Concept of the comprehension level of situation awareness using an expert system

J Nine, S Manoharan and W Hardt
Department of Computer Engineering, Technische Universität Chemnitz, Chemnitz, Germany
E-mail:julkar.nine@informatik.tu-chemnitz.de

Abstract. One of the major hurdles in high level autonomous driving is the integration of software that has the ability to understand its surrounding situations. Situation awareness is to perceive information, take actions based on that perception and to predict the future events based on the taken actions. Mica R. Endsley introduced the term "Situation Awareness" where the concept was applied only for humans. Expert Systems, on the other hand is a rule based system that is used to make critical and fast decisions. In this paper a concept on the comprehension (understanding) level of situation awareness using an expert system is presented for high level autonomous vehicles. This concept will be realized on the "CE-Box" component of the hardware demonstrator "BlackPearl" from the department of Computer Engineering in TU Chemnitz. Perception is done with the help of sensing data from the sensors from the ECU's of "BlackPearl". This data is provided to the comprehension level to make an assessment and take a decision. The goal of this paper is to provide a conceptual method to extract these decisions based on an expert system.

1. Introduction

Millions of people around the world have lost their lives or became physically disabled as a consequence of road accidents. One of the purposes of making driving autonomous is to eradicate the number of accidents and improve road safety. As per the report was given by the European Transport Safety Council (ETSC) every one million inhabitants around 51 lost their lives due to road accidents in the year 2016 [1]. According to the statistical report in the European Union between 2015 and 2016 (figure 1), the total number of road deaths was decreased by 2% [1]. The human driver being unaware of the situation as result in majority of the road deaths. With the advancement in the automated driving, rate of death can be reduced every year if situation awareness is integrated properly. In the future, death rates can be reduced even further with the help of autonomous driving. Finally, at Level 5 of the autonomous driving death rates due to situation unawareness of drivers will be nullified [1].

As we move up the ladder of autonomous driving that is, from Level 0 to Level 5 the complexity of the software increases. Mica. R. Endsley highlighted in her paper [3], automotive companies have a software failure rate of 82 percent. Most of this failure lies due to the lack of understanding of surrounding situations of the vehicle. Collecting the information to identify the driving situation will further intensify the software’s complexity. This has been the motivation to develop the concept of the classification of the driving situations. The classification of the situation might help to decrease software complexity. The Situation Awareness [2] is defined as how the environmental elements or events are perceived, comprehended the meaning and future actions are projected with respect to space and time.
There are three levels in situation awareness [4]:

(i) Perception: Observing and attention to the information such as dynamics, features, and status which are important and relevant to the situation [4].

(ii) Comprehension: All the information are processed. This also involves the integration of the situation’s complete picture from different elements which results in the understanding the importance situation [4].

(iii) Projection: Reacting according to the result of comprehension in a timely manner. This also predicts future actions on the source of the situation comprehension [4].

Disengagements from the autonomous driving include (1) detect a problem in the software of automation and itself gets disengaged, passes the control back to the driver, (2) driver disengages the automation and takes over manually [3]. The reasons for disengagements of the autonomous driving includes failures of software (81.6%), issues in the hardware (14%), conditions of the road surface (0.4%) and weather (4 %) [3]. From the table 1 [3], Waymo [3] is the best performing technology in autonomous driving. In Waymo vehicles, intervene of the human driver needed on an average of 5,600 miles approximately, or before some detected on vehicle and gets disengages by itself.

| Manufacturer       | Miles per Disengagement |
|--------------------|-------------------------|
|                    | 2015 | 2016 | 2017 |
| Google/Waymo       | 1244 | 5128 | 5596 |
| Nissan             | 14   | 146  | 208  |
| Delphi             | 41   | 17.5 | 22.4 |
| Mercedez-Benz      | 1.5  | 2    | 4.5  |

The system should automatically detect the objects and surrounding features, also analyzing the features and predicting the near future development from the current situation data. These methods can be briefed as situation awareness, which was proposed for the aviation domain. According to the traffic situation, the system have to identify and interpret the objects correctly and also have to react to these elements by predicting and giving respective command to the ECU. The system have to understand other elements such as traffic signs, signal or weather conditions act to their relevance for safe driving.

An expert system [4] is a computer program which pretends as the decision and action are made by a human. The system has expert understanding and knowledge in a specific field. The inputs are
provided to the end user by entering the data or selecting from a list which has one or more answers. The system which makes the decision and solves the problem based on the (1) logical rules, (2) task’s knowledge and, (3) procedural knowledge.

2. Literature Survey

2.1. Situational Awareness
Endsley, M.R. [5], situation awareness in the system operation is one of the major concern in which decisions are made based on an eloquent view of the different state is proposed. Here initially, several individual and environmental factors are explored and their relation between situation awareness. The situation awareness deals with the complexity of system, features and operator automation. The model of the situation awareness introduces the classification of errors and design implications are generated to enhance the operator situation awareness.

2.2. Levels of Situation Awareness
Rashaad E.T. et al. [6], presented a paper on the situation awareness levels. The first level perceives the important factors in the environment, the second level tries to recognize the meaning of the perceived factors, and finally, the third level will predict the near future with the perceived situation.

2.3. Expert System
Niehaus A. et al. [7] assumed that vehicle consists of sensors for the detection of traffic, signs, geometry of the road, and also actuators and control logic for controlling the brakes, steering wheel's angle and throttle. The expert system is given with the situation of the traffic, signs and driver’s strategy such that commands are issued to the controllers (Figure 2). The system comprises of a rule base that provides the required knowledge for driving. The reasoning was performed by backward chaining inference, and the reasoning process is optimized by the compiler. The vehicle is controlled either by the expert system or preset strategy.

2.4. Backward Chaining Algorithm
Yuliadi E. [8] used a backward chaining algorithm to start with a goal and end with fact leading to a new goal. This algorithm starts with the last element in the chain and continues until it reaches the first element. Depth- and breath-first search algorithms are used by backward chaining method. The depth-first algorithm is used to search the data in the deeper level of the tree before backtracking. The breath-first algorithm is used to search the data in adjoining nodes and then moves to a deeper level of the tree. This method provides answers dynamically for the inference engine, computation effort, etc.

2.5. Rete Algorithm
Charles L. Forgy [9] presented about Rete Match Algorithm in this article. This algorithm is very efficient in comparing the massive collection of patterns and objects as not iterating over the sets. The
iteration over the patterns does not occur in this program as it contains a network which works on tree-structure sorting or pattern index and efficiently matches all the objects with the pattern. The algorithm is used as the interpreters in the production system and used for the systems which consist of hundreds or thousands of objects and patterns.

In this study, a conceptual method for the comprehension level is highlighted, focusing on an expert system which uses Rete algorithm as its inference engine.

3. System Design

3.1. System Architecture

Videos are capture from Raspberry Pi camera which is provided as the software architecture layer of the Raspberry Pi. The application layer will provide image processed data as input whereas sensory data through CAN message are also taken as the input. Different image processing algorithms can be used to detect traffic light, traffic signs, pedestrian, curvature, and lane. Once the process of the assembler and the expert system is done, the respective CAN message command is given to the ECU of the vehicle. With the help of PiCAN 2 mounted on the Raspberry Pi, this can be achieved. The architecture of the system and data flow control is shown in Figure 3 and Figure 4.

![Figure 3. System Architecture Layer.](image)

![Figure 4. System Architecture.](image)

3.2. Software

Raspbian Stretch operating system is used in each Raspberry Pi of the CE-Box. Open CV library is installed on Raspberry Pi with Python and C++ dependencies for performing the computer vision and image processing applications. The C Language Integrated Production System which in short known as CLIPS. The CLIPS is used for the implementation of the expert system.
3.3. Hardware

The "BlackPearl" is one of the hardware demonstrators of the Department of Computer Science in TU Chemnitz and is used for the applications. The hardware unit of CE-Box consists of multiple racks of Raspberry Pi mounted with the PiCAN 2. The CAN messages are sent and received using the PiCAN 2. The Raspberry Pi of the CE-Box is programmed with the assembler and expert system. Through the CAN message, the raw data are taken as input from the ECU, generate the fact list and corresponding action commands are given back to the ECU. The hardware setup is shown in (Figure 5).

**Figure 5.** Hardware Setup using CE-Box (left) and BlackPearl (right).

There are 8 LEDs, 1 Ethernet port, 5 USB ports, and 1 switch for power and reset option (Figure 6). The power and reset functionality are designed using the Finite State Machine (FSM) (Figure 7). For the safety restart of the Raspberry Pi, the switch need to be pressed and hold for more than 5 seconds. The single press of the switch will disconnect the power from the source and shut down the Raspberry Pi.
The Assembler code and expert system are coded onto the Raspberry Pi. The CAN message from ECU received and transmitted using PiCAN 2 which mounted on the Raspberry Pi. The CAN messages are perceived and converted into raw data. The raw data is given as an input to the assembler. In assembler, data is changed into fact and given to the expert system which in turn give back command (action) to control the operation of the ECU.

The capability of the CAN Bus is given to the Raspberry Pi via PiCAN 2 board. It uses Microchip MCP2515 and MCP2551 for the CAN controller and transceiver. The PiCAN Board can be powered via DB9 connector or screw terminal. This can be programmed using C or Python.

4. Conceptualization
The situation awareness deals with the complexity of system, features and operator automation. Several individual and environmental factors are collected in the perception level. From the perception level, collected elements and converted into facts which are used as the input for the comprehension level. The comprehension level will consist of a rule-based system which will match the element with the set of rules. The comprehension level will also predict the near future action. Finally, action commands are used to control the operation of the vehicle in the projection level.

The main steps of the conceptualization are: (1) getting the sensory data from the ECU through PiCAN 2, (2) getting the Image Processed Data (IPD) from the Raspberry Pi camera which is controlled by the Raspberry Pi Camera Controller (RCC), (3) the assembler will generate fact list from the raw data, (4) the facts are matched with the predefined rules in the Expert System Shell (EXS), (5)
the expert system will update the fact and give the command request to the assembler, (6) the assembler will send the command as CAN message to the ECU, (7) ECU executes the command. All these steps will run continuously in a cyclic manner.

The ECU / Raspberry Pi will perceive the required data or element from the vehicle sensor and environment factors. These elements are used to classify the situation. The collected data is provided to the assembler where the understanding of the data will take place. Once, all the elements are recognized by the assembler, the assembler will generate the initial fact list. The generated facts are given to the expert system. The expert system will match the facts with the predefined set of rules. Then, action commands are given back to assembler from the expert system. Upon receiving the command from the expert system, the assembler will convert this command into CAN message and send it to the ECU and action will take place.

The main role of the assembler is to act as the interface between the ECU and expert system. The raw data from the ECU will be received by the assembler via PiCAN 2. The received raw data as CAN message will be processed and changed into facts. These facts are sent to the expert system. Then, the commands are received back from the expert system. The received command is converted back into CAN message by the assembler. Finally, the command as CAN message is given back to the ECU and the ECU will act on it.

The main functionality of the assembler is: (1) Fusing all the sensory data received via CAN bus from the ECU, (2) With the data, initial fact lists are generated and sent to the expert system, (3) Receive an action request from the rule-based expert system, (4) Send the received command from expert system to the ECU via CAN message, (5) Then acknowledge the received command, (6) Finally, update the fact list.

The rule-based expert system (Figure 9) does the process of situation comprehension. This system will receive the fact list from the assembler. Upon the received fact, the expert system will match the fact with the predefined set of rules. When there is a match between the fact and rule, then the rule is fired. As per the fired rule, the projection of command and predicted action will be generated. The generated command is given to the assembler. The predicted action for the near future will be based on the data collected from the current situation.

The complete set of rules are sub-categorized. The generated facts are given as input to every rule of one sub-category. The facts are matched with the rules and commands are given back to the assembler. On acting upon the rules, we may transit between Rule 1 to Rule 2 for the criteria 1 → 2

![Figure 9. Flow Chart of Expert System.](image-url)
and Rule 1 to Rule 3 for the criteria 1 \(\rightarrow\) 3. This transition between rules takes place based on the different situation criteria which are depicted as FSM (Figure 10).

![Figure 10. Rule's FSM.](image)

5. Future Work
Implementation of the presented concept using Raspberry Pi, PiCAN 2 and checking the output. In order to increase the performance, use a Jetson Nano board which comes with an in-built GPU. Using Assembler to compare the sensory data and generate the fact list may not be realistic. Hence, assembler and expert system design can be modified such a way that assembler will convert the CAN message into data and send the data to an expert system. The expert system will compare and give the command to the assembler. Finally, assembler sends the command to the ECU. Therefore, most of the computation and comparison part is moved to the expert system which makes the overall design feasible for the real-time applications.

6. Conclusions
In this paper, a new concept for the situation awareness in the autonomous driving is proposed. This concept can make the system aware of the situations by collecting the required data, recognizing the data and giving the command to control the ECU according to the current situation. This setup can be carried on Raspberry Pi 3B model with a PiCAN 2 mounted. The hardware demonstrator called BlackPearl with the CE-Box of the department of computer engineering in TU Chemnitz can be used. Once the implementation is realized, an evaluation can be done and presented as well.

As part of the future aspects of this study, a concept for the projection level can also be carried out, completing the three levels of situation awareness.

References
[1] ETSC (European Transport Society Council). Available at: https://etsc.eu/category/publications/
[2] Krems J F and Baumann M R K 2009 Proceeding of International Conference on Human Centered Design 986–994
[3] Endsley M R 2018 Proceedings of the 20th Congress of the International Ergonomics Association) pp 303-309
[4] Kumar Y and Jain Y 2012 International Journal of Computing & Business Research 2229-6166
[5] Endsley M R 1995 Toward a Theory of Situation Awareness in Dynamic Systems (Human Factors The Journal of the Human Factors and Ergonomics Society) vol 37(1) pp 32-64
[6] Jones R E T and Connors E S and Endsley M R 2009 Incorporating the Human Analyst into the Data Fusion Process by Modeling Situation Awareness Using Fuzzy Cognitive Maps (12th
International Conference on Information Fusion Seattle (USA)) pp 1265-1271

[7] Niehaus A and Stengel R F 1991 An expert system for automated highway driving (IEEE Control Systems Magazine) vol 11(3) pp 53-61

[8] Erdani Y 2012 Developing Backward Chaining Algorithm of Inference Engine in Ternary Grid Expert System (International Journal of Advanced Computer Science and Applications) vol 3(9) pp 241-245

[9] Forgy C L 1982 Rete: A Fast Algorithm for the Many Pattern / Many Object Pattern Match Problem (Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213, U.S.A) pp 17-37