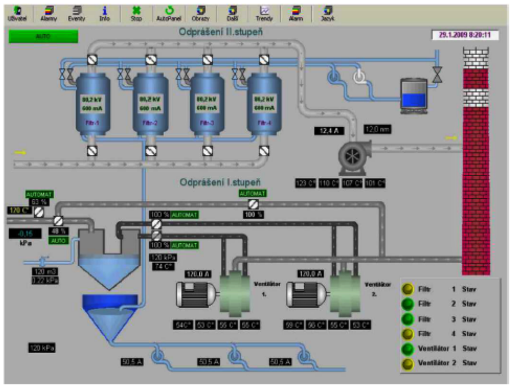


Fig. 2. Example of graphic display for controlled process [8]

HMI or Human Machine Interface is a device that allows you to visualize and work with collected process data. In most cases, this is done through a graphical user interface (GUI), such as a computer monitor, which serves the supervisory system operator and thus creates an interface between the operator and the SCADA software. This human-machine interface provides process information to operating personnel graphically in the form of diagrams and trends, which are a schematic representation of the controlled process. It is also used to display important recorded events and alarms. HMI panels are connected to the main SCADA computer, which provides real-time data for creating live diagrams, alarm displays and trend graphs.

Displayed diagrams consist of line graphics and schematic symbols that represent process elements, or they may consist of digital photographs of process equipment accompanied by animated symbols.

When controlling the operation of equipment through HMI, operators can enter commands using a mouse, keyboard or touch screen. For example, an animated tank symbol can show when the tank is full, and a colored valve symbol can show whether liquid is flowing into the tank. The operator can control the valve by clicking the mouse or touching the screen. The HMI will then display the change in liquid level in the tank in real time (Fig. 2).

In smaller SCADA systems, the master station may consist of a single computer - in which case the HMI is part of that computer. In larger SCADA systems, a master station may contain several HMIs hosted on client computers, multiple data acquisition servers, distributed software applications, and disaster recovery sites [4, 5].

E. Database system

The database system consists of two parts that can be operated by the database application or the end user:

- database,
- database management system.

The database management system handles all user requests for access to the database - it can be adding a new file to the database, deleting it or working with the data itself (creating, reading, updating or deleting data). Among the most used commercial products are Microsoft Access, Microsoft SQL Server, MySQL or Oracle Database.

A database is a shared integrated computer structure that includes end-user data and metadata, i.e. data about the structure of the database.

User data is organized into tables that represent a set of data describing the data model of the database. Table columns define individual properties (attributes) of a given data type, i.e. data type and name. The most common data types are text (alphanumeric), numeric (numeric), logical (TRUE/FALSE), and date (dates). Rows (records) of tables represent entries in the database, and each entry contains data from all defined columns in a precisely given structure.

A set of tables, each of which represents a certain type of data, can be called a relational database. Relational databases make it possible to capture not only data about researched objects, but also their mutual relationships, which makes them closer to the real world. Each record of the table is uniquely identifiable by a primary key, and the order of the records does not matter. A primary key is a property (attribute) or combination of properties that uniquely identifies a record, i.e. this column must be different in each record [10].

III. Microsoft Azure

Azure is a professional cloud-based platform provided by Microsoft. Thanks to the advanced functionality, it is possible to operate virtual machines, manage SQL databases, create backup copies of resources and many other actions without worrying about failure or out-of-date equipment and software. All operating systems and a wide variety of programming languages are supported.

MS Azure provides access to integrated services, including computing power, storage options, applications and data, thanks to which it is possible to quickly manage the infrastructure, implement a larger number of tasks and, above all, achieve significant financial savings. MS Azure is a combination of IaaS (Infrastructure as a Service) and PaaS (Platform as a Service) and was designed for access to all services, computers and applications without interrupting availability. The Azure system provides an SLA contract guaranteeing operation at the level of 99.95%, continuous monitoring of the services used and 24-hour permanent support. Azure offers access to services from anywhere in the world. In addition, this system is based on a worldwide network of data centers that is constantly expanding [11].

IV. Case Study

The equipment used in this article is a basic Siemens S7-1200 series PLC equipment that controls a laboratory model of a production line. We monitor the course of the production process in a visualization running on a personal computer connected to the PLC.

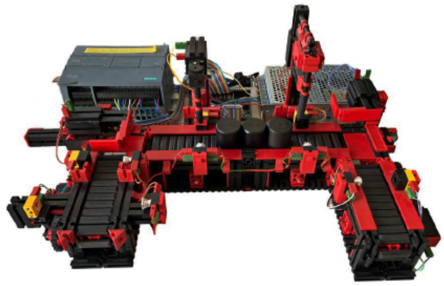


Fig. 3. Laboratory model of the production line

We created the control program in the TIA Portal tool due to compatibility with the used PLC device. Programming consisted of ladder diagram programming. In addition to control, we also ensured diagnostics of the line in the form of alarms, which were triggered when individual motors were started for a long time and stopped the entire production.

Another task was the design and creation of a visualization application. For this purpose, we chose SCADA software Promotic because of its simple, intuitive operation, high scalability, reliability and previous experience with this software.

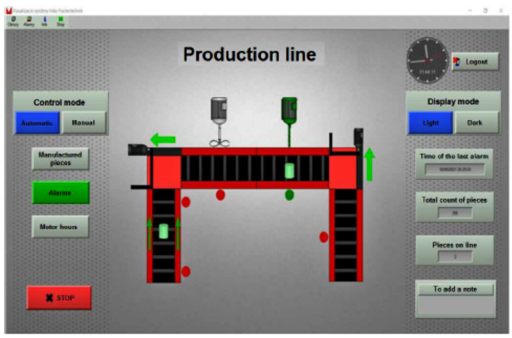


Fig. 4. The main visualization panel

Our SCADA application consists of two main screens – first one with automatic mode and the other with manual mode. From these main screens, it is further possible to switch to modal pop-up screens that display alarms, the number of manufactured pieces and an overview of the machine hours.

The application also works with a local database, in which generated alarms, engine hours and the total number of manufactured pieces are recorded. After launching the application, the first thing that appears is the login screen, which prompts the user to enter a login name and password. After entering the login information, the "Main Panel" will appear.

The main home screen of our application consists of a graphic visualization of the production line, which runs in fully automatic mode. On the graphic shown, we can start the production process and monitor the current position of the workpieces on the line, i.e. specifically at which of the optical sensors the workpiece is located. In case of movement of any belts or sliders, this movement is represented by green arrows next to the given object.

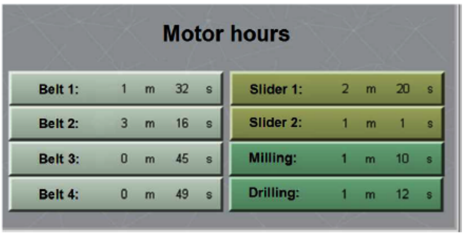
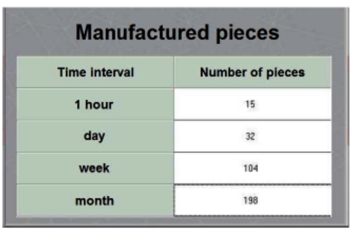


Fig. 5. Modal watch window

By pressing the "Motor hours" button, we open a new modal pop-up window, which contains an overview of the output devices and their total operating time since the PLC was turned on. Activity time is displayed in minutes and seconds for the needs of our project. In real operation, this information can be used for device diagnostics and maintenance planning.

The "Manufactured pieces" button opens a modal window displaying a table with production statistics at intervals of hour, day, week and month. In real operation, this information is essential for the higher corporate layer and production planning.


Manufactured pieces

Time interval	Number of pieces
1 hour	15
day	32
week	104
month	198

Fig. 6. Modal window of manufactured pieces

As a cloud service, we chose Microsoft Azure, specifically the SQL Database, which is used to create and manage databases. Subsequently, we chose to create a new database. Here we chose to simultaneously create a new SQL server, for which we defined the name, data center location, capacity and administrator account. There are also other setup configurations to choose from, such as security features, restoring a database backup, or adding tags. Azure offers an integrated SQL editor for managing databases and executing queries, but for better clarity we will use a different tool. We will access our cloud database using Microsoft SQL Server Management Studio (SSMS), which is clearly the most robust SQL database server management tool. It is a fully compatible and verified database management tool, but similar editors such as HeidiSQL or DBeaver can also be used.

In the next step, it was necessary to create tables in our database, in which we will write individual process data. Similar to the local database, here too we have created tables for the number of manufactured pieces, motorcycle clocks and alarms. To each table we assigned the appropriate attributes with the corresponding data type.

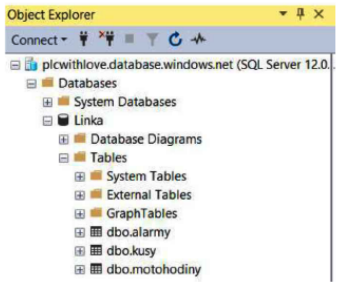


Fig. 7. Database tables

We will use the dataFEED OPC Suite tool to connect the PLC to the database and then record the process data. The configuration concept of this tool is based on so-called data sources and data targets. The source is PLC variables (tags) storing process data (engine hours, number of manufactured pieces, alarms) and their target is individual database tables.

In data destinations, there are various services to choose from such as MQTT Broker, Siemens MindSphere, OPC clients and data files to which data can be sent or written. We chose SQL Database as the data target, since we will send the data to the SQL database. First of all, we had to set up the ODBC driver, with which the communication will take place. ODBC (Open Database Connectivity) is a standardized, open interface for accessing various database management systems. Through ODBC drivers, applications can directly issue instructions to databases or execute queries. Here we have set up our SQL server address, login information and the specific database we will be connecting to. Successful execution of the database connection test completes the process. In the end, we just wrote the process data into the connected database at certain time intervals or when the value of the variable changed. We chose a new database action, where we had the choice of direct preconfigured writing with the INSERT command, connecting a stored procedure or executing a custom SQL command. In the case of recording the number of pieces and engine hours, we used the user-friendly option of direct input, in the case of recording alarms, we executed our own SQL command.

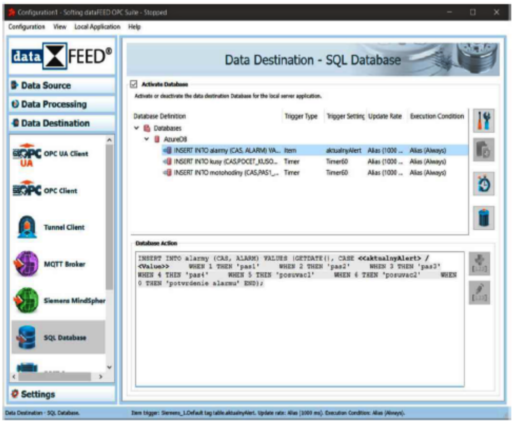


Fig. 8. Configuring the SQL database as a data target

V. Conclusion

The goal of the work was to design and implement a complex SCADA system for managing a reduced model of the production line and subsequent remote collection of process data using cloud database. The architecture of this system is composed of technological, application and cloud layers. To create it, we used available hardware, network and software components found in the school laboratory, which are used for educational purposes. The technological layer of the system consists of a PLC machine and a controlled model of the production line, which can work in automatic mode based on the designed control program. The entire production process is monitored through visualization on the main SCADA station, which fulfills the control and diagnostic function and thus forms the application layer. Acquired process data on alarms, manufactured pieces and equipment hours are stored in a cloud database on a virtual server.

In addition to the required properties, the visualization also contains additional functions such as a login screen, adding notes or changing the color display mode. The implemented complex SCADA system can be used in the educational process in the topics of visualization systems, PLC programming and storage of process data either in the cloud or locally. A more complex solution to the current system design would be to completely transfer the system to the cloud and remotely connect to the control network.

Acknowledgment

This paper was supported by the Slovak Grant Agency VEGA 1/0107/22 and KEGA 016STU-4/2020, 039STU-4/2021, by the Scientific Grant APVV-17-0190 and project No. ZDPVCC.

References

[1] D. Bailey and E. Wright, "Practical SCADA for Industry", Newnes, 2003. ISBN 9780080473901. Available: https://doi.org/10.1016/B978-0-7506-5805-8.X5000-4.

[2] B. R. Mehta and Y. Reddy, "Industrial Process Automation Systems", Butterworth-Heinemann, 2014. ISBN 9780128010983. Available: https://doi.org/10.1016/C2013-0-18954-4.

[3] G. Clarke and D. Reynders, "Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems", Newnes, 2004. ISBN 9780080480244. Available: https://doi.org/10.1016/B978-0-7506-5799-0.X5015-3.

[4] "Master Terminal Units (MTU) in SCADA Systems", In: ElectricalVoice. Available: https://electricalvoice.com/master-terminal-units-mtu-in-scada-systems/.

[5] A. Gerer, " HMI – new knowledge and the best skills for the design of operator's interface. ATP Journal [online]. Available: www.atpjournal.sk/rubriky/prehladove-clanky/hmi-nove-poznatky-a-najlepsie-skusenosti-pri-tvorbe-operatorskeho-rozhrania-2.

[6] W. Bolton, "Programmable Logic Controllers", 6th ed. Newnes, 2015. ISBN 9780081003534. Available: https://doi.org/10.1016/C2014-0-03884-1.

[7] E. Alphonsus and M. Abdullah, "A review on the applications of programmable logic controllers (PLCs)", Renewable and Sustainable Energy Reviews, ISSN 1364-0321. Available: https://doi.org/10.1016/j.rser.2016.01.025.

[8] "What is PROMOTIC", In: Promotic. Available: https://www.promotic.eu/cz/pmdoc/WhatIsPromotic/WhatIsPromotic.htm.

[9] S. McCrady, "Designing SCADA Application Software", Elsevier, 2013. ISBN 9780124170353. Available: https://doi.org/10.1016/C2012-0-06034-6.

[10] D. Kroenke and D. Auer, "Database", Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

[11] "What is package of Microsoft Azure?", In: Senetic. Available: https://www.senetic.sk/azure/.

[12] "What is SCADA?", In: ECAVA INTEGRAXOR. Available: https://www.integraxor.com/what-is-scada/.

Authors

Ing. Filip Žemla
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
filip.zemla@icloud.com
Currently a student of doctoral studies at Slovak University of Technology in Bratislava. The main focus of his studies is oriented to virtualization and optimalisation of modern manufacture processes. His main skills are SCADA systems, database systems and front-end programming.

Ing. Ján Cigánek, PhD.
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
jan.ciganek@stuba.sk
He was born in 1981 in Malacky, Slovakia. He received the diploma and PhD. degree in Automatic Control from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology (FEI STU) in Bratislava, in 2005 and 2010, respectively. He is now Assistant Professor at Institute of Automotive Mechatronics FEI STU in Bratislava. His research interests include optimization, robust control design, computational tools, SCADA systems, big data, and hybrid systems.

ISSN: 2453-7497 (online)
ISSN 1338-6468