User-friendly tools for new digital education in computer sience

Alžbeta Kanalikova
Department of Control and Information Systems
University of Zilina
Zilina, Slovakia
alzbeta.kanalikova@feit.uniza.sk

Eva Rakovská
Department of Applied informatics
University of Economics in Bratislava
Bratislava, Slovakia
eva.rakovska@euba.sk

2024 International Conference on Emerging eLearning Technologies and Applications (ICETA) | 979-8-3315-2771-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICETA63795.2024.10850856

***Abstract*— In today's world of pandemic era, it is important to develop new ways of e-learning. E-learning is also associated with finding creative innovation as a part of digital learning, which includes a variety of IT tools as multimedia and smart tools. The article is focused on preparing and implementing innovative tools in computer science teaching. Computer science teaching is traditionally based on explicit exact knowledge. It leads to memorizing the steps or searching the ready solutions on websites from the students' side. In the article, we analyse the problems of computer science teaching online and the experience of using a few innovative elements in e-learning courses last year. Based on gained experience we choose appropriate tools for collaboration, interactivity, tools that record the sequence of steps, and tools for creating scenarios, focusing on gamification in teaching. The aim of the article comes from such experience and shows examples of how to implement multimedia and tools in a creative way for algorithmization and programming. We also try to use the interaction with the chatbot, which responds directly to the students' suggestions and helps them with basic terms. The specific results of our analyses and practical use are discussed and supported with figures.**

Keywords*— *Algorithmization, Programming, Computer science teaching, Chatbot, Digital learning, Programming.

I. INTRODUCTION

The last year shows the necessity to use computers and various types of software at an advanced level. The computers start to be the daily gadgets for more people than in previous times. Computer skills are needed for everyday life especially for young people. Computer literacy starts to be the required human feature in society. The boom of using IT as a daily routine brings a new challenge: the necessity of preparation new well-educated IT experts as developers of new applications, information systems and systems which use the artificial intelligence. Although this challenge may seem "great", it is not an easy task, as the hardware is evolving very fast and the required software for the new technology needs to be developed quickly. The software is the "soul" of ubiquitous technique, so we need more and more qualified IT people, as the labour market shows us. For this reason, the teaching of informatics should be paid considerable attention. Not only in technical schools, universities but also in more humanities-oriented universities. The trend at Slovak universities is to decrease the number of hours addressed to informatics. Many universities come from an assumption that the students should learn basic computer skills in secondary school. For example, skills in how to use the office suite (e.g. Microsoft Office) or how to deal with operation systems (usually MS OS Windows). This assumption seems to be incorrect because the secondary schools have not enough qualified IT teachers. How to teach informatics on different levels for different students in the digital era and in the "covid online time" requires enormous pedagogical effort and excellent technical skills.

The teaching of informatics etc. whether in schools or various courses, has its specifics and the problems, which can be summarized as follows [1]

- The concept of informatics is not always clearly understood. Often the content of the course depends on the opinion and knowledge of teachers, especially at secondary schools. Sometimes informatics is understood as technical principles of computer hardware, other times it is the perception of information transfer and management in companies or organizations (information management) or how to use the web and other Information Systems etc. From such an ambiguous understanding of the term "informatics" in teaching, it is not easy to find common guidelines on how to teach informatics and which digital tools to use for teaching.
- The teaching of computer subjects (whether of a technical or non-technical nature - mathematical, cognitive, social, managerial, etc.) cannot follow all details in long-term processes because the development of IT is too fast. These rapid IT changes are not possible to add to the teaching process, because the teachers have not enough time and opportunities to study all novelties. It is necessary to build the knowledge core and then to show how the core is involved in all methods, techniques, and principles of informatics.
- Informatics as a subject is offered from small children to seniors so it is necessary to accept the age, the ability of self-education, the ability to deal with technology. Mainly by using digital teaching. Young children often work on the principles of experimentation and easily remember their actions and processes. It is often in contrast to older people, who are more conservative, afraid of experimentation, they need more instruction and memorization of learned procedures. Sometimes it also depends on the style of their work in the past, or on the directive approach in their past education.
- The groups of students are often very heterogeneous in terms of previous computer literacy, in relation to technology, in logical reasoning, etc., but also in terms of motivation to learn in the field of informatics.
- IT subjects cannot be taught only with a "pencil and paper" or by talking about IT. Thus, information technologies are also a tool for teaching about themselves. Teaching algorithmization, programming,

development of applications, etc. requires to use of other software and it is not easy every time. Information technologies can act not only as a presentation tool or support for the teacher, but they can even be a teacher (as a virtual tutor).

There are certainly more problems related to the teaching process of informatics, but the aim of the work is to focus on teaching informatics at the university level. If we take a closer look at the experience with teaching the study programs, which include several subjects concerning IT, we can characterize students from several perspectives [1] which have strong impact on the way of studying. Their expectations of the course also affect their learning style (memorizing, logical style, analytical style etc.).

The specific attribute of the IT students is, that they do not like reading books including the guidelines. As it was mentioned before, the group of students usually contains different students. Therefore, to find the appropriate way how to teach them and to satisfy most of them, needs creativity, good skills in communication, and often abstract thinking of teachers. The face-to-face teaching supported by various IT and software is the optimal way how to show the students what is important, to help them find the errors, to lead them to logical thinking, and support their creativity in many ways. It is easier to solve unexpected situations in the class than by online teaching. Mainly when the teacher has a big group (more than twenty students).

The pandemic era forces the teachers to search new ways of using the various digital tools in student friendly mode. Plenty of digital tools could be used for teaching informatics and the internet offers dozens of freeware and commercial learning tools. Pedagogic art is the art of good choice. Sometimes is better to start with a low number of digital tools and then add the other tools step by step. Therefore, our paper is focused on research on how to add and implement software tools that, in addition to motivation in informatics, focus on:

- Interaction - tools that interact with the student during teaching as Chatbot, that explains the content of the course, orientation in the course, the basic definition in informatics, the differences between programming languages, and the practical implementation of the algorithm.
- Creativity in algorithm design and programming - using scenario - notation, sequence of steps, notation or representation of ideas, notation of possible solutions, conditional division, etc. Suitable tools for this area are, for example, the software tools of Scenario and planning or web application PS Diagram [2].
- Visualization and gamification - for example, graphical representation of the program result, visualization of the context of concepts or relationships, graphical output of the program. For example, the tool displays the result of programming immediately on the output graphics area. Gamification introduces elements of play, tension but also competitiveness into teaching.

Although, the programming and algorithmization of the tasks seem to be the only sequence of logical steps and there is no place for creativity, it is not true. The main idea is to offer freedom to students by using the mentioned tools, try to solve some problems in more ways, and discuss them. It is possible by using chatbots, various software which helps to develop scenario-based learning and see the visualized result of programming. On the other hand, the usage of the mentioned software could help teachers evaluate the projects and assignments in more effective ways and to find the "bottleneck" in understanding the topics within the group.

II. METHOD OF HOW TO IMPLEMENT SOFTWARE TOOLS INTO ELEARNING ENVIRONMENT

In the previous section was mentioned, that the choice of appropriate software for teaching informatics is very important. Here it is necessary to answer a few questions:

- Whether the software is compatible with the used Learning management system or the other online environment, and whether we can implement the software into the LMS or to find another possibility of using the software (open access and link),
- Whether the software is user-friendly and understandable for students; whether they can work with software without enormous effort and the software has an accessible user guide in the language of teaching,
- Whether the software tools allow to save the various students` solution and compare it, whether it allows seeing the mistakes in procedures or logical steps in program etc.

The main principle of creating an e-learning course comes from software engineering which could be described as a cyclic sustainable process [3]. This life cycle of e-learning courses is based on adding increments (technical) and new knowledge (content). The increments can be not only modules, that are built-in in LMS (e.g. in LMS Moodle modules for adding materials, modules for student activity, assignments, statistics for teachers, track progress, etc.), but as plugins or apps (e.g. multimedia plugins, antivirus plugins, plagiarism plugins, scan QRs, mobile app, etc.).

So, we present two approaches on how to implement applications into the LMS Moodle course and we point out the first experiences with some tools from the previous academic year. First is the direct implementation of chatbot application into the course environment and the second is only as a link of open-source software into the course because the policy of university e-learning does not allow to implement applications directly into the course.

III. INTERACTIVE TOOL - CHATBOT

A chatbot is an interactive tool that interacts with people and data in various online services or various applications. Modern chatbots serve as virtual assistants for customers and are based on natural language processing. A chatbot is considered as an agent or a tool of artificial intelligence called Conversational Artificial Intelligence. The large Internet companies (Google, Facebook, Microsoft) assume that chatbots are a popular technology that has a future. Microsoft and Facebook integrated the chatbots into their messaging already in 2016 and the companies provided resources for creating chatbots already in 2016 [4].

A. Chatbot principle and selection of chatbot tool

The basic principle of a chatbot was introduced in the Turing test by Alan Turing in 1950 as an imitation game. In the game, one person observes the communication of two respondents (human and computer) and the observer`s aim is to decide, who is the computer and who is the human. Turing supposed, that if the computer has been able to respond very similarly to humans, we can call the computer "a computer with artificial intelligence" [5].

Today exist more tests, that are used for testing the intelligence of software or test whether the user is not a machine. It is Marcus Test or the reverse Turing test (human tries to convince a computer that he is not a computer, e.g. CAPTCHA) or the Lovelace Test 2.0 which can create art. One of the best-known chatbots was the ELIZA as a Rogerian psychotherapist created by Joseph Weizenbaum, or later the Ultra Hal machine, Elbot, Cleverbot, A.L.I.C.E, JFred, and the others [4].

Chatbots can be divided into three classes according to a sophisticated approach [6]:

- Purposeless mimicry agent - provide the illusion of conversation. Former chatbots as ELIZA was based only on the syntax of words (coming from the theory of the formal languages). Newer chatbots from this class are based on deep learning over the dialog datasets (e.g. Ubuntu Dialog Corpus), but still, they have no knowledge about language understanding.
- Intention-based agents - these agents have a simple understanding of commands (following the intent and then try to refine details to take an action). Such agents are Amazon's Alexa, Google Home, etc. The used technologies can be again specific deep learning methods or dealing with context-free languages and appropriate grammars (theory of formal languages from Noam Chomsky).
- Conversational agents – "expand on intention-based agents to have multi-turn conversations. To do this, they must keep track of the state of the conversation and know when the person wants to talk about something else" [6]. These agents are not common, it is a challenge for developers to create a dialog manager with dialog stack and expectation agenda to watch the multi-turn conversation. minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

The offers of ready-made chatbots are growing. When looking for a suitable chatbot for teaching, we focused on well-known platforms of companies such as Google, IBM, Amazon, and Microsoft, and for comparison, we also choose one open-source platform, Wit.ai [7]. When comparing the tools, we focused mainly on the characteristics in the creation of chatbot knowledge (Figure 1 wit table) because it is important for teaching.

Based on a comparison of the features in the selected chatbots and the possibility of easily integrating the Chatbot into the LMS, we chose a chatbot from Google called Dialogflow. It is an Intent-based chatbot, limited and partially paid. Its advantages are simple implementation, validatable, built-in cloud security, and its own API (Application Programming Interface) with the possibility of expansion. On the other hand, it also contains ready-made, pre-trained templates. It is implemented in more than 50 languages and can be integrated into social networks and websites of any kind.

	Dialogflow Google	IBM WATSON	AWS Chatbot	Microsoft Azure bot	Wit.ai
Regular expressions/patterns	Yes	Yes	Yes	Yes	No
NLP for phrase match	Yes	Yes	Yes	No	Yes
Number of languages:	>50	>50	English	>50	>50
Dialogue structure	Intents	intents	intents	intents	intents
Specification of chatbot answers	Yes	Yes	Yes	Yes	-
Integration with social networks/ websites	>10	<10	<10	>10	>10
Chatbot usage via API	Yes	Yes	Yes	-	Yes
Version control: native, code-based	Yes	Yes	Yes	Yes	Yes
Validation support	Yes	-	-	-	Yes
Cloud security	Yes	Yes	Yes	Yes	-
Cost plan	Free +paid	Free up to 10k SMS/month	$0.00075 /text	Free + $29/month	Free

Fig. 1. Figure table of comparison of Chatbot platform.

B. Google chatbot and its realization

As mentioned before, Dialogflow is a Google cloud platform [8] with understanding the natural language, which allows easy design and integration of a conversational interface for websites or social networks. Dialogflow provides two different virtual agent services. Each of them has its own interface, libraries, and documentation. One is Dialogflow CX that is an advanced agent type suitable for complex agents and the second is Dialogflow ES, which is the standard one, suitable for small and simple agents. We decided to use a simpler variant of the agent - Dialogflow Essential (ES) and implement it into the LMS environment. It is possible to "train" the agent to be able to handle the conversation in various domain areas based on Intents. Therefore, it was necessary to create some pilot Intents that characterize various levels of conversation. In Intent is necessary to set [8]:

- Training phrases of respondents and actions.
- Parameters which are set according to the user-end expressions and are structured data that can be used to perform logic or generate responses.
- Agent answers as the resulting recommended answers to several questions (continue with dialog, terminate the conversation etc.).

Textual, visual, but also speech responses are defined in Dialogflow (it has speech data embedded). We decided to create a conversational chatbot for teaching in the LMS Moodle environment because the taught courses are implemented in the LMS Moodle at our universities. First, the chatbot was created and added in a "C # programming course for beginners". So, the chatbot can be integrated into the Moodle course via external integrated HTML code. You

can set up HTML block and add source code for integration, that is generated in the Google Cloud API. The resulting chatbot is implemented within the entire LMS Moodle system (Figure 2).

Here is a possibility to create several Intents for the chatbot in LMS Moodle, in addition to the basic Intents as (Hello Intent, Introduction Intent, Emotion Intent, etc.). The additional Intents have been created when applied to a specific course. In our case the course "C # programming course for beginners" and intents are:

- Intent concerning Programming - this intent introduces the basic concepts of programming (what is the program, algorithm, procedure, function, etc.)
- The Intent for basic input and output commands - this intent teaches input and output command in this case for the course Programming in language C # for beginners.

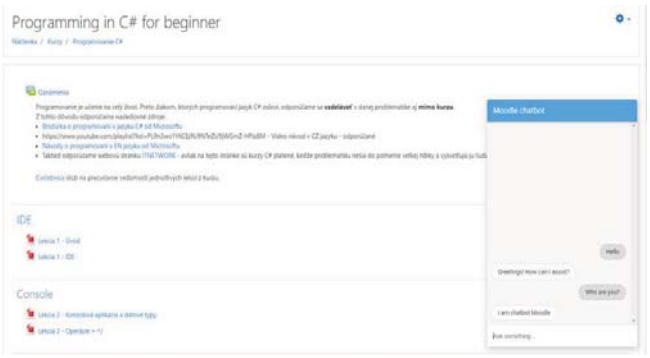

Fig. 2. Implementation Dialogflow chatbot in LMS Moodle (Source: Own)

Some other Intents have been added via Javascript. For example, Intent to recognize a programming language (C, Java, Python, etc.) and a spoken language (English, German, French, etc.). At the same time, the Entity for programming languages was created, the chatbot can distinguish the classic language mutation through a system entity. We created also the Intent for selected mathematical operations.

The Figure 3 represents a three main Intents of the Chatbot (introduction, basic definition, and basic command Intents).

Fig. 3. Chatbot Intents for course Programming for beginners (Source: Own)

The main task of using the chatbot is to create an interactive teaching guide for courses in LMS Moodle step by step. Full deployment of a chatbot requires additional time to create Intents and train the chatbot. Currently, the chatbot is in the initial phase of building. The deployment of the chatbot will be evaluated after its use in several courses. As [4] write, the main motivation of using the chatbot is productivity and then entertainment. These are both reasons for using the chatbot.

IV. SUPPORT CREATIVITY TOOL

The important task of teaching algorithmization and programming is how to support logical thinking, problem analyzing, and also how to support creativity. In reference [9] writes that students are not led to solving problems with the "integrative thinking" approach. Naiman sentence, "These young employees have never learned how to focus on the important problems - we haven't adequately taught them how to think", expresses exactly the problem of why young people are not able to find the core of the problem. Therefore, to teach a creativity is important for finding solutions to real problems that should be modelled in Information Systems. Although many developers' web pages as GitHub, Stack Overflow, etc. offer huge number of ready-made solutions for various cases, the creativity is necessary for composing the complex tasks from partly ready-made functions or solutions.

One of the tools for the development of integrative thinking is a scenario. The scenarios are appropriate tools for problem-based or case-based learning, where the students need to solve complex or badly structured problems from real life. Such problems are usually described by storytelling or presented by videos. Many research articles define the scenario in various ways, but in the context of Informatics, we focus on one type of definition. It is when "the scenario is an instrument for explaining and communicating a training project" [10]. This is from the engineering perspective and fits the best way of analyzing the tasks on various levels to a final sequential algorithm, which can be processed by computer. The scenario in computer science describes communication between users and Information systems in a functional way. The software developers have some methods and techniques for decomposition of the task to the final procedure (for example the Unified Modeling Language).

Though these techniques are well-known for developers, the students do not understand very well why and how to use these techniques. Therefore, the visualization of the task, verbal explanation of the problem, and leading the students to the solution step by step is very important. The scenarios can be presented in various ways [11]:

- As a story (storytelling, narrative description of the problem e.g. the problem of the competitors in the finish and their order).
- As a visualization by using video (robot activity, arrangement of numbers - different types of sorting, the problem of Hanoi towers, various logical games, etc.).
- As a situation (situation described by the final aim, usually a final state of the problem in the tasks where we need to search the way how to achieve the aim, state-space search problems, etc.) The situation can be presented as a game (chess problems) or as real-life problems (optimization problems).
- As a structure by using various diagram techniques (Flowcharts, UML diagrams, etc.).

979-8-3315-2771-6/24/$31.00 ©2024 IEEE

- As a sequence of the logical steps (algorithms of elementary instructions).

- As a simulation using virtual reality or augmented reality. It is more suitable for working in groups for advanced programming and solving complex problems.

We focus on leading the students from description or visualization of the task across description the final situation (final state of the task) to analyzing the solving process, decomposing the process to various levels (from structure to elementary commands), and from concrete scene to generalization. The example of teaching steps for solving concrete task (sorting problem) presented by a video are:

- Watching the video with the concrete type of a sorting problem (insert sort, bubble sort, merge sort, binary sort, etc.), where we use the videos from Sapientia University, Tirgu Mures (Marosvásárhely), Romania [12], (Figure 4). Writing the steps how to sort the numbers (usually from 0 to 9) based on watching.

- Generalization of the problem by written text, discussion how to generalize it.

- Finding the appropriate cycle for solving the process and verifying the process in studying materials.

- Finding the adequate programming code and translate it to a flowchart.

- Discussion about problems with generalization and effectivity of sorting algorithm.

Fig. 4. Shell sort in video [12].

Another example is to use the video of the logical game (Hanoi towers, missionaries and cannibals, painting the map with four colours, Japanese IQ test-crossing the river etc.), where we analyze all possibilities how to search the aim in the state space and how to use breadth-first search and depth-first search in state space for solving the problems. Finally, we find adequate programming code and discuss algorithm effectivity. Analyzing the scene is only one aspect of how to handle and how to understand the "computers steps". Important is to write effective programming code and so to design solutions for easy tasks (usually based on simple programming structures as some types of cycles, conditions, etc.). In this case, we use the application PS diagram which was developed by M. Bartyzal in 2012 as a bachelor thesis [2] and the application is still in the cycle of development. By using this application, the students can create and edit their own flowcharts and thanks to the built-in layout, it is very fast way how to do it. The application has three modes – editing mode, preview mode, and animation mode. The animation mode allows watching how the flowchart is running which makes it a good tool for fast errors correction. Animation is also used for assessing the students' homeworks. Another possibility, how to use the PS Diagram is to export the flowchart into programming code in Pascal (Figure 5) so the students can learn the main commands and the structures in programming. Also, the programming code can be imported to the PS Diagram and converted to the flowchart.

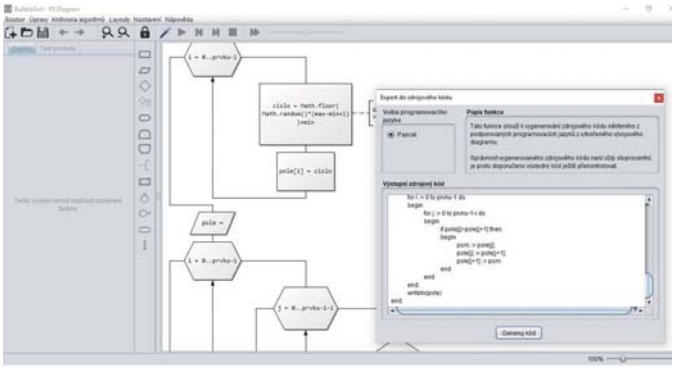

Fig. 5. Flowchart export to programming Language (Source: application PS Diagram).

As mentioned above, the e-learning course development is a cyclic process and chatbots, new applications and multimedia for visualization involved in the course can be the first phase of further course development. The IT students are usually very critical to all novelties, so choosing suitable tools needs long-term teaching experience and courage to make changes.

V. Discussion and Conclusion

To change the way of teaching is always a long-term process. Adding the new technologies means watching the students` manners, their way of learning, communicating with them, and finding the compromise solution for all. Then, it is necessary to search on web appropriate tools, compare them by various features, and select one. And it is not enough. The implementation of a new IT tool brings sometimes unexpected situations. Sometimes, it is necessary to learn how to add the knowledge into the IT tool. We decided to introduce the interacting IT tool - Chatbot and its usage in the teaching (learning) process. We started with the chatbot in the programming course, where the students prefer short and exact explanations and definitions. The students can avoid lengthy web searches of definitions, solutions, and programming commands by using the chatbot. It is more effective by solving projects and homework. We take-into account that IT students do not like to read, they prefer short exact explanations and then use the working method „trial and error ". Such a working method can be an advantage when we want to develop their creativity and analytical thinking. Therefore, we focus on scenario-based learning. The students must analyse, describing concrete video or initial state of the game, then search for a specific solution, and generalize it. In this process, they develop their creativity in discussion and answering decision questions. As [13] writes, such questions are: Which type of data can be used? Which type of cycle is better for the concrete tasks? Which algorithm is the best (more effective)? Is my algorithm general (or reusable)? etc. Finally, thanks to the application

979-8-3315-2771-6/24/$31.00 ©2024 IEEE

PS Diagram, we can teach better the algorithmization and the programming language Pascal. The further changes of PS diagram are prepared in [14], where the PS diagram includes a new model and can translate (parse) the flowchart to programming language JAVA and Python. The author Šulek suggests also the possibility to parse more complex flowcharts.

ACKNOWLEDGMENT

This paper was supported by the KEGA grant No. 008ŽU-4/2024 SCORE4AI: Slovak Collaborative Open, Research, and Education Ecosystem for AI of the Ministry of Education, Science, Research and Sport of the Slovak Republic.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of [1] Rakovská, E., Humanistic approach in learning informatics: possibilities of applying Rogers' approach In Ecoletra.com: scientific eJournal. - Lewes: Ecoletra.com LLC, 2016., vol. 2, no. 4, pp. 78-95 [online] https://ebook.ecoletra.com/2016-04-ecoletra.com-scientific-ejournal.

[2] Bartyzal, M., PS diagram, 2021, [online], http://www.psdiagram.cz/.

[3] Rakovská, E., Kanáliková, A.,Development and Implementation of E-learning Course by Using Knowledge Engineering and Software Engineering Methods. In Proceedings of the 13th International Conference, Microsoft Prague, 2018, p. 73-97 [online] https://disconference.eu/wp-content/uploads/2017/01/DisCo-2018-Overcoming-the-Challenges-and-Barries-in-Open-Education-13th-conference-reader-1.pdf.

[4] Brandtzaeg P.B., Følstad A., Why People Use Chatbots. In: Kompatsiaris I. et al. (eds) Internet Science. INSCI 2017. Lecture Notes in Computer Science, vol 10673. Springer, Cham.

[5] Russell, S., Norvig, P., Artificial Intelligence: A Modern Approach 3rd Edition. Prentice Hall, 2016.

[6] Mugan, J.,Chatbots: Theory and Practice. Medium,2017 [online], https://medium.com/intuitionmachine/chatbots-theory-and-practice-3274f7e6d648.

[7] Wit.ai, Build Natural Language Experiences,2020, [online], https://wit.ai/.

[8] Dialogflow documentation of google cloud,2021, [online], https://cloud.google.com/dialogflow/es/docs/basics.

[9] Naiman, L., How to Educate More Creative Problem-Solvers, 2012, [online],https://www.creativityatwork.com/2012/03/06/how-to-educate-more-creative-problem-solvers/.

[10] Khaldi, M., Barhone, J., Erradi, M., Khaldi, M., The educational scenario architecture of a learning situation. Global Journal of Engineering and Technology Advances, 2020,3(1),p. 027–040.

[11] Alexander, I., Maiden, N., Scenarios, stories, use cases: Through the systems development life-cycle. Wiley 2004.

[12] AlgoRythmics,From dance to code, 2011, [online], https://www.youtube.com/user/AlgoRythmics/about.

[13] Wu, J., 30 Ways Programmers Can Be Creative. Medium 2019, [online] https://betterprogramming.pub/30-ways-programmers-can-be-creative-6c10ddf2df4b.

[14] Šulek, J., Parsing algorithms into a selected programming language (in Slovak language) Thesis superviser: Eva Rakovská. Bratislava, 2021.