Kamil WOŁOSZYN[1], *Emil MICHTA*[2], *Krzysztof PIOTROWSKI*[1]
[1]*IHP – Leibniz-Institut für innovative Mikroelektronik, Frankfurt (Oder), Germany*
[2]*Uniwersytet Zielonogórski, Instytut Metrologii Elektroniki i Informatyki*

MODULAR AI APPLICATIONS FOLLOWING THE SENS4U APPROACH

This paper presents the concepts of a universal, modular tool that will enable rapid development of AI applications. The goal of the conceptual tool is versatility in terms of changing the environment, as well as integration of different types of sensors to perform a specific task. The use of the Sens4U approach will facilitate the entire process of building AI applications and simplify the testing process.

MODUŁOWA APLIKACJA SZTUCZNEJ INTELIGENCJI BAZUJĄCA NA ROZWIĄZANIU SENS4U

W artykule przedstawiono koncepcje uniwersalnego, modułowego narzędzia, które umożliwi szybką budowę aplikacji AI. Celem koncepcyjnego narzędzia jest uniwersalność w zakresie zmiany środowiska, a także integracja różnych typów czujników w celu wykonania określonego zadania. Zastosowanie rozwiązania Sens4U ułatwi cały proces budowy aplikacji AI oraz uprości proces testowania.

1. MOTIVATION

Artificial intelligence (AI) has grown in popularity in recent years. AI has been applied in many scenarios, from intelligent assistants to advanced real-time vision systems. AI enables devices to make decisions based on data collected from cameras or various sensors. This data is processed and the device is then able to react based on it. This paper presents the concept of a universal, modular tool that will enable rapid development of AI applications. The goal of the conceptual tool is versatility to change the environment and also to integrate different types of sensors to perform a specific task. The concept of the proposed tool using the Sens4U approach [1] will improve the development process, testability and reliability of the application. The idea of Sens4U is based on accelerated application development using the modularity of available libraries for non-experts in the field.

Currently, there are many tools available to train deep neural networks and many sample program codes for creating intelligent robots or implementing trained models. However, combining them with other sensors requires a lot of time to be fully functional. While working on the SpaceRegion project, the initial focus was on building a stationary robot, without the ability to move. The robot was equipped with video cameras and servos to control the direction of the cameras. During the course of the work, it was decided that, additionally, a mobile robot will be built for the project. Initially, the mobile robot had the same functionalities as the stationary robot, but during the implementation of the code for the mobile robot it turned out that most of the functionalities were the same, only the code responsible for controlling the robot's chassis had undergone major changes. This gave rise to the idea of building a universal tool to speed up the process of building the application. This approach can also be used to transfer one application from one scenario to another one, like a consumer solution to become a space solution, without re-implementing the application from scratch.

2. RELATED WORK

While it is possible to find systems that provide users with libraries and tools to help build robot applications such as Robot Operating System (ROS) [2], or CubeSystem [3]. It is difficult to find systems or collections of libraries that have standardized functionality for sensors of the same type. Complicating the operation of the application is a number of additional installations of software, libraries

and sometimes the required system, which can be problematic for a user unfamiliar with the field. Bearing in mind the existing software and libraries, it seems reasonable to present a concept of a tool that is functional, modular and cross-platform, supporting the development of a ready application.

3. PROPOSED APPROACH

The concept of the tool for fast and modular development of the application is shown in Fig 1. As it is known, almost every available sensor on the market has ready-made hardware drivers as well as available software.

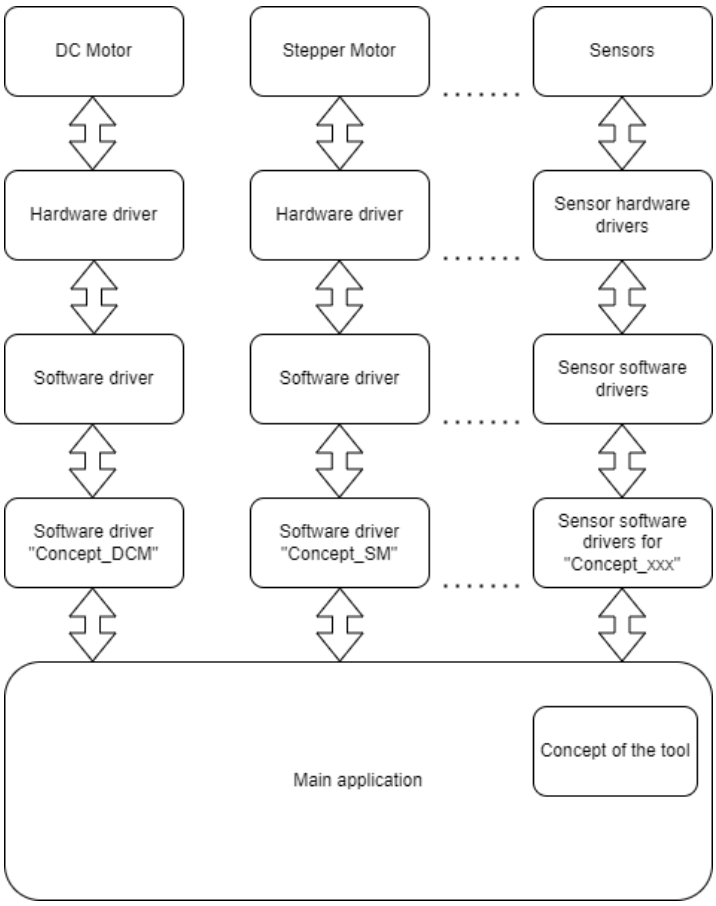

Fig. 5 Concept of the proposed tool
Rys. 1 Koncepcja proponowanego narzędzia

The aim of the proposed tool is to create software, which will be a set of dedicated libraries, but the introduced modification in the conception tool will allow to perform the same action on different sensors. As a result, it will always return the same output, assuming that all the requirements specified during the development of the conceptual tool are met. The idea for the proposed tool emerged during work on the development of a computer vision application. The programming language used to build the application is Python version 3 [4]. Python is a high-level programming language, and thanks to its simplicity and versatility, it has become an ideal tool for building AI systems. The base AI application that was created for the project had to observe the environment, recognise people in the same room and check from time to time whether the person is still in the same place. Based on the people in the room, the robot estimated the office it is located in. Then, in the course of the work carried out, an idea emerged

to create a mobile robot "Łazik", whose task would be, e.g., to follow a detected object. While building the mobile platform, it was found that the code responsible for computer vision could be directly transferred, and only the variables responsible for resolution or camera port were modified. The code responsible for controlling the robot was completely changed. The stationary robot had servos that control the camera direction, and the mobile robot had DC motors to move it. Thanks to these changes, an idea emerged to create a conceptual tool that would allow to accelerate work, and transferring the code to a new platform for users would not be so problematic. And the created application would not need significant changes made by the user.

Further, the plan was also to investigate the possibility for easy transfer of the example applications to more demanding environments (like space). In order to support that, the application should realize its tasks when the original sensors and actors are replaced by devices that can withstand the new environmental conditions. The application should thus use generic application programming interface (API) to use the functionalities of the devices and for instance adapt to the physical features the devices provide (video resolution of a camera module or the resolution of a motor turn). Replacing a device with another one does not change the application implementation although it can change its internal behaviour and features. The API provides the application with the control over the device and provides information about the features of the device.

The application developer who uses the conceptual tool will be able to obtain a list of available functions for a given sensor. When building the tool, it would be necessary to define the communication interface with the user and the parameters that the user would have to enter for the tool to work correctly. For example, a user building a robot uses built-in DC motors. Using the API functionality of the conceptual tool, the user would be able to call, for example, the getFeaturesDCmotor() method. This method will return data to the user with information about what parameters the user should enter and available functions for DC motors e.g., turnForward(), turnBackward(), stop(). Another functionality of the tool concept is its modularity, which aims to support the modification of the project. An example is the modification of the mentioned robot by the user. The user replaces the standard DC motors with stepper motors, which brings a number of changes to the created main application. However, when using the proposed conceptual tool, the only modification that would be on the user's side would be a possible change of the name of the motors used. The basic functionality would be consistent for both motors and the operation of the program after the exchange of the motors would not be disturbed. What's more, after modifications, calling the getFeaturesSmotor() method for stepper motors will display more available functions because besides turnForward(), turnBackward(), stop() stepper motors can also have functions like turnForwardAboutAngle(xxx_angle), turnBackwardAboutAngle(xxx_angle) and turnLeftAboutAngle(xxx_angle), turnRightAboutAngle(xxx_angle), setStartPosition() while maintaining the requirements related to the motor mounting direction specified during the tool development.

The proposed tool will allow expansion with new libraries depending on the users' needs. The main advantage of the conceptual tool would be reliability to environmental changes, and unification of functionality for sensors of the same type would allow rapid modifications in the design. During the development of the tool, it will be possible to extend its operation to support different types of sensors. Moreover, sample functionalities such as object detection on the provided video stream or real-time object tracking from trained deep learning models can be implemented in the tool for artificial intelligence purposes. For devices with low processing power, the proposed conceptual tool will be able to implement functionality to support AI accelerators. The AI accelerators will enable the satisfactory performance of artificial neural networks combined with computer vision. AI accelerator is, for example, the Intel Neural Compute Stick 2 [5], which will enable artificial neural networks to achieve satisfactory results when combined with computer vision. One advantage of the proposed tool will be its cross-platform capability, which would allow the tool to run on both Microsoft Windows and Linux platforms.

4. CONCLUSIONS AND FURTHER WORK

The paper presents the concept of the tool, which is still under development and may change during implementation when new problems and challenges are discovered within the SpaceRegion project. Different scenarios based on terrestrial and space solutions are currently developed and implemented. Based on the developed scenarios, the capabilities and requirements of the proposed tool will be adjusted. We will also analyse and compare implemented code for different scenarios in order to find common parts, which will be eventually transferred to the proposed tool. A library will be created consisting of several sensors of the same type in order to test the behaviour of the application during its modification. Furthermore, sample applications related to artificial intelligence, robotics and computer vision using the conceptual tool will be created. In addition, an analysis of the application for modification when scenarios change will be performed. The main goal of the proposed conceptual tool is to simplify the development of AI applications.

ACKNOWLEDGMENTS

This work was supported by the European Regional Development Fund within the BB-PL INTERREG V A 2014-2020 Program, “reducing barriers using the common strengths”, project SpaceRegion, grant number 85038043 and by project SmartRiver, grant number 85029892. The funding institutions had no role in the design of the study, the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

REFERENCES

1. Piotrowski K., Peter St.*: Sens4U: Wireless Sensor Network Applications for Environment Monitoring Made Easy*, in Proc. 4th International Workshop on Software Engineering for Sensor Network Applications (SESENA 2013), 2013.
2. ROS, *ROS-Robot Operating System, https://ros.org/,* last viewed 22.03.2022
3. A. Birk.: *Fast robot prototyping with the cubesystem,* IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004
4. Python, *Python™, https://www.python.org/*, last viewed 22.03.2022
5. Intel Corporation, Intel® Neural Compute Stick 2 (Intel® NCS2), *https://www.intel.com/content/www/us/en/developer/tools/neural-compute-stick/overview.html,* last viewed 22.03.2022