Software Architecture for Intelligent Ground Vehicles

Mu Guo, Xinyu Zhang, Wei Han, Siyi Zheng, Lifeng An

Department Of Computer Science and Technology
Tsinghua University, Beijing 100084, China.
Guom08@Gmail.Com

Abstract **- With the rapid development of sensor, navigation, control and artificial intelligence, human society has inevitably entered the era of intelligent driving, which may have a chance to release us from low-level tedious driving activities, hence create an ultimate driving experience. There are many topics of intelligent driving, software architecture is among the most important ones. However, engineers are now overwhelmed by challenges of different vehicle models, sensors, programming languages, driving strategies, perception algorithms, massive data etc. while software architecture is rarely studied. In this paper, we focus on the software architecture for intelligent ground vehicles. Based on detailed analysis of current systems, we propose three design criteria and key techniques like radar map and virtual switch, then we give a novel and universal software architecture design. On-vehicle experiments have shown their advantages.**

Keywords - intelligent driving; software architecture; C^4ISR; radar map; virtual switch

I. INTRODUCTION

Automobiles have been invented for hundreds of years, and have shown really deep impact on human life. Nowadays, with the rapid development of automotive electronics, navigation, automatic control and artificial intelligence, mobile wheeled robots that meet the vehicle dynamics is coming to us. Those robots, also named intelligent ground vehicles or autonomous ground vehicles, would completely change the traditional style of automobiles. They would re-create automobiles via intelligent or autonomous driving.

Currently, large amount of research groups, automobile companies and even internet companies have been investing considerable effort to the research and development of intelligent ground vehicles. Many of them have partially fulfilled the goal of intelligent driving. Groups from all over the world have been doing research based on all kinds of automobile platforms and sensor configurations. The traditional style of developing an intelligent ground vehicle is to perform various algorithms including perception, navigation, decision-making and automatic control. In past few years, with the great effort of groups from all over the world, large quantities of codes and software including perception, navigation, decision-making and automatic control have been realized by variety of programming languages. With the accumulation and evolution of all those algorithms and codes, the kernel problem of automobile software development is changing from writing codes to the designing of software architecture. Design of software architecture with highly adaptive to different automobile platforms, sensor configurations and all kinds of programming languages have become the most urgent requirement for development of intelligent ground vehicles.

Excellent software architecture provides solid guarantee to the quality of overall software system. It brings great convenient to the development procedure with considerable improvement on the scalability, reusability and maintainability of software architecture. Meanwhile, the defective software architecture would make really bad effect on the whole system. With defective software architecture, it would be hard to add, modify or reuse the units of software system. Moreover, little modification may bring crash to whole system. Defects in software architecture are all faulty error, may lead to irreversible consequences. For this reason, design a flexible, scalable software architecture with little defect have been the top problem to be solved.

This article proceeds as follows. In section 2, we make systematical analysis for the system of intelligent ground vehicles. Section 3 lists three principles for designing software architecture. A reference software architecture is shown in Section 4. And this paper concludes with a review of summary and future directions.

II. SYSTEM ANALYSIS OF INTELLIGENT GROUND VEHICLES

A. Comparison with C^4ISR

C^4ISR is a military term, short for the combination of command, control, communications, computers, intelligent, surveillance and reconnaissance. It’s the kernel of modern high technical military confrontation. The command system is the headquarters of C^4ISR, taking charge of the power distribution and decision making of the overall battleground. The control system is the feet and hands of C^4ISR, doing execution of decision made by

DOI 10.5013/IJSSST.a.17.36.15 ISSN: 1473-804x online, 1473-8031 print

command system. The communication system is the neural networks of C^4ISR, transferring the situation from battle field and commands from headquarters. The computer system is brain of C^4ISR, forming the technical basis of whole system. The intelligent, surveillance and reconnaissance systems are the eyes and ears of C^4ISR, collecting all information and intelligence from the battle field.

The system of intelligent ground vehicle contains all elements in C^4ISR, although it's much simpler compared to real C^4ISR. When driving on the road, intelligent ground vehicles would confront complex surroundings. We need to equip variety of sensors including camera, radar etc. , in order to percept the environment precisely, rapidly. These sensors constitutes the ears and eyes of system. Communication network and computer system are also equipped on the vehicle. Raw data collected by the sensors would be transferred to computer system via communication network. Several intelligent ground vehicles can also be connected by communication network, sharing information between nearby vehicles. Decision-making unit running on the computer is the headquarters of intelligent ground vehicles. It determines the expected state of vehicles based on systematical analysis of surroundings, navigation information and human command. Decisions would be send to control system, which automatically operates the wheel, throttle and brake of vehicle, making vehicle approaching its intended state.

The system architecture of intelligent vehicles can be divided as the C^4ISR, shown in Figure 1.

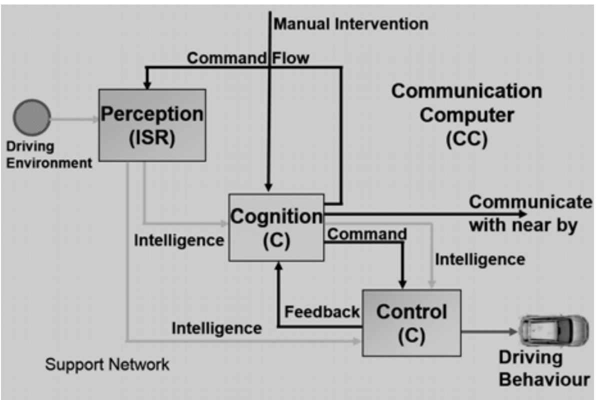

Figure 1 C^4ISR Architecture for Intelligent Ground Vehicles

B. Strategy for Intelligent Driving

The implementation of intelligent ground vehicle is a huge system engineering, involves many technical details from all kinds of subjects. Even so, from an overall perspective, the strategy for intelligent driving is composed of autonomous driving, integrated navigation and manual intervention.

Autonomous driving means vehicles collect and analyze the information from surrounding environment, get perception and knowledge of driving situation, make decision based on this situation and perform automatic control to make vehicle driving autonomously. Autonomous driving can achieve normal driving behavior instead of human drivers, including following, taking over, going through intersections etc.

Integrated navigation means location based service via cloud computing. It provides real-time navigation with centimeter-level accuracy for vehicles by using real-time kinematic GPS, GLONASS, Beidou navigation systems and continuous operational reference system.

Besides, Human driver can express intent and handle the unexpected situations via manual intervention.

The combination of autonomous driving, integrated navigation and manual intervention forms the overall strategy for intelligent driving. But the priority of these three is different. Reasonable manual intervention has the highest priority, while the autonomous driving has higher priority than integrated navigation.

C. ISR System of Intelligent Vehicles

The capability of percept the surroundings rapidly, precisely and comprehensively is the precondition for intelligent driving. In the perspective of C^4ISR, this part would be the ISR system of intelligent ground vehicles. The ISR system of intelligent ground vehicles includes posture sensors detecting the pose of host vehicles, perception sensors getting the surroundings, position system calculating the absolute location of host vehicles and priori knowledge including geographical environment, road facility and other driving information.

The sensor configuration is the hardware basis of intelligent ground vehicles. Some research groups tend to percept the environment mainly based on cameras. One of the representations is the intelligent vehicle group from VisLab, Parma University, Italy. Their vehicles team achieved a long journey from Milan Italy to Shanghai China in 2010 with limited manual intervention. The total length of their journey is about 13,000 kilometers. Meanwhile, most of the research groups get the environment information mainly using radar and lidar. Driveless car developed by Google is a good example. Google's car use a group of lidars to detect obstacles, self-position and plan path. Cameras are only responsible for detecting traffic lights, lane marks and traffic signs. Besides, there is no uniform configuration of sensors, the model and mounted-position of sensors vary a lot.

There exist many thinking and ideas for sensor configuration. For camera sensors, we can use only one fish-eye camera to detect lane-marks, traffic light, traffic signs and obstacles simultaneously, while we can also equip several cameras, one with unique task. On the configuration of sensors, we need to do trade-off between

DOI 10.5013/IJSSST.a.17.36.15

ISSN: 1473-804x online, 1473-8031 print

cost of calculation resource and proper redundancy for reliability. We don't have unique solution right now, perhaps no final solution as well.

In our group, we concurrently use cameras, radars and lidars for perception. Integrated positioning system based on GPS, Beidou and IMU are also equipped. Configuration of sensors are shown in figure 2 and figure 3.

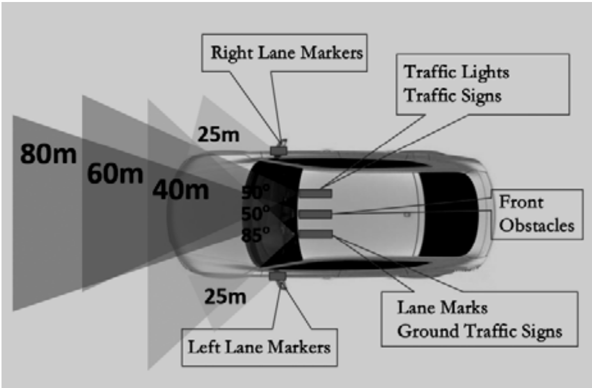

Figure 2 Sensor Configuration-Camera.

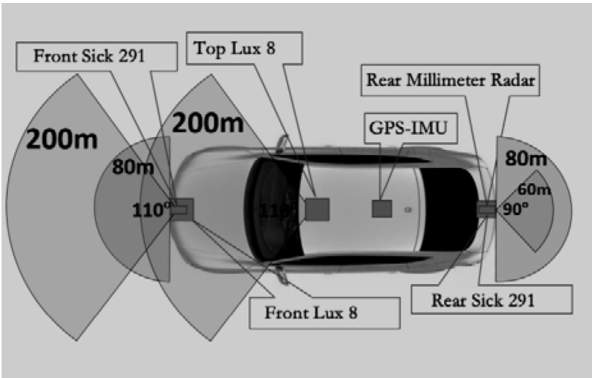

Figure 3 Sensor Configuration-Radar and Positioning.

D. Information Flow

For most intelligent ground vehicle, the sequence of information flow follow perception-decision-control.

In perception step, intelligent vehicle make comprehensive analysis based on posture sensors, perception sensors, positioning sensors and priori knowledge. The posture sensors provide speed, acceleration, yaw, pitch and roll of host vehicle. Inertial Measurement Unit, odometry are all post sensors. Besides, we can get some more information directly from the can-bus of vehicle. Perception sensors include cameras, lidars, millimeter radars, infrared radars etc. Raw data are processed by all kinds of algorithms to provide driving information including obstacles, lane-marks, traffic signs and traffic lights etc. This two kinds of sensors provide dynamic, real-time information of host vehicle and its surroundings. Besides, we can get high-precision location from the positioning system. Based on accurate location and priori knowledge of geographical environment and other driving information, we can get some static driving information such as speed limit and slope of road.

Decision making step get the real-time and static information supplied by perception step, determine the expected state of host vehicle with comprehensive consideration of priori driving knowledge, integrated navigation and manual intervention. Control step puts the expected state as input, operates the wheel, throttle and brake of host vehicle using automatic control theory, to make the host vehicle close to the expected state.

The information flow of intelligent vehicles is show in Figure 5.

III. SOFTWARE ARCHITECTURE DESIGN PRINCIPLES

In the process of designing, we need to follow principles of system, expression and modularity, to guarantee the scalability, reusability and maintainability of software architecture, shown in Figure 4.

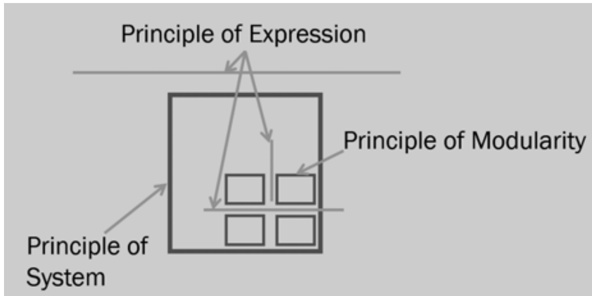

Figure 4 Software Architecture Design Principles

A. System Principle

Before the design of software architecture, we need to clarify the boundary of system first. That is we need to make clearly that which parts are in the boundary of system that we need to accomplish while other parts are not and we don't need to care about them.

As for the development of intelligent ground vehicle, at least three parts are out of our boundary:

Any modification related to the dynamic performance of host vehicle is out of boundary. We won't enhance or weaken the dynamic performance of the host vehicle. Intelligent vehicle would totally follow the factory index about the dynamic performance.

Perfect solution for all road and weather condition is not our goal. We won't pursue the absolute autonomous driving. The development of intelligent vehicle would be an evolution process

There won't be any mandatory usage for specific platform, programming language, debug or test tools. The software architecture must be open and scalability.

B. Expression Principle

As a complex system, there are usually tens of modules in the software architecture. These modules are distributed in different levels and no natural expression framework exists. The core of expression principle is to build a uniform framework under which modules in same or different levels can be well expressed.

Based on the cognition of human driver, we design a framework named driving situation map, as the uniform framework for all modules. Driving situation map is a round map, with coarse-grained grids in the far region and fine-grained grids in the near region. This design well fits the sensor perception and human cognition accuracy. Information from posture sensors, perception sensors and priori knowledge can all be mapped into driving situation map, forming an entire knowledge of driving status. The origin point of driving situation map is superposition with the center of host vehicle. Content of driving situation map changes dynamically, reflecting the changes of surroundings and is the basis of path planning and decision making. We choose driving situation as uniform framework for all modules and entire system.

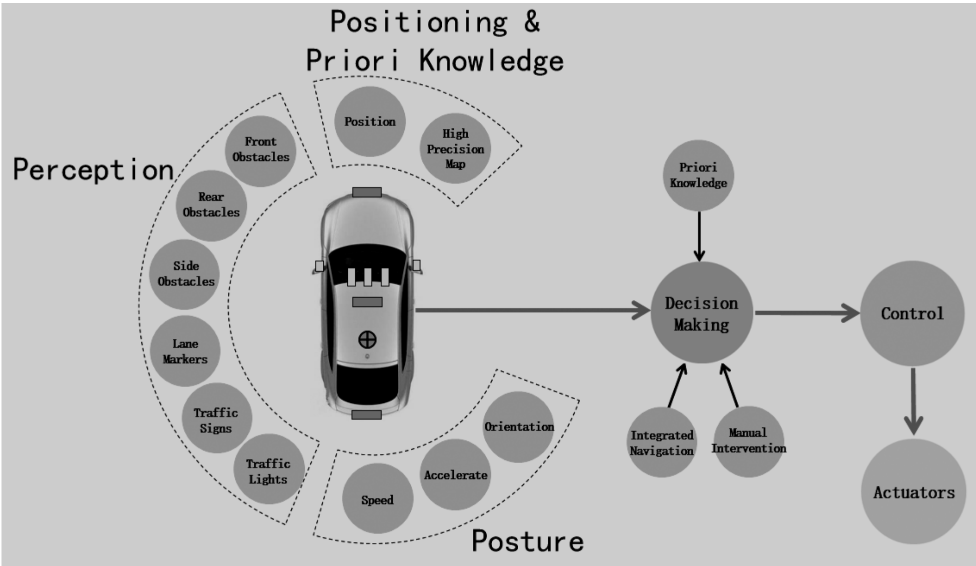

Figure 5 Information Flow of Intelligent Ground Vehicle.

One typical design of driving situation map is shown as Figure 6.

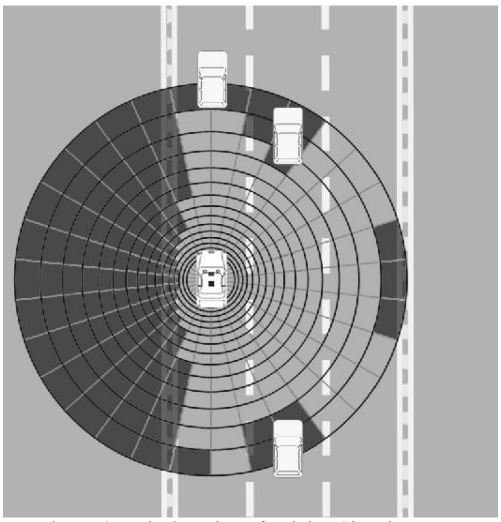

Figure 6 Typical Design of Driving Situation Map.

C. Modularity Principle

Module is the basic unit of software architecture. There would be quantities of modules for an intelligent ground vehicles. We need to follow the principle of modularity, make wise division to maximize the independence between modules. The principle of modularity requires each module has independent function and simple interface so as to minimize the difficulty for coding, comprehension and maintain.

There are some principles in traditional software architecture designing, which are also suitable for use in the design of intelligent vehicles. One of the most important is high cohesion and low coupling. We should follow the single responsibility principle to make sure that every module has single function. Module should only use local variables, except for parameters. When two modules correlated with each other, the best choice is always data coupling. Control and common coupling should be limited and content coupling is forbidden.

For intelligent ground vehicles, we may divide the entire software architecture into several layers. Modules in the same layer are independent or have one-directional data coupling. Cross call of same layer modules is forbidden. Modules in lower layer may be called by upper layer modules, to achieve data interchange. The bottom supporting modules should be stable. When we change upper layer modules due to new algorithms or different sensor configuration, supporting modules should not have any modification.

Principle of modularity makes the delta step of debugging come true. In the development of intelligent vehicles, joint debugging all modules is not realistic in the beginning. Make wise division of modules, do verification incrementally is the scientific method.

IV. SOLUTION TO SOFTWARE ARCHITECTURE

As has been noted, there is no unique solution to sensor configuration now, and perhaps there won't be a final solution either. Software architecture is tight related to sensor configuration, so software architecture for different vehicles won't be the same.

However, no matter how the sensor module and quantity changes, the system boundary determined by the principle of system is fixed. Follow the principle of modularity, choose driving situation map as uniform framework, we can divide the intelligent software into support layer, driver layer and application layer, from bottom to top.

Support layer is the bottom layer, containing operating system and three support modules including virtual switch, process monitor and logger. These three support modules are basic common modules, providing services such as data interchange, running status monitor and log record. Virtual switch is like a bus-controller, creating several virtual data-bus. Every module could publish messages to and subscribe messages from one or more data-buses. Virtual switch convert possibly common coupling, control coupling into loose data coupling, significantly improve the module independence, strengthen the scalability and maintainability of overall system. Process monitor module judges the running status of modules by receiving and analyze the heart-beat data from them. If the heart-beat data is lost or much slower than normal, the modules may not work properly. Logger module recording messages on the bus real-time, so as to playback and reproduce the situation when encountering abnormal.

Driving layer consists of driver modules of each sensor. The content of driver modules is different according to sensor configuration. However, every driver module's function should contain parameter setting, data calibration and data transmission.

Application layer is the most changeable layer. There are various kinds of modules including data processing, data fusion, decision-making, automatic control, human intervention, navigation and so on. We achieved data fusion via driving situation map. The data fusion modules subscribe messages from sensor data processing modules and make fusion under the framework of driving situation map. The driving situation map is also used as a public data pool, providing sufficient information for decision making and other modules.

A reference solution of software architecture is show in Figure 7. All three intelligent ground vehicles in our team are developed under this architecture, with a little modification according to different platforms and sensor configurations.

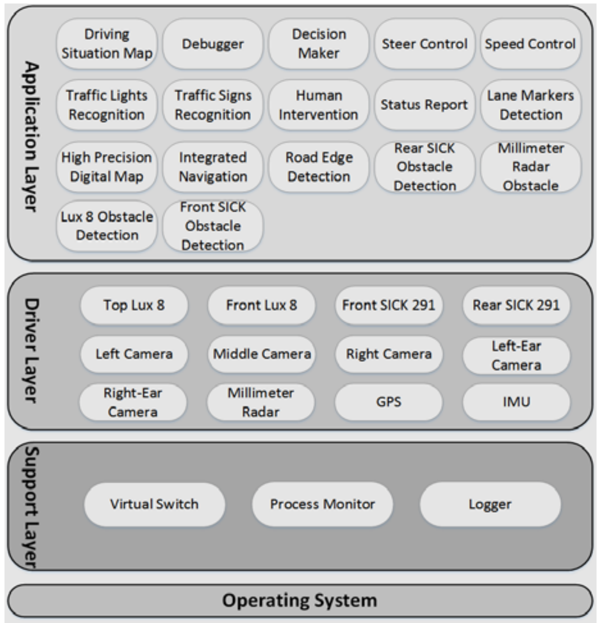

Figure 7 Reference Solution to Software Architecture

Intelligent ground vehicles of our team achieved 114km highway driving without any human intervention with an average speed 87.1km/h in 2012. In the 4th Future Challenge of Intelligent Vehicles in China, our intelligent ground vehicles won the first place.

V. CONCLUSION

In this paper, we first made an overall analysis of the system architecture, driving strategy and sensor configuration of intelligent ground vehicles. We concluded three principles of software architecture design and proposed a reference software architecture that highly adaptive to different platforms, algorithms and sensor configurations. The software architecture can be divided into support layer, driver layer and application layer, with driving situation map as uniform framework for expression. Based on this software architecture, developers could conveniently verify different combination of modules and build the whole system incrementally. This software architecture have shown its advantages in the development of series intelligent ground vehicles.

With the fusion of information technology and traditional automobile industry, intelligent ground vehicles is coming to us. It's expected that human driver will be emancipated from the low level, tedious driving. And hope we would live in a world without driving accident and hurt.

REFERENCES

[1] Ibañez-Guzman J, Laugier C, Yoder J D, et al. Autonomous Driving: Context and State-of-the-Art[M]. Handbook of Intelligent Vehicles. Springer London, 2012: 1271-1310.

[2] Levis A H, Wagenhals L W. C^4ISR architectures: I. Developing a process for C^4ISR architecture design[J]. Systems engineering, 2000, 3(4): 225-247.

[3] Li Deyi, Zhang Tianlei, Huang Liwei. A down-to-earth cloud computing: Location-Based Service[J]. Acta Electronica Sinica, 2014, 42(4): 786-790. (in Chinese)

[4] Bertozzi M, Broggi A, Coati A, et al. A 13,000 km intercontinental trip with driverless vehicles: The VIAC experiment[J]. Intelligent Transportation Systems Magazine, IEEE, 2013, 5(1): 28-41.

[5] Thrun S. Google's driverless car[J]. Ted Talk, Ed, 2011.

[6] Li Deyi. Intelligent vehicle using variable size row radar map method of information fusion [P]. China, 201310128508, 2013.

[7] Bass L, Clements P, Kazman R. Software architecture in practice[M]. Addison-Wesley Professional, 2003.