

A blackboard approach to the mission management for autonomous underwater vehicle

E.A.P. Silva, F.L. Pereira & J. Borges de Sousa

Institute of Systems and Robotics (I.S.R.) and

D.E.E.C. - Faculdade de Engenharia da

Universidade do Porto, Rua dos Bragas, 4099

PORTO CODEX, Portugal

ABSTRACT

In this paper we propose a blackboard approach to the coordination stage of a practical Mission Management System which is composed of three hierarchic levels:

The organization level takes place off-line and produces a coherent set of subplanes permitting to achieve the mission goals in a desirable fashion. The resulting plan will be considered as a reference basis for future actions once the mission has started.

As the mission develops, a blackboard based coordinating structure (in the intermediate hierarchic level) will adopt the adequate mission achievement and safety oriented behaviors as a response to continuously monitored mission state and detected internal and external events.

The lowest level of the hierarchy consists in the execution units which perform the required behaviors.

INTRODUCTION

In this article, we address issues arising in the Vehicle and Mission Management System (VMS) providing the underwater vehicle with competence to autonomously accomplish a given mission in the defined operational environment while ensuring its integrity either under adverse unexpected conditions (e.g., unforeseen obstacles or currents) or when unexpected events occur (e.g., system failures).

Typical missions include travelling between given points and, at each site, performing a sequence of tasks. Examples of such activities

are: material sampling, environmental data collection, video tape recording, inspection of underwater structures and surveillance.

Therefore the vehicle should be provided with the following functions:

Task performance (navigate to reach task site, perform task and move to next rendezvous point);

Safety (detect potentially dangerous situations, activate adequate contingency plans);

Reactivity (perform adequate replanning, react in real-time to either expected or unexpected events);

Cost effectiveness (efficient use of hardware, low mission programming costs, adequate compromise between mission achievement and integrity risk, efficient preprocessing and/or storage of collected data and/or samples).

This paper addresses the VMS coordination system which is the intermediate level of a three level hierarchical structure. While the top level takes place off-line and performs the validation, interpretation and organization of the user specified mission, the other two operate as the mission develops. The coordination level includes not only the assessment of vehicle systems, mission achievement and environmental conditions but also all the making of decisions that ensure the activation of the functional modules required to carry out the mission subject to the vehicle's integrity. The lowest level is constituted by the execution systems.

We present, in this paper, a blackboard system permitting the real-time global coordination of all the various subsystems constituting the underwater vehicle so that a feasible given mission may be autonomously accomplished via an intelligent like behavior. This includes not only the monitoring and evaluation of the mission performance and status of all the various subsystems constituting the underwater vehicle (such as navigation, control, vehicle support system, communications, external devices interface and sensorial integration) but also the dispatching and scheduling of functional modules permitting the motion and task performance, task and path replanning and the real-time obstacle avoidance and reflexive trajectory planning.

The proposed software architecture will be implemented in a multiprocessor environment where a unique processor is responsible for the control blackboard and the knowledge sources are executed on other parallel processors. This hardware architecture provides the adequate environment for real-time execution of this software architecture (see Erickson et al. [9]). By embedding the trigger

conditions of the knowledge sources in the control blackboard mechanism and locating their execution on other processors we envisage to separate the firing mechanisms from their execution.

This article is organized as follows:

In the next section, we present the main motivation of our approach after a brief description of the state of the art concerning VMS architectures and architectures for AUV management and control using blackboard systems. A detailed description of the architecture of the AUV software system is presented in the following section. In section 4, a detailed description of the blackboard based coordinating scheme and an analysis of its features are given. Finally, some conclusions are drawn and perspectives for future work presented in section 5.

STATE OF THE ART

Research on the design and implementation of Mission Management Systems for Autonomous Mobile Vehicles constitutes a challenging task involving advanced concepts from a variety of areas such as AI, Robotics, Real-World Modeling, Planning and Intelligent High Level Control. Intelligent High Level Control represents a generalization of the concept of control in order to manage complex systems in uncertain environments by using cognitive engineering systems and the power of available hardware and software technology.

There are two main approaches to the design of software systems for autonomous operation of robotics vehicles.

G. Saridis [20] has proposed an analytic formulation of such systems based on the principle of increasing intelligence with decreasing precision (Saridis [21],[22]). Such an intelligent machine is structured in three levels (Saridis [23]): the Organization level, the Coordination level and the Execution level. J. Albus [3] noticed that the structure of a hierarchical controller is similar to the structure of the brain functioning and that the hierarchy is generated as a result of "task decomposition". He outlines for the area of robotics the structures of brain functioning/ hierarchical control as three interacting hierarchies of task decomposition, world model and perception. Motivated by these developments A. Meystel [16] proposes a control architecture "Planner-Navigator-Pilot" for robots. He established the theoretical foundations of decision making in a class of control systems which allows for using nested representations. As a result nested hierarchies of multiresolutional (multiscale, multigranular) control structures are generated.

As an alternative to the above mentioned hierarchic architectures the subsumption approach, Brooks [6], avoids the use of an explicit world model and instead implements both primitive and complex behavior by a more direct coupling between perception and action. It appears to be better suited for reaction to unforeseen events, although less predictable in more routine circumstances. The central issue is the degree to which abstract concepts and symbolic world models are needed to obtain intelligent behavior. Brooks argues that the "world is its own best model" and that complex and apparently purposeful action can arise from the competition of layered behaviors according to a predetermined priority scheme without recourse to an internal world model. Behaviors at a higher level of abstraction are said to subsume those at a lower level. Tests of layered control for AUV's indicated that the complexity of the architecture increases significantly as the number of required behaviors increases. In order to overcome the performance sensitivity arising from interactions between actuating behaviors Bellingham [4] proposed a state configured form of layered control. This architecture adds a high level of control which is responsible for the activation of the right behaviors at the right time and with the right priority. This high level takes the form of a state table which determines the vehicle state by configuring the layered control structure.

Active investigation is taking place in order to define alternative system software architecture organizations. Many possibilities are available which might incorporate multiple intelligent agents, low-level behaviors, expert systems and blackboard paradigms. The incorporation of the blackboard paradigms in existing architectures represents a promising effort pursued by several researchers. Most of this research effort is inspired in the pioneering work of Hayes-Roth et al. [10], Lesser et al. [15] and Erman et al. [8].

Elfes [7] defined a Distributed Control Architecture for an Autonomous Mobile Robot responsible for scheduling and coordinating multiple concurrent activities. In this architecture Expert Modules communicate through messages and maintain globally relevant information in a blackboard.

Harmon [10] adopts a different approach by using a distributed blackboard system to coordinate the sensor, control and planning systems which constitute the vehicle management system of a Ground Surveillance Robot.

The latest approach developed by Ericksson [9] consists of a multiple KSAR trigger/KSAR execution running on a parallel processor and a dynamic control mechanism. This approach directly relates to ours since we are also interested in parallel processing techniques.

Researchers from SINTEF, Rodseth [17, 18], considered a mixed hierarchical-heterarchical architecture for a tethered underwater vehicle where state variables, representing aspects of the vehicle state are interconnected on a blackboard. However the operator specifies the plan in the form of a sequence of commands via a user interface and the rest of the system just has to follow them.

Honeywell, (see Kramer et al [14]), developed a simulation testbed for autonomous submersible research including an interactive off-board planning system and an autonomous on-board mission execution system. The information system includes sensor-processing and fusion algorithms organized on a blackboard centralized at the lower levels of the architecture.

A hierarchical architecture was developed at Heriot-Watt University, Russell et. al. [19], where the lower, the higher the frequency of control activities. Extensive recognition-oriented modeling is used and information is stored in a distributed knowledge-based blackboard system coordinated by a central kernel.

A remotely operated autonomous robot with a hierarchical structure was considered at LAAS, Alami et. al. [1], where a real-time operator system at the high levels of the architecture communicates through a low bandwidth link with the vehicle. Recognition-type modeling is performed by situation assessment routines on a blackboard system.

Researchers at Linköping University, Hultman et. al. [13], propose a very general three level hierarchical architecture where activities with a short response time are distributed at the lower level and the long response ones reside at the higher level. All information about the vehicle and the environment is stored in a blackboard accessible to any part of the architecture.

Albus [2] proposed a truly hierarchical structure where complexity, abstraction, and time scale of information increase with the hierarchical level. Knowledge of the past, present and projected future are used in each of the planning, modeling and information processing components of the hierarchy being the required information represented in formats adjusted for each level. Objects, relationships, features, regions, tasks and events are represented in a global blackboard and associated with a convenient frame.

VEHICLE MANAGEMENT SYSTEM ARCHITECTURE

We propose a VMS with a flexible structure which may reconfigure itself according to the class of events arising as the mission develops. Hierarchic flow of data and commands determines the vehicle's

behavior either in normal or exceptional situations. Horizontal communication occurs within some subsystems.

The hierarchy of the VMS consists of three levels as indicated in the following figure:

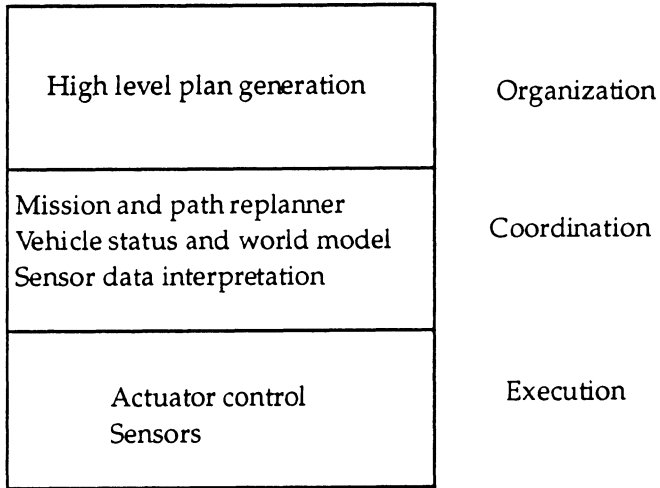


Fig. 1. Vehicle Management System Architecture.

Organization Level

It is the top level and takes place off-line. It receives mission specifications via a User Interface module. After mission interpretation and validation, this module organizes into a complex set of behaviors, so that it may be understood by subsystems at the coordination level.

Coordination Level

In this stage, decisions are taken in order to accomplish the set of behaviors established by the organization level. Of course, this stage must have a capability to assess the vehicle's and the mission state and define the required set of the actions. Typical decisions include: activation of vehicle subsystems, modification of the current set of behaviors required to achieve the mission's goals. On the assessment side, it is crucial to detect whether a mission behavior has been achieved with success or not, the admissibility of the next mission behavior or whether the vehicle is in correct working conditions. In particular there should be a Path and Task Replanning module in order to adequately modify the current mission if required. This level consists of the following conceptual modules (see figure 2).

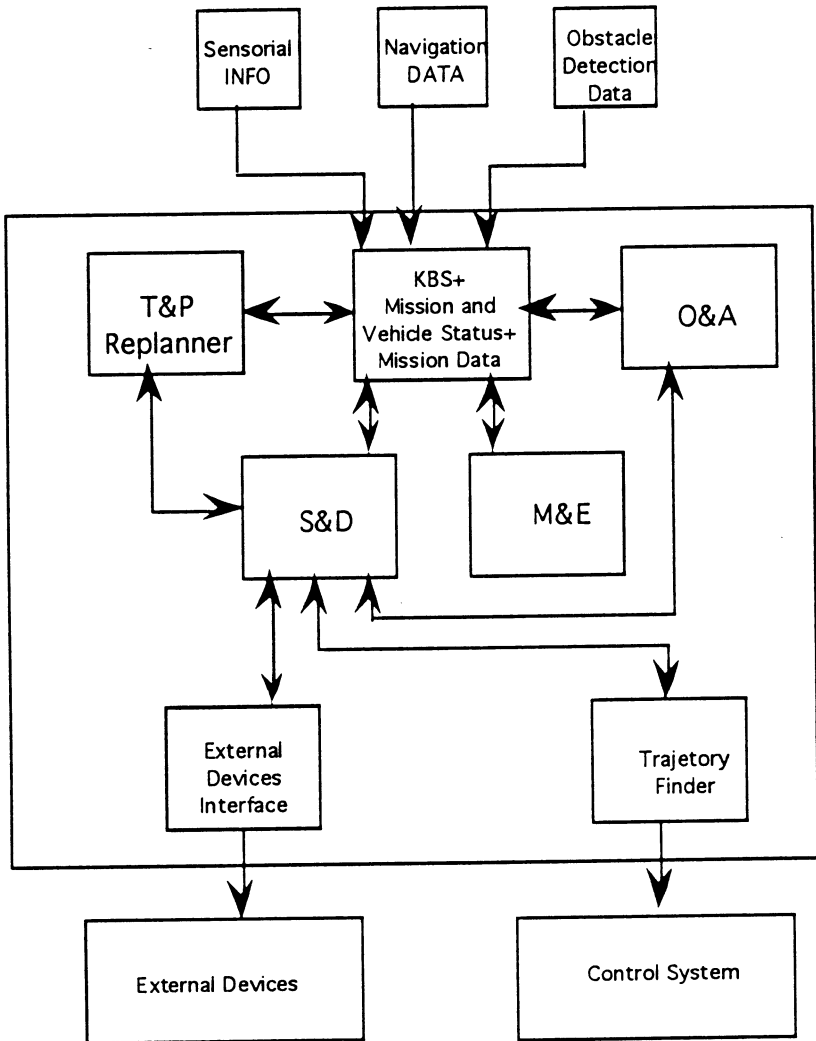


Fig. 2. Conceptual Scheme of the Coordination level.

Knowledge Based Systems The activities of the functional modules are supported by a knowledge based system. The following components should be included to accomplish this goal: world map, entities and respective attributes, internal and external commands, evaluation criteria, models of some entities and subsystems, plans, rules of specialized knowledge and information rules.

Monitoring & Evaluation (M&E) The monitoring function will address every subsystem at the functional level of the VMS (via the knowledge based system). Information concerning the current mission

plan and mission and vehicle status is used in order to evaluate situations or detect and diagnose causes of events. This data will be related by an organized subset of rules.

Scheduler & Dispatcher (S&D) This module is responsible for the scheduling and dispatching of all actions required to accomplish the desired behavior taking into account the output of the M&E, the current behavior and a set of the control rules. These actions take place at the appropriate functional modules which are activated upon the reception of messages sent by this module.

Path & Task Replanner (PTR) It is activated by the S&D whenever M&E detects a significant mismatch between the planned and actually occurred events. For example, significant path deviation is detected, blocked planned path that can not be tracked, etc. Whenever activated, this module will respond with the most adequate local plan to take the system from current unplanned situation to the desired one.

Trajectory Finder (TF) This module accepts a set of commands (a functional language) in order to compute a "local" path to be travelled during the next short elementary time interval and the associated velocity to be provided as a reference to the vehicle's controller. During this short elementary time interval the supervisory loop is considered open since there is no time to act via the coordination level, i.e. the vehicle is served on the controller reference.

Obstacle Avoidance (O&A) This functional module is activated whenever the M&E system detects the presence of an obstacle. This module is responsible for the generation of the most appropriate behaviors for obstacle avoidance thus guaranteeing the accomplishment of the specified goals.

External Devices Interface This functional module will be activated whenever the current or future behaviors require the use of some device such as loading or unloading mechanism, video camera, warning device, etc.. This module accepts a set of commands (a functional language) in order to independently generate the most adequate behavior of the required device.

Execution level

This level consists of actuators and sensors. At each given time a well defined specific reference behavior is produced by the corresponding coordinator and input to the actuator or sensor in order to generate the appropriate action. Obviously, this reference behavior is defined by taking into account the model of interaction between the external environment and sensor or actuator.

BLACKBOARD IMPLEMENTATION OF THE COORDINATION LEVEL

The development of Mission Planning and Execution modules for Autonomous Underwater Vehicles represents a challenging task since it involves the consideration of a set of simultaneous requirements of diverse nature in an isolated real-time environment. In order to accomplish the desired autonomous behavior in an uncertain environment the development of true intelligent like behavior is required.

To achieve intelligent behavior in a real-time environment special hardware and software architectures should be considered.

Although our work addresses the planning and execution of mission management systems this paper is specially focused on the execution of a given plan in the adequate real-time environment. Plan execution in an uncertain environment requires the consideration of replanning capabilities and imposes severe restrictions on the upper limit of reaction times. The available technology permits to achieve these goals in real-time when mission planning and execution are disjoint in time.

In what follows we are especially concerned with the implementation of the functions associated with the coordination level of the VMS architecture.

The diversity and complexity of the requirements (event-driven and goal-driven behavior response in real-time) posed on the coordination level makes a blackboard architecture an attractive one for real-time response. The proposed model of the coordination level involves the consideration of a blackboard architecture which is directly related to the pioneering work developed by Hayes [11]. We consider a domain blackboard for information sharing of the knowledge sources and a centralized control blackboard system where the knowledge sources are activated semantically. The approach of Hayes-Rooth [12] was extended by considering the parallel execution of the knowledge sources and the blackboard control on different processors. This development allows the incorporation of richer heuristics for the blackboard control. The proposed approach is directly related to the one presented by Ericksson [9] where a multiple KSAR trigger/KSAR execution parallel architecture is proposed.

From the vehicle point of view the structure of the blackboard is represented in the figure 3.

The blackboard structure data

It is organized into three main areas: the KBs, Mission and Vehicle Status and Mission Data. These are hierarchically organized into several levels of abstraction.

KBs - Includes objects defined by a set of attributes and relations between objects.

Mission and Vehicle status-Includes a set of pairs value-attributes.

Mission Data - Contains a set of rules of the problem and mission domain.

The knowledge sources

The knowledge sources of the domain problem are expert programs which may execute actions whenever a request is formulated. These requests are formulated in the functional language of the corresponding module. The blackboard is updated by the knowledge sources during their execution in order to schedule the appropriate tasks or behaviors.

We consider the following knowledge sources: obstacle avoidance, task and path replanner, trajectory finder and external device interfaces as indicated in the figure 3:

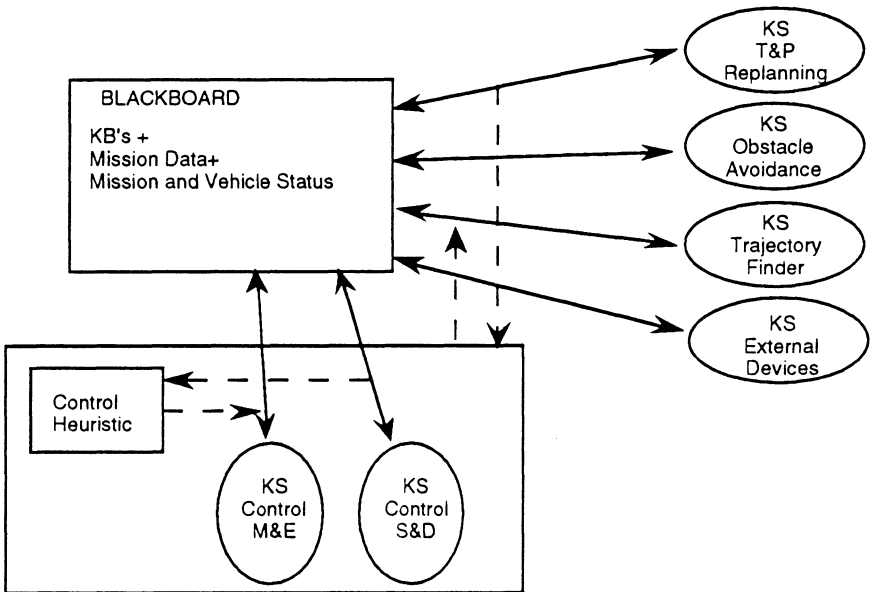


Fig. 3.Global Blackboard Structure.

Control Structure of the blackboard domain:

Activities of blackboard domain are coordinated by a blackboard control system. This blackboard control system is composed of a knowledge source, a coordinating control heuristic and a blackboard control data embedded in the global blackboard.

The knowledge sources of the control blackboard play the role of the M&E and S&D and are implemented by a set of rules and logical assertions.

In order to cope with the real-time requirements imposed by the asynchronous arrival of external events the control mechanism maintains a queue of events awaiting for processing.

The solution method is initiated by the selection of an event via a heuristic selection rule (such as priority and type event). The blackboard data is updated after this event selection takes place. Then the control mechanism selects all potential triggered control rules which are evaluated in the natural order defined by a context derived criterion. This evaluation may produce modifications in the blackboard data, which, in turn, may add other control rules to the list of potential control rules. This process maintains a list of potentially successful domain rules and goes on until the list of control rules is empty. Then, for a given heuristic mechanism, the control algorithm successively picks up and evaluates the domain rules until the corresponding list is empty, or the control is transferred to the first phase by a global heuristic. This last possibility may happen because of the arrival of an asynchronous event or the triggering of a high priority rule.

The above described software architecture was implemented in a multiprocessor environment with the following organization (figure 4).

A unique processor is responsible for the classical sequence of processing of the control blackboard: blackboard update, KS trigger processing and KS control execution. Simultaneously it picks up messages from the other processors, updates the blackboard and promotes the triggering of the KS domain.

The KS are implemented in the other processors. This hardware implementation of the software architecture envisages the fulfillment of real-time response requirements.

The functional modules are activated on the basis of phrase interpretation of a specific language. A language for rule description was specified through a very simple format whose main feature consists in having the specific functional languages as subsets.

AUV Process

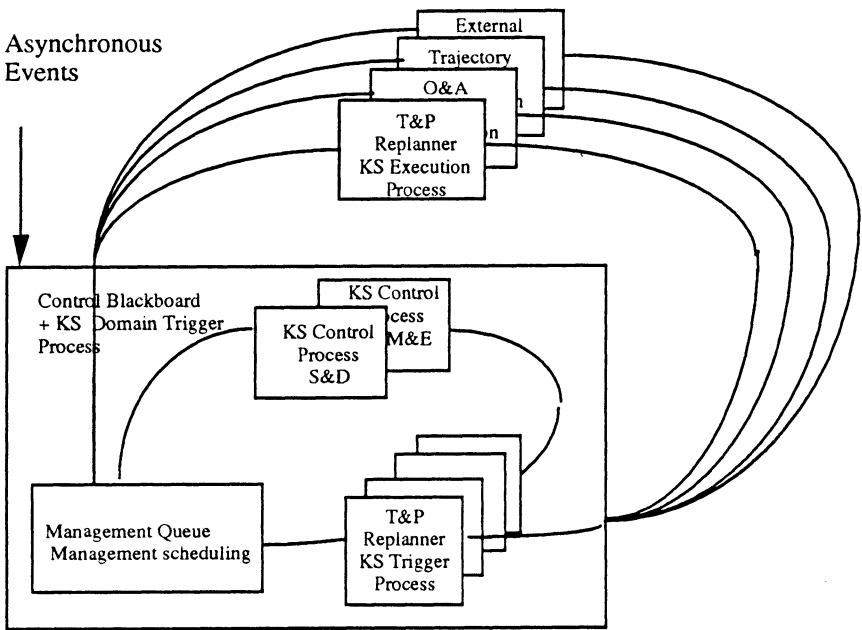


Fig. 4. Parallel KS Execution.

CONCLUSIONS

In this work we demonstrate the feasibility of the proposed architecture where a blackboard system is used at the heart of the coordinating system. Simulated missions show that the proposed architecture has the real-time capability to process data and activate a sophisticated set of behaviors oriented towards mission completion subject to safety constraints.

An important property of the proposed system consists in its modularity. Since this modularity is achieved through an integration of control and domain rules in the blackboard structure it permits an easy trade-off between efficiency/specificity and generality in the design of the coordinating system.

The arbitration mechanism allowing the composition of several activated behaviors enabled by the blackboard system endows the VMS with an extended reactive capability usually present in heterarchic structures. This permits to pick the most adequate reactive behavior in the optic of mission achievement subject to the vehicle's integrity. On the other hand, the intrinsically hierarchic nature of the

VMS guarantees the globally stable behavior, in the sense that, whenever feasible, mission gradually approaches its completion. The proposed blackboard based architecture endows the vehicle with a context-configured behavior by picking the most adequate compromise between goal and event driven behaviors.

Therefore, as an alternative to mission concept driven behavior (provided by truly hierarchic structures) and to event-state driven behavior (characteristic of subsumption architectures), we propose a mission context driven behavior where functional modules are activated so that survival actions are picked among those that approach mission completion if any.

There are several avenues through which this research work will proceed:

The blackboard control algorithm should be enhanced so that more efficient results in terms of mission achievement subject to vehicle's integrity are obtained.

Linguistic improvements and task organization procedures should be subject to continued research so that less requirements have to be imposed on the off line stage. Additionally this research effort will bring the possibility of incorporating more complex behavior description.

REFERENCES

1. Alami, R., Chatila, R. and Freeman, P. "Task Level Teleprogramming for Intervention Robots", *Proceedings of Mobile Robots for Subsea Environments*; International Advanced Robotics Program, Monterrey, CA, 1991.
2. Albus, J. "System Description and Design Architecture for Multiple Autonomous Undersea Vehicles", *NIST Technical Note 1251*, Washington, DC, Sept., 1988.
3. Albus, J. "Outline for a Theory of Intelligence", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.21, N. 3, pp. 473-509, 1991.
4. Bellingham, J. G., Consi, T. R., Beaton, R. M. and Hall W. "Keeping Layered Control Simple", pp. 3-6, *Proceedings of the Symposium on Autonomous Underwater Vehicle Technology*, Washington, 1990. IEEE Publication 90 CH2856-3, 1990.
5. Blidberg, D. R. and Chappell, S. "Guidance and Control Architecture for the EAVE Vehicle", *Autonomous Mobile Robots*, pp. 436-448, IEEE Computer Society Press, 1991.



6. Brooks, R. "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation*, Vol. RA-2, 1, March 1986.
7. Elfes, A. "A Distributed Control ARchitecture for an Autonomous Mobile Robot", *Artificial Intelligence*, Vol.1, N. 2, 1986.
8. Erman, L. D., Hayes-Roth, F., Lesser, V. R. and Reddy, D. R. "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty", *ACM Computing Survey* 12 (2), pp.212-253, 1980.
9. Erickson, W. K. and Baum, L. S. "Real-time Erasmus", *Proceedings of the Fifth Annual AAAI Workshop on Blackboard Systems held at the Ninth National Conference on Artificial Intelligence*, Anaheim, California, July 1991.
10. Harmon, S. Y. "The Ground Surveillance Robot (GSR): An Autonomous Vehicle Designed to Transit Unknown Terrain", *IEEE Journal of Robotics and Automation*, Vol. RA-3, N. 3, 1987.
11. Hayes-Roth, B., Hayes-Roth, F., Rosenschein, S. and Cammarata, S. "Modeling Planning as an Incremental Opportunistic Process", *Proceedings IJCAI 79*, pp. 375-83, Morgan Kaufman, San Mateo, California, 1979.
12. Hayes-Roth, B. "A Blackboard Architecture for Control", *Readings in Distributed Artificial Intelligence*, pp. 505-540, Morgan Kaufman, San Mateo, California, 1988.
13. Hultman, J., Nyberg, A., and Svensson, M. "A Software Architecture for Autonomous Systems", *Sixth International Symposium on Unmanned, Untethered Submersible Technology*, Durham, NH, 1989.
14. Kramer, A., Toms, D., Schrag, R. and Johnson, D. "Operational Planning and Programming of Autonomous Underwater Vehicle", *Sixth International Symposium on Unmanned, Untethered Submersible Technology*, Durham, NH, 1989.
15. Lesser, V. R. and Erman, L. D. "A Retrospective View of the Hearsay Architecture", *Proceedings IJCAI 77*, pp. 790-800, 1977
16. Meystel, A. "Nested Hierarchical Control", *An Introduction to Intelligent and Autonomous Control*, pp. 129-161, Kluwer Academic, 1992.
17. Rodseth, O. "Software Structure for a Simple Autonomous Underwater Vehicle", *Proceedings of the Symposium on AUV Technology*, Washington D.C., 1990.

18. Rodseth, O. "Object-Oriented Software System for AUV Control", *Proceedings of Mobile Robots for Subsea Environments*; International Advanced Robotics Program, Monterrey, CA, 1991.
19. Russell, G., and Dunbar, R. "Intelligent Control and Communication Systems for Autonomous Underwater Vehicles", *Proceedings of Mobile Robots for Subsea Environments*; International Advanced Robotics Program, Monterrey, CA, 1991.
20. Saridis, G.N. "Toward the Realization of Intelligent Controls", *IEEE Proceedings* 67, N. 8, 1979.
21. Saridis, G.N. "Foundations of the Theory of Intelligent Control", *Proceedings IEEE Workshop on Intelligent Control*., pp. 23-28, 1985.
22. Saridis, G.N. "Analytic Formulation of the Principle of Increasing Intelligence with Decreasing Precision for Intelligent Machines", *Automatica*, Vol. 25, N. 3, pp 461-46, 1989.
23. Saridis, G.N. "Architectures for Intelligent Machines", *Cirsse Rep. N. 9*., 1991.