
Design and Realization of System Structure based on Blackboard System

RZ Chu*, SJ Sun, L Yang

Hebei University of Engineering, HanDan Guangming S. Main Street No. 199, 0310 8579377

*Corresponding author, e-mail: crzcrzcrz@163.com

Abstract

This paper discusses design environment of software system structure of blackboard style, which ascertain out the job pattern the blackboard decision-making model. At last, let's list regarding a mobile robotics architecture primary need and realize the characteristic to the blackboard system pattern in the quality demand to carry on the analysis, taking floor intelligence robot software system as example. Finally the quality and ability of the four patters in architecture classification has been compared.

Keywords: blackboard model, blackboard system structure, intelligence robot, quality demand

Copyright © 2013 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

Structuring the software controlling the floor intelligence robot is a difficult duty. The floor intelligence robot is also called the floor self-service doing battle platform, the pilotless vehicles etc. It is one kind of non-crew member the intelligent ground weapons, which can also replace a soldier to accomplish the intellectualized various tactics mission. Robotic platform of floor is the outcome of multidisciplinary synthetical, taking IT as core. Intelligence robot's core is computer system, which mainly is used in the robot sensation information processing, the understanding, the decision-making and the guidance. Navlab project is one part of AVL (Autonomous Land Vehicle) in Advanced Research Project Agency (be ARPA), aiming at developing out the the vehicle able to run and stride over the open country on the road automatically. This project has a lot of significant technology breakthrough in a lot of aspects such as computer vision, robotic route plans, autocontrol and obstacle identification, making the intelligence robot jump over up a brand-new flight of steps [1-2].

2. Blackboard Style

Software system structure style is the customary model describing the way of system organization in some specific application domain. System structure style has defined a system family, which means that a system structure defines a glossary and a group of constraint. System structure style has reflected common structure and the semantic characteristic of a lot of middle systematic field institutes, and guided on how the various modules and subsystems will be effectively organized into a complete system. In this way to understand, software architecture style defines the rule used in describing system's terminology table and a set of guide component system.

The warehouse style has two kind of different components: central data structure that describes the current status and independent component implementation in the central data storage. The interreaction of warehouses and external components in the system will be big changes [3]. The control principle's selection has two main subclasses. If a class of time in an input stream of transactions trigger, the selection of the execution of a process, the repository can be a traditional database. On the other hand, if the current state of the central data structure is the main trigger of the selection of the execution of a process, the repository can be a blackboard.

The blackboard system structure is a skill of rule-based reasoning and problem solving. It's necessary to coordinate the behavior of members in social organization. Every members all

act as certain roles, complete one selected low layer behavior and a high-level objective. It requires cooperation of different low-rise behaviors, which usually is implicit [4].

Figure 1 is the constitution of blackboard system. The traditional application of blackboard system is the signal processing, such as voice and pattern recognition. Another application is loose coupling agent data shared access. Blackboard system is made up of three major parts.

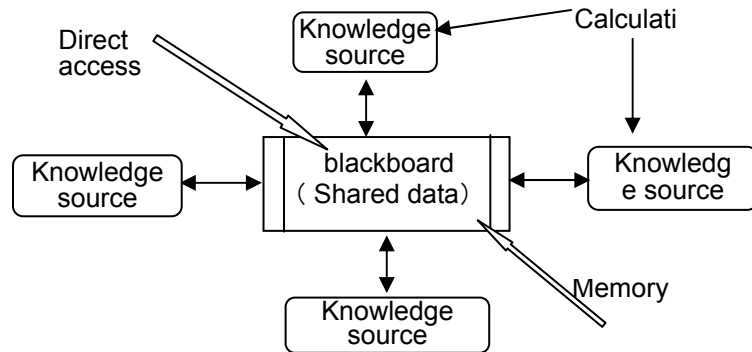


Figure 1. The Composition of the Blackboard System

The knowledge sources: the knowledge sources include knowledge and its processing unit which is independent, separate and be related to .Their interreaction is only completed by blackboard.

The blackboard data structure: blackboard data is problem-solving state data according to association with the application level. Knowledge sources solve the problems by making changes in the blackboard data incrementally [5].

Control: driven entirely by state of blackboard. Blackboard state decides the used selected knowledge.

3. Intelligent Control System

The floor intelligence robot decision-making subsystem belongs to the model intelligent control system, generally using the lamination hierarchical control method. According to classification intelligent control system, it is composed of three intelligence (perception) level mainly [6-7] .It's exactly shown in Figure 2.

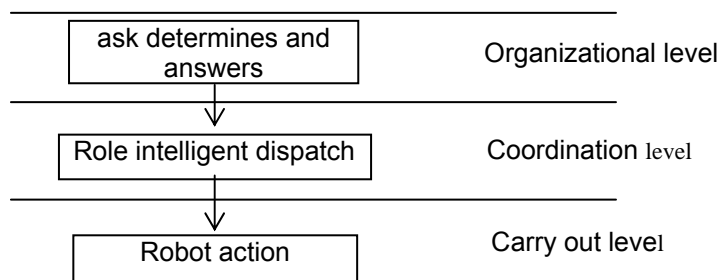


Figure 2. Classification of Intelligent Control System

First level: the organizational level. It represents the dominant ideology, and is held by artificial intelligence, which means the current problem solving and problem decomposition.

The second level: coordination level. It is the interface between the higher level (the first level) and the lower level (the third level) being controlled by artificial intelligence and

operational research. That is, intelligent scheduling or portfolio can find reasonable and be the most effective distribution scheme, and it is also a multi-agent collaboration at the core of the mechanism.

The Third level: It is the intelligent control systems of the lowest level, requiring a high level of precision and controlling by control theory.

Thinking that the characteristic of ground intelligence robot decision-making subsystem, we use the opportunity of reasoning and blackboard-knowledge source system structure to sort the task of solving subtask that will dispatch system perform when dispatch system according to priority. It always invokes the appropriate knowledge sources to address the highest priority task [8]. At the same time, through the blackboard monitor, it reschedule some parts as the right answer for an unexpected event occurs for. Its structure is shown in Figure 3.

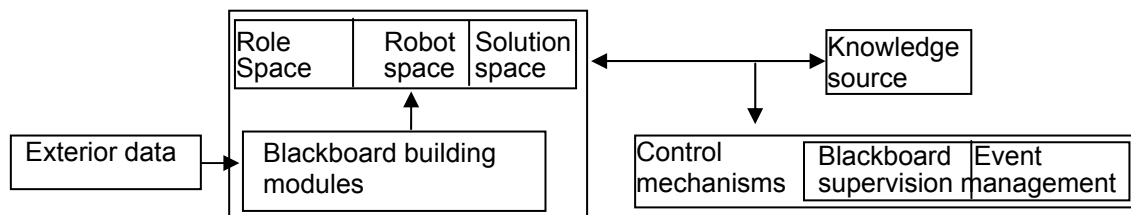


Figure 3. Role Intelligent Scheduling of Blackboard Structure

4. Case Study

The features of ground robotic software generally include getting the input provided by the sensor signals, controlling wheels and other removable parts and calculate next walking path. There are many factors that can make these tasks complicated: obstacles standing in the way of walking path, the sensor incomplete input, robot power failure and the robot action accuracy restricted by mechanical principle. Besides, robots need to handle dangerous goods. Moreover, It is necessary to reponse quickly to unpredictable event. In many robot system design paradigm, the following lists the mobile robot architecture of basic needs: robot needs to coordinate the two aspects of the behavior to complete the set destination task to subtle action steps. On the other hand, it is necessary to response to a abnormal environmental effect timely. Schema must take the uncertainty, the robot operation and inherent danger in the environment into account. Schema must give designers flexibility [9-10]. The development of ground robotic application need to be continually experimented and reconfigured regularly.

Requirements of Mobile Robotics: (1) Requirements for the mobile robot's architecture. (2) The architecture must accommodate deliberative and reactive behavior. (3) The architecture must allow for uncertainty. (4) The architecture must account for dangers inherent in the robot's operation and environment. (5) The architecture must give the designed flexibility [11].

5. Solution

Solution 1: Control Loop

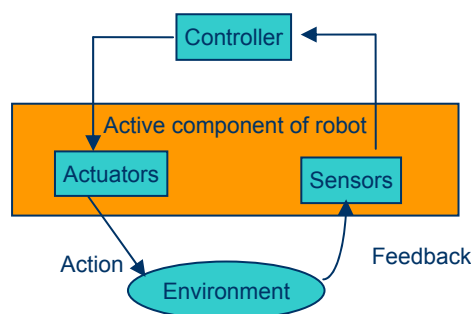


Figure 4. Control Loop of Mobile Robotics

Traditional robots say ones welding on an assembly line don't have the amount of uncertainty that mobile robots do. (1) Mobile robots require a Feedback loop .(2) →sensors→ controller→actions →feedback. Control Loop of Mobile Robotics is shown in Figure 4.

Notes on Requirements: (1) Feedback loop not suited for large amounts of uncertainty; it assumes continuity in changes in the environment. (2) Fault tolerance can be supported by duplication because of the simplicity of the feedback loop. (3) Refinements will typically occur inside the modules at a level the architecture will not capture.

Solution 2: Layered Architecture

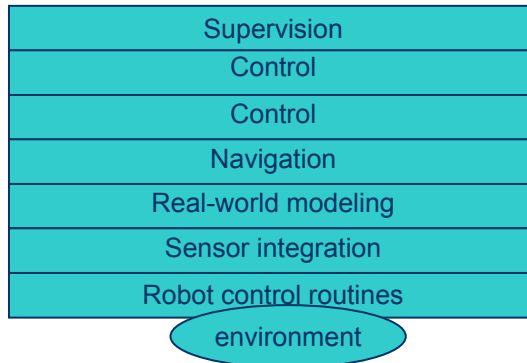


Figure 5. Layered Architecture of Mobile Robotics

Layered Architecture of Mobile Robotics is shown in Figure 5. This organizes the components well, but restricts the actual data flow and control flow. Notes on Requirements: (1) Feedback loop not suited for large amounts of uncertainty; it assumes continuity in changes in the environment. (2) Abstraction layers address need for uncertainty. (3) Fault tolerance are served by the abstraction mechanism. (4) Interlayer dependencies are an obstacle to easy replacement and addition of components. (5) Breaks down when greater level of detail is required.

Solution 3: Implicit Execution

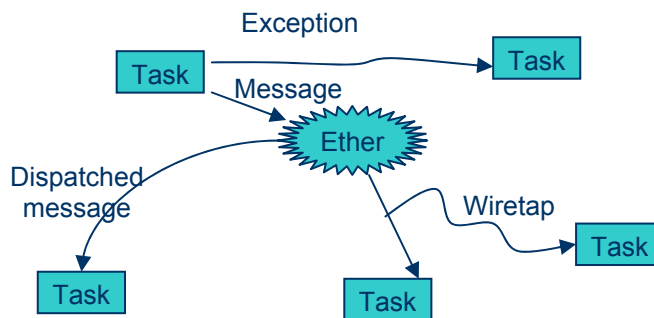


Figure 6. Task-Control-Architecture (TCA) of Mobile Robotics

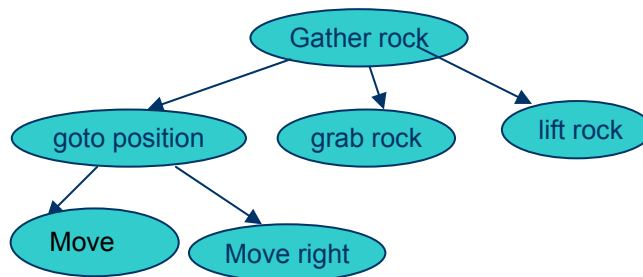


Figure 7. Task Trees of Mobile Robotics

Task-Control-Architecture (TCA) can be used to control several robots. Figure 6 shows The main frame of TCA. Figure 7 shows A simple Task trees. It shows the sub-task is activated by the parent task.

TCA's implicit invocation support 3 functions: (1) Exceptions—conditions cause the execution of an exception handler. (2) Wiretapping—messages can be intercepted and superimposed on an existing task tree. (3) Monitors—read information and execute actions.

This organizes the components well, but restricts the actual data flow and control flow. Notes on Requirements: (1) Task trees and functions permit clear separation of action and reaction. (2) Uncertain how TCA handles uncertainty. (3) Fault tolerance achieved by redundancy with multiple “handlers” associated with signals. (4) Incremental development and replacement of components is easy. (5) TCA's task trees match the domain well.

Solution 4: Blackboard

Figure 8 shows the architecture of the blackboard model. CODGER system a “whiteboard architecture” as it works with abstractions Components of CODGER:

Captain: overall supervisor;

Map navigator: decision maker walking on the high-level way;

Lookout: the one who monitors the environment looking for landmarks;

Pilot: low level path planner and motor controller;

Perception subsystem: the unified consistent explanation that comprised from the primitive input by many sensors.

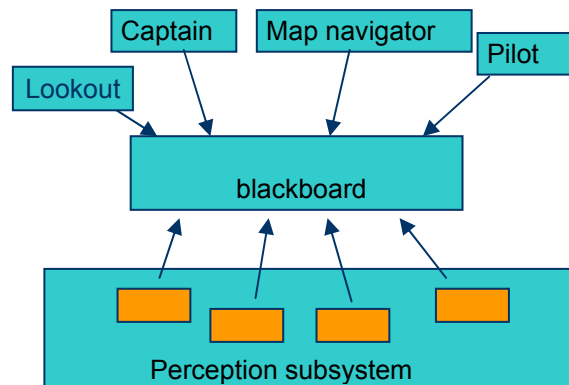


Figure 8. Ground Robotic Blackboard Mode Solutions

Analyse 4 fundamental need:

Requirement 1: Component communicate by Blackboard mode system shared database. Each module specified their own particular types of information. A database return appropriate data directly or when other modules plunge the required data into the database. For example: a supervisor is responsible for paying attention to the specific geographical characteristics of the way. The database is able to notice a supervisor during the period of the subsystem receiving the image according to the characteristic from perception.

Requirement 2: Blackboard components are also means resolving the conflict and uncertainty from the perspective of Robotics logic. For example, a supervisor of landmark detection provides an on-site examination, that is, directly through the store on the Blackboard to sense data to estimate distances [12-13]. Function modules and perception subsystem data are stored in the same Blackboard to solve the problem that the settlement uncertainty modules registered to the database when getting the desired data. (The sense data "fusion" feature is a good example, reconciling the input data in different sensing element by perception subsystem).

Requirement 3: They are similar between the adoption of the Blackboard data communications and control loop TCA (task control model) schema via a centralized information server communication. TCA accidents solving, putting a listening and monitor mechanisms to ensure the robot reaction speed, security and reliability —The CODGER construction realizes the similar request through definition separation module. These components monitor the database in which the accident concurrent or the indication that difficult position will start.

Requirement 4: Blackboard mode system that supports concurrency intercept reception and transmission by design. It contributes to system maintaining.

In short, Blackboard mode system has the ability of collaborative modeling tasks, providing you with a flexible way to implement task collaboration and the ability to resolve the uncertainty [14]. This is attributed to the contents of a database based on the Blackboard implicitly invocation mechanism.

6. Resolve Scheme Comparison

Judge 4 kind robots for the mode to meet basic needs with comparative table of assessment. The four substantive different architectures has virtual differences in control policy, means of communication and component definition, which affected the basic requirements are met [15].

Table1. 4 Mode Quality Demand Satisfaction Comparison

	Control loop	Layers	Implicit Invocation	Blackboard
Task Coordination	+-	-	++	+
Dealing with uncertainty	-	+-	+-	+
Fault intolerance	+-	+-	++	+
Safety	+-	+-	++	+
Performance	+-	+-	++	+
Flexibility	+-	-	+	+

The Blackboard system has the following advantages: support on changeable and serviceability; reusable knowledge sources; support for fault tolerance and robustness, etc.

Conclusion: Implicit invocation mechanism of Blackboard database determines the blackboard model system's ability to collaborate and model the tasks. It realize the ability to accomplish tasks and solve uncertainty by flexible way. The Blackboard architecture has the advantage of being operational unit to roughly the same way the message is delivered by Blackboard. Their behavior is opportunistic, cooperative, consonant and extensible. In ground robotic systems using the Blackboard architecture, Blackboard command directly, realizing information shared flat, command and control changed and diversified, system connection seamless, response time minimization and network access liberalization. It increases control and coordination among the various units. Besides, it is beneficial to the army in the battlefield to anticipate the enemy earlier.

References

- [1] Chen CH, WU Tao, Luis G. Occena. Knowledge Organisation of Product Design Blackboard Systems Viagraph Decomposition. *Knowledge-Based Systems*. 2002; 15: 423- 435.
- [2] CHANG Yan, CHEN Mao-gou, YUAN Jie. Study on the Architecture of MAS-Based Distributed Group Decision Support System. *Computer Engineering and Applications*. 2003; 18: 100– 102.
- [3] Xiang Yang, Shen Hong. Intelligent Decision Support System Based on Agent. *Computer Engineering*. 2006; 32(16): 167-169.
- [4] NIIHP. Blackboard systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architecture. *The AI Magazine*. 1986; 17(2): 38-53.
- [5] WANG Gang. Blackboard Architecture to Unify Encapsulate Multiple Computer Communications Interfaces. *Ship Electronic Engineering*. 2009; 29(9): 38-41.
- [6] Gilmore JF, Roth SP, Tynor SD. A Blackboard System for Distributed Problem Solving. San Diego, CA: Academic Press. 1989.
- [7] McManus JW. Design and Analysis Techniques for Concurrent Blackboard Systems. *IEEE Trans On System, Man and Cybernetics Part A: Systems and Humans*. 1996; 26(6): 669-680.
- [8] Xiong N, Svensson P. Multi-Sensor Management for Information Fusion: Issues and Approaches, Information Fusion. *Information Fusion*. 2002; 3(2): 163- 186.
- [9] XIA Hong, LIU Qun. Knowledge-Based Diagnostic Systems of Turbine with Faults using the Blackboard Mode. Proc of the 1997 IEEE Conf on Intelligent Processing Systems. Beijing. 1997: 1516- 1520.

-
- [10] Hari Lawrence Pfleeger. *Software Engineering Theory and Practice*. Higher Education Press Pearson Education. 2003: 383-443.
- [11] A Petrenko, S Boroday, R Groz. Confirming Configurations in EFSM Testing. *IEEE Trans. Software Engineering*. 2004; 30(1): 29-42.
- [12] J Darlington, RM Burstall. A Transformation System for Developing Recursive Programs. *J.ACM*. 1977; 24(1): 44-67.
- [13] M Harman, L Hu, R Hierons, J Wegener, H Sthamer, A Baresel, M Roper. Testability ansformation. *IEEE Trans. Software Engineering*. 2004; 30(1): 3-16.
- [14] Cui Cheng-chen, Lu Zhi-cheng, Zhu Zhu-bing. Analysis on Seismic Response of UHV Electric Equipment-Support Structure System. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(8): 2101-2107.
- [15] Mardiyono Mardiyono, Reni Suryanita, Azlan Adnan. Intelligent Monitoring System on Prediction of Building Damage Index using Artificial Neural Network. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(1): 155-164.