ANNALS

OF THE

UNIVERSITY OF CRAIOVA

Series: AUTOMATION, COMPUTERS,
ELECTRONICS and MECHATRONICS

Vol. 16 (43), No. 1, 2019

ISSN 1841-0626

Editura Universitaria Craiova

ANNALS OF THE UNIVERSITY OF CRAIOVA
Series: AUTOMATION, COMPUTERS, ELECTRONICS AND MECHATRONICS
Vol. 16 (43), No. 1, 2019 ISSN 1841-0626

***Note*: The “Automation, Computers, Electronics and Mechatronics Series”** emerged from **“Electrical Engineering Series” (ISSN 1223-530X)** in 2004.

Address for correspondence: Liana STĂNESCU
University of Craiova, Faculty of Automation, Computers and Electronics
Al.I. Cuza Street, No. 13, RO-200585, Craiova, Romania
Phone: +40-251-438198, Fax: +40-251-438198
Email: stanescu@software.ucv.ro

We exchange publications with similar institutions from country and from abroad

Chatbots in E-learning: An Overview of the Literature

Ștefania-Carmen Dobre

Computers and Information Technology Department, University of Craiova, Romania
cdobre@software.ucv.ro

Abstract:

A lot of the technology that we are using today involves Artificial Intelligence: there are also AI chatbots for human-computer interactions that can offer learning support, help students and improve learning. Chatbots have potential all along the learning journey, in many of the learning activities. They can be considered as a virtual 'teacher' used for reducing the workload of teachers, the trainers from the traditional learning. This paper describes a proposal for an educational chatbot and presents an overview about the AI features used for educational chatbots, an architecture model and a literature review about the chatbots used for improving the learning process.

Keywords: educational chatbot, artificial intelligence, learning

1. INTRODUCTION

A bot is a software that is designed to interact with humans using language-based interfaces to perform some automatic tasks. Chatbots are becoming more and more common nowadays, from personal chatbots, personal assistants (can assist in conducting business, meetings reminder, managing to-do lists), healthcare and business industries to educational chatbots (e.g. Botsify [Khan, A. (2019)]). Due to its flexibility, a chatbot can be considered as a trending system, that has been widely used in various fields. There are several companies that embedded chatbot technology into their system environment: Facebook (has implemented Facebook Messenger with the support of chatbot system), Microsoft (Cortana), Samsung (Bixby), Apple (Siri) and Google (Google Assistance).

In e-learning, chatbots offer a personalized experience for students, provide support to acquire knowledge being available 24/24 hours and enable the students to get everything instantly. They are very helpful for students' learning activities. Chatbots are an innovative approach to automate user personalization messages (Akma et al. (2018)). The purpose of this paper is to present an overview of the literature about the chatbots used in education, how AI can improve the learning process and a proposal for a chatbot for education, a smart student assistant. Our proposal for an AI chatbot comes to improve student interaction and collaboration and to act as a virtual teacher assistant in the innovative ed-tech world.

The rest of the paper is structured as follows: section 2 presents an overview of AI features used for educational chatbots, section 3 describes the general features of the chatbot system, an architecture model, few of the methods applied for chatbots development, section 4 includes some examples of chatbots used in education, section 5 contains few details about the proposed chatbot and section 6 highlights some conclusions.

2. AI AND CHATBOTS IN EDUCATION

Chatbots are often called artificially intelligent conversational tools. They are acting as a game changer in the innovative ed-tech world and are built to improve student interaction and collaboration (Singh (2018)).

In the education field, AI is present right across the learning journey (Clark (2018)) to support the students to achieve their learning goals. Chatbots can play a useful role for educational purposes, because they are an interactive mechanism for learning, compared to traditional learning systems as Kowalski et al. (2011) considered.

AI Chatbots can simulate a real human conversation, with real-time responses in natural language based on reinforced learning. They either use voice, text messages or both. There are already artificially intelligent conversational tools able to find things, create and curate content, allow natural language input and output, deliver personalized answers to questions, to enable online assessment and to have adaptive learning features for students' needs.

A chatbot is a useful tool used in education because it can have a lot of features to help teachers to deliver the learning content in a pleasant way or to provide information about the students to adapt the teaching activity (for example a chatbot to answer questions will help the teacher to see what questions students ask, where students have problems, it can identify the student's weaknesses or the learning path). For students, the interaction with chatbots will bring them a lot of benefits: can offer them needed information, receive instantly answers to their questions, support them with learning/administrative topics and last, but not least, to offer a personalized learning experience.

In the last years, chatbots have started to be used in education due to the benefits they bring to both students and teachers. In what follows, there will be presented some of the ways that chatbots and artificial intelligence are influencing the education (Singh (2018), McElvaney (2018), Spilka (2017)).

From these ways can be mentioned:

Learning Through Chatbot: Artificial intelligence technology used for building chatbots can be used to teach the students a lecture by turning it in a series of messages to make it easier to be read and to look like a standardized chat conversation. The bot will present the next part of the lecture when the student will understand the concept. Unlike the traditional learning method, where a teacher cannot track the level of understanding of the lesson for each of the students present in the classroom, the chatbot can offer this. It will lead to a higher percentage of assimilation of the information presented and implicitly, to achieve the student's learning goals.

Enhanced Student Engagement: The instant messaging platforms and social media tools are used daily by nowadays students. So, the best ways to communicate with peers or to find information about school topics are these platforms or a virtual assistant that can easily provide to the students the needed information about the assignments, due dates or any other events. Thus, it can lead to increase the engagement of students in a subject and can enhance the learning process.

Use Bots as Trainers: Students need continuous support for learning, which makes necessary a resource that can answer any question as quickly as possible.

AI chatbots, are the best option, in this case, because, chatbots can be trained and they will continually learn from students' questions to enrich their knowledge database and to have answers to more and more questions.

Artificial intelligence chatbot can reinforce by learning while doing.

Efficient Teaching Assistant: In a modern learning environment, the repetitive tasks of the teachers can be replaced with a virtual teaching assistant. It should be able to answer the students' questions about the courses, lesson plans, assignments and deadlines. Chatbots deliver instant access to expert knowledge and advice all the time. Also, the bot needs to be knowledgeable enough to provide feedback to students and to analyse the students learning path and to recommend the learning content to them accordingly.

Instant Help to Students: Technology has enabled the students to get everything instantly. Sometimes, a lot of students need the same information, which can be a very time-consuming task for teachers. Chatbots can be used to convert this time-consuming task of replying to each query personally into an automatic one. This will save a great deal of time of the students and they do not need to wait to receive the information, they will instantly receive the chatbot response.

Concentrate on the Learner: To facilitate learning and to increase student engagement, take ownership of their activities, it is useful to use a chatbot. Students will access the information and the learning content via a chatbot and will decide which topic to cover in a learning session. They will perceive this activity as having a personal teacher, who will provide support for the desired lesson/subject.

Smart Feedbacks: In education, feedback is playing an important role: on one hand, students' feedback offers information about their gaps and thus, the teachers can improve their teaching activity; on the other hand, the teachers' feedback allows the students to identify the areas where they need to do some extra work. A smart option to provide feedback, is to use a chatbot, where students can ask several questions, and the feedback will be automatically sent to the teacher, to be analysed and to change/improve their activities accordingly.

Better Student Support: Chatbots can offer a huge value to the educational institution if they are used to keep students informed about faculty facilities, if students have access to all the necessary information about the courses or the modules. Chatbots can also act as campus guides and help the students as they arrive at the campus.

This can increase institutions trust, because it seems that they really care about the students and facilitate their access to the information.

These bots can collect a lot of data about students' perception about school/faculty services and thus the services can be improved.

Personalized Learning: One of the main benefits of e-learning is that the course can be structured to cater for learners with different needs and abilities. Sometimes, some students need to learn deep some modules, but others have other gaps and need an adaptive learning environment. A chatbot can build a comprehensive picture of the learner as well as devising a highly personalized learning pathway. Thus, they will continue to monitor the progress of the learner and can provide the course information only when it is needed.

3. CHATBOT FEATURES

3.1 Chatbot architecture

A chatbot is created to perform some tasks. Regardless of the responsibilities that the chatbot must have, the architecture should contain the following components marked in the figure 1 as component blocks. The chatbot will always communicate with a user and will perform the activities requested by him/her.

The overall conceptual architecture of chatbots is shown in Figure 1.

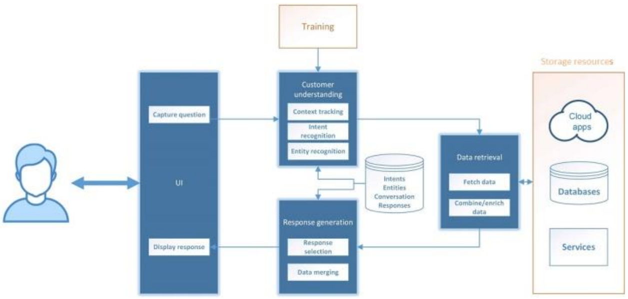

Fig.1 Chatbot architecture (De Scheerder (2018))

In fig.1, there can be identified the following functional components:

User interface (UI): The chatbot should have a responsive UI, good looking and attractive. It is the only visible component for the end-user, the way that the user interacts with chatbot functionalities.

User understanding: This block is responsible for few tasks: it tries to identify the intents (what the user is trying to do), to extract entities (the subjects that the user is talking about; these provide more information about the intents), it uses the conversation context to track what has been said. The bot should be able to learn from these actions. It is trained to match intents and entities and to acquire knowledge to become even better over time.

Data retrieval: If the bot understands what users want, it must find and retrieve the correct data in the storage resources. The resources can be retrieved using web-services or database calls or can be reached out from cloud.

Response generation: The chatbot must select from several good alternatives for responses to provide them to the user. The response(s) will be displayed to the user.

Conversation and context: This component can be programmed with conversation flows (the 'user understanding' module responsible for matching intents) matching the purpose of the bot. This way, it can track and update context and provide a meaningful and natural way to engage in a conversation with the user. Without this component, the bot would revert to zero-state after each question (De Scheerder (2018)).

3.2 Work Methods of Chatbots

3.2.1 Pattern Matches

Chatbots use pattern matching to classify the text and to give a suitable response to the users. For these patterns "Artificial Intelligence Markup Language" (AIML) is used as a standard structure model (Elupula (2019)).

Example for pattern matching:

```
<category>

<pattern>Who old are you </pattern> <template>I am 25</template>

</category>

<category>

        <pattern>what are you called</pattern>

        <template>

        <srai>what is your name</srai>

  </template>

</category>
```

The chatbot will give the answer for the user question because the pattern contains a part of the question. It will react only to anything related with the correlate patterns. But it can't go past the related pattern. To go beyond the associated pattern, there are used algorithms. The algorithms are utilized for text classification and can produce several pattern combinations to make a hierarchical structure.

3.2.2 Natural Language Understanding (NLU)

NLU (Natural Language Understanding) is an artificial intelligence technique used for matching the sentences from the users into intents, based on the intelligence behind the system, a supervised intent classification model created on a range of sentences as input and intentions as target. NLU has 3 specific concepts which will be described below.

Entities are represented by the chatbot purpose (e.g. student assistant chatbot, language learning chatbot).

Intents are the chatbot actions performed when the user says something. The bot should identify the user's intentions, to extract the main idea from these, because it is possible that users ask for the same thing, but with different inputs.

Context is a way for mapping user questions to intents, without being necessary to store conversation/questions history, each of the question will have assigned a flag (e.g. if a question is: "What's time in Romania", the flag can be 'Romanian time').

3.2.3. Natural Language Processing (NLP)

Natural Language Processing Chatbots are smart enough to find a way to convert the user's speech or text into structured data for choosing a relevant answer.

NLP can translate human language into information that contains text as well as patterns that can be useful in discovering applicable responses. Usually, the NLP chatbots are connected to a database, which is used to sustain the chatbot for providing the appropriate responses to every user.

3.3 Elements of a chatbot

Chatbots must have the following essential components to be a conversation partner (Nieves, 2018):

- *Conversational artificial intelligence*, the basic source of chatbots, thanks to which all management and natural language processing (NLP) occur. The first chatbots focused on the interpretation and recognition of patterns and rules. The more advanced chatbots implement deep learning processes to analyze the human input, learn from conversations and generate as suitable a response as possible.

- *User experience* (UX), which allows a natural, intelligent and coherent conversation to be established.
- *User interface* (UI), whereby the user can see or hear the conversations with the chatbot.
- *Conversational design,* which allows an artificial interaction to be equipped with human logic.

4. CHATBOTS FOR IMPROVING THE LEARNING PROCESS

Depending on their nature, there can be distinguished two types of chatbots in education: with and without an educational intentionality (Garcia Brustenga et al. (2018)). Chatbots without educational intentionality are responsible for teaching tasks of an administrative nature (student guidance and personal assistance) and of a support nature (to answer FAQs). From the other category, are mentioned chatbots that are designed to foster teaching and learning directly: tutors, that provide support for the learning process (can adapt, select and sequence contents according to the student's needs and provide learning motivation), and exercises or practice programs for skills acquisition (chatbot gives immediate feedback to the student when the student answers questions).

Lately, the need for the use of chatbots in education implies their development according to the new technologies that respond better to the needs of students and teachers. In the following, some examples of chatbots used for educational purposes will be presented.

Otto chatbot was developed by Learning Pool company (Clark (2018)). It aims to enhance the student-content interaction and it is integrated in an LMS.

Ani (Garcia Brustenga et al. (2018)) is an educational chatbot that was designed for learning and to replace some tasks of teachers. The goal was to provide personalized tutoring and mentoring that students become more involved in the learning activity. It includes the ability to adapt to the user's needs using automatic learning algorithms, as well as elements of assessment, motivation and immediate feedback.

Duolingo (Sawers (2019)) is designed for language learning (many of the world's most common languages) using conversation through gamification techniques.

Botter (Garcia Brustenga et al. (2018)) is a physical robot able to provide student support, it works as cognitive technology for learning, for the promotion of student behavioural change. It can interact with students through few ways: using light signals, movements, sound messages for motivation and disappointment or to help students to monitor their learning progress.

Differ (McNeal (2016)) is a chat application for higher education. The aim is to create a space for students where they can ask everything. Differ can have communities that bring together students in similar situations, it publishes relevant messages, reminders for increasing commitment and involvement.

CourseQ (Garcia Brustenga et al. (2018)) was designed at Cornell University (USA), to help the students, college groups and teachers by providing them an easy platform to communicate. The chatbot's functions include obtaining information for faculty and students as well as giving reminders regarding submission dates, timetables, material and events.

Botsify (Khan, A. (2019) is an education chatbot that presents a specific topic to the students and after learning the topic, students take quizzes and submit the results to their teachers. It is used, mainly, to help teachers to easily monitor the students' performances.

Ivy (Garcia Brustenga et al. (2018)) is an artificially intelligent self-service chatbot, designed for colleges and universities. It enables financial services, career services, management of admissions, technological services such as email access, Wi-Fi connection and app installation.

Di Blas et al. (2019 a, b) propose iMOOC and iCHAT, two chatbots created as an innovative approach to adaptive learning and for the using of the conversational interfaces in education. They implemented adaptive learning via conversational empathic interfaces, to help the learner deal with complex materials.

Unsupervised machine learning techniques were used by Ndukwe et al. (2019) for the task of automated grading chatbot, able to ask students questions and require written responses. In contrast with this approach, a different communication from the questionnaire procedure, Vladova et al. (2019) proposed a solution, a chatbot that acts as a teacher with natural language capabilities and used an informal way of obtaining the information about students through the chat addresses.

5. TOWARDS AN INTELLIGENT CHATBOT FOR EDUCATION

Due to continuous improvements in technology, the educational field should also be up-to-date with current trends. That is why chatbots (educational virtual assistants) are used more and more to perform automatic tasks for teachers and to support the learners to acquire knowledge.

Chatbots are considered a "way to improve the learning process, by helping the learner to deal with a complex material, tailoring the learning experience to specific needs" (Di Blas et al. (2019 a)).

An educational chatbot should be powerful enough to provide immediate and customized instruction or feedback to the learner, a unique, highly focus learning path and individual learning program for each student based on data gathered throughout the training process.

A combination of different AI techniques such as Natural Language Processing, Machine Learning and Semantics Understanding can drive the desired results for the needed requirements. From the above, we can consider that an AI chatbot for educational environment is an intelligent tutoring system for students, that has adaptive and personalization capabilities. There are already many chatbots used in education with AI features (Di Blas et al. (2019 a, b), Vladova et al. (2019), Ndukwe et al. (2019)) created to assist the students.

In the following, will be presented an intelligent chatbot for education, some features and problems that will be solved by our system proposal.

Our AI chatbot proposal will include features from two main areas: Intelligent Tutoring Systems (ITSs), the way in which AI techniques will be applied to education and adaptive and personalized learning, the way in which AI techniques will identify what the learners' needs and how will be improved the learning experience.

The chatbot will be a "virtual teacher", to support one-to-one tutoring, with natural language understanding and natural language processing capabilities. The advantage of this chatbot is that the communication/responses in real time will be through text or voice as it is in the traditional classroom or in case of face-to-face training.

By using AI techniques, the system will be able to interpret complex language and to extract the intents from the questions, will reply with natural responses and due to that, the learner needs to have a learning progress, the bot must remember the context of the entire conversation. From the informal conversation with the students and from answers, the bot will know the starting point for learning a topic and will identify the cognitive students' skills for organizing the course information that will be provided to the learner. An important role in the learning process is represented by the assessments and the feedback related with the acquired knowledge. To complement the grading system mentioned in Ndukwe et al. (2019)), where short answer text provided by a student are matched with at least one correct answer, machine learning techniques for text processing will be used when the students answer will be a paragraph or a document. For collecting data used for personalization, all students‘answers will be stored to get an overview about the learning style, how the learning content should be provided (text, video) and how the student can manage routine tasks easily (e.g. reading the course information and taking a quiz to check for knowledge).

6. CONCLUSIONS

In this paper, we have presented a literature review about the chatbots used in education, an overview of AI features used for educational chatbots, general features for chatbots and few details about a chatbot proposal. The next step is to design and implement this chatbot and to conduct an experimental evaluation about the chatbot usage for learning.

REFERENCES

Akma, N., Hafiz, M., Zainal, A., Fairuz, M. and Adnan, Z. (2018). Review of Chatbots Design Techniques. *International Journal of Computer Applications*, 181, 7-10.

Anteunis, J. (2017). How to Build A Powerful Enterprise Chatbot? Retrieved from: https://cai.tools.sap/blog/enterprise-chatbot-methodology/. (last accessed on 1 august 2019)

Clark, D. (2018). The Fallacy of "Robot" Teachers. Donald Clark Plan B. Retrieved from: https://donaldclarkplanb.blogspot.com/search?q=10+uses+for+Chatbots+in+learning+(with+examples).

De Scheerder, M. (2018). Building a chatbot: a reference architecture. (2018). Retrieved from: https://www.loqutus.com/building-a-chatbot-a-reference-architecture/

Di Blas, N., Lodi, L., Paolini, P., Pernici, B., Raspa, N., Rooein, D. and Renzi, F. (2019 a). Sustainable Chatbots supporting Learning. *Proceedings of EdMedia + Innovate Learning*, 1376-1381

Di Blas, N., Lodi, L., Paolini, P., Pernici, B., Renzi, F. and Rooein, D. (2019 b). Data Driven Chatbots: A New Approach to Conversational Applications. Retrieved from http://ceur-ws.org/Vol-2400/paper-24.pdf

Elupula, V. (2019). How do chatbots work? An overview of the architecture of chatbot. Retrieved from: https://bigdata-madesimple.com/how-do-chatbots-work-an-overview-of-the-architecture-of-a-chatbot/. (last accessed on 1 august 2019)

Fryer, L., Nakao, K. and Andrew, T. (2019). Chatbot learning partners: Connecting learning experiences, interest and competence. *Computers in Human Behavior,* 93, 279-289.

Garcia Brustenga, G., Fuertes-Alpiste, M., Molas-Castells, N. (2018). Briefing paper: chatbots in education. *Barcelona: eLearn Center. Universitat Oberta de Catalunya.* Retrieved from: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/80185/6/BRIEFING_PAPER_CHATBOTS_EN.pdf

Khan, A. (2019). How Education Industry Is Being Improved by AI Chatbots? Retrieved from: https://medium.com/botsify/how-is-education-industry-being-improved-by-ai-chatbots-4a1be093cdae. (last accessed on 1 august 2019)

Kowalski, S., Hoffmann, R., Jain, R., Mumtaz, M. (2011). Using Conversational Agents to Help Teach Information Security Risk Analysis. *SOTICS 2011: The First International Conference on Social Eco-Informatics*, pp 91-94. Retrieved from: https://www.thinkmind.org/download.php?articleid=sotics_2011_4_30_30106

McElvaney, P. (2018). 10 Secrets to Transforming L&D with A Chatbot. (2018). Retrieved from:

https://elearningindustry.com/transforming-l-d-with-a-chatbot-10-secrets (last accessed on 1 august 2019)

McNeal, M. (2016). A Siri for Higher Ed Aims to Boost Student Engagement. https://www.edsurge.com/news/2016-12-07-a-siri-for-higher-ed-aims-to-boost-student-engagement. (last accessed on 1 august 2019)

Meyer von Wolff, R., Hobert, S., Schumann, M. (2019). How May I Help You? - State of the Art and Open Research Questions for Chatbots at the Digital Workplace. *Proceedings of the 52nd Hawaii International Conference on System Sciences 2019*.

Ndukwe, I.G., Daniel, B.K., Amadi, C.E. (2019). A Machine Learning Grading System Using Chatbots. *Artificial Intelligence in Education. AIED 2019. Lecture Notes in Computer Science*, volume 11626, pp 365-368.

Roos, S. (2018). Chatbots in education: A passing trend or a valuable pedagogical tool? Retrieved from: https://pdfs.semanticscholar.org/533e/bc0255c36749e1f6b8d3662464d6ee5d4f0.pdf

Sawers, P. (2019). How Duolingo is using AI to humanize virtual language lessons. Retrieved from: https://venturebeat.com/2019/07/05/how-duolingo-is-using-ai-to-humanize-virtual-language-lessons/. (last accessed on 1 august 2019)

Singh, R. (2018). AI and Chatbots in Education: What Does the Future Hold? (2018). Retrieved form: https://chatbotsmagazine.com/ai-and-chatbots-in-education-what-does-the-futurehold-9772f5c13960 (last accessed on 1 august 2019)

Sjöström, J., Aghaee, N., Dahlin, M., Ågerfalk, P. (2018). Designing Chatbots for Higher Education Practice. *International Conference on Information Systems Education and Research,* Proceedings 4

Spilka, D. (2017). 4 Ways for Using Chatbots for eLearning. Retrieved from: https://elearningindustry.com/chatbots-for-elearning-4-ways-using. (last accessed on 1 august 2019)

Vladova, G., Haase, J., Rüdian, L.S., Pinkwart, N. (2019). Educational Chatbot with Learning Avatar for Personalization. Retrieved from: https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1352&context=amcis2019

Winkler, R. and Söllner, M. (2018). Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis. *78th annual meeting of the academy of management*, at Chicago

Rust, a Memory-safe Alternative to C

Mădălin Mămuleanu*

*Automation and Electronics Department, University of Craiova, Craiova, Romania
e-mail: mamuleanu.madalin@gmail.com

Abstract: Applications written in unsafe languages like C or C++ can introduce serious security issues due to memory errors such as buffer overflows, dangling pointers or reads of uninitialized data. In this paper, we are analyzing the most common security vulnerabilities found in software applications written in C or C++ and whether these issues could be avoided by using Rust, a modern programming language focused on safety.

Keywords: software security, memory safety, undefined behavior, embedded systems, C programming, Rust

1. INTRODUCTION

One of the oldest problems in computer security is memory corruption. Software written in C or C++ is prone to these type of issues. The lack of memory safety in these languages offers hackers the possibility to exploit memory bugs and alter the software's behavior. These bugs can be very hard to reproduce, debug and potentially expensive to correct. This war in memory safety is fought on two sides. One side is trying to find new ways to exploit software bugs in order to change the application's behaviour. And, on the other side, researchers and programmers are working to develop new protections and to write a safer code.

According to the Common Weakness Enumeration (CWE™) Top 25 Most Dangerous Software Errors, the most dangerous issue found in software is related to memory safety. C programming language facilitates writing high performance code through lightweight abstractions close to the hardware. However, low-level control undermines security. Support for manual memory management and unchecked memory access introduces opportunities for human errors. In some cases, these errors may escalate to catastrophic failures. The C programming language was created to provide proximity to the underlying hardware in order to write low-level code. C has control over the memory layout and minimal runtime support. However, some features of C, like pointers, pointer arithmetic, pointers to middle of objects, unchecked array indexing can cause simple programming error to corrupt the values of arbitrary memory locations. Memory corruption can cause a program to crash either immediately or in a non-deterministic manner, producing wrong results. This can be the root cause of multitude of security bugs. This article is structured as following: In section 2 the common security bugs related to memory safety are analyzed.

Section 3 describes Rust's approach to memory safety and the code from section 2 is analyzed and section 3 presents conclusions from the presented work.

1. *Spatial safety violations*

Memory safety violation can be either a spatial safety violation or a temporal safety violation. A spatial safety violation occurs when the code accesses a memory location outside the bounds of the object associated with the pointer or the array. Any pointer that points outside of it's associated object may not be dereferenced. Dereferencing these pointers results in a spatial memory safety error and undefined behavior.

Example of spatial safety violation:

```
//...
char *p = malloc(24);
for(int i=0; i<27;i++)
{
  p[i] = i+0x41;
}
//...
```

2. *Temporal safety violations*

A temporal safety violation occurs when a pointer is used to access a memory location after the object has been deallocated. When an object is freed, the underlying memory is not longer associated to the object and the pointer is not longer valid. Dereferencing these type of pointers will result in a temporal memory safety error and undefined behavior.

Example of a temporal safety violation:

```
//...
char *p = malloc(24);
free(p);
for(int i=0; i<27;i++)
{
   p[i] = i+0x41;
}
//...
```

2. COMMON SECURITY BUGS

The serious security vulnerabilities facilitated by the lack of memory safety in C and C++ are well known.

This is the underlying root cause of many security vulnerabilities. The memory corruption often allows an attacker to insert malicious code into the system and taking control of the entire system.

2.1 Integer overflow

In the context of programming, an integer is a variable capable of representing a number with no fractional part. Like all variables, integers are regions of memory. When we talk about integers, we usually represent them in decimal because this is the numbering systems that we are used to. Computers are not dealing with decimal. So, internally, an integer is stored in binary. Because it's necessary to store also negative numbers, there is a mechanism to represent them using binary. This is accomplished by using the most significant bit (MSB) to determine the sign. Because an integer is a fixed size, there is a maximum value it can store. When a value greater than the maximum value is written to this type of data, and integer overflow occurs. This overflow causes undefined behavior. An integer overflow cannot be detected after it happened, so there is no way for an application to tell if a result calculated previously is in fact reliable. This can become very dangerously if the calculated value has to do with the size of a buffer or an index of a buffer.

Example of a integer overflow bug in libssh2 library:

```
 if(session->userauth_kybd_num_prompts) {
     session->userauth_kybd_prompts =
=LIBSSH2_CALLOC(session,sizeof(LIBSSH2_USER
AUTH_KBDINT_PROMPT) *           session-
>userauth_kybd_num_prompts);
// This may overflow


}
```

The bug fix provided by the libssh2 team:

```
   if(session->userauth_kybd_num_prompts >
100)
{
        _libssh2_error(session,
LIBSSH2_ERROR_OUT_OF_BOUNDARY,"Too many
replies for keyboard-interactive prompts");
     goto cleanup;
}
if(session->userauth_kybd_num_prompts)
{
    session->userauth_kybd_prompts =
LIBSSH2_CALLOC(session,
sizeof(LIBSSH2_USERAUTH_KBDINT_PROMPT) *
session->userauth_kybd_num_prompts); //
This will not overflow
                /* ... */
}
```

2.2 Dangling pointers

In computer programming, dangling pointers are pointers that are pointing to a memory address that has been deleted (or freed). Dangling pointers often arise during object destruction, when a reference of an object has been deleted or deallocated without modifying the value of the pointer, so that the pointer still points to the deallocated address. In some situations, the system may have reallocated the previously freed memory and unpredictable behavior may result because the memory contains completely different data.

Example of a program that creates a dangling pointer:

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    char **strPtr;
    char *str = strdup("Hello!");
    strPtr = &str;
    free(str);
    //strPtr now becomes a dangling pointer
    printf("%s", *strPtr);
    return 0;
}
```