

***IKON - IT Kompetenz Netzwerk
zur Stärkung der Grenzregion
Sachsen-Tschechien 2020***

Prof. Dr. Dr. h. c. Wolfram Hardt (Hrsg.)

TUD*press*

IBS Scientific Workshop Proceedings

Herausgegeben von Stiftung IBS, Wolfram Hardt

Band 12

***IKON - IT Kompetenz
Netzwerk zur Stärkung
der Grenzregion Sachsen-
Tschechien 2020***

Prof. Dr. Dr. h. c. Wolfram Hardt (Hrsg.)

TUD*press*
2021

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im
Internet über <http://dnb.d-nb.de> abrufbar.

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Na-
tionalbibliografie; detailed bibliographic data are available in the Internet
at <http://dnb.d-nb.de>.

ISBN 978-3-95908-236-5

© 2021 TUDpress
THELEM Universitätsverlag
GmbH und Co. KG
<http://www.tudpress.de>

Alle Rechte vorbehalten. All rights reserved.
Gesetzt von den Herausgebern.
Printed in Germany.



IBS Scientific Workshop Proceedings

Am 4. November 2019 startete mit dem Kick-Off Meeting das internationale und EFRE-geförderte Projekt „IKON-IT Kompetenz Netzwerk zur Stärkung der Grenzregion Sachsen –Tschechien“, welches in Kooperation mit der Technischen Universität Chemnitz (TUC) und der Jan Evangelista Purkyně-Universität (UJEP) durchgeführt wird, in den Räumen der Stiftung Internationales Informatik- und Begegnungszentrum Sachsen (IBS) im sächsischen Laubusch in die Umsetzungsphase. Als Leadpartner hat Prof. Dr. Dr. h.c. W. Hardt (TUC) große Ziele für die Teilnehmenden, welche sich insbesondere durch eine exzellente Praxisqualifizierung auszeichnen, gesteckt. Nach diesen Maßstäben und unter Mithilfe von einem Team aus wissenschaftlichen Mitarbeitern wurden seit Projektstart eine Vielzahl von Seminar- und Workshoptagen planmäßig durchgeführt. Die Erarbeitung und praktische Verwirklichung von zukunftsweisenden Aspekten der Informatik stellt dabei das oberste Ziel für die Zusammenarbeit junger Studenten verschiedener Nationalitäten dar.

Das Projekt gliedert sich in drei große Themenblöcke, welche parallel zueinander bearbeitet und in einer zweiten Durchführungsphase auch wiederholt werden. Im Block I „Automotive“ beschäftigten sich die TeilnehmerInnen rundum mit dem Thema Automotive Software Engineering – spezifischer gesagt mit der Umsetzung von Strategien des autonomen Fahrens beispielsweise durch automatische Fußgängererkennung, Spurhalteassistent, Verkehrsschilderkennung und weiteren notwendigen Methoden. Der Block II „Digital Processes“ umfasst ein anderes Themenfeld, das Mentoring von Studenten während des Lernens, insbesondere hierfür online gestützte Ansätze. Im Block III liegt der Fokus auf dem Thema Künstliche Intelligenz (KI) und den damit einhergehenden verschiedenen Techniken wie beispielsweise „Deep Learning“. In der ersten Durchführungsphase erfolgte die praktische Umsetzung am Beispiel des autonomen Fahrens.

Mit der Beendigung des letzten Blockes im Juli 2020 blicken wir auf eine erfolgreiche erste Projektzeit zurück. Die Ergebnisse der Arbeit, welche die teilnehmenden StudentInnen geleistet haben, lassen sich nun in den jeweiligen Berichten nachlesen.

Dne 4. Listopadu 2019 byl oficiálně zahájen projekt podpořený v rámci Programu přeshraniční spolupráce Česká republika – Sasko v období 2014 – 2020 s názvem „IKON – IT síť kompetencí k posílení příhraničního regionu Česko – Sasko“, na jehož realizaci se podílí Technická univerzita Chemnitz (TUC) a Univerzita Jana Evangelisty Purkyně v Ústí nad Labem (UJEP). Úvodní konference projektu se konala v prostorách Nadace Mezinárodního informačního a potkávacího centra Sasko (IBS) v saském městečku Laubusch.

Prof. Dr. Dr. H.c. W. Hardt (TUC) jako zástupce leadpartnera projektu vytyčil před přítomnými vysoké cíle, spočívající zejména v excelentní praktické kvalifikaci. Dle těchto měřítek za pomoci týmu vědeckých pracovníků bude plánovitě uskutečněna celá řada seminářů a workshopů. Dosažené výsledky a jejich praktická realizace s ohledem na budoucí určující aspekty v oblasti informatiky jsou nejvyšší prioritou v rámci spolupráce mladých studentů různých národností.

Projekt se dělí do tří velkých tematických bloků, které paralelně na sebe navazují a proběhnou opětovně ve dvou fázích. V bloku I „Automotive“ se zabývají účastníci tématem automobilovního softwarového inženýrství – výstižněji řečeno s uplatněním strategií izdy bez řidiče na příkladech rozpoznávání chodců, asistent zachování jízdního pruhu, rozpoznání dopravních značek a dalších metod. Blok II „Digitální procesy“ se zabývá další oblastí témat spojených s monitoringem studentů během výuky se zvláštním zřetelem na online výuku. V bloku III spočívá podstata v oblasti umělé inteligence a s ní spojené různé techniky jako např. „deep learning“. V první realizační fázi proběhla praktická ukázka na příkladu autonomního řízení.

S ukončením posledního bloku v červenci 2020 se můžeme ohlédnout za úspěšným prvním obdobím projektu. S výsledky prací zapojených studentů se můžete v dalším textu seznámit.



Inhaltsverzeichnis

1. Prototypical System for Situation Awareness Using the “CE-Box” of Computer Engineering, TU Chemnitz	1
1.1 Abstract.....	1
1.2 Motivation.....	1
1.3 Introduction.....	2
1.4 Implementations and Demonstrations.....	3
1.5 Results.....	4
1.6 Conclusion	7
2. Realtime Detection of Hand Movement by EMG signal.....	8
2.1 Abstract.....	8
2.2 Motivation.....	8
2.3 Introduction.....	9
2.4 Research questions and methods	9
2.5 Implementation and demonstration.....	11
2.6 Results.....	13
2.7 Reference	14
3. Digital Processes for interactive Mentoring Approaches.....	15
3.1 Abstract.....	15
3.2 Introduction.....	15
3.3 Database.....	15
3.4 Research Methodology	17
3.5 Guide to Task Completion	18
3.6 Theoretical framework.....	18
3.7 Research Objective	19
3.8 Task Implementation and Results	19
3.9 Results & Discussion	24
3.10 Conclusion	28
3.11 Acknowledgement	28
3.12 References.....	28
4. Implementation of digital processes via Reliance program from Scada.....	30
4.1 Abstract.....	30
4.2 Motivation.....	30
4.3 Introduction.....	31



4.4 Research questions and Methods	31
4.5 Implementations and Demonstrations.....	33
4.6 Results.....	38
4.7 References.....	38
5. Automotive II.....	39
5.1 Abstract.....	39
5.2 Motivation.....	39
5.3 Introduction.....	39
5.4 Research questions and Methods.....	40
5.5 Implementations and Demonstrations.....	40
5.6 Results.....	41
5.7 Conclusion	42
5.8 References.....	42
6. Deployment of Self-checking and Self-diagnosis in Complex Systems	43
6.1 Abstract.....	43
6.2 Motivation.....	43
6.3 Introduction.....	44
6.4 Research questions and Methods.....	44
6.5 Task solving and Results.....	45
6.6 Conclusion	55
6.7 References.....	55



1. Prototypical System for Situation Awareness Using the “CE-Box” of Computer Engineering, TU Chemnitz

Julkar Nine

1.1 Abstract

During the last decade, leading automotive companies are moving their focus towards high level autonomous vehicles. In order to make automotive vehicles highly autonomous, it needs be given the ability to understand its surrounding environments and situations properly. Only once it is able to estimate the situations, it can be made aware of them. This project concentrates on a prototype of this research field called Situation Awareness for highly autonomous vehicles. The prototype is realized on a custom hardware called the “CE-Box”, which is consisted of multiple Raspberry Pi 3b+ models.

1.2 Motivation

The focus of this topic is to realize a situationally aware system via a prototypical manner using the “CE-Box”. Situation Awareness has three major levels: a) Perception (Extracting features & Information), b) Comprehension (Understanding the Information & taking action) & c) Projection (Predicting the next state based on the action taken).

For this project, this topic is divided into 5 individual tasks. They are:

Task 1: Pedestrian Detection using Haar Cascade Classifiers.

Task 2: Lane Detection and Curvature Estimation using Image Processing.

Task 3: Traffic Sign Detection and Recognition using Deep Learning Techniques.

Task 4: Facts Generation from Sensory Synthetic Data using PiCan 2 Module.

Task 5: Facts Generation from Image Processing Algorithms using PiCan 2 Module.

Tasks 1 to Task 3, belongs to the perception level of situation awareness. Perception is done through two different ways, perception through image processing application, which is done by Tasks 1, 2, 3 & perception through sensory synthetic data, which is done by Task 5. Additionally, Task 5 had also the sub task to do some work for the comprehension level using a technique called Expert Systems. The responsibility of Task 4 was to synchronize the image processing applications and provide information of Tasks 1, 2, & 3 to Task 5 via CAN messages using the PiCan 2 module, which is mounted on each of the raspberry pi's in the CE-Box. The task design can be seen in the figure 1 below:

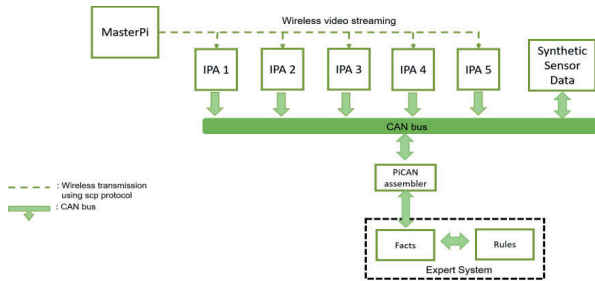


Figure 1: Concept of a Situationally Aware System Prototype

1.3 Introduction

Tasks description for each of individual tasks are given below:

Task 1: Pedestrians Detection using Machine Learning/Image Processing

Detecting pedestrians in urban scenes remains to be a challenge in computer vision despite recent rapid advances. The main goal of this internship is to develop an algorithm using Image processing or machine learning which will used to detect and recognize the pedestrians. Direct approaches of pedestrian detection rely on pedestrian segmentation and tracking of each individual in the scene, normally requiring strong computer vision algorithms.

Task 2: Lane Curvature Detection using Image Processing

Real-time Lane Detection for Driving System Using Image Processing Based on Edge Detection and Hough Transform has been developed to aid a driver in the lane departure decision-making, to reduce a loss of concentration and to prevent an accident while driving. In this internship, propose a method for detecting the lane markers in semi-real time. The Edge detection is a technique which used to identify points into a set of curved line segments called edge. Hough Transform method is used to detect the lanes in an image for increasing the accurate lane detection and the safety driving system.

Task 3: Driving Scenario Based Traffic Sign Recognition using Image Processing/Machine Learning

The main goal of this internship is to develop an algorithm using Image processing or machine learning which will used to detect and recognize the 4 different traffic signs (excluding speed signs). The objective of detection methods is localizing regions of interest containing traffic sign, and we divide detection methods into three main categories: color-based (classified according to the color space), shape-based, and learning-based methods (including machine learning).

Task 4: Facts Generation from Sensory Synthetic Data using PiCan 2 Module

Due to the recent advancement of the Image processing and machine learning, there are various algorithms have been developed for the visual perception of the information. But it can put a lot of dependency on the camera sensors itself, which in the case of any error can be really problematic. In order to prevent these kind of scenarios, perception from other sensors of the vehicle is also required. The main objective of the internship to generate a synthetic sensory data of 4 or 5 sensors for the automotive vehicles. The perceived data are send to comprehension via PI CAN 2 and received back the corresponding actions / projections.



Task 5: Facts Generation from Sensory Synthetic Data using PiCan 2 Module

In recent years, there are various available Image Processing algorithms for the perception of the outside world to ease driving. The main aim of this internship is to combine few of the available algorithms together and run in a single raspberry pi. The output of the algorithms are converted into facts and send to comprehension level via PI CAN 2.

1.4 Implementations and Demonstrations

The implementation was done on individual Raspberry pi 3b+ models, and showcased using the „CE-Box“, which part of a is a demonstrative hardware unit called the „BlackPearl“ from the department of computer engineering at TU Chemnitz. It is consisted of six slots for Raspberry Pis. Each of the Raspberry pi has a PiCan2 Module mounted on them. It also has a additional touch screen for output visualization, which is also ran by a raspberry pi. The „CE-Box“ is showcased in the figure below:



Figure 2: „CE-Box“ (left) & „Black-Pearl“

1. Pedestrian Detection using Haar Cascade Classifiers

The dataset for the training of the pedestrian detection collected from INRIA dataset. The dataset are necessary selected from the traffic scenarios to detect the pedestrians. The training of the model done using the Haar Cascade Classifier. Two sets of images used to train the Haar Cascade Classifier, the positive images consists of the only pedestrian and negative images consist of car, bike, etc. The training has been done for 20 stages, and at the end of the training, a final classifier is generated in the XML format. The output of the classifier has been implemented in Raspberry Pi 3B+.

2. Lane Detection and Curvature Estimation using Image Processing

Lane curvature detection uses the image processing method. Image processing methods can be sub-divided into a few steps. Initially, the input video frames converted into the HLS color threshold and gradient threshold. Then both color and gradient threshold are combined and changed into the Bird-Eye view. In the Bird-Eye view, the sliding window search used to detect the lanes. Finally, the polyfit function used to find the curvature



of the lane from the detected lanes. The output of the polyfit function determines the curve is right or left. The output of this method is implemented on the Raspberry Pi 3B+.

3. Traffic Sign Detection and Recognition using Deep Learning Techniques

The dataset for the traffic Sign detection collected the German Traffic Sign Dataset Benchmark (GTSDDB). In the collected dataset, four traffic signs separated for the training. All four traffic signs are labeled using the labelImg tool. A new model developed for training the four traffic signs using the TensorFlow and Keras framework. Then the developed model is trained and tested in the CPU. For the implementation of the model in the Raspberry Pi 3B+, a preprocess of cropping the traffic sign from the entire image is implemented. The model will predict the cropped traffic sign from the image.

4. Facts Generation from Sensory Synthetic Data using PiCan 2 Module

The generation of the synthetic sensor data done using the vehicle dashboard GUI. The vehicle dashboard GUI developed with multiple sensors and Knob controller. The sensor values are adjusted automatically based on the clock event or manually using Knob Controllers. The sensor values sent to the expert system as CAN message using PiCAN2. The expert system gives back the command based on the current values of the sensors, and the actuation of the sensors takes place based on the decision made by the expert system. The implementation of the GUI and the expert system is done on two racks of the CE Box.

5. Facts Generation from Image Processing Algorithms using PiCan 2 Module

The image processing algorithms namely traffic sign, traffic light, and autobahn sign provided. The provided algorithms are optimized using the frame skipping method and splitting algorithm. The skipping of the frame helps to reduce the number of frames processed by the algorithms. The Adhoc network implemented with one master and several slaves. Each slave contains one image processing algorithm and implemented on Raspberry Pi. A separate Raspberry Pi used to implement the master. All the slaves are connected to the master wirelessly. The master will provide the video feed to the slaves and triggers all the slaves at the same time. The slaves will process the video frames and generate the output. The output sent to the expert system via the CAN message.

1.5 Results

1. Pedestrian Detection using Haar Cascade Classifiers

The performance of the Haar-cascade was evaluated on Raspberry Pi perfectly without any difficulties. And it has provided a successful detection of the pedestrians for different traffic scenarios videos with different resolutions. The cascade was trained several times with several data set and with different training parameters which is very time-consuming. On the other hand, the detection rate not only depends on the training parameters it also depends on the OpenCV built-in function's parameters. The output of the pedestrian detection is shown in Figure 1.

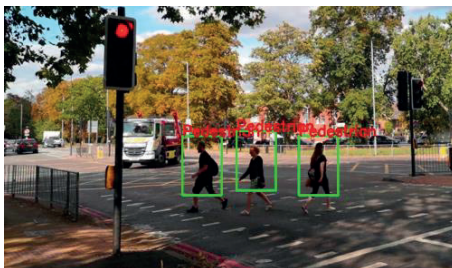


Figure 4: Pedestrian Detection using Haar Cascade

2. Lane Detection and Curvature Estimation using Image Processing

The algorithm accurately detects lane curvature and show warning message for curvy road condition. The algorithm tested on highway video where the lanes are detected correctly. A warning message is displayed while there is a curvy lane moving to the right or left. Figure 2 shows the result for both 'curve to the left' and 'curve to the right' respectively.

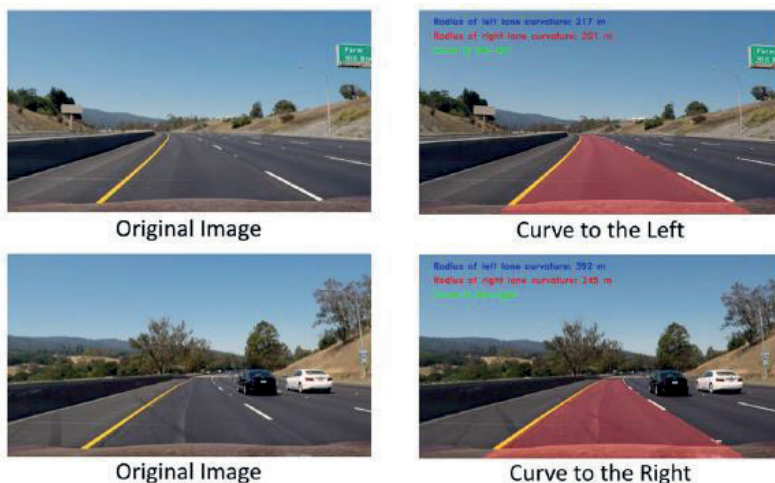


Figure 5: Lane Curvature Detection

3. Traffic Sign Detection and Recognition using Deep Learning Techniques

The pre-process method and model used to predict the traffic sign in the on the images. The complete methodology implemented on Raspberry Pi 3B+. The test image provided to the methodology able to predict the traffic sign. The model also predicts traffic signs as unknown which are not used for training. The output of this methodology is shown in Figure 3.



Figure 6: Traffic Sign Detection

4. Facts Generation from Image Processing Algorithms using PiCan 2 Module

Several Image processing algorithms were provided to the students of this group. Their task was to run these image processing algorithms individually in different raspberry pis of the “CE-Box”. They also had to optimize these algorithms and send the outputs of these algorithms to the Master pi using CAN messages. The system was designed successfully and CAN protocols were used as a mode of communication for taking algorithm outputs to an expert system. All image processing algorithms have their predefined CAN message format and arbitration ID which decides the priority of CAN message. In our case, the highest priority was given to traffic light algorithm then traffic sign and autobahn detection algorithm have the least priority. At the comprehension level received CAN messages are identified based on the arbitration ID. A python script decodes received CAN messages and a fact is updated in the current fact file. The CAN payload and generated facts are shown in Figures 4 & 5.

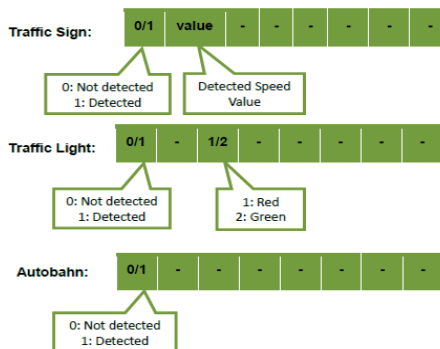


Figure 7: CAN payload

```
(state(context Traffic)(Type Traffic Sign)(value 80))
(state(context Traffic)(Type Traffic Light)(value Green))
```

Figure 8: Generated Facts

5. Facts Generation from Sensory Synthetic Data using PiCan 2 Module

The Vehicle Dashboard GUI was implemented and synthetic sensory data are generated from the GUI. The design of vehicle dashboard and backend communication are shown in the Figures 6 & 7. The implemented was done on Raspberry Pi and observed the overshooting of CPU load reaches constantly above 130% whenever the GUI was running. This overshooting of CPU load was happening because there was no synchronization between the client, middleware and the server. And later the issue was fixed by using shared memory to establish the interconnection between the client, middleware and the server using Flask extension such as Flask_Restful and Flask_cors for handling Cross Origin Resource Sharing (CORS). After this, there was a successful generation of facts that fed to the expert system to guide the suggestion for optimal autonomous driving.



Figure 9: Vehicle Dashboard GUI

```
{ 'speed': 115, 'EngineSpeed': 3335, 'VehicleBattery': 12, 'Engine Temperature': 65.25, 'Fuellevel': 63 }
127.0.0.1 - - [19/Mar/2020 01:19:11] "B[37mGET / HTTP/1.1]0m" 200 -
{ 'speed': 115, 'EngineSpeed': 3335, 'VehicleBattery': 12, 'Engine Temperature': 65.25, 'Fuellevel': 63 }
127.0.0.1 - - [19/Mar/2020 01:19:11] "B[37mGET / HTTP/1.1]0m" 200 -
{ 'speed': 115, 'EngineSpeed': 3335, 'VehicleBattery': 12, 'Engine Temperature': 65.25, 'Fuellevel': 63 }
127.0.0.1 - - [19/Mar/2020 01:19:12] "B[37mGET / HTTP/1.1]0m" 200 -
{ 'speed': 115, 'EngineSpeed': 3335, 'VehicleBattery': 12, 'Engine Temperature': 65.25, 'Fuellevel': 63 }
127.0.0.1 - - [19/Mar/2020 01:19:13] "B[37mGET / HTTP/1.1]0m" 200 -
{ 'speed': 115, 'EngineSpeed': 3335, 'VehicleBattery': 12, 'Engine Temperature': 65.25, 'Fuellevel': 63 }
127.0.0.1 - - [19/Mar/2020 01:19:13] "B[37mGET / HTTP/1.1]0m" 200 -
{ 'speed': 115, 'EngineSpeed': 3335, 'VehicleBattery': 12, 'Engine Temperature': 65.25, 'Fuellevel': 63 }
127.0.0.1 - - [19/Mar/2020 01:19:14] "B[37mGET / HTTP/1.1]0m" 200 -
{ 'speed': 115, 'EngineSpeed': 3335, 'VehicleBattery': 12, 'Engine Temperature': 65.25, 'Fuellevel': 63 }
127.0.0.1 - - [19/Mar/2020 01:19:14] "B[37mGET / HTTP/1.1]0m" 200 -
{ 'speed': 115, 'EngineSpeed': 3335, 'VehicleBattery': 12, 'Engine Temperature': 65.25, 'Fuellevel': 63 }
127.0.0.1 - - [19/Mar/2020 01:19:14] "B[37mGET / HTTP/1.1]0m" 200 -
{ 'speed': 115, 'EngineSpeed': 3335, 'VehicleBattery': 12, 'Engine Temperature': 65.25, 'Fuellevel': 63 }
127.0.0.1 - - [19/Mar/2020 01:19:15] "B[37mGET / HTTP/1.1]0m" 200 -
{ 'speed': 115, 'EngineSpeed': 3335, 'VehicleBattery': 12, 'Engine Temperature': 65.25, 'Fuellevel': 63 }
127.0.0.1 - - [19/Mar/2020 01:19:16] "B[37mGET / HTTP/1.1]0m" 200 -
{ 'speed': 115, 'EngineSpeed': 3335, 'VehicleBattery': 12, 'Engine Temperature': 65.25, 'Fuellevel': 63 }
```

Figure 10: Backend Communication of Vehicle Dashboard GUI

1.6 Conclusion

The purpose of this project was to create a prototypical system to realise the concept of situation awareness. The major aspects were to perceive information and to comprehend those information. All the groups were able to do their respective tasks with great success. As each of the groups work was depended on the others, they all worked a part of a larger team supporting each other, which can be highlighted in their final results.



2. Realtime Detection of Hand Movement by EMG signal

Marek Beran, František Čmuchař, Jan Ďurdák, Jiří Fišer, Petr Jahoda, Marek Helis, Václav Klíma, Matyáš Moudrý, Jan Pařízek, Ondřej Pátrovič, Jan Žaloudek

2.1 Abstract

The research in the field of processing and interpretation of electrical physiological signals by mean of machine learning has potential to change communication with our bodies. The wearable sensors, data processing and analyze in embedded devices and integration with local or global information system can provide a new level of health care or human-machine cooperation. Our research are focused on real-time detection of simple hand movement – grasp for driver using steering wheel.

A electromyography signal from forearms muscles are taken by Myoware muscle sensor, digitalized and preprocessed by Arduino and saved by Raspberry Pi (Raspberry will be used for in final detection chain). Design and training of the supervised machine learning model is provided by notebooks or Nvidia Jetson Nano microcomputer.

The main challenge for research is preparation and implementation of signal measurements. For training of machine learning models we need sufficient amount of input data, which should be acquired in a realistic environment and annotated (i.e. the the signal is augmented with timestamps of start of grasp movement). Currently we have produced data which are adversely affected by muscle fatigue, non-uniform reaction time and imperfect wire fastening.

The proposed model which is based on prediction of function which maps narrow and compressed signal window to value characterizing usability of detection (balancing the timeliness of detection and its accuracy) are tested for simple EMG signals (free of other hand movements e.g. wheel turning).

2.2 Motivation

The measurement and processing of physiological signals (ECG, EEG) has more than 100 years of history. However, for a long time it was limited to medical diagnosis and extensive and also mostly closed research in the field of physiology a computer science (for example gesture recognition allowing the input of physical action to a computer as a form of human-computer interaction [1]). This was due to the high price of equipment for the acquisition of high-quality signals and the necessary know-how in the field of medicine and electrical engineering. That has been changing lately, because with the development of electronics, cheap sensors are available, computing resources for (online) signal preprocessing and especially hardware and software for recognizing patterns through machine learning. One of the now classic tasks of machine learning deployment is to recognize hand gestures based on EMG signals (generated by muscle activity) through surface sensors. The Myoware Muscle sensor can be used for measurement (see <http://www.advancertechnologies.com/p/myoware.html>), which can be purchased for less than \$50, basic signal processing (simple resampling and normalization) can be provided by Arduino (\$20) and the application of some of the machine learning methods that require higher computing power can be provided by Raspberry PI (\$40). The process of training the selected machine learning model can be carried out on common desktop computers or specialized devices such as Jatson Nano.

Our research group consists of first year and third year students. At the beginning of the project, students had only minimal experience in the target area. Only a minority of students had python programming experience and none of them were using Raspberry PI to read and process sensor data (however, approximately half used Raspberry PI as a multimedia box or as a desktop or web computer). Students



only got acquainted with mathematically oriented software as part of a numerical mathematics course (only in the Matlab and R environments). However, everyone expressed interest in machine learning. A third identified it as a knowledge that computer science graduates should be required to have, the rest as knowledge that is good to know.

A significant part of the first workshops were therefore focused on getting acquainted with Python and its numerically oriented libraries (NumPy resp. SciPy), for signal processing (using the signal-pipe library created at the Department of Informatics) and especially machine learning libraries (scikit-learn, keras).

2.3 Introduction

Electromyography (EMG) is one of diagnostics method based on recording of electrical activity of human (animal) body. EMG records electrical activity of skeletal muscles. An electromyograph detects the electric potential generated by muscle cells and motor neurons, which transmit electrical signals that cause muscles to contract and relax.

The use of EMG is limited by the nature of the signal in which the superposition of signals from many muscle fibers and motor neurons occurs (compared to other physiological signals mainly ECG). This problem is particularly typical for surface EMG, where electrodes are placed on the skin, not using electrodes located directly in the muscle (intramuscular EMG). The surface EMG signal is also strongly influenced by noise, which can be produced by surrounding sources of electromagnetic interference.

The absence of simple patterns and the complexity of the EMG signal (see frequency spectrum obtained from the signal of the professional electromyograph Fig. 1) predisposes EMG signals for processing with machine learning. Machine learning is primarily used for classification and can therefore be used for example to classify various human movements based on the EMG signal produced by the participating muscles.

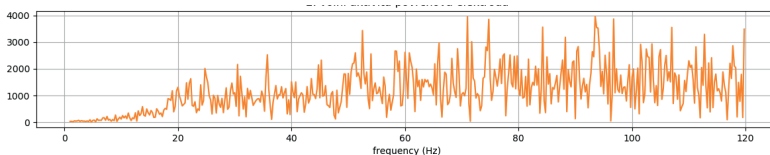


Fig. 1 Frequency spectrum of a EMG signal

2.4 Research questions and methods

The main objective of the project is to verify the usability of low-cost and potentially wearable devices for recording EMG log, which would allow to detect driver reactions in real time. For this reason, research is done on simple hardware devices, the limitations of which are determined by the following requirements for real deployment:

- contactlessness – the EMG sensor cannot be fixed to the optimal location on the driver's skin in real time (not to mention the skin preparation)
- independence from external power supply – signal pre-processing and subsequent detection must use limited computing resources
- wireless communication – the bandwidth is relatively narrow and offers only a small bit rate (because of that a low sensor frame rate)



- integration into the car information system

The test hardware device does not meet these conditions, but is suitable for simulation because of these reasons:

- *Myoware* muscle sensor is a touch sensor, but compared to professional sensors its capabilities are limited (moreover, precise location cannot be achieved without detailed anatomical knowledge)
- the maximum frame rate is significantly smaller than that of professional sensors (of the order of 200 Hz, compared to 50-100 kHz for professional devices) as well as the resolution of the AD converter (9 bits), resulting in a maximum baud rate of less than 1kpbs.

Similarly limited is the device designed for signal processing. Arduino allows only simple normalization and resampling of the signal (this also applies to the wearable incarnation of the Arduino named LilyPad, which could be used in real-world deployment). Classification using a pre-trained machine learning model cannot be performed on Arduino (this is mainly due to the limited amount of data memory, although most fixed parameters can be placed in persistent flash memory). However, the significantly more powerful Raspberry Pi or other devices in this category have enough memory and performance for online signal classification.

Myoware muscle sensor based researches focus primarily on relatively easily recognizable hand movements, as is the case, for example, in article [1], where the neural network is used to recognize five hand gestures including, but not limited to, cylindrical grasp and both types of twist (left and right). In this case, a vector of features is obtained from the signal. Features are statistical quantities obtained by transformation (mean absolute value [MAV], root mean square [RMS], waveform length [WL]) and discrimination is carried out on their basis. A similar approach is chosen by Article [2], which focuses on grasping movements. A little more information is provided by article [3], which uses the same hardware, but uses more methods for obtaining scalar features for processing and, above all, uses multiple channels obtained from three independent sensors located on the forearm. While for one channel the success rate of classification is 80%, the use of three channels increases the success rate above 90%.

All the above approaches offer offline solutions i.e. classification happens only long after the recording has been taken. The online classification is described, for example, in [4] although this article uses a professional multi-channel sensor.

With the limitations of the sensor used, as well as the need for real time detection, we decided to focus the research on only one hand movement. After discussion in our group and consultation with German students and experts with experience in the automotive industry, we have decided for online detection of initial phase of grasp movement for the following reasons:

1. in case of strong pressing, all muscles of the forearm are connected (i.e. the exact location of the sensor is not mandatory)
2. detecting this movement when using the steering wheel can be potentially useful as a strong grip of the steering wheel may indicate that the driver has registered a hazard or other strong emotion that may affect the driving of the vehicle

The project has several stages. In the first stage the simple hardware appliance has been assembled. Secondly testing data are acquired and saved in plain CSV format on desktop computers or Raspberry Pi (including annotation of signal by time labels which defines timestamps of initial phase of grasp movement). The third stage is focused on design of several machine learning models and tuning of

model metaparameters. In the last stage optimal model can be deployed on more realistic environment and demonstrated (see Figure 2).

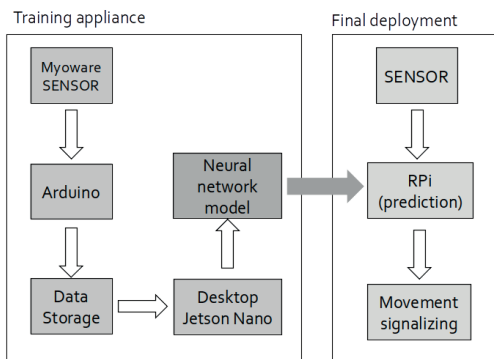


Fig. 2 Schema of training appliance and potential deployment

2.5 Implementation and demonstration

conducted research on the subject of “Deployment of self-checking and self-diagnosis in complex systems”. This task was divided on several sub-tasks and students worked on their own sub-task. These sub-tasks include solving such problems as organization test execution, using mathematic and real simulation for assessing suggested

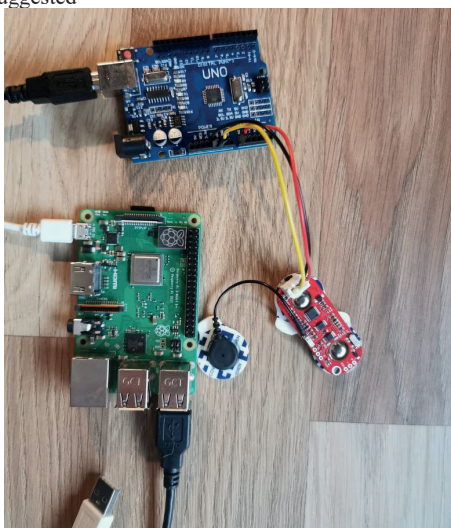


Fig. 3 Training appliance

The biggest problem turned out to be the connection of the wires to the connectors. That is why we have expanded the connectors with the JST XH 3-Pin to enable a fixed JST cable connection (see Figure 3). However, the most difficult task turned out to be the actual organization of measurements. Some problems were due only to a lack of appropriate resources. While the first measurement (second workshop) was carried out using a half-filled plastic bottle, the second measurement was already carried out with the gaming steering wheel (see Fig.4). This measurement also used a sound signal, specifying the moments when a grasp movement is to be performed. The signal time was recorded because the time signals were generated at a random interval to avoid reflex pressing. More than 100 individual grasp motions have been recorded so that machine learning can be used. However, the selection of test subjects was reduced – only two students from our group.



Fig. 4 Data collection

However, subsequent processing has shown that serious issues remain:

- tagging of start-up edges is difficult due to the large variation of reaction times (from 0.2 to 1.2 s). Synchronization between EMG recording times and beep times also poses a problem (provided by two non-synchronized devices with limited accuracy)
- some parts of the signal are difficult to interpret, the reason is probably muscle fatigue or a problem with contact fixation on the sensor (adjustment has not yet been made during this measurement).

In the field of signal processing, a solution has been chosen where the input to the machine process is directly the signal (more precisely, the fixed width window) with minimal adjustment:

1. elementary resampling: a chosen number of equidistantly spaced points within the window is selected
 - normalization: signal amplitude is normalized to an interval of $[-1,1]$ (maximum and minimum values are known in advance)

The resulting signal, including annotations, shows Figure 5 (without normalization). Note that the time between the label that defines the time of the sound signal and the beginning of the muscle reaction (i.e. reaction time) oscillates in a wide range.

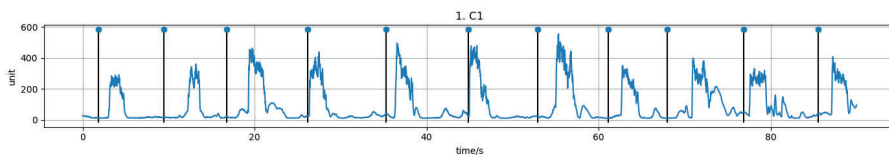


Fig. 5 Annotated EMG signal

The standard approach based on scalar features cannot be used because only the rising edge of signal and its immediate surroundings must be taken into account for the training from the rapid response request. Most features are thus irrelevant (e.g. those that describe a zero-point transition) or do not provide much new information compared to the original data.

For machine learning, a single-hidden layer neural network is used in the current implementation to predict a detection function that evaluates the detection success rate for a given window. Maximum reaches at the point of the vertex of the leading edge (i.e. in the place where the detection can be interpreted as well-timed), non-zero only its immediate surroundings with a possible shift towards the beginning of the leading edge (detection of initial phase of grasp movement) or vice versa in the direction of continued muscle extension (slightly delayed detection, which may, however, provide more relevant results).

2.6 Results

Current results are currently influenced by the impossibility of unambiguous annotation of moments when the initial phase of grasp movement occurs, i.e. it is not possible to precisely define the function whose prediction is the goal of machine learning. For this reason, the prediction has so far only been verified for data where annotation is easier, i.e. in the case where the subject does not perform any further movements and measurements are not affected by muscle fatigue.

See, for example, results visualized in Figure 6. EMG signal is represented by light line, light peaks are detection function and predicted function is drawn in black. Higher function values (e.g. above 0.33 can be interpreted as grip origin detection).

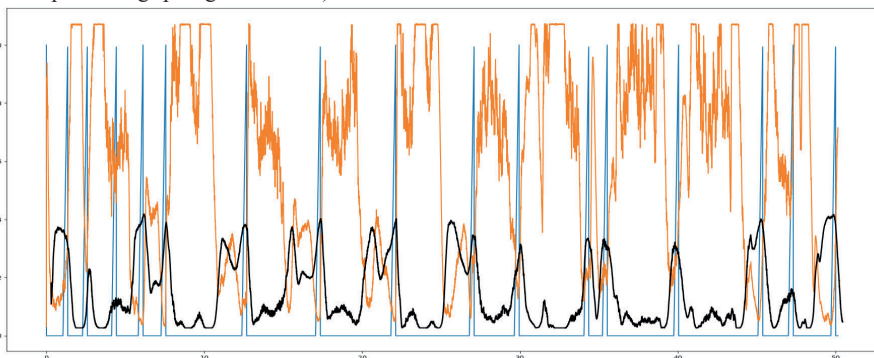


Fig. 6: Prediction of detection function

The graph shows that the approach described above is applicable, but it should be admitted that in the case of a simple signal (without further movements and without noise), the detection of leading edges



can be done in simpler ways. More practical verification of the usability of the method (including the application of more complex types of machine learning such as recurrent neural networks) requires better and more extensive input data and, above all, quality data annotation.

2.7 Reference

- [1] Camacho Navarro, Jhonatan; Leon-Vargas, Fabian; Barrero Perez, Jaime. Emg-Based System for basic hand movement recognition. Medellín, v. 79, n. 171, p. 41-49, Feb. 2012 . Available from <http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0012-73532012000100005&lng=en&nrm=iso>. access on 19 Sept. 2020.
- [2] Abu, Mohd Azlan & Rosleesham, Syazwani & Suboh, Mohd Zubir & Syazwan, Mohd & Yid, Md & Kornain, Zainudin & Jamaluddin, Nurul Fauzani. (2020). Classification of EMG signal for multiple hand gestures based on neural network. 17. 256-263. 10.11591/ijeecs.v17.i1.pp256-263.
- [3] Putra D.S & Weru Y.U.W and Fitriady. Pattern recognition of electromyography (EMG) signal for wrist movement using learning vector quantization (LVQ). IOP Conference Series: Materials Science and Engineering, Volume 506, 1st South Aceh International Conference on Engineering and Technology (SAICOET) 8–9 December 2018, Politeknik Aceh Selatan Campus, Indonesia
- [4] Hassan Jaffar, Y., & Hussein Ali Al-Timemy, A. (2018). Investigation of Pattern Recognition System Based On Electromyography Signals for Optimal Electrodes' Number and Positions. Association of Arab Universities Journal of Engineering Sciences, 25(2), 174-188. Retrieved from <https://jaaru.org/index.php/auisseng/article/view/142>
- [5] Zhang, Z., Yang, K., Qian, J., & Zhang, L. (2019). Real-Time Surface EMG Pattern Recognition for Hand Gestures Based on an Artificial Neural Network. Sensors (Basel, Switzerland), 19(14), 3170. <https://doi.org/10.3390/s19143170>



3. Digital Processes for interactive Mentoring Approaches

Ummay Ubaida Shegupta, René Schmidt

3.1 Abstract

Mentoring, specifically blended mentoring is considered as one of the best teaching methods based on the recent scientific evidences. In this paper, it is described how a mentoring strategy has been integrated into the project work of computer engineering education with the objective of introducing students the scientific way of working. The students work intensively on their own topic, which in turn is a part of a complex system. Throughout the entire duration of the project, the students are supported by the mentors in order to maintain the scientific way of working and to achieve their project goals. Various technical systems are used to support the students ensuring a lively support flow. The results show the effectiveness of the method.

3.2 Introduction

Mentoring is considered to be one of the best didactic methods worldwide, but it is currently rarely used in high education context. Recent research focuses mainly the fields of science, technology, engineering and mathematics (STEM)[1]. Furthermore, particular application-oriented disciplines of computer science education are addressed [2], [3]. The mentor uses lessons characterized by a rich understanding of the cognitive process and the mentee's individual needs and interests to generate a mutual mentoring process [4]. In this context, this block analyses a mentoring approach in project work. In order to explore mentoring approaches, mentoring opportunities are necessary. These are created by providing particular tasks for the students in the context of learning data extraction and processing. The large fraction of learning data is generated in the learning management system (LMS) "OPAL" of TU Chemnitz. It provides the ability to access the data to acknowledge student's learning experience and taking data driven decisions to improve the quality of education. This is known as learning analytics [5], which can give an in-depth insight about the implemented mentoring approach with contrast to students' competence gain. For this reason, we are investigating possibilities to extract feasible data from the LMS "OPAL". Furthermore, the state of the art of data analytics using artificial intelligence methods needs to be analyzed compared to analyzed rule based approaches providing the basis of data usage approaches. The mentoring objective is to bring students closer to the scientific way of working through close cooperation supported by technical methods increasing the interactivity and intensifying the mentoring process. This report aims to explore a blended mentoring approach in context of a workshop in combination with the video conferencing tool Big Blue Button and the team communicator Slack to ensure scientific working methodology during the project work of the students.

3.3 Database

The objective to bring students closer to the scientific way of working, therefore 6 Tasks in the context of learning data extraction has been defined, based on working fields identified in cooperation with the tech4comp project. Reasoned by data protection law conditions, the person data cannot be made available to students. For this reason an evaluation course in OPAL has been created, which contains all necessary course elements. The students can use it freely to generate all necessary data as examples for artificial data generation following the given data structure. This course has mainly 6 types of elements: Registration-Element, Upload-Folder- Element for the files from supervisors, 6 types of Calendar-Elements, two types of Blog-Elements, five types of Checklists-Elements and five types of ONYX Test-Elements. Log data can be acquired from all course elements and these different types of elements provide different types of learning experience data of the students. For conducting learning analytics, these data needs to be exported from the data archiving tool



and exported function from the "Course evaluation" and "Statistics" functions of the course. The data are exported as XLS Files. To perform Learning analytics it is necessary to extract and evaluate the data providing the particular task basis. The Task description are as follow:

Task 1: Analysis and Implementation of Methods for iCal Calendar Data Extraction

Opal offers the possibility to integrate iCal calendars into courses. For data analytics it is necessary to automatically read the data of the calendar and store it in a database. Therefore a script or program has to be developed which reads all information of the calendar, prepares it and stores it in a database. In the first step, the data must be understood and the information that can be extracted has to be identified. On this basis, the database design is to be jointly defined and implemented in cooperation with the other teams. The data will then be automatically extracted and prepared for database storage. Finally, the data will be automatically stored in the database.

Task 2: Analysis and Implementation of Methods for ONYX Test Evaluation

Opal offers the possibility to integrate ONYX Tests into courses. These tests can be extracted by the archive function from Opal as Excel files. AFor data analytics it is necessary to automatically read this data and store it in a database. Therefore, a script or program has to be developed which reads all information of the exported data file, prepares it and stores it in a database. In the first step, the data must be understood and the information that can be extracted has to be identified. On this basis, the database design is to be jointly defined and implemented in cooperation with the other teams. The data will then be automatically extracted and prepared for database storage. Finally, the data will automatically be stored in the database.

Task 3: Implementation of a GUI for Data Visualization

The data exported from task 1 to 4 stored in the database are to be visualized in this task. Therefore, a database design has to be developed together with the colleagues of the other tasks. Afterwards the data should be presented as web components on a website. For this purpose statistics have to be created to visualize the temporal evolution (e.g. by semester) and the current state of the semester.

Task 4: AI for Automatic Test Data Evaluation

Knowledge tests are very important in the blended learning environment. However, a lot of data has to be processed. One goal of such data processing is the generation of individual feedback for the respective pupil/student. To create a good feedback it is necessary to understand the connections between tasks and to be able to draw conclusions about the actual misconceptions. At this point artificial intelligence should be used to identify such misconceptions. The task is to conduct a systematic literature survey (with evidence), to analyse concepts for the identification of misconceptions and, if necessary, to extract the required conditions. On this basis, guidelines should be developed enabling AI-based evaluation and feedback generation.

Task 5: Comparison of AI Algorithms for Data Analysis

In recent years, AI has become more and more important in the processing of data masses. The importance is also increasing in the teaching environment where new blended learning approaches generate considerable amounts of data. The task is to create a systematic literature survey (with evidence) which AI based algorithms can be used for data processing in the teaching environment, which information can be extracted with which method and which necessary conditions have to be fulfilled for the KI systems to work correctly. On this basis, guidelines should be developed enabling AI-based data evaluation in the learning context.

Task 6: Data Acquisition and Generation of Statistical Data from Log Files



For data analytics it is necessary to automatically read the log data of the course that has the access time and usage frequency of different course elements and store it in a database. Therefore a script or program has to be developed which reads all information of the exported data file, prepares it and stores it in a database. In the first step, the data must be understood and the information that can be extracted has to be identified. On this basis, the database design is to be jointly defined and implemented in cooperation with the other teams. The data will then be automatically extracted and prepared for database storage and statistical analysis. Finally, the data will automatically be stored in the database.

3.4 Research Methodology

Student Background

There are six students registered to complete the tasks of block 2. They are all Master students of Chemnitz university of Technology (TUC). They are mainly from two faculties - Fakultät für Informatik and Fakultät für Elektrotechnik und Informationstechnik. The gender ratio can be considered as balanced with 4 men and 2 women. Two of the students are enrolled in first semester whereas the other students are enrolled within a range of 3 to 7 semesters (Table 1).

Table 1: Students' demographic details

Student	Gender	Faculty	Field of studies	Current Semester
1	Female	Fakultät für Informatik	Master Automotive Software Engineering	1
2	Female	Fakultät für Informatik	Master Automotive Software Engineering	1
3	Male	Fakultät für Elektrotechnik und Informationstechnik	Master Embedded Systems	3
4	Male	Fakultät für Elektrotechnik und Informationstechnik	Master Embedded Systems	5
5	Male	Fakultät für Informatik	Master Automotive Software Engineering	6
6	Male	Fakultät für Elektrotechnik und Informationstechnik	Master Embedded Systems	7

The six students registered for this block have chosen 5 of the described topics. The voluntary chosen topics are:

1. Analysis and Implementation of Methods for iCal Calendar Data Extraction
2. Analysis and Implementation of Methods for ONYX Test Evaluation
3. Data Acquisition and Generation of Statistical Data from Log Files



4. Implementation of a GUI for Data Visualization
5. AI for Automatic Test Data Evaluation

3.5 Guide to Task Completion

The **first** step is to recognize the data and filter the information. On this basis, the database should be defined and implemented as team across the groups and filled manually with artificial data once. Afterwards, the data has to be

automatically extracted and prepared for **database storage** and statistical evaluation. The exported data has to be visualized in the **fourth** task using the manually data of the database for the prototyping process.

Subsequently, the

data has to be presented as components of a website. For this purpose, statistics on the temporal evolution and the current status of the semester have to be conceptualized in advance.

Knowledge queries are very important in a mixed learning environment, but a lot of data has to be processed. One goal of data processing is the individual feedback from the pupils. In order to give good feedback, it is necessary to

understand the connection between tasks and draw conclusions about possible misunderstandings. Task **five** required that a systematic literary survey be carried out, with clues for the analysis of concepts that serve to identify misunderstandings. With the help of the results, it should be possible to generate ratings and feedback with the help of an **AI**.

3.6 Theoretical framework

Mentoring has already been proven effective to provide support to university students to facilitate their learning and to provide guidance [6]. When situated within the context of a complex real-world task, it is one of the most successful ways for learners to develop mastery [7]. The pedagogical key of academic mentoring is its use of modeling to help a student explore the inner craft of the intellect [8]. In IKON internship, the students of this block has been supervised and facilitated based on the pedagogy of mentoring. All the three mentors have tried to illustrate and transmit meta-cognitive strategies, techniques for scaffolding the students' own thinking and learning process. Mentoring focuses on relationship that is supported by the education theory of "Zone of Proximal Development" (ZPD) by Russian psychologist Lev Vygotsky [8]. Vygotsky emphasizes the social nature of learning [8]. The student learns by participating in challenging activities with a "more competent other" who supports the elements of the work that the student cannot do individually. Gradually the student takes over direction and moves toward independence [8]. Following the theory of ZPD, the students got frequent guidance at the beginning of the internship. Two face to face residential workshop respectively three days and five days long happened at IBS Laubusch, Germany, in order to provide intensive mentoring support. The students completed the task of understanding the requirements of the internship, state of the art analysis and developed their first plan of implementation in these sessions. The scaffolding techniques of the mentor helped them to get gradual support of consultation, instructions, motivation along with providing continuous feedback to their learning activities. Afterwards, the mentoring process steered the students to take over the ownership of their own learning, as it is supported by ZPD. Due to the global pandemic situation and formal obligation the face to face mentoring sessions of the IKON intership had to be modified. As the theory of ZPD was followed in this mentoring process, it had helped to easily cope the situation to render the mentoring support via e-learning tools. The communication between the mentor and the internship students were maintained via Email and SLACK. Whenever the students needed in-depth support, the online meeting tool Big Blue Button had been used. These communication enabled students to keep track of their respective implementation tasks with the minimum possible guidance from the mentors. Finally, in the last official online



workshop the students were able to show their implementation individually by themselves. Here they presented themselves as competent individuals who could perform the difficult tasks by their own as stated by the theory of ZPD.

In the above-mentioned methodology, it indicates that the mentoring process has occurred with combination of traditional and online sessions, it can be called blended mentoring approach accordingly [8]. This blended approach has enabled the mentor to provide task related continuous feedback and to facilitate the students to sustain their motivation to complete their assigned tasks.

3.7 Research Objective

Reasonably, the mentees' view of the applied blended mentoring approach is necessary to continue this support in the next round of IKON project. Besides, the student's task oriented goal, which is already described in the task description section, the research objectives are to explore the mentees' perception about the mentoring support, about their cognitive, affective and skill development, their preferences of the mentoring modes, mentor's roles and the effectiveness of different consultation media. Along with these, finding out the shortcomings of the mentoring process to improve it in future is an underlying objective of this research.

3.8 Task Implementation and Results

Task 1: Analysis and implementation of the methods for iCal Calendar data extraction

The calendar data from OPAL are exported to a file in .ics format by using the ICS-URL. To transfer the data in .ics file to a database, selecting an intermediate file format was presented with choice of JSON / XML / CSV files. Based on the universality, flexibility and ease of parsing, JSON was chosen. For conversion to JSON format, the following options were explored:

- cloud solution (<https://ical-to-json.herokuapp.com/>)
- Node JS (<https://github.com/adrianlee44/ical2json>)
- Python (https://pypi.org/project/ical_dict/)

The first option was discarded due to data privacy concerns. The Node JS solution is rejected due to the dependence on additional runtime environment. Finally, Python ICAL_dict Package was chosen combined with Jicson package

(<https://github.com/CalyFactory/python-jicson.git>). Furthermore access to a relational database is required. MySQL Database and Firebase Realtime database were considered, but MySQL was chosen as it could be used by all groups.

The general implementation is shown in the **Figure 1**. The first step is the export of the ICS File. Afterwards it is parsed by the Python Package to JSON format subsequently the extracted information are proceed and stored in the database using MySQL.

In summary, calendar data in the iCal .ics format was processed using the implemented Python script, which involved filtering of data, modifying the data format to suit the destination and elimination of blank data. This was then converted to a JSON file. The data in the JSON file could be imported into a table in a MySQL local database. This import process was repeated successfully to verify whether an integration into a common database was possible. This sort of data extraction can also be used for any calendaring application which uses the .ics format. A seamless integration ensures that this concept can be utilized to store larger data from different sources and can be further analyzed using visualization tools.

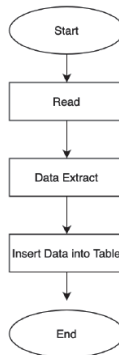


Figure 1: General implementation flow of Task 1.

Task 2: Analysis and Implementation of Methods for ONYX Test Evaluation

This task has been solved by implementing a system where users can upload a XLS-file exported by OPAL containing ONYX-Test results. At the beginning of the process the system extracts data from the uploaded files and insert into a database. For the development process Java Script (JS) has been used. For front-end development ReactJS [9] has been used. It is a very popular JS library to efficiently create user interfaces. The back-end has been realized with NodeJS [10]. NodeJS is used, reasoned by its code reusability for the front-end. Along with it ExpressJS [11] has been used extending the features used for mobile devices. Furthermore, the database was realized as MySQL DB [12].

The overall process flow is shown in **Figure 2**. The user uploads the extracted XLS-File by Opal. Afterwards the extraction of the data is taking part. Therefore, necessary student information have to be extracted by an additional member list exported by Opal. This is necessary to identify the correct student and matching it with the other tasks work. Afterwards the ONYX-Test data files are processed extracting required data like:

- Student Name
- Matriculation number
- Dates of Test execution
- Results per task
- Given answers

This provides the basis for the front-end team to calculate the term of study from this year. In the next step, the system checks the existence of the student in the DB by the registration number. If the student exists in the DB then the system updates the specific test column with the test score value. On the other hand if the student does not exist in the DB the system insert the personal data with the test score and student details value. In this system insertion will be occurred only one time, when the DB is fully empty that time this insertion will be occurred. After that, whenever the system runs the DB will be updated.

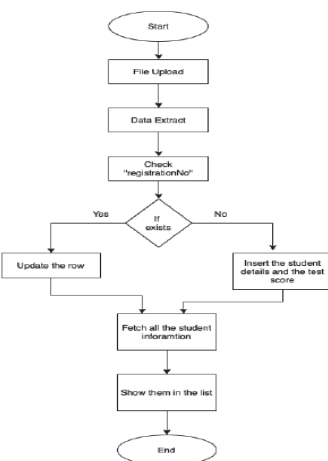


Figure 2: General implementation flow of Task 2.

Task 3: Implementation of a GUI for Data Visualization

This task has been solved in teamwork of two students. One student solved the task by using a commercial tool “Tableau”. As input the extracted Excel Files from ONXY and Opal has been converted to CSV and loaded as Database into the tool. Afterwards the included Tableau features Columns Renaming, Splitting, Removing, Filtering, Pivoting, Aliasing, Data type Manipulations etc. are used to order and manage the data. Based on the organized data filter and statistical functions are used to generate plots as shown in Figure 3.



Figure 3: Data visualization in Tableau.



On the contrary, the 2nd student generated a similar dashboard based on Python 3.7 and the corresponding Dtale package. Dtale is a web based data analytic tool for visualizing and interpreting data library for Python. The general process flow is identical to the 1st approach since Dtale provides similar evaluation methods resulting in an alternative visualization shown in **Figure 4**.

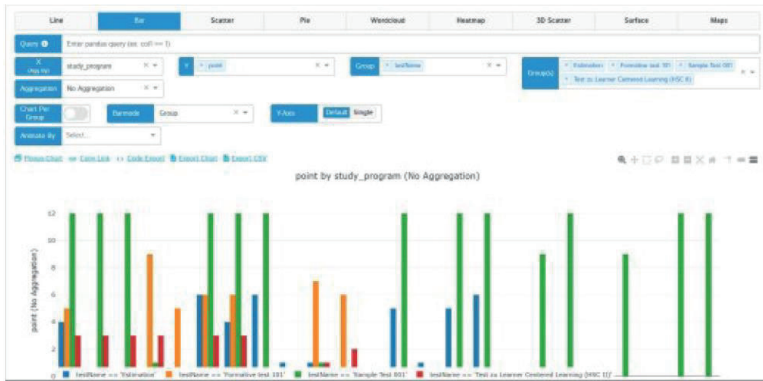


Figure 4: Data visualization in Dtale.

Task 4: AI for Automatic Test Data Evaluation

In Task 4 a structured literature review has to be done. The basis of of the survey was an initial literature search with the keywords AI, automatic test and evaluation. The gathered documents has been analyzed by the keywords used in the keyword section of the paper. The most significant keywords have been comprised in the following **Figure 5**.

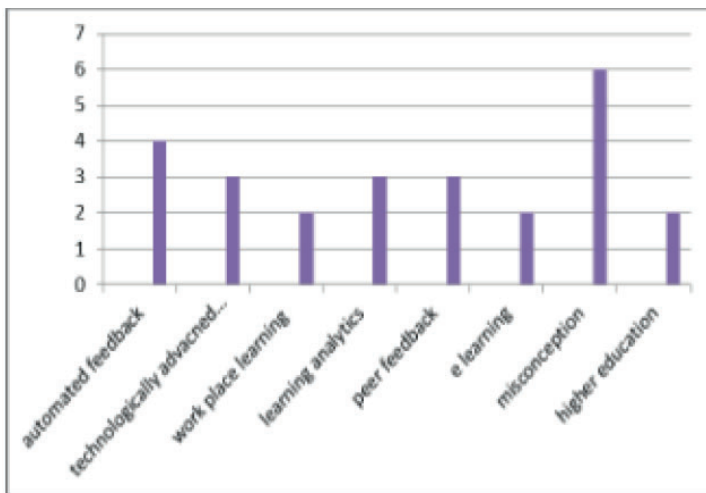


Figure 5: Keyword frequency of 1st literature survey

It is evident that, the two most frequent keywords are “automated feedback” and misconception. The general approach of automated feedback bases on test cases or criteria. In test case based feedback the input sent in by students, is judged based on the basis of certain test cases. If the given input satisfies the design test case a positive feedback is given. If they do not the failed test case is given out as feedback. Eg. If for a task to be successful it should satisfy 3 cases a, b and c. And say it is not able to satisfy the case b. output is given as “The solution fails to satisfy case B.” Alternatively peer based feedback is to be considered. In peer based feedback a number of individual working on the same or similar type of topics are linked together. They work with each other sharing their concepts and thinking of possible or definite solutions.

The 2nd most used keyword is Misconception. The term “Misconception” has various definitions. Flower & Jaoude (1987) defined misconceptions as, “an inaccurate understanding of a concept, the misuse of concept the incorrect

Classification of concept examples, confusion between differing concepts, improper Hierarchical relationships or on- or under – under generalizing of concepts”. The following general types of misconceptions are distinguished:

1. Preconceived notions: Making a concept with incomplete observation.
2. Nonscientific beliefs: Concepts learned from nonscientific sources
3. Conceptual Understanding: Misapplication of general concepts.
4. Vernacular misconception: Misunderstanding of the meaning.
5. Factual misconception: These are the misconception picked up in early age, which are carried into adulthood.

Furthermore, sources of misconception are represented by:



1. Books: The books refereed might not have enough evidences of the concept.
2. Teachers: The language of the teacher or the way the concept is taught might lead to misconception.
3. Family: Family being most influential part of life in early age has a major impact on a person. The misconceptions from the members of family might carry forward to next generation,
4. Peers: The misconceptions can be picked up from fellow student or peer.
5. Media: Media being a very powerful tool can affect the way a person thinks.
6. Cultural beliefs: Superstations and cultural beliefs which come from nonscientific source can cause misconception.
7. Personal experience: Some incidents in day to day life might result in incomplete observation and might lead to misconception.

In Summary, the automated feedback helps in removal of misconception using various tools and methods. Its helps in better learning and understanding of a concept for a student.

Task 5: Comparison of AI Algorithms for Data Analysis

This topic hasn't been chosen by a student, so no results and summery can be provided.

Task 6: Data Acquisition and Generation of Statistical Data from Log Files

The purpose of the data extraction and generating statistical data from log file of OPAL course is to generate the total accessing courses with the name of the course element in categories of days, week, monthly etc. The idea is to combine in a useful tool a didactically theory with users' data and serve the needs of stakeholders. The approach of generating statistical data from the given log files bases on the analysis of learning activities in online OPAL courses from a dedicated point of view, thus going beyond than simply counting and visualizing the numbers of clicks. For this purpose, a model of the log file analysis was proposed. These requirements have been collected through exported excel files where all necessary data are provided. Based on this, a model composed by a concept map, a solution map and use cases was developed. Implementation started from these use cases. The statistical data are available for deployment and exploitation in academic institutions.

The realization has been done using MySQL. The first step is represented by the creation of the database and the corresponding data table. The data table contains seven columns:

1. columnsCreationdate: Date and Time where the event has been created.
2. userName: empty due to privacy policy regulations from OPAL.
3. actionCrudType column mentioned the reading of data.
4. actionVerb: what happened mostly launching the website.
5. actionObject: defining the object type as course element.
6. parentResName: basically the course name.
7. targetResName: the course element name.

After creating the data table the exported Opal data where converted in CSV file and imported in MySQL database. From the imported data element and total access column which is the counting or sum of the accessing number. The maximum access is 96 and the name of course element is "Test course for IKON Internship". On the other hand, the minimum total accessing is 11 and the name of the course element is "OPAL Check-list Data Extraction 4".

3.9 Results & Discussion

Mentor's Perspective



The mentoring process conducted through the mentors has been based on the theory of Zone of Proximal Development (ZPD). The interns have been mentored intensively during the first stage of IKON internship. According to ZPD, the scaffolding techniques have been used during these intensive consultation sessions. The mentors have tried to model similar examples of their assigned tasks and answered all the possible queries of the mentees to clarify the confusions of the tasks. The mentees have got frequent direct constructive feedback for the first three months of their internship. Then they have been working on the theoretical basis and conceptual development of their assigned task, as these are fundamental for the implementation of the technical part.

Overall, the mentors are satisfied with the guidance that they provided. All the mentee have been able to complete their internship tasks. Since all the mentee have to works in whole groups, it has been found out that all of them are supportive to each other. They have been successfully worked in the team because of their sound communication and social skills. As they have to finish their task with less possible guidance during the last months of the internship, there has been a reduction in the role of ‘Instructor’ as mentor.

In course of mentoring, the mentors have realized that most of the mentees lack in programming experience resulted in lower quality of the implementation part. The implementation have been successful though the expectation of the mentors is higher.

Mentee’s Perspective

The responses of the mentees from the close-ended questionnaire about the mentoring support shows diversified prospect about the provided mentoring process. All of them reported about being mentored via face-to-face meetings, online synchronous and asynchronous communication media.

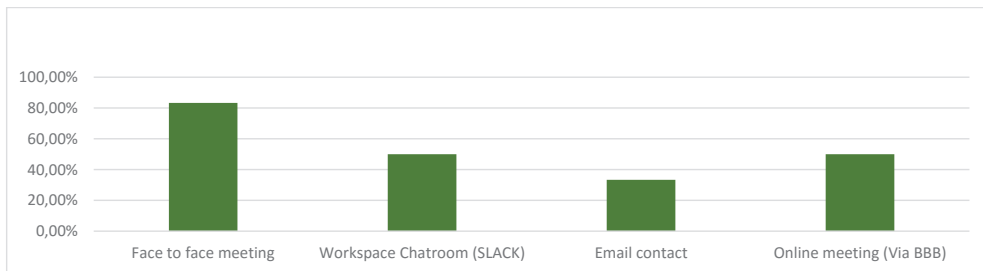


Figure 6: Effectiveness of the blended mentoring media

They have opined that the face-to-face meetings provided the most effective mentoring support (See Fig.6). Among the other modes of communication, Slack and online meeting room Big Blue Button have been reported as effective mentoring tool. Their responses show that they have been most benefited by the mentoring support in increasing their self-confidence. The Fig.2 below shows their perception that their problem solving skill and presentation skill have been improved following their self-confidence.

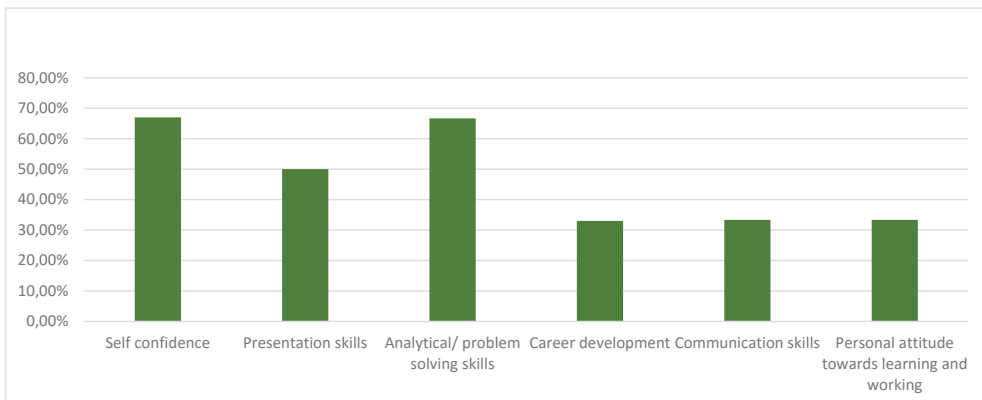


Figure 7: Mentee's improvement via mentoring support

The Fig.7 shows their affective and skill development in this whole mentoring process. However, the mentees have realized the differentiated mentor's roles according to their needs shown in Fig.8.

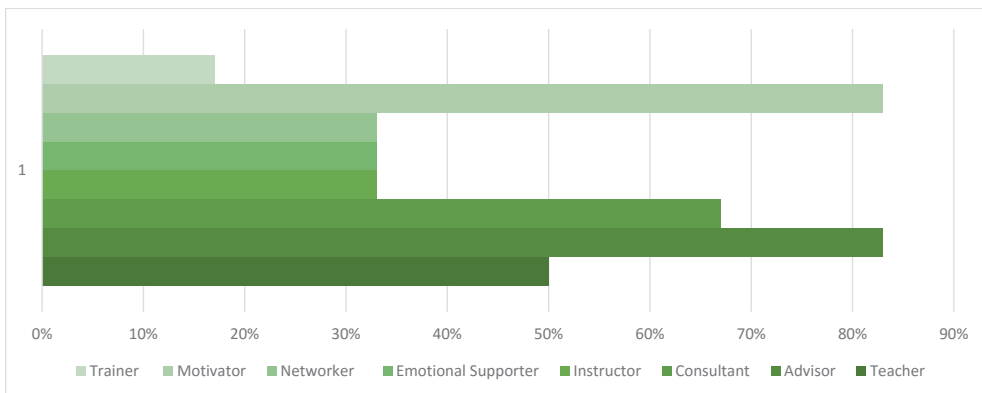


Figure 8: Roles of Mentor

The Fig.8 above shows that the mentors have taken mostly the role of advisor and motivator in all through the internship period. The other significant roles that the mentors have played are consultant and teacher. The role that the mentor have played the least is as the trainer. This supports the theory of ZPD in rendering the scaffolding support via mentoring as motivator and advisor and less as a trainer.

They have also been asked about their competency achievement according to the Information Technology Competence model (Fig.9).

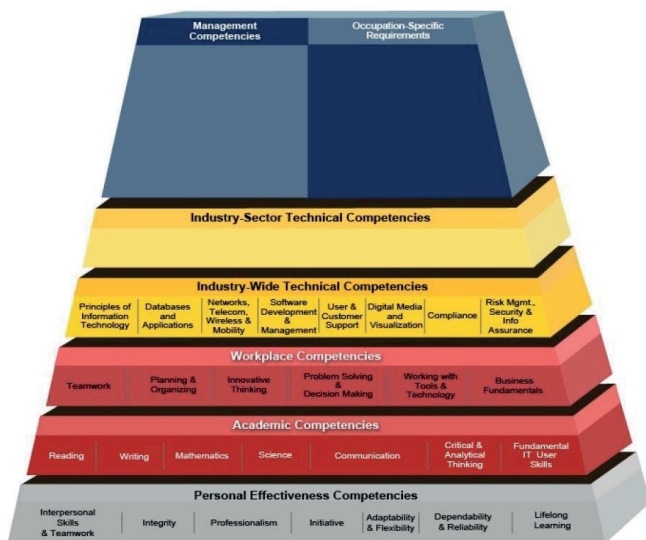


Figure 9: Information Technology Competency Model

The reported responses show an average group-wise gain of ‘Industry-Wide Technical Competencies’ 37.5%, whereas the ‘Workplace’ and ‘Academic’ competencies are respectively 61% and 40.5%. Most of the mentee’s about 61.9% have reported their ‘Personal Effectiveness Competencies’ gain. This result shows altogether the cognitive, affective and skill development from mentee’s point of view. In addition, it is according to our objective to bring the academic working style closer to the students (see research objectives). In this process of mentoring, the mentors characteristics have been one of the crucial criteria that was asked to the mentees. Among those, the following table 2 shows the most significant ones from mentee’s responses.

Table 2: Mentee’s opinion of the mentors’ characteristics

Mentor’s Characteristics	Opinion in Percentage
Approachable	83%
Accessible	50%
Supportive	83%
Encouraging	83%
Provides Constructive feedback	42%
Provides guidance	42%
Provides timely response	33%
Provides comprehensive response	33%
Suggests appropriate resources	50%

The mentees have responded towards agreement of the statement shown as a summary in table 2. The mentors have been accessible whenever they needed and recommended them supplementary resources as stated by



50% of the mentees. To 83% of the mentees the mentors have been easily approachable, supportive and encouraging while only 42% of them reported that their mentors provide guidance and constructive feedback. Lastly, 33% have responded that they have had timely and comprehensive response from their mentors. This finding indicate the role of mentors mostly as motivator and advisor that is in line with previous result.

Another important finding from the recommendation of the mentees are that they have needed more support in their technical skill development. This result is in line with the abovementioned mentor's perspective that the mentees have less programming experience.

Above all, the findings of both the task related results and mentoring questionnaire indicate that the mentees have received adequate mentoring support. However, some amendments are needed in the next round in terms of considering the individual needs of the mentees. Otherwise, the blended mentoring support has helped the mentees to complete their assigned tasks.

3.10 Conclusion

In conclusion, we can say that the mentoring process has been a success based on the results. Nonetheless, the insufficient programming skills of the students, the mentors have been able to help the students finishing their programming tasks. The findings indicate that the mentees are introduced with the scientific way of working in this mentoring process. Moreover, the mentees' have realized their inner improvement in analytical thinking. Skills like this represent the key competences essential to achieve in the field of Computer Engineering.

As mentioned before, the mentoring support has to be modified and adapted due to the global pandemic situation. Even though, this process of blended mentoring has been successful reasoned by well-chosen technology.

3.11 Acknowledgement

We gratefully acknowledge the cooperation of our project partners and the support of tech4comp founded by BMBF (16DHB2106)

3.12 References

- [1] H. Stoeger, X. Duan, S. Schirmer, T. Greindl, and A. Ziegler, "The effectiveness of a one-year online mentoring program for girls in stem," *Computers & Education*, vol. 69, pp. 408–418, 2013.
- [2] J. McCarthy, "International design collaboration and mentoring for tertiary students through facebook," *ustralasian Journal of Educational Technology*, vol. 28, no. 5, 2012.
- [3] E. H. Trainer, A. Kalyanasundaram, and J. D. Herbsleb, "E-mentoring for software engineering: a socio technical perspective," in *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*. IEEE, 2017, pp. 107–116.
- [4] A. Ziegler, "Mentoring: Conceptual basics and effectiveness analysis," *Mentoring: Theoretical Background, empirical Findings and Practical Applications*, pp. 7–29, 2009.
- [5] KD Mattingly, MC Rice and JL Berge, "Learning analytics as a tool for closing the assessment loop in higher education," *Knowledge management & e-learning: An international journal*, vol. 4, no. 3, pp. 236-47, 2012.



- [6] G. Crisp, I. Cruz, “Mentoring college students: A critical review of the literature between 1990 and 2007”, Research in higher education, vol. 50, no. 6, pp. 525-45, 2009.
- [7] L. Akin and J. Hilbun, “E-Mentoring in Three Voices,” Online Journal of Distance Learning Administration, vol. 10, no. 1, 2007. [8] E. Gleibermann, A Pedagogy of Mentoring, 2014.
- [8] W.M. Murphy, “From e-mentoring to blended mentoring: increasing students' developmental initiation and mentors' satisfaction“, Academy of Management Learning & Education, vol.10, no. 4, pp. 606-622, 2011.
- [9] facebook/react: A declarative, efficient, and flexible ... - github, <https://github.com/facebook/react> , (accessed Jun. 23, 2020)
- [10] Guides | node.js, <https://nodejs.org/en/docs/guides/> , (accessed Jun. 22, 2020)
- [11] Expressjs tutorial - tutorialspoint, <https://www.tutorialspoint.com/expressjs/index.htm> , (accessed Jun. 27, 2020)
- [12] Mysql documentation - mysql, <https://dev.mysql.com/doc/> , (accessed Jun. 27, 2020)



4. Implementation of digital processes via Reliance program from Scada

RNDR. Krejčí Jan, Ph.D.

4.1 Abstract

Data visualization is essential for better monitoring and control of data especially in industrial processes. In this paper, it is described how to simply implement such software to visualize data. During the whole project students were supported by their project supervisor and tutor in case of any questions. All the technical systems needed for the project were given to the students.

4.2 Motivation

The team for block 2 consisted of 14 students. All are in bachelor's degree programme. Three of the students are enrolled in the first year, two in the second year and the rest are in the third year of bachelors. All of the students are from Faculty of Science at Univerzita J. E. Purkyně. Out of 14 students, 4 are female students (Table 1).

Student	Gender	Faculty	Field of studies	year
1	Female	Faculty of Science	Bachelor's in Applied Computer Science	1
2	Male	Faculty of Science	Bachelor's in Applied Computer Science	1
3	Male	Faculty of Science	Bachelor's in Applied Computer Science	1
4	Male	Faculty of Science	Bachelor's in Applied Computer Science	2
5	Male	Faculty of Science	Bachelor's in Applied Computer Science	2
6	Male	Faculty of Science	Bachelor's in Applied Computer Science	3
7	Male	Faculty of Science	Bachelor's in Applied Computer Science	3
8	Male	Faculty of Science	Bachelor's in Applied Computer Science	3
9	Male	Faculty of Science	Bachelor's in Applied Computer Science	3
10	Male	Faculty of Science	Bachelor's in Applied Computer Science	3



11	Male	Faculty of Science	Bachelor's in Applied Computer Science	3
12	Female	Faculty of Science	Bachelor's in Applied Computer Science	3
13	Female	Faculty of Science	Bachelor's in Applied Computer Science	3
14	Female	Faculty of Science	Two-subject bachelor's in Geography and Informatics	3

Table 1

4.3 Introduction

Reliance is a Supervisory Control And Data Acquisition software and Human-Machine Interface (SCADA/HMI) designed for visualizing and controlling various industrial processes. Not only that, this software is also used at many universities and educational institutions all around the world. It's also used in scientific research and organizations. There are many modules of Reliance (e.g. Reliance Design, Reliance Control, Reliance View, Reliance Server, Reliance Control Server, Reliance Web Client etc.) Our team built a small environment using L2 (also called Data link layer) and L3 (also network layer) of ISO/OSI model to implement and demonstrate Reliance software modules Reliance Design and Reliance Server.

4.4 Research questions and Methods

The research question is how to implement Industrial SCADA/HMI system software Reliance. First step to do that is to link up Data link layer, which is the second layer of ISO/OSI model or Open Systems Interconnection model. The model [1] was designed to enable dissimilar computers in multi-vendor environments to share information transparently. The OSI structure calls for cooperation among systems of different manufacture and design. There are seven layers of the OSI model that communicate between one end system and another. The layers cover nearly all aspects of information flow, from applications-related services provided at the Application Layer to the physical connection of devices to the communications medium at the Physical Layer. All seven layers have long since been defined and ISO protocols ratified for each layer, though extensions have been made occasionally. Although the model has changed the way we look at networking, the dream of complete OSI-compliance has not come to fruition. The causes are varied, but this is essentially because OSI protocols are too expensive and too complex compared with other protocols that have become de facto standards in their own right , the model has formed the way we think of the structure of networks, and the model itself is always referred to in internetworking matters. Here is a table showing all 7 layers with their names and a short description of purpose.



The Seven Layers of OSI

Layer	Name	Purpose
7	Application	Applications and application interfaces for OSI networks. Provides access to lower-layer functions and services.
6	Presentation	Negotiates syntactic representation for the Presentation Layer and performs data transformations.
5	Session	Coordinates connection and interaction between applications. Establishes a dialog, manages and synchronizes the direction of data flow.
4	Transport	Ensures end-to-end data transfer between applications, data integrity, and service quality. Assembles data packets for routing by Layer 3.
3	Network	Routes and relays data units among network nodes.
2	Data Link	Transfers data units from one network node to another over a transmission circuit. Ensures data integrity between nodes.
1	Physical	Delimits and encodes the bits onto the physical medium.

Table 2

As we can see in the table above the Data link layer is responsible for communication between devices. It does that by exchanging frames (data packets) between both nodes. [2] It describes methods for moving information between multiple devices within the same logical network based on physical device addressing (MAC address). It makes the physical layer, a raw transmission facility, to a reliable link. It makes the physical layer error free to upper layer (network). The basic purposes for Data Link Layer protocol implementations are:

- Organize Physical Layers bits into logical groups of information called frames.
- Detect errors.
- Control data flow.
- Identify computers on the network.



After setting up L2 our team ‘upgraded’ it to L3, which is Network layer of ISO/OSI model. Basic task of this layer is to transfer data packets from one node to another. [2] The Data link layer oversees the delivery of the packet delivery between two systems on the same network, whereas the network layer that each packet gets to the final destination. If systems are connected to the same link, there is no need for network layer. If systems are attached to different networks then it is inevitable. This layer defines:

- Logical Network structures and addressing.
- Route discovery and selection.
- Network layer flow control and error control.
- The network connectivity hardware associated with network layer.
- Routers.
- Network layer protocols.
- IP (Internet Protocol) etc.

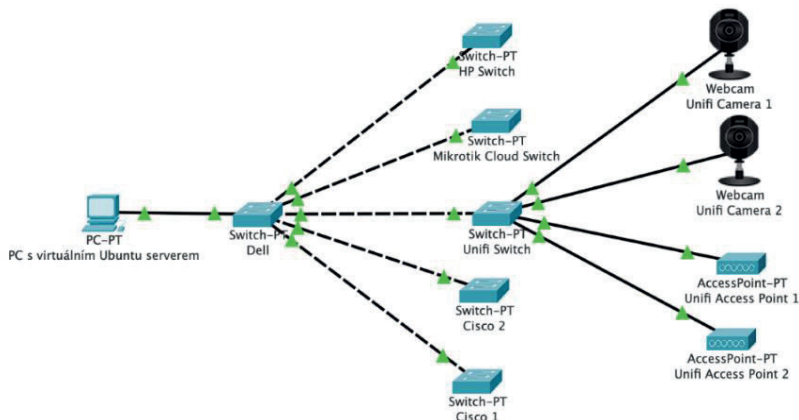
4.5 Implementations and Demonstrations

In the lab we used multiple devices. All of them are listed below.

Devices	Company
6 switches	Cisco, Hewlett-Packard (HP), Dell, MikroTik, UniFi, NetGear
2 AP (access point)	UniFi
2 cameras	UniFi
5 routers	DrayTek, Cisco, MikroTik

Table 3

First thing to do was connecting all of the devices together, and configuring all the necessary parameters. Shown below is a lab figure created in Cisco Packet Tracer to demonstrate connectivity of all the used devices during the first meeting in Laubusch (Figure 1). Next step was configuration of Ubuntu server running on a virtual machine. We had implemented two NIC (Network interface card/controller) cards one for internal network communication and second for internet connectivity.



Since routers were the next devices to be added, during the second meeting in Laubusch our team enhanced the lab to L3. Again, the figure below shows L3 connectivity, made in Cisco Packet Tracer (Figure 2) with the addition of five routers and two Aps.

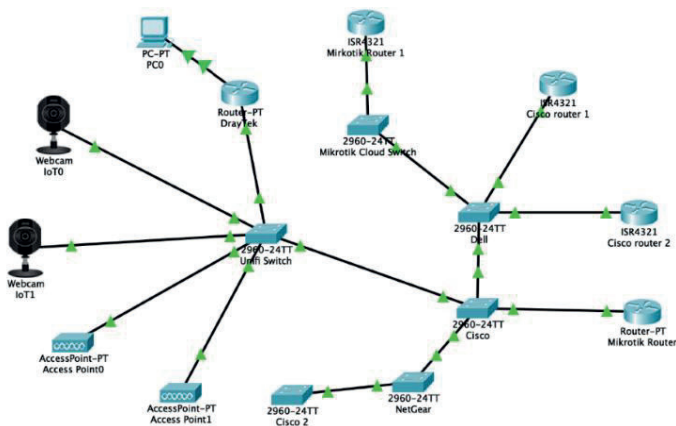


Figure 2



Implementation of software Reliance in steps:

1. Installation of Reliance 4 modules Design and Servery with all the needed runtime modules for remote visualization of the project.
2. Implementation of SNMP communication drivers for remote communication with network components.
3. Creating objects of individual network components by which we wanted to connect. Configuring all the compulsory parameters, e.g. IP address, port, SNMP protocol version, Community string and password (Figure 3).
4. A problem occurred when we were trying to switch between SNMP protocol versions. After contacted the protocol support we were told that Reliance only supports version 1 SNMP, but the problem was that not all the newer devices supported the first version due to its poor security. Besides that, variables in the first version are not named as string but rather as OID (e.g. 1.3.6.1.4.1.6574.2.1.1.6.0).
5. Due to upper mentioned issue we had to find needed OID tables of all the variables, where we chose compulsory components again.
6. Creating a template of data structure and merged it with the components template (Figure 4).
7. Every network object was given a data structure according to the template. We only added the OID addresses of every variable used (Figure 5). Thereby automatically a window was created showing variables in an unpleasant form for testing the functionality of program (Figure 6).

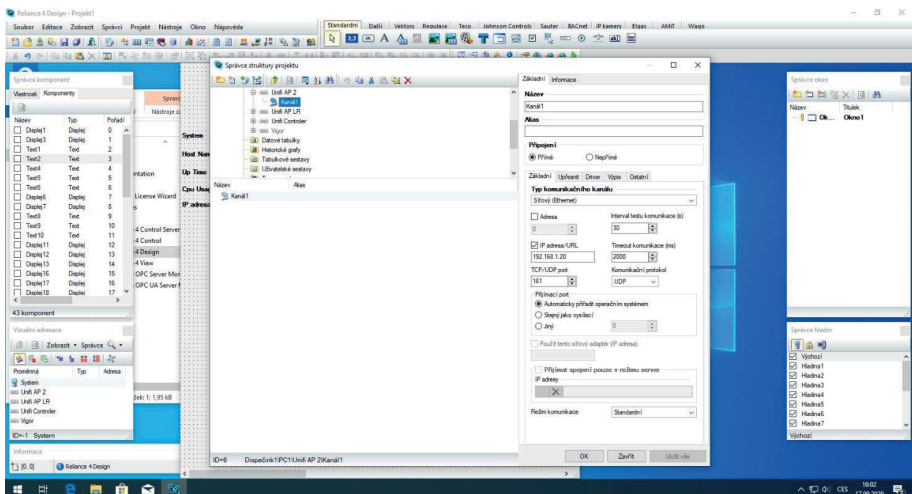


Figure 3

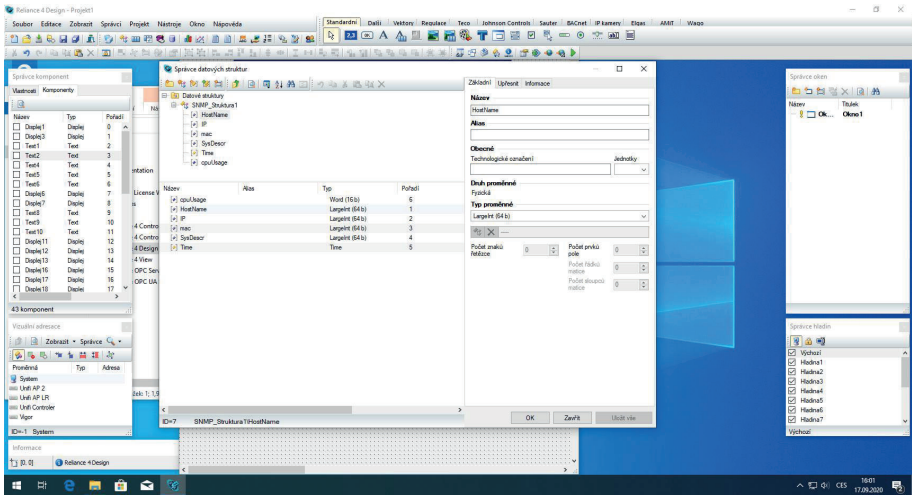


Figure 4

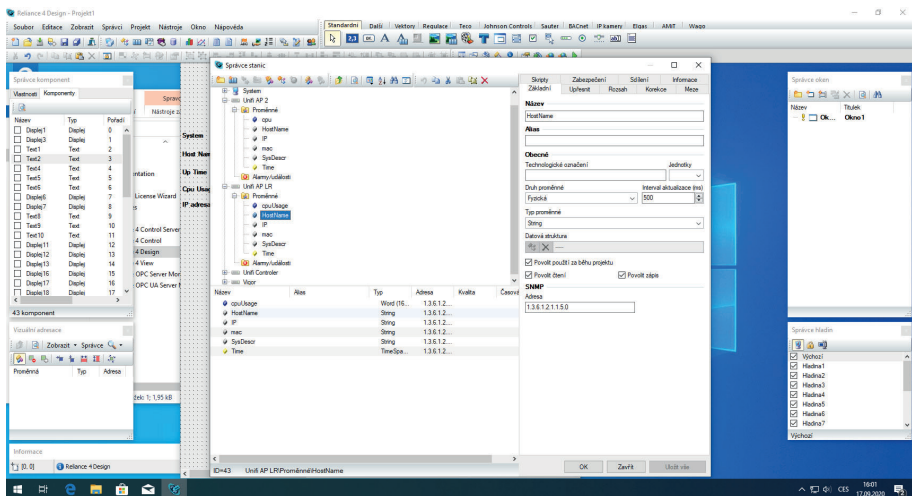


Figure 5

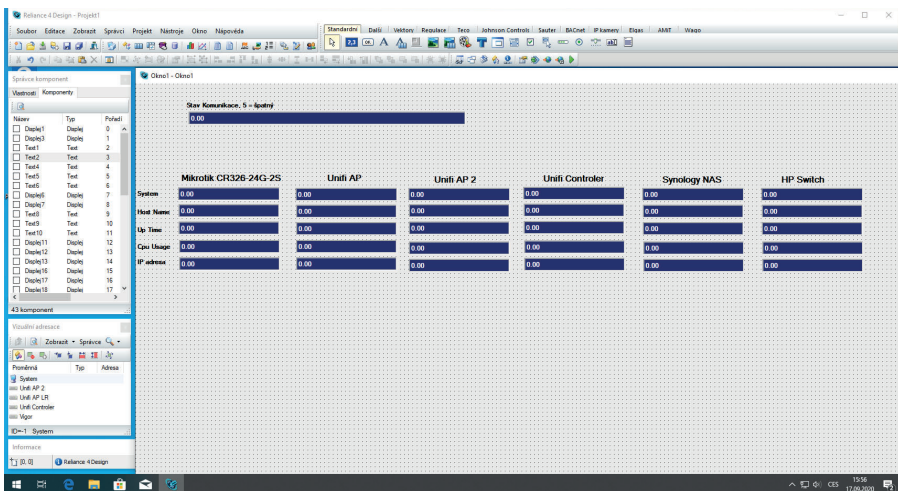


Figure 6

All the 7 steps of Reliance and L3 were done during the second meeting. The last meeting unfortunately wasn't held in Laubusch due to covid pandemic and time load on the UJEP's side. Beneath are the steps we would have carried on with or will be carried on the next year.

8. Creation of SQL database and setting data logging for every component in the created database.
9. Linking graphs to the database which would show process of individual variables in time.
10. Creating alarms to warn about the unwanted situation in our testing network.
11. Wrapping all above mentioned into a well-tuned UI, and that would be executable remotely via thin client using .jar (java application) or using a web application.



4.6 Results

In conclusion implementation of the software was not completed due to already mentioned reasons. Other than that, the project was very useful for students as they obtained many skills such as management skills. Students also got the experience of using Open Systems Interconnection model in practice not only at university. Three of our freshmens got the opportunity to work and learn from seniors and with Chemnitz students, which can prove helpful for their coming years in university and after.

4.7 References

- [1] POPESCU-ZELETIN, R. 1983. Implementing the ISO-OSI reference model [online]. Berlin, [retrieved 2020-09-10]. Retrieved from: http://www.bitsavers.org/pdf/datapro/communications_standards/2783_ISO_OSI.pdf. [2] JASUD, V. P. 2017. The OSI Model: Overview on the Seven Layers of Computer Networks [online]. Nagpur, [retrieved 2020-09-12]. Retrieved from: https://www.academia.edu/35484011/The_OSI_Model_Overview_on_the_Seven_Layers_of_Computer_Networks.



5. Automotive II

Shadi Saleh

5.1 Abstract

The automotive industry is currently experiencing a major paradigm shift from conventional, human-driven to autonomous vehicles driven by artificial intelligence. Automated vehicles offer a safe, reliable, efficient and cost-effective solution that will dramatically redefine the future of human mobility. Deep learning techniques have proven very successful in overcoming perception problems such as objects/obstacles classification, recognition, and segmentation. On the other hand, deep learning models usually require specialized hardware due to higher memory and computing demands. The solution is either more efficient hardware or less complex deep neural networks. A balance between these two aspects is crucial to achieve an optimal result. Therefore, the real question is: Which type of edge hardware and which type of network should we bring together to maximize the accuracy and speed of deep learning algorithms?

5.2 Motivation

The enrolled students came from the two domains: Automotive Software Engineering and Embedded Systems. In both domains, students have a background in theory and practice of computer science, linear algebra, mathematics and programming languages (Python/C++). They also have the fundamental knowledge of artificially intelligent and deep learning techniques. In this section, different topics are offered in the context of a deep learning approach which give the students an overview and comprehensive understanding of state-of-the-art engineering practices used in the self-driving car industry.

Students will learn how to develop deep learning models and how to optimize the performance for self-driving car perception applications such as multi-object detection and distance estimation, optical flow estimation, and vehicle detection and tracking. They will be able to develop and train deep learning models running on an embedded GPU (graphics processing Unit) to identify objects in offline video/ live stream cameras, such as roads, pedestrians, other vehicles, and vehicle speed. Students deployed the deep neural network on Nvidia Jetson Nano board platform. At the end of the section, students will have the skills necessary to create AI applications for a variety of autonomous driving scenarios, including traffic navigation, obstacle avoidance, and robotic.

5.3 Introduction

In this section, we focused on the Convolutional Neural Network (CNN) [1], which is a type of deep learning, and it is an approach for learning classification and feature extraction from training samples. A Convolutional Neural Network (CNN, or ConvNet) is a particular type of multi-layer neural network that is designed to recognize visual patterns directly from pixel images with minimal preprocessing. CNN's ability to identify patterns, but since this pattern recognition is done as part of algorithms running on neural networks, a major concern is that the system is a "black box". Once the system is trained, data can be fed to it and a useful interpretation of this data will come out.

But the actual decision-making process that goes on between the input and output stages is not necessarily something that a human being can intuitively understand.



5.4 Research questions and Methods

Various deep learning-based image recognition techniques such as Faster R-CNN [2], YOLO [3], SSD [4] and Optical Flow [5] are being studied and developed based on OpenCV[8], TensorFlow [6] and Keras [7] libraries and frameworks in the context of the topics offered. Moreover, understanding the dataset as it contains the images for training. Developing a CNN Architecture and using other deep learning libraries to estimate optical flow, object detection, vehicle recognition and tracking.

- **Collision Warning Based on Multi-Object Detection and Distance Estimation.**
- **Optical Flow Estimation Based Deep Learning Approaches.**
- **Vehicle Recognition and Tracking Using Front Face Monocular Camera.**
- **Vehicle detection and tracking with a front face monocular camera.**

5.5 Implementations and Demonstrations

The students reviewed the appropriate literature and provided a good overview of the state of the art on each topic. They then implemented their models for in-depth learning with cloud servers or stand-alone GPUs. In the evaluation phase, they tested their CNN model on NVIDIA Jetson Nano. For example, Chirag Rellan used SSD MobileNet v2 architecture for training on PC accelerated using NVIDIA GeForce GTX 1060 with Max-Q design, then a live Camera Demo of object detection implemented on Jetson Nano. while Shubham Pareshkumar Surana trained his object detection model on system GPU Nvidia Gtx1660ti in the darknet framework. After that, he converted his model to TensorFlow format to run this model for object detection with the distance estimation part on embedded GPU Nvidia Jetson Nano. on the another hand, another student such as Shehbaaz Shaik implemented his model using Google Colaboratory on a cloud service to predict the vehicle velocity, and his method was evaluated on a Comma.ai dataset.



5.6 Results

The following figures show the implementation results of our project:

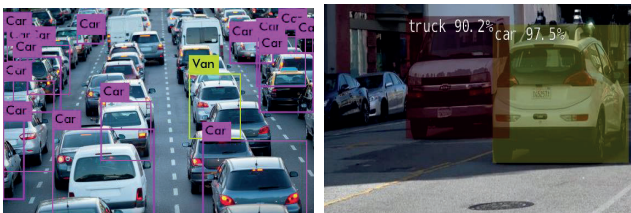
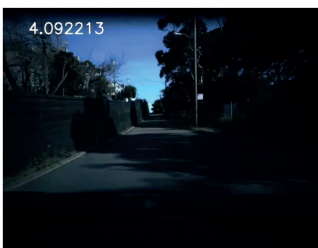


Fig1: output from the objects detection model implemented on Jetson Nano



a)



b)

Fig.2. Prediction of Vehicle Velocity with Optical Flow, b) Prediction of Vehicle Velocity without Optical



Fig.3.Vehicle Recognition and Tracking from Front Face Monocular Camera



5.7 Conclusion

In this part of the IKON project, the students are able to develop some functions related to the self-propelled car, driven entirely by deep learning, and they also gain good skills in neural networks that they can apply to any project or company in the future. They learned how to employ computer vision techniques through OpenCV to process many offline video/live-stream camera and identify the region of interest. They learned how to train convolutional neural networks, such as multiple object recognition in real time as they work on Jetson Nano board. This section is designed to guide students with fundamental programming/mathematics experience to accomplished deep learning developers.

5.8 References

- [1] Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 9, 611–629 (2018). <https://doi.org/10.1007/s13244-018-0639-9>
- [2] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.
- [3] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
- [4] Liu W. et al. (2016) SSD: Single Shot MultiBox Detector. In: Leibe B., Matas J., Sebe N., Welling M. (eds) *Computer Vision – ECCV 2016*. ECCV 2016. Lecture Notes in Computer Science, vol 9905. Springer, Cham.
- [5] A. Dosovitskiy et al., "FlowNet: Learning Optical Flow with Convolutional Networks," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 2758-2766, doi: 10.1109/ICCV.2015.316.
- [6] Abadi, Martín et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems." *ArXiv abs/1603.04467* (2016).
- [7] Gulli A, Pal S. *Deep learning with Keras*. Packt Publishing Ltd; 2017.
- [8] OpenCV. *Open Source Computer Vision Library*. 2015.



6. Deployment of Self-checking and Self-diagnosis in Complex Systems

doc. RNDr. Viktor Maškov, DrSc.

6.1 Abstract

Self-checking and self-diagnosis can be considered as an effective method of complex system checking and diagnosis in the situations when using external diagnosis facilities is difficult or even impossible. Students conducted research on the subject of “Deployment of self-checking and self-diagnosis in complex systems”. This task was divided on several sub-tasks and students worked on their own sub-task. These sub-tasks include solving such problems as organization test execution, using mathematic and real simulation for assessing suggested organization, development of system self-diagnosis algorithm, evaluation of efficiency of the suggested algorithm, recommendations for deployment of the suggested organization in real complex systems. The obtained results show the applicability and efficiency of the suggested method of complex system diagnosis.

6.2 Motivation

Student background

There are seven students registered to complete the tasks of Block 3. They are all Bachelor students of University J.E. Purkyně (UJEP). They are all from faculty of Science. The gender ratio can be considered as balanced with 5 men and 2 women. Five of the students are enrolled in fourth semester whereas the two other students are enrolled in sixth semester (Table 1).

Table 1: Students’ demographic details

Student	Gender	Faculty	Field of study	Current semester
1	male	Faculty of Science (FS)	Information systems (IS)	4
2	male	FS	IS	4
3	male	FS	IS	4
4	male	FS	IS	4
5	male	FS	IS	4
6	female	FS	IS	6
7	female	FS	IS	6

Seven students registered for this block have chosen 4 of the described topics. These topics are:

1. Developing algorithm for tests execution.
2. Modeling and simulation of tests execution.
3. Mathematical modeling of tests execution by using programming languages.
4. Developing diagnosis algorithms.

Results obtained by solving these problems can be combined and should be considered as a theoretical background for performing real simulation of self-diagnosis in a complex system represented by Jetson Nano units.



6.3 Introduction

Nowadays complex systems are applied nearly in all spheres of human activities, such as production (manufacturing); transportation; communication, etc. For example, in transportation, airplanes, trains and even private cars are equipped with and controlled by complex systems. Malfunction of these complex systems may directly impact on safety and may cause serious consequences on human life and on environment. In view of this, checking of complex systems state (i.e., error detection) plays very important role during system's life span.

Error detection can be performed either before or after normal operation or during system normal operation (so-called, concurrent error detection). Error detection can be performed by using different methods. One option consists in that system units can be tested by external tester (or diagnoser). Another option consists in using self-testing. This kind of testing can be conducted either by a special system unit which will perform diagnostic tasks only, or by system units themselves (i.e., by using mutual testing) [1], [2], [3], [4]. Several requirements must be met in order to conduct mutual testing in complex systems [5], [6]. First of all, every unit must be capable of performing a test on any other unit. Any two units must have a link through which a test could be executed. Moreover, every unit should be able to perform the role of diagnoser (i.e., to perform a diagnosis algorithm and to signal/messaging a diagnosis result to the system environment).

Mutual testing has several advantages, e.g., ability to perform testing for a long period of autonomous work of system; ability to provide fault-tolerance of testing; ability to perform diagnosis correctly even when separate system units have low reliability.

In order to realize these advantages of mutual testing, it is necessary to determine its optimal parameters. This can be done with the help of appropriate modeling and simulation of mutual testing in complex systems [7], [8], [9].

6.4 Research questions and Methods

To involve students in research work smoothly and make their work more concrete, the whole problem was divided on several tasks. It is planned to integrate the obtained result and form the single method of how to implement the suggested method of diagnosis in particular complex system.

Task 1: Assessment of the suggested organization of test execution by using mathematic simulation (with the help of tool SHARPE).

For this, at the beginning, the states of the system are presented in the form of Petri Net. Then by using some tool (e.g., tool SHARPE) it is possible to simulate the test execution in the system. The main goal of such simulation is to reach to the system state when all tests needed for diagnosis have been executed. Usually, the event that all needed tests are executed, can be reflected in Petri Net (which corresponds to system state transition) via token in special place denoted as P0. Petri Net allows to simulate the test execution for different system parameters, such as:

- different values of rate of system transfer from one state to another;
- different values of testing time (i.e., different duration of each particular test);
- different initial system states.

Task 2: Assessment of the suggested organization of test execution by using mathematic simulation (with the help of computer program written in appropriate programming language).

Another possibility of how to perform mathematic simulation of test execution consists in using computer programs written in any modern programming language. The essence of such approach consists in the following: as an input is used Petri Net with initial marking. Transfer of the system from one state to another is reflected in Petri Net as changing of marking. Therefore, after certain period of time (so-called, clock) the



new system state or new marking of Petri Net is determined. We are interested in the system state which corresponds to the situation when all needed tests are executed (or the marking of Petri Net with token in the targeted place P0 is reached). As soon as place P0 has a token, the algorithm is terminated. Output value of the algorithm gives the information of how long testing procedure lasts (i.e., information about duration of testing procedure).

Task 3: Developing diagnosis algorithm and evaluation of its efficiency.

Before providing self-diagnosis, it is necessary to perform several steps, such as:

- choose the diagnosis strategy (i.e., unique, sequential, excess or probabilistic diagnosis);
- make decision about allowable faulty situations;
- decide on the method of how to interpret the test results.

Depending on the made decision on each step, different diagnosis algorithms can be developed. Currently, the following diagnosis algorithms are widely used for diagnosis of complex systems:

- algorithms for unique diagnosis;
- probabilistic algorithms;
- algorithms for diagnosis of intermittently faulty units.

Students focused on algorithms for unique diagnosis. Particularly, they developed the algorithm based on the table of potential syndromes and evaluated their efficiency.

Task 4: Developing algorithms for tests execution and performing real simulation of diagnosis procedure.

To perform a test each system unit should have additional data and algorithms related to system diagnosis. Word “additional” means that these data are redundant in relation to the main tasks performed by the system units. Test can be complex or very simple. The algorithms are needed to send a test and read the reply from the tested unit, process the received data, form the test result and output the result of diagnosis. It is necessary to decide about the time when the tests should be executed. In the chosen diagnosis, it is assumed that the tests are executed in the periods when units are free from performing their assigned functions. Thus, the time when a test will be send is random value that can follow exponential or any other distribution. Test reply can also be complex or simple. Reply can be sent when a tested unit is free (i.e., not perform the assigned functions according to its specification). For the chosen organization of tests execution each system unit should have algorithms of how to process the received data (i.e., input values related to test or test reply). Usually, system self-diagnosis uses special structures of testing graph (e.g., structures with only one edge between any two vertices). In view of this, the algorithm should ensure the following restrictions:

- only one test between any two system units can be executed (i.e., without replication);
- do not process the reply from the tested unit which has already sent his reply to the previous test;
- terminate the processing input values when pre-defined condition is met.

Solving these tasks allow to adjust the suggested organization of system diagnosis to self-diagnosis of each particular complex system so as to achieve the desired efficiency.

6.5 Task solving and Results

Task 1: Assessment of the suggested organization of test execution by using mathematic simulation (with the help of tool SHARPE).

Since mutual testing has many parameters which are probabilistically defined, it was used Stochastic Timed Petri Nets (STPN) for modeling and simulation of mutual testing.

Modeling of mutual testing is presented and explained with the help of simple examples.

In the very simple case, mutual testing is performed in the system which has only two units (see Fig. 1).

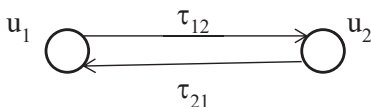


Figure 1: Mutual testing in the system consisting of only two units

In the given case, Petri Net should be able to reflect and model the following goal and properties of mutual testing:

- the goal of mutual testing is the execution of only one test. It doesn't matter which test is executed, either τ_{12} or τ_{21} ;
- units do not have priority in performing the test (i.e., choosing the testing and tested units is random procedure);
- no restrictions on test execution are imposed.

In Fig. 2, the Petri Net presentation of such mutual testing is shown.

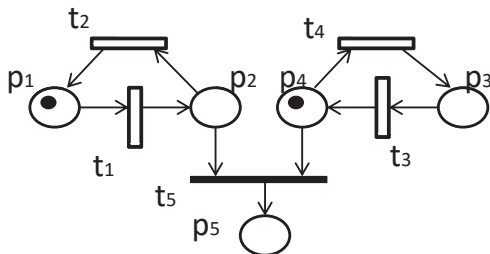


Figure 2: Petri Net modeling of mutual testing in the system with only two units

In Fig. 2, places p_1 and p_3 are associated with engaged states of units u_1 and u_2 , respectively. Places p_2 and p_4 , in turn, are associated with idle periods of units u_1 and u_2 . For the given example, several initial markings are possible (Table 2).

Table 2: Possible markings and their interpretations

Initial markings, M_0	Meanings
1 0 1 0 0	both units u_1 and u_2 are engaged
1 0 0 1 0	unit u_1 is engaged, and unit u_2 is idle
0 1 0 1 0	both units u_1 and u_2 are idle
0 1 1 0 0	unit u_1 is idle, and unit u_2 is engaged



Let, for example, initial marking be (10010). It means that, at the beginning, unit u_1 is engaged with working tasks, whereas unit u_2 is idle and is ready to participate in testing procedure. After some time unit u_1 will transfer to idle state and, thus, will be ready to test or to be tested. It depends on the state of unit u_2 which also can change its state from idle to engaged state. If this transition has occurred, no tests will be enabled. Otherwise, both units participate in testing procedure and test will be executed.

Now, we proceed with a slightly modified example. We change the goal of mutual testing and introduce the priorities for the order of execution of tests. In the given case, Petri Net should be able to model the following goal and properties of mutual testing:

- the goal of mutual testing is the execution of two tests (both test τ_{12} and test τ_{21}), so that both units must be tested;
- test τ_{12} has priority over test τ_{21} ;
- both units change their states after execution of test τ_{12} ;
- only one test τ_{12} must be executed. It also concerns test τ_{21} ;

It is assumed that at the beginning unit u_1 is in engaged state, whereas unit u_2 is idle. For this case the Petri Net is shown in **Figure 3**.

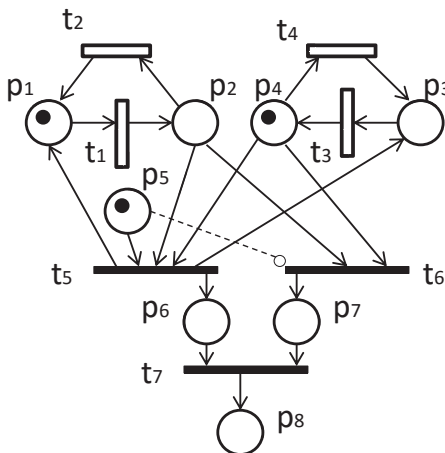


Figure 3: Petri Net for the case when both units must be tested

With the increasing number of units, the task of modeling of the priorities and restrictions on tests executions becomes more complex.

In the next example, system consists of three units (see Fig. 4). Tests have priorities (first τ_{12} , then τ_{23} and then τ_{31}). Each test τ_{12} , τ_{23} and τ_{31} must be executed only once. All units change their states immediately after execution of test.

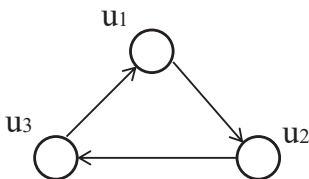


Figure 4: Testing graph for system with three units

It is assumed that initially all three units are in engaged state. For this case, the Petri Net is shown in Figure 5.

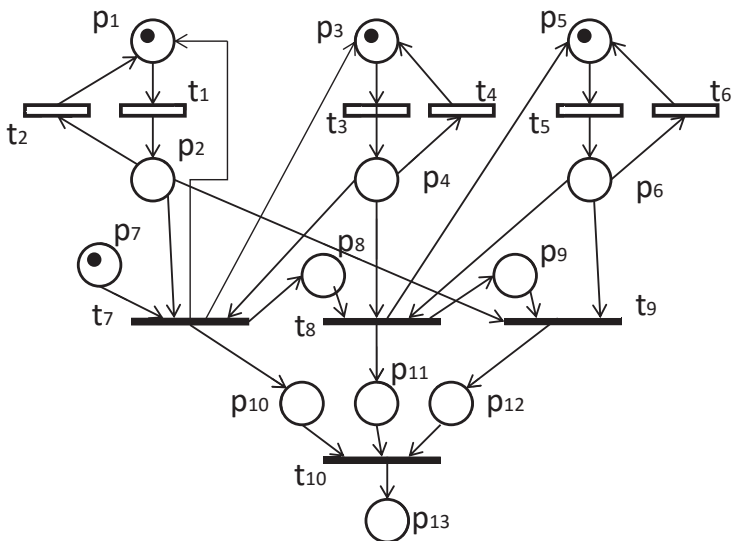


Figure 5: Petri Net which models the mutual testing in the system with three units

Places p1, p2 and p3, p4 and p5, p6 model engaged and idle periods of units u1, u2 and u3, respectively. Place p13 models the situation when all system units have been tested.

For providing simulation of mutual testing, the tool SHARPE was chosen.

All Petri Nets considered in the presented examples can be plotted directly in the project window of SHARPE (see Fig. 6). Petri Net considered in the first example (see Fig. 2) was processed by SHARPE under different input data. Particularly, we set initial marking M_0 and rates r_1, r_2, r_3 and r_4 that correspond to the transitions t_1, t_2, t_3 and t_4 . As a result of simulation, probability of the event that place P_5 has a token at time t , was determined. This probability corresponds to the probability of achieving the goal of mutual testing.

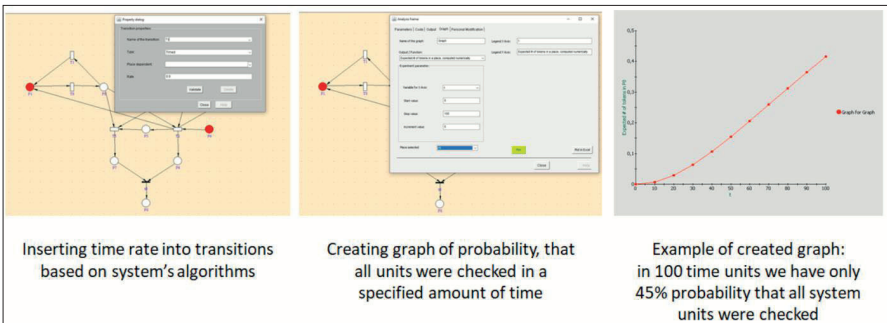


Figure 6: SHARPE usage

Simulation has shown that initial marking (except marking $M_0(01010)$) influences the sought probability insignificantly (has insignificant influence on the sought probability). Initial marking $M_0(01010)$ leads immediately to reaching the goal of mutual testing since both units immediately perform the required test. SHARPE enables input of rates r_1, r_2, r_3 and r_4 in conditional units. It means that one can select any values from the real world. For example, rates r_1 and r_3 can be set to "1", whereas rates r_2 and r_4 can be set to the values 2, 10, 30, 50 and 80, respectively. Thus, rates r_2 and r_4 are 2, 10, 30, 50 and 80 times greater than rates r_1 and r_3 . For each particular complex system, one can set the real values of rates r_1 and r_3 (for example, for many-core processor these rates could be set to $1,5 \div 2,5 \times 10^6$ 1/s).

Task 2: Assessment of the suggested organization of test execution by using mathematic simulation (with the help of computer program written in appropriate programming language).

At the beginning, the simple case when complex system consists of two units was considered. The goal of mutual testing is performing one test, either test τ_{12} or test τ_{21} (see Fig. 7).

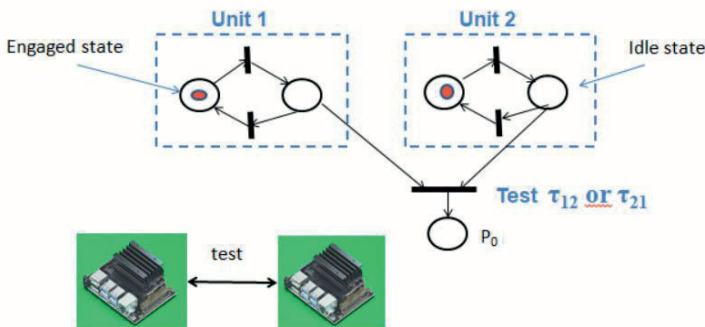


Figure 7: Mutual testing in complex system consisting of two units

Simulation of such mutual testing is performed by using Petri Net (see Fig.7) and can be described by computer program (algorithm). In Figure 8, the block diagram of the algorithm is shown.

The main part of this algorithm is Step 1: marking determining. Determining the new marking requires consideration of several issues, such as, for example, time of firing of transition (see Fig. 9). Timed transition t_1 doesn't fire immediately. It takes some time to fire denoted as t_A . As a rule clock τ (i.e., time increment) is lesser than mean value of t_A . It means that place P_2 will have a token after several Steps k (i.e., $k=t_A/\tau$). Time t_A is random value which can have exponential distribution (see Fig. 10) or other distribution.

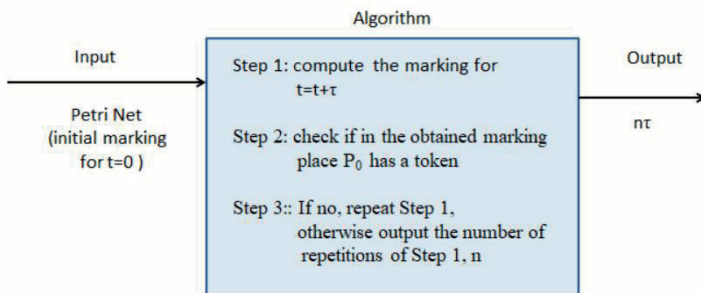


Figure 8: Block diagram of the algorithm

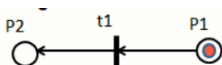


Figure 9: Firing of timed transition t_1

In **Figure 10**, the functional dependence between probability of firing of transition t_1 and time t_A is shown.

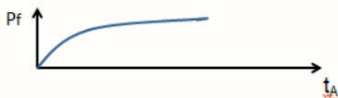


Figure 10: Probability of firing of timed transition, P_f

For each value t_A probability P_f can be determined. Knowing P_f , it is possible to perform simulation of mutual testing by using the function “rand”. Function “rand” returns the result which allows to make decision if transition t_1 has fired or not. Similarly, all timed transitions in Petri Net which models the mutual testing in the complex system can be examined. Follow this way the new markings can be determined. In order to perform such simulation of mutual testing it is important to choose the correct value of clock τ . For the case when random value t_A has exponential distribution with mean value $t_{mean}=1$, there was determined the average time of transition firing depending on the value of clock (Table 3).

Table 3: Impact of clock on the simulation of transition firing

$k=t_{mean}/\tau$	Average time
1	2



2	1
4	0.5
6	0.5
10	0.5

As it follows from the table the value of clock should be lesser than or equal to 0.25.

Next problem that should be considered is assessment of impact of initial marking on the time of mutual testing. For the example presented in Fig. 2 the following results (see Fig. 11) were obtained.

In this figure, denotation $P[i]=1, i=1,..,m$, means that place P_i has a token. If $P[i]=0$, place P_i doesn't have a token. When t_1 approximately equals to t_3 the average time of mutual testing can be very long.

In the given case, some modifications of the Petri net should be made. It is suggested to add new places to the Petri net (see Fig. 12).

For the real complex system it means that in each system unit the additional process or thread should be introduced. This process is responsible for storing the data received from another system unit. Now, unit after transition from engaged to idle state can immediately process these data.

Initial marking	k	t _{mean}	Average time
P[0]=0	4	1	0.5
P[1]=0	4	10	2.5
P[2]=1	4	20	5
P[3]=0			
P[4]=1			

Initial marking	k	t _{mean}	Average time (when timed transitions $t_1, t_2 \gg t_3, t_4$)
P[0]=0	4	1	2
P[1]=1	4	10	15
P[2]=1	4	20	35
P[3]=0			
P[4]=1			

Figure 11: Results of simulation of mutual testing for different initial markings

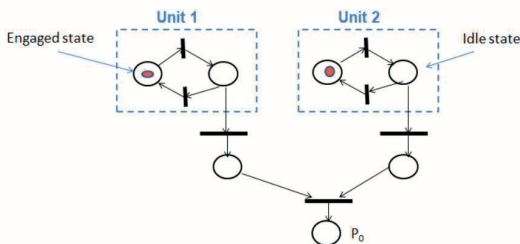


Figure 12: Modified Petri net



Task 3: Developing diagnosis algorithm and evaluation of its efficiency.

There was chosen the approach based on unique diagnosis. Particularly, the approach, which exploits the table algorithms with parameter t . Value of t corresponds the maximum possible number of faulty units allowed in the system. In order to make decision about efficiency of the developed algorithm and about its possibility to apply in complex systems, the simulation of diagnosis procedure should be performed. For this the web application can be developed. The main parts of this application are as follows:

- syndrome generation;
- diagnosis algorithm;
- data recording;
- evaluation of efficiency of diagnosis.

Syndrome generation exploits the table of potential syndromes and, thus, it accepts the assumptions of this table. Particularly, the assumption about maximum number of faulty units in the system.

Syndrome generation can be performed:

- with or without accounting the unit reliability;
- taking into account the assumptions relating to test results (mainly it concerns the results produced by faulty units).

Essentials of the algorithm are as follows:

- table of potential syndromes is formed before the testing procedure begins;
- table contains all possible syndromes that can be obtained under allowable faulty situations.

Number of allowable faulty situations Q is restricted and depends on the value t .

$$Q = \sum_{i=1}^t \binom{n}{i}$$

Where t is diagnosability measure which reflexes capability and quality of diagnosis of the system given particular testing assignment.

For example, for the system with testing graph shown in Fig. 13 the table of potential syndromes is shown in Fig. 14.

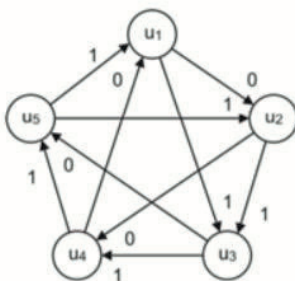


Figure 13: Testing graph for the system with five units

Diagnosis consists in comparing syndrome obtained after tests execution with the syndromes of the table of potential syndromes.

The main part of the diagnosis algorithm is to carry out the comparisons of obtained syndromes (real or generated) with the syndromes of the table of potential syndromes. Such comparison can be performed in



different ways. Each way can be evaluated by the number of required elementary comparisons, which influences the total complexity of the diagnosis algorithm. At the beginning, there was chosen the method based on the comparisons by columns. Although, other methods of comparisons can also be considered and, thus, the best method of comparisons can be determined. Example of comparing by columns is explained in the slide (see Fig. 14).

Our generated syndrome is "0001101100"

(Firstly we are comparing with the first 5 rows - just 1 faulty unit)

- 1) First element of generated syndrome = 0 comparing with r_{12} → Selection of matching syndromes (S_1, S_2, S_4, S_5)
- 1) 2nd element = 0, comparing with r_{13} → Matching syndromes are S_1, S_4, S_5
- 1) 3rd element = 0, comparing with r_{23} → Matching syndromes are S_1, S_4, S_5
- 1) 4th element = 1, comparing with r_{23} → Matching syndrome is S_4
- 1) Comparing the rest of the elements of generated syndrome with the elements of the S_4 row

S_i	faulty units	potential syndrome	test results										
			r_{12}	r_{13}	r_{23}	r_{24}	r_{34}	r_{35}	r_{45}	r_{41}	r_{51}	r_{52}	
S_1	s_{11}	R_1^1	X	X	0	0	0	0	0	0	1	1	0
S_2	s_{21}	R_2^1	1	0	X	X	0	0	0	0	0	0	1
S_3	s_{31}	R_3^1	0	1	1	0	X	X	0	0	0	0	0
S_4	s_{41}	R_4^1	0	0	0	1	1	0	X	X	0	0	0
S_5	s_{51}	R_5^1	0	0	0	0	0	1	1	0	X	X	X
S_6	s_{61}, s_{62}	R_6^1	X	X	X	X	0	0	0	1	1	1	1
S_7	s_{71}, s_{73}	R_7^1	X	X	1	0	X	X	0	1	1	0	0
S_8	s_{81}, s_{84}	R_8^1	X	X	0	1	1	0	X	X	1	0	0
S_9	s_{91}, s_{95}	R_9^1	X	X	0	0	0	1	1	1	X	X	X
S_{10}	s_{10}, s_{13}	R_{10}^1	1	1	X	X	X	0	0	0	1	1	1
S_{11}	s_{11}, s_{14}	R_{11}^1	1	0	X	X	1	0	X	X	0	1	1
S_{12}	s_{12}, s_{15}	R_{12}^1	1	0	X	X	0	1	1	0	X	X	X
S_{13}	s_{13}, s_{14}	R_{13}^1	0	1	1	1	X	X	X	X	0	0	0
S_{14}	s_{14}, s_{15}	R_{14}^1	0	1	1	0	X	X	1	0	X	X	X
S_{15}	s_{14}, s_{15}	R_{15}^1	0	0	0	1	1	1	X	X	X	X	X

Figure 14: Slide from student's presentation

The developed web application allows evaluating the efficiency of the developed algorithm by using recorded data, such as:

- number of comparisons for each generated syndrome;
- total number of generated syndromes;
- average number of elementary comparisons, etc.

Evaluation of the efficiency of the suggested diagnosis algorithm should be performed for different system parameters and characteristics (e.g., number of system units, different interpretation of test results, unit reliability, etc.)

Task 4: Developing algorithms for tests execution and performing real simulation of diagnosis procedure.

The next step of the research was to perform real simulation by using five Jetson Nano units (see Fig. 15).

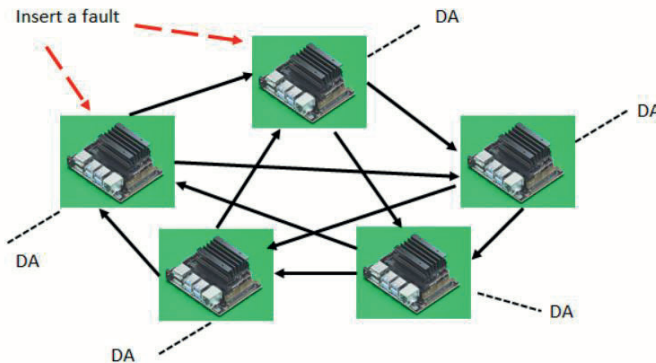


Figure 15: Using Jetson Nano for simulation of mutual testing

It is possible to insert one or two faults in the system units. Each unit (Jetson Nano) should be equipped with diagnosis algorithm (DA). It means that each unit can execute DA and can deliver the result of diagnosis to the system environment. The goal of real simulation is to evaluate the quality of diagnosis algorithm and the quality of the whole mutual testing. Real simulation allows determining the time needed to perform system self-diagnosis. Based on the diagnosis time, it is possible to estimate the types of complex systems in which the suggested self-diagnosis can be implemented. If needed, the detailed recommendations of how to improve the diagnosis procedure can be given.

As the first step towards developing algorithms for providing execution of tests among system units, the algorithm for performing communications among system units should be developed.

To solve this task there were used the following software (see Fig. 16 and Fig. 17). For some reasons at the first stage, the Raspberry pi was exploited. Further it is planned to provide simulation by using Jetson Nano.

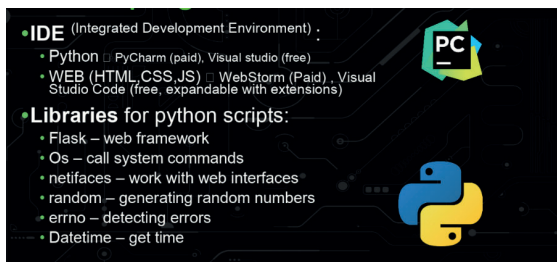


Figure 16: Developing SW

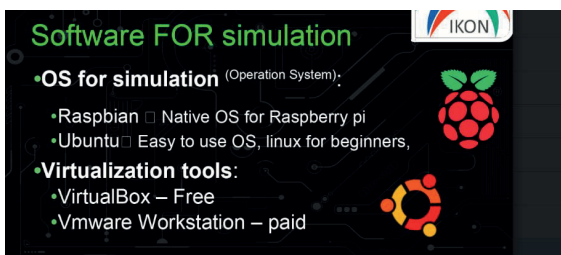


Figure 17: SW for simulation

6.6 Conclusion

Student research deals with the tasks of complex system self-checking and self-diagnosis. Such self-checking and self-diagnosis can be performed both when system carries out its main functions (according to specification) and when system is idle. One of the main advantages of such self-checking and self-diagnosis is that these procedures can be performed without external facilities. In the given case, system can work for a long time autonomously in the harmful environment when it is difficult to perform system diagnosis by using external sources. Examples of complex systems in which using of the suggested diagnosis is efficient and gives good results are as follows: sensor networks; many-core processors; multi-agent systems; smart matter systems; many-robot systems; etc. Application of the suggested system diagnosis in complex systems requires solving several problems and tasks. Students solved these tasks by using modeling and simulation. For this purpose they used Petri Nets, mathematic simulation, computer simulation (with the help of web application) and made first step towards real simulation. Results obtained by students will allow to make concrete recommendations about deployment of mutual testing in the particular complex system. Moreover, these results will help to assess the efficiency of exploiting mutual testing in particular complex system.

6.7 References

- [1] V. Gostev, V. Mashkov, O. Mashkov, „Self Diagnosis of Modular Systems in Random Performance of Elementary Tests“, Cybernetics and Computing Technology (Discrete Control System). No, 105. Allerton Press, pp. 104-111, 1995.
- [2] V. Mashkov, O. Barabash, „Self-checking and self diagnosis of modular systems on the principle of walking diagnostic kernel“, Engineering simulation. Vol. 15, pp. 43-51, 1998.
- [3] A. Friedman, „A new measure of digital systems diagnosis“, In Proc. Int. Symp. Faulttolerant computing. Paris (France), pp. 167-170, 1975.
- [4] V. Mashkov, „Selected problems of system level self-diagnosis“, Lviv: Ukrainian Academic Press., ISBN 978-966-322-365-0, 184 pages, 2011.
- [5] F. Preparata, G. Metzger, R. Chien, „On the Connection Assignment Problem at Diagnosable System“, IEEE Transactions on Electronic Computers. vol. EC-16, No. 12, pp. 848-854, 1967.
- [6] T. Barsi, T. Grandoni, P. Maestrani, „A Theory of Diagnosability of Digital Systems“, In IEEE Trans. on Comp. Vol. C-25, No. 6, pp. 585-593, 1976.
- [7] V. Mashkov, J. Barilla, P. Simr, „Applying Petri Nets to Modeling of Many-Core Processor Self-Testing when Tests are Performed Randomly“, Springer, Journal of Testing, 2013, DOI: 10.1007/s10836-012-5346-8.
- [8] J. Fiser, V. Mashkov, V. Lytvynenko, „Representation of system level self-diagnosis in Python programming language“, Int. Journal „Electrotechnic and Computer Systems“, No. 17(93), pp.48-54, 2015.



**Europäische Union. Europäischer
Fonds für regionale Entwicklung.
Evropská unie. Evropský fond pro
regionální rozvoj.**



Ahoj sousede. Hallo Nachbar.
Interreg V A / 2014–2020

IKON - IT Kompetenz
Netzwerk zur Stärkung der
Grenzregion Sachsen -
Tschechien

[9] V. Mashkov, P. Simr, J. Barilla. Chapter 5 „Applying Petri Nets to Modeling of Complex Systems Diagnosis“ in book „Advances in Engineering Research“ Ed. V.M. Petrova. NOVA science publishers Vol.13, pp.81-120, 2016. (URL: <https://www.novapublishers.com>)

