Design of a Mission Management System for the Autonomous Underwater Vehicle MARIUS

Claire Bizingre[a], Paulo Oliveira[b], António Pascoal[b], Fernando Lobo Pereira[c],
Jean-Paul Pignon[a], Eduardo Silva[d], Carlos Silvestre[b], João Borges de Sousa[c]

[a]THOMSON SINTRA Activités Sous Marines, 1 Avenue Aristide Briand,
94117 Arcueil CEDEX, France

[b]Department of Electrical Engineering and Institute for Systems and Robotics (ISR),
Instituto Superior Técnico (IST), Av. Rovisco Pais 1, 1096 Lisboa CODEX, Portugal

[c]Department of Electrical Engineering and Institute for Systems and Robotics (ISR),
Faculdade de Engenharia da Universidade do Porto, Rua dos Bragas, 4099 Porto CODEX, Portugal

[d]Department of Electrical Engineering and Institute for Systems and Robotics (ISR),
Instituto Superior de Engenharia do Porto, Rua de S. Tomé, 4200 Porto Portugal

Abstract - **This paper describes the initial design phase of the software control architecture for the AUV MARIUS. This research effort is part of a long term project that aims to enable the prototype vehicle to carry out environmental surveying missions in coastal waters, in a completely autonomous mode. Throughout the project, MARIUS will be used as an advanced testbed to test developments in the areas of Mission Planning and real-time Mission Execution Systems for AUV's.**

I. INTRODUCTION

This paper describes the initial design phase of a software control architecture for the Autonomous Underwater Vehicle (AUV) MARIUS. The vehicle was designed to serve as an autonomous station for environmental surveying in coastal waters and future applications will include data collection, environmental inspection and surveillance in a fully autonomous mode. For a complete description of the vehicle's design and construction see [18] and the references therein.

The design of control architectures for intelligent machines (see [37], [38] and [41]) is a major challenge that involves tools from a variety of disciplines such as Artificial Intelligence, Control Systems Theory, Computer Science and Operations Research. The design problem is further complicated when it comes to the definition of control architectures for autonomous vehicles to operate in unstructured, partially unknown, and hostile environments. In this case, the most important design issues are mission reliability and safety. These requirements are directly linked to the adaptivity of the behavior to external and internal (in case of malfunctions) stimulae. In order to endow the AUV with this ability to adapt to external stimulae it is necessary to install in it a rich variety of sensors which, depending on the context, will permit the control software either to directly react to events or to identify and analyze the situations and subsequently take the most appropriate decisions. These considerations imply that the AUV control architecture will be a complex structure whose operational behavior depends not only on a set of plans defined by the operator but also on the unique set of circumstances arising during the course of the mission. These issues pose a major challenge in what concerns the design of the architecture and motivated the birth of several approaches. Although there is a strong interpenetration of the work of several research groups, a generally accepted classification distinguishes the following main tendencies or approaches: Purely reactive (see [11]), Hierarchical (typified by [12], [13], [20], [27], and [39]), and Hybrid (see [5], [6], and [16). The mentioned references are by no means exaustive and, for more details, see [9] or [10].

The remainder of this paper is organized as follows: In section 2, a typical mission scenario allowing the consideration of a set of requirements is described. After the presentation of the systems organization of MARIUS in section 3, key issues and main options on the design of the control architecture are discussed in section 4. In section 5, the various modules integrating the control architecture, including their interactions and linguistic organization, are described. This section also includes considerations concerning methodologies for analysis, development and modelling for validation. Finally, several conclusive remarks are presented.

II. A MISSION SCENARIO FOR MARIUS

This section describes one possible mission scenario for the AUV MARIUS. The envisioned mission takes place in the Atlantic, off the west coast of Portugal. It focuses on civilian applications, and addresses environmental problems that require non-traditional underwater surveying techniques. The mission was defined in the course of a multidisciplinary study that involved marine biologists and geologists. The

---

This work was supported in part by the Commission of the European Communities under contracts MAST-CT90-0059 and MAS2-CT92-0021 of the MAST-I and II Programmes and in part by JNICT under contract PMCT/C/TIT/702/90

0-7803-1808-0/94 $4.00 © 1994 IEEE

reader is referred to [7] and [33] for two other mission scenarios in the North Sea/Skagerrak - Kattegat Area of Denmark, and in the region of Ile D'Yeu - France, respectively. In their present form, these scenarios serve the dual purpose of:

- helping disseminate the concept that AUVs can play a major role in the future exploration, surveillance and rational exploitation of the ocean,
- setting realistic goals against which to compare the technical performance of the vehicle under development.

Mission Scenario - west coast of Portugal

Due to an increasing demand of sand and gravel for construction and the depletion of their land deposits, there is a strong need to extract those resources from deposits on the continental shelves (see [21]). The definition of an extraction policy requires a clear evaluation of the size of the deposits and of the environmental impact of the extraction process. In particular, it is fundamental to answer the following questions:

- Can one develop efficient and economically feasible methods to assess the location, distribution and quality of the deposits?
- To what extent can the negative impact of the extraction process on the environment be minimized?

As a first step to study these issues, the Marine Geology Department of the Geological Survey of Portugal (Instituto Geológico e Mineiro de Portugal) conducted reconnaissance surveys in 1980 and 1981 to evaluate the sand and gravel deposit potential of the Portuguese continental shelf. The results reported in [1] and [2] describe the experimental methods used to collect data and contain a wealth of information on the location and quality of potential exploitation sites. This survey was carried out during repeated cruises aboard two research vessels and involved the analysis of 500 samples taken along the Portuguese Atlantic coast.

At each inspection site, the position of the vessel was determined, bathimetric readings were taken and samples of the seabed were acquired for analysis. The maximum water depth was in the range of 150m. The distance from the inspection sites to the coast never exceeded 50km. On one of the cruises [2], samples were taken on a rectangular grid with the dimension of 6km x 18km and an approximate distance of 2km between inspection sites. Moreover, seismic analysis was performed along a continuous path located inside the rectangle in order to estimate the vertical extension of the deposits. According to geologists, this type of surveys is bound to provide only rough estimates of the extension and quality of the deposits. In fact, due to economic reasons the sampling grid is necessarily coarse. On the other hand, seismic analysis does not provide a direct measure of the quality of the deposits. This requires the taking of samples from the seabed, which slows down the surveying process considerably.

As suggested in [22], these problems can be partially solved by resorting to AUVs. The key point is the realization that it is possible to obtain good estimates of the size and distribution of gravel and of the percentage of carbonates on the seafloor surface without resorting to direct sampling. In fact, the size of gravel can be determined from video images of the seabed, the carbonate (i.e., shell) concentration can be estimated using dedicated sensors, and finally photometer readings complemented with video images can help distinguish the deposits with a high concentration of silicates (red or gray colour) from those where the carbonates abound (white colour). This motivated the design of the following mission for the AUV MARIUS, off the coast of Portugal:

- Performing a survey along a grid consisting of parallel lines traversed at a constant speed of 1 knot with respect to the seabed. Area to be inspected: 2kmx2km. Size of the grid: 100m. Maximum sea current: 1 knot.
- Recording video images as well as colour and carbonate sensor data at regular intervals on the site grid, in order to determine the horizontal extent of the deposits; performing accurate navigation and avoiding obstacles on the seabed.

In a future scenario, the AUV will be required to adjust the search pattern during the mission, determining and following the boundaries of the regions with higher concentrations of silica. The data acquired will help determine promising sites for future exploration and select specific sites for seismic probing. The two types of data are fundamental to obtain a 3-D image of the deposits.

It is important to stress that the extraction of materials from deposits on continental shelves may disturb the physical and biological balance of the environment. According to [19], the physical impact can be minimized by restricting the exploration to older relict sediments that are located farther away from the coast. At the present sea level, these sediments do not play an active role in the dynamics of the seabed. The biological impact, however, is of a very complex nature and manifests itself in many facets, as the following examples show.

- The extraction of raw materials may release high concentrations of clay that can remain suspended in the water for prolonged periods of time. The clay acts as a barrier against light penetration, thereby inhibiting the growth of phytoplankton that is essential to the equilibrium of the fauna.
- Potential extraction sites may harbour nesting grounds for the benthic fauna. It is important to study the reproductive cycles of the benthic communities throughout the year and to fully understand their evolution following excavation (e.g., how long does it take for a particular community to return and settle down after being temporarily dislodged?). The conclusions of this study could lead to a map with regions marked in red (no excavations allowed), yellow (excavations allowed at some times of the year) and green (excavations allowed all year long).

These tasks seem particularly suited for AUVs, which could perform the following functions:

- Traversing repeatedly the neighborhood of a potential deposit site following excavation and collecting data on the turbidity and oxygen content of the water - this will help determine the time constant associated with the rate of deposition of clay.
- Idem, but taking video images and stroboscopic images periodically around that site throughout the year, in order to determine the level of recuperation of the benthic fauna.

The mission scenario described and the ones reported in [7] and [33] have greatly impacted on the design of the MARIUS vehicle, which is described in the next section. The envisioned role for MARIUS as an operational vehicle for environmental surveillance, monitoring and inspection permits to specify a set of requirements that will guide the design of the control architecture.

This set of requirements should include, on the one hand, autonomy, flexibility and mission reliability through intelligent decision capability, and, on the other hand, safety through adequate reactive mechanisms. A delicate balance between these partially conflicting requirements must be found and makes architectural options the object of careful consideration.

Other requirements such as fault-tolerance (achieved through partial redundancy), error recovery capabilities, testability and behavior predictability play an important role to ensure the practical meaning of such a vehicle. A friendly user interface (amenable to mission definition by mission specialists which are not be experts in computer science), the possibility of some exchanging of information and commands (albeit a limited one) between the vehicle and the operator, and the ability to carry a payload composed of sensors and actuators which, during the course of the mission system, may interfere with the motion of the vehicle are fundamental requirements for the success of an operational vehicle.

The proposed missions scenarios impose some added features to the above requirements, such as the use of sensor guided navigation algorithms, mission dependent dynamic controllers, simultaneous navigation algorithms including conflict arbitration, and mechanisms allowing to perform missions guided by the on-board sensors carried as payload.

Finally, given the youth of this field and the fast evolution of the range of applications and available technologies, it is desirable to build a modular and expandable control architecture which, besides the replacement of current algorithms by more powerful and adequate ones, should permit the future incorporation of MARIUS in more complex systems where it might be necessary to cooperate with other vehicles, to perform more sophisticated tasks, or interact more strongly with human beings (e.g., assisting divers).

III. THE MARIUS VEHICLE. SYSTEMS ORGANIZATION.

This section contains a short description of the MARIUS vehicle and of its functional organization. The reader is referred to [17] and [18] for complete details and to [29] and [30] for earlier work.

The vehicle is 4.5m long, 1.1m wide and 0.6m high. It is equipped with two main back thrusters for cruising, four tunnel thrusters for hovering, and rudders, elevator and ailerons with hinged flaps for precise vehicle steering. The vehicle has a dry weight of 1400 kg, a payload capacity of 50 kg, and a maximum operating depth of 600m. Its maximum rated speed with respect to the water is 2.5m/s. At the speed of 1.26m/s, its expected mission duration and mission range are 18.4h and 82km, respectively.

The functional organization of the vehicle is depicted in Fig 1. The computer system for its implementation is described in [28]. Detailed information on the computer hardware, actuators and sensors can be found in [17], [28], and [40]. The following systems can be identified:

Vehicle Support Systems (VSS) - The Vehicle Support Systems control the distribution of energy to the remaining systems in the vehicle and monitor the energy consumption. Upon detection of an emergency situation (e.g. leak in a pressure tube), they force the vehicle to surface.

Propulsion System (PS) - The Propulsion System is responsible for controlling the speed of rotation of the propellers and the deflection of the ailerons, rudder and elevator. Set points are provided by the dynamic control system. Propulsion data (speed of rotation of the propellers, surface deflection angles and current supplied to the actuators) are fed back to the dynamic control system for closed loop control and to the Mission Management System for vehicle status assessment. (See [23].)

Navigation System (NS) - The Navigation System provides estimates of the linear position and attitude of the vehicle and corresponding speeds. It merges information provided by a Long Baseline Positioning System and a Motion Sensors Integration System. The motion sensor package consists of 2 pendulums, 3 accelerometers, 3 rate gyros, 1 gyrocompass, 1 depth cell, 1 echo sounder and 1 paddle wheel. A Doppler Sonar will be installed in the near future. The outputs of the navigation system are fed back to the Guidance and Control System and sent to the Mission Management System for performance assessment. (Refer to [3] for more details.)

Obstacle Detection System (ODS) - The Obstacle Detection System is responsible for detecting and providing a geometrical characterization of the obstacles that lay ahead of the vehicle. This system provides high level descriptions of

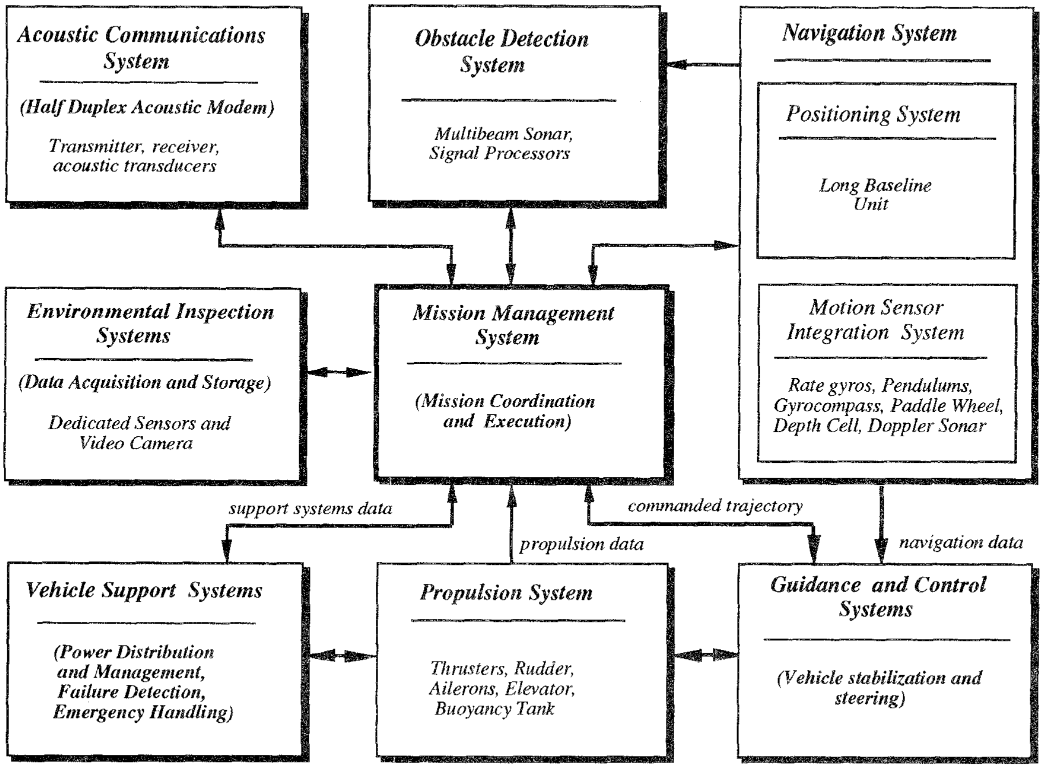

Fig. 1. The Functional Organization of MARIUS

the obstacles to the Mission Management System, which is responsible for modifying the vehicle's reference trajectory.

Guidance and Control System (GCS) - The Guidance and Control System accepts as inputs the reference trajectories issued by the Mission Management System and the navigational data provided by the Navigation System. It outputs commands to the propulsion system (set points for the speed of rotation of the propellers and deflection of the surfaces) so that the vehicle will achieve robust, precise trajectory following in the presence of shifting sea currents and vehicle parameter uncertainty.

Environmental Inspection System (EIS) - The Environmental Inspection System consists of a suite of environmental sensors to measure conductivity, temperature, pressure, turbidity, fluorescence, oxygen and pH. A video camera is incorporated to provide close-up images of the seabed. Data acquisition is triggered by the Mission Management System. Selected data is stored for post-mission analysis.

Acoustic Communications System (ACS) - The Acoustic Communications System is a bilateral digital acoustic link that is used by the operator to send new mission directives to the Mission Management System, and by the vehicle to relay back information regarding its own status. Typically, short messages are sent across the acoustic channel, such as sensor readings, mission definition commands and malfunction warnings. (For details, see [3] and [4].)

Mission Management System (MMS) - The Mission Management System is in charge of guaranteeing the integrity of the vehicle . Based on a mission plan provided by the operator, the MMS coordinates and executes the tasks that are necessary to execute that mission.

IV. KEY ISSUES ON THE CONTROL ARCHITECTURE DESIGN

The lack of a well established methodology for the design of control systems for complex objects such as MARIUS, implies the need to consider some fundamental architectural options based on the set of proposed requirements. These organizational options are of diverse nature covering the arrangement of subsystems into hierarchies, the specification of linguistic structures, and the definition of mission synthesis and error handling and recovery mechanisms.

A. Hierarchy

The requirements mentioned in the previous section suggested us the selection of a hierarchic (possibly distributed) control structure endowed with strong reactive capabilities (see [24], [25] and [31]). We consider three levels, Organization, Coordination and Functional Layer, subordinated to the Principle of Increasing Precision, Decreasing Intelligence as stated in [37], [38] and [40].

This option corresponds to the need to have well defined planning and control mechanisms which are at the heart of hierarchical approaches [13]. In these systems the planning module predetermines the actions to take in order to achieve a certain goal based on some assumptions on the real environment. The control of the execution is an entirely decoupled system.

The existence of an independent planning system requires a representation of the world which embodies the knowledge that, together with some planning guidelines (mission doctrine), is required by the planner to produce the most appropriate plan. This plan will constitute the operational reference and will provide options for execution.

The highly unstructured environment of the considered mission scenarios, where unexpected events may occur with a high probability, implies the need to include alternative mission subplans or execution modalities whose selection will take place in real time. Simultaneously, it is necessary to endow the vehicle with reactive behaviors that will ensure its safety and do not represent planned actions but functionalities that are activated under certain conditions.

B. Linguistic Structure

The hierarchic structure will rely on the definition, for each level, of independent linguistic units (see [36]). This allows the definition of a module structure, where each module corresponds to a syntactic unit of the language used, independent of the linguistic structure of the system. Decomposability, composability and protection are the main design criteria fulfilled by this approach.

Additionally, the following goals are accomplished: ease of mission specification by using a high level language; parallel and independent module design and programming; ease of system expansion by extending the vocabulary of the corresponding level and defining additional translation procedures; introduction of future enhancements or new sensors and actuators due to the decoupling of the various hierarchic levels; overall coordination achieved via language translation; and use of grammars amenable to plan generation.

C. Mission Synthesis

Full advantage of the system capabilities is obtained by designing each mission as the composition of primitive functionalities which are associated with the hardware and software resources of MARIUS.

Since most of the missions of MARIUS correspond to the composition of primitive functionalities involving sensing and control, maximum flexibility can be achieved if the high level capabilities offered by mission specification language correspond to the composition of primitive functionalities.

A library of functionalities whose combination permit to obtain more complex behaviors can be built to support the mission specialist in optimizing the use of MARIUS resources.

D. Error Handling and Recovery

The highly unstructured environment of MARIUS and the strong requirements concerning safety and mission reliability imply the need of carefully designed error handling and recovery procedures (see [25] and [31]).

Based on continuous knowledge feedback regarding the current state, flexible recovery options and logic flow control should be provided by the software in order to cope with the difficult error recovery due to the interaction with the real world.

An additional parallel hierarchy of error handling devices should be designed in such a way that a global error handler takes care of errors which were not solved by the corresponding module error handling device. By considering a list of contingency modes of operation and by using some redundancy achieved by partial functional overlapping of computational resources and different sensors and actuators, it is possible to preserve some operating capabilities in the presence of malfunctions or failures.

V. GUIDELINES FOR DESIGN AND IMPLEMENTATION OF A CONTROL ARCHITECTURE.

The design phase is based on an object oriented methodology (see [15] and [26] for general details, and [14] for application to mobile robot control). Besides being extremely helpful in dealing with complexity, this methodology allows the streamlining of analysis, synthesis and programming by adopting an uniform representation. By analyzing the class of missions to be performed by MARIUS and the set of requirements to be met subject to the existing hardware and software constraints, a set of objects has to be specified through the definition of associated static, dynamic and functional models. While the first kind of models describes the various structures of the system by introducing classes of objects and relations between them, dynamic models pertain to all the temporal aspects including the associated events and states, and the functional models specify how outputs are generated from the inputs for all the available services.

In the next subsection, it will be briefly describe how the various modules in the three hierarchic levels of the control software architecture interact. Corresponding to these three levels, three main components of the global MARIUS System may be distinguished (see [35]), where the first two are centered around an user friendly Man-Machine Interface (MMI):

- Mission Preparation System (MPS), being in charge of the mission validation and generation (at the Organization level),
- Mission Teleoperation System (MTS), allowing the enduser to permanently assess the vehicle situation and

mission achievement, and to intervene in the course of the mission at the Organization level), and

* Mission Execution System (MES), installed onboard the vehicle and being in charge of the mission execution and vehicle integrity (at the Coordination and Functional Level).

In Fig. 2 we show the control architecture which is described in subsections A, B and C. The contents of these subsections appear with more detail in [8], [25] and [31]. In [32], special emphasis is given to the coordinating structure.

A. Organization Level

Once specified in a high level language via the user interface, the mission has to be conveniently organized before being loaded into the vehicle's computer for execution. This means that the mission specific data provided by the user, together a set of more generic data concerning characteristics of the vehicle and the environment, and implicit behavior specifications have to be combined to produce a coherent feasible Mission Plan.

This Mission Plan, consists of a valid mission description which will guide execution at the coordination level. It is built by the Plan Generator by taking into account the mission specification and the mission doctrine in a form amenable to transmission by the low bandwidth acoustic communication channel.

The mission plan consists of a set of commands specifying the various activities of the vehicle related to motion, safety, communications and payload. The motion component will require the definition of: set of waypoints (from mission data), trunk travel attitudes (specified from both mission data and doctrine), reduced capacity strategies, monitoring rules and execution rules (the last three arising from mission doctrine).

Defined by the system manager, the Mission Doctrine consists of a set of scripts specifying automatic and implicit operational tactics. The need of this module arises from the use of a high level user interface language for the mission specification by the operator. This doctrine will include trunk travel attitudes, reduced capacity strategies, monitoring rules, execution rules, replanning strategies and temporal constraints.

The Plan Generator takes inputs from the mission doctrine and mission specification. Its role consists in:

a) Validating the mission specification;
b) Checking conflicts with the mission doctrine; and
c) Building the mission plan.

This last functionality makes use of a general planner which, given the importance of motion control in MARIUS missions, is supported by a Path Planner. This specific module provides the path for the plan generator.

Since the concepts of teleoperation and task level programming, allowing the interruption of the mission, the suppression or modification of segments of the mission, and the monitoring of relevant variables will be considered in this project, it is necessary to include the Mission Supervision module which is responsible for coordinating these activities. This module plays an important role not only during the development stage but also for future uses of MARIUS as a testbed facility.

B. Coordination Level (of the Mission Execution)

At this level, a global coordinator schedules and dispatches commands whose execution will contribute to the achievement of the set of goals specified in the mission plan. Given the uncertainty involved and the complexity of the tasks, the coordination is only possible with careful monitoring and evaluation of the effects of actuators and the status of the vehicle. If these do not agree with what should be expected in the plan, then appropriate action should be taken at this level. The considered on-board systems require the global coordination of the following specialized coordinators: motion (including navigation, guidance and control), communications (synchronous or asynchronous with all the systems external to the vehicle), payload (data and sample collection), and vehicle support system (monitoring all internal and external variables relevant to the safety of the vehicle and as constraints to the accomplishment of the mission). During the execution of the mission, the global coordinator not only dispatches commands to the pertinent coordinators, but also resolves potential conflicts and establishes cooperation among local coordinators whenever required to accomplish the mission. The activities of coordination involve the interaction of the following modules:

Plan Interpreter

The global coordinator decomposes the current segment of the mission plan into a set of subplans and sends them to the corresponding local coordinators. By taking into account the vehicle and environment status, this module generates, at each local coordinator, a detailed plan expressed in the coordinator language. This will serve as a reference for the plan supervision.

Plan Supervisor

The plan supervisor is in charge of monitoring the execution of the tasks involved in the plan. It checks whether the actual execution of tasks produces indeed the anticipated effects, and provides the adequate reaction in case of significant discrepancy.

Execution will be globally managed by this supervisor that schedules tasks, sending the associated orders to the corresponding task coordinator that controls their execution, possibly after some refinement.

In order to allow the plan supervisor to monitor task execution and to act on them while they are executed, each task is modeled as a finite state automaton (FSA). The FSA associated to a task not only models task execution but also nonnominal situations as well as the different actions which should be taken by the plan supervisor.

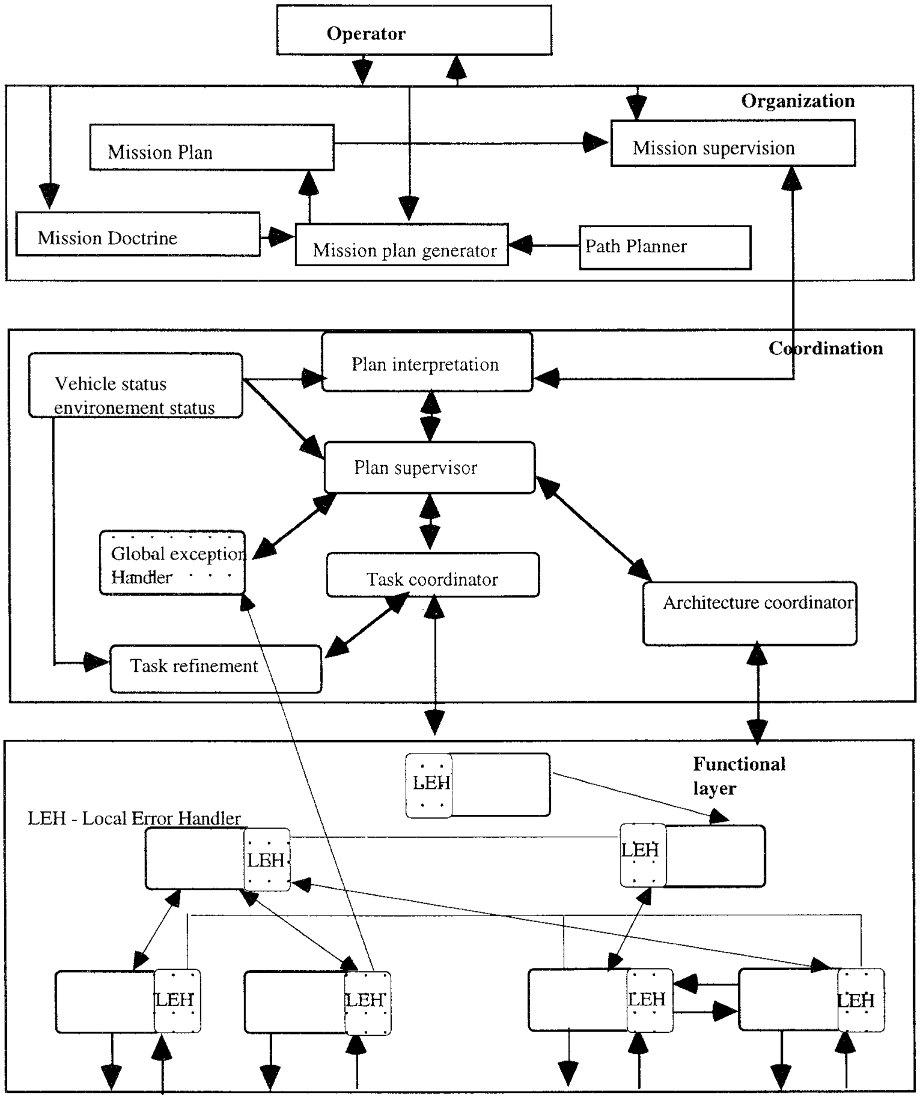

Fig. 2 Proposed Control Architecture

Task Coordinator

This is an intrinsically distributed module since there is a task coordinator for each task type. The task coordination activity consists of:

a) Managing task refinement;
b) Checking pre-conditions and co-conditions required for the execution of each action;
c) Activating the relevant functional modules to execute the actions resulting from task decomposition; and
d) Sending typed events to the plan supervisor.

The plan supervisor may use an FSA in order to model the execution of each task and to interact with its execution.

Task Refiner

Task refinement transforms a task into specific actions that are adapted to the actual context. The various modes correspond to the execution of different scripts. Script selection is based on testing conditions using acquired data. Scripts have variables as arguments that are instantiated at execution time.

This module should take into account the real world and certain pre-conditions not explicit in the operators. Task refinement requires the decomposition of tasks into primitive actions which takes into consideration information about the world and the available capabilities. Furthermore, it should be powerful enough in order to permit the management of tasks which have a decomposition into non-deterministic actions. There are three cases where the planning action must be completed by an instance at the moment of execution:

a) Choice between alternative actions which produce the same modifications in the world.
b) Possibility to assign an uncertainty to an action.

c) Reactive capabilities before a world not entirely known.

Architecture Coordinator

This module maintains a representation of the configuration of the functional layer. It is also responsible for the configuration of the communication channels of the functional layer. The current configuration is used by the plan supervisor in order to achieve the mission execution.

Vehicle and Environmental Status

The physical and logical status of the vehicle and a world model are maintained by this module. This data will be required for the plan interpretation, plan supervision, global exception handler and the task refinement.

Global Exception Handler

The exception handling system is responsible for the generation of error recovery procedures by classifying error patterns, providing a diagnosis to the plan supervisor and invoking contingency operation modes. It has a hierarchic structure in the sense that each module has its own means of detecting, diagnosing and recovering from errors and anomalous situations. If the nature of the error is such that it can not be dealt with locally (because, either the required detection and diagnosis data is of a global character or recovery procedures involve activities of several modules), then the methods of this module are invoked.

C. Functional Level (of the Mission Execution)

As in [13] we consider the functional layer composed of two types of functional modules: primitive and complex.

Primitive functional modules

These are responsible for direct sensing and actuation. The primitive modules may be divided into two classes:

a) actuator interface.
b) sensor interface.

They operate at the time scale of the corresponding actuators or sensors and reactivity in the most immediate sense should come from these functional modules. In order to provide a diversity of reactive behaviors, these functional modules should correspond to the most primitive actions in terms of sensing and actuation.

The connections between these modules are defined by the coordinator level where the architecture coordinator module keeps the state to which the current structure corresponds to. Different types of reactive behaviors can be achieved by considering different states of the architecture. These behaviors occur with no intervention from the coordination level which only knows what might occur. These links between the modules are also characterized by their rate and synchronism. This structure provides consistency for the system operation. Specifically, if all actuator or sensor commands are only invoked at the primitive modules, the existence of conflicts is prevented by the coordination level which permits to avoid logic conflicts.

Complex functional modules

These modules represent a way for achieving more complex behaviors and their invocation should correspond to syntactic units of the language of the coordination level. By isolating the more primitive functionalities, flexibility for future definition of new complex functional modules is preserved. To add more complex behaviors to this layer it is only necessary to augment the vocabulary of the coordination level and to extend the control logic.

The way these modules are invoked depends on the configuration of the connections between the primitive functional modules, i.e., the state maintained by the architecture coordinator. Here, a discrete-event formalism is the more adequate to describe state evolution. The consideration of a layer of complex functional modules intends to accomplish the following goals:

a) Modularity.
b) Expandability.
c) Architecture modification.
d) Definition of complex behaviors characterized by the use of services from either primitive modules or other complex modules.
e) Simplification of tasks at the coordination and organization levels. The basic idea is to define more complex behaviors by composing the basic functionalities.

D. Modeling, Simulation and Validation

In order to simulate and validate the Mission Management System of MARIUS, the various entities of the Mission Execution System are being modeled as a Multi-Agent structure within a Distributed Artificial Intelligence framework. The progress in this direction is described in detail in [34] and [35].

This approach has the advantage of permitting to use available generic tools. The agents constituting the system share a common generic structure integrating the following components:

* Supervisor - It receives the plan to be executed from the upper hierarchic level, asks the Refiner to instantiate it, and organizes the global surveillance tasks. It has the capability to define reactions and it passes the control to the upper level whenever unable to handle the situation.
* Refiner - When activated by the Supervisor, it receives orders and produces a script fully specifying procedures tailored to the current context.
* Executive - It schedules and dispatches procedures sent by the Supervisor by activating the required functional modules. It activates the specific surveillances and, whenever an event occurs, it activates the diagnosis Manager module.
* Surveillance Manager - It manages all monitoring activities associated with the hardware and software of

the vehicle and, besides informing the Supervisor, it may activate some reflexive activity.

- Diagnosis - It is activated by the Supervisor, whenever an error occurs. It proposes some corrective action resulting from analysis.

The Multi-Agent model specifies the interactions among the various agents in such a way that the set of functionalities of the Mission Execution System described in the previous subsections satisfies the needed requirements. In this framework, a Mission Command and Control Agent will play the role of the global coordinator and will be supported by the following assistant command and control agents: Navigation, Communications, Payload and Vehicle whose specializations correspond to those of the above mentioned local coordinators.

These specialized agents will act independently but in a coordinated fashion so that behavior predictability will be ensured. On the other hand, the decentralized concept behind the implementation of this structure will provide high reliability, strong extendibility and will contribute for a good management of complexity by localizing specific knowledge.

VI CONCLUSIONS

This paper presents a design approach for the control software of the AUV MARIUS. Although the presented framework is built on a field rich of ideas, much work remains to be done in order to demonstrate the fulfilment of the stated requirements. Careful consideration of developments along a number of directions may contribute not only for the improvement of the quality and success of the design, but also to extend the range of missions to be executed by MARIUS so that future more sophisticated ones may be encompassed. These directions of development include:

- Development of an environment amenable to a better and more extensive intervention of mission specialists with no special background in computer science. The proposed architecture should also allow the incorporation of future support of teleoperation by a more sophisticated operator-vehicle interface.
- Enhanced error and failure detection and recovery should be better embedded in the whole architecture.
- Development of a simulation environment allowing the incorporation of hardware in the loop permitting to fully test the proposed architecture and its future refinements.

ACKNOWLEDGMENT

The authors would like to thank Hipólito Monteiro for their contribution to Section 2.

REFERENCES

[1] J. Alveirinho Dias, J. Hipólito Monteiro and L. C. Gaspar, "Gravel and Sand Deposits on the Continental Shelf of Portugal", (in Portuguese), Comun. Serv. Geol. Portugal, t. 66, 1980, pp. 227-240.

[2] J. Alveirinho Dias, J. Hipólito Monteiro and L. C. Gaspar, "Gravel and Sand Deposits Southwest of Peniche,", (in portuguese), Comun. Serv. Geol. Portugal, t.67, fasc. 1, 1981, pp. 65-83.

[3] G. Ayela and J. Michel Coudeville, "The Navigator and the Communication System for MARIUS", MARIUS-Technical Report, ORCA Instrumentation, August 1993.

[4] V. Barroso, P. Geada and João Gomes, "Real Time DSP Based Underwater Communications", Proceedings of the 1993 International Conference on Signal Processing Applications and Technologies, pp. 1584 - 1593, Santa Clara, California, USA, September 1993.

[5] J. Bellingham, T. Consi, R. Beaton, and W. Hall, "Keeping Layered Control Simple", pp. 3-6, Procs of the Symp. on Autonomous Underwater Vehicle Technology, Wash. DC, 1990, pp. 3-6.

[6] J. Bellingham and T. Consi, "State Configured Layered Control," Procs of Mobile Robots for Subsea Environments. International Advanced Robotics Program, Monterey CA, 1991, pp. 75-79.

[7] A. Bjerrum, "Kattegat Benthic Survey", MARIUS -Technical Report, COWIconsult, March 1993.

[8] J. Borges de Sousa, F. Lobo Pereira and E. Pereira da Silva, "A Dynamically Configurable Architecture for the Control of an AUV", FEUP and ISR-Porto Technical Report, October 1993.

[9] J. Borges de Sousa, F. Lobo Pereira and E. Pereira da Silva, "Mission Management Systems for Autonomous Underwater Vehicles: A Survey", MAST-2-CT92-0021 Technical Report, ISR-Porto, October 1993.

[10] J. Borges de Sousa, F. Lobo Pereira and E. Pereira da Silva, "Software Architecture for Autonomous Mobile Vehicles: A Survey" to appear in Autonomous Underwater Vehicles, Kluwer Academic, 1994.

[11] R. Brooks, "A Robust Layered Control System for a Mobile Robot", IEEE J. of Robotics and Automation, Vol. 2, March 1986.

[12] R. Byrnes, D. MacPherson, S. Kwak, R. McGhee and M. Nelson, "An Experimental Comparison of Hierarchical and Subsumptiom Software Architecture for Control of an Autonomous Underwater Vehicle", IEEE Oceanic Eng. Soc. Symp. on AUVs, Wash. DC, June 2-3, 1992, pp. 135-141.

[13] R. Chatila, R. Alami, Degallaix and H. Haruelle, "Integrated Planning and Execution Control of Autonomous Robot Actions". Procs of the IEEE Intern. Conf. on Robotics and Automation, Nice, France, 1992.

[14] H. Chochon, "Object-oriented Design and Distributed Implementation of a Mobile Robot Control System", in Procs of the Workshop On Architectures for Intelligent Control Systems, IEEE Intern. Conf. on Robotics and Automation, Nice, France, 1992, p. 32-36.

[15] P. Coad and N. Jill, Object-Oriented Programming, Yourdon Press Computing Series, 1993.

[16] J. Connel, "SSS: A Hybrid Architecture Applied to Robot Navigation", Procs of the 1992 IEEE Intern. Conf. on

Robotics and Automation, Nice, France, 1992, pp. 2719-2724

[17] P. Egeskov and A. Bjerrum, "Vehicle Design", MARIUS-Technical Report, COWIconsult, February 1994.

[18] P.Egeskov, A.Bjerrum, A.Pascoal, C.Silvestre, C.Aage and L. Wagner Smitt, "Design, Construction and Hydrodynamic Testing of the AUV MARIUS", Proc. AUV 94, Cambridge, Mass., 1994.

[19] K. Emery, "Some Potential Mineral Resources of the Atlantic Continental Margin", U.S. Geol. Survey Professional Paper 525C, 1965, pp. C157--C160.

[20] Healey, A. J.,McGhee, R.B., Christi, R., Papoulias, F.A. , Kwak, S.H. Kanayama, Y. and Lee, Y., "Mission Planning, Execution and Data Analysis for the NPS AUV II Autonomous Underwater Vehicle", Procs of 1st IARP Workshop on Mobile Robots for Subsea Environments, Monterey, California, 1991, pp. 177-186.

[21] J. Hill, "Offshore Mineral Prospecting and Evaluation", Quarry Management and Products, 1978, pp. 51--56, July.

[22] J. Hipólito Monteiro, "AUVs for Geological Exploration", communication presented at the Workshop on Autonomous Underwater Vehicles, Porto, Portugal, September 1993.

[23] B. Jacobi and A. Bjerrum, "Propulsion Test Planning", MARIUS-Technical Report, COWIconsult, June 1992.

[24] F. Lobo Pereira, J. Borges de Sousa and E. Pereira da Silva, "System Analysis and Specification for a Mission Management System of the Marius Vehicle" to appear in Autonomous Underwater Vehicles, Kluwer Academic, 1994.

[25] F. Lobo Pereira, J. Borges de Sousa and E. Pereira da Silva, "A Study of a Control Architecture Underwater Vehicles", ISR-Porto, MAST-2-CT92-0021 Technical Report, October 1993.

[26] B. Meyer, Object-Oriented Software Construction, Prentice Hall, 1988.

[27] F. Noreils, Contrôle d'Exécution de Plans d'Actions et Architecture pour Robots Mobiles. Thèse de l'Université Paul Sabatier, Toulouse, France, Laboratoire d'Automatique et d'Analyse des Systèmes (CNRS), Novembre 1989.

[28] P. Oliveira, C. Silvestre and A. Pascoal, "Computer System", MARIUS-Technical Report, Institute for Systems and Robotics, January 1994.

[29] A. Pascoal, A. Bjerrum and J. Michel Coudeville, "MARIUS (Marine Utility System): An Autonomous Underwater Vehicle for Environmental Surveying", Proceedings of the Mast Days and Euromar Market, Commission of the European Communities, Brussels, Belgium, March 1993, pp. 746-758.

[30] A. Pascoal, F. Lobo Pereira, A. Bjerrum, K. Christiansen, J. Paul Pignon, G. Ayela and C. Petzelt, "Development of a Self-Organizing Underwater Vehicle", Procs of the Mast Days and Euromar Market, C. E. C., Brussels, Belgium, March 1993, pp. 760-769.

[31] E. Pereira da Silva, J. Borges de Sousa and F. Lobo Pereira, "On the Design of the Mission management System for an Autonomous Underwater Vehicle", to appear in Autonomous Underwater Vehicles, Kluwer Academic, 1994.

[32] E. Pereira da Silva, J. Borges de Sousa and F. Lobo Pereira, "A Configuration Kernel for a Dynamically Configurable Architecture for the Control of a Mobile Platform", Euriscon'94, Malaga, August 22-24, 1994.

[33] C. Petzelt, "Survey of the Coastal Waters of Ile D'Yeu, France by the SOUV MARIUS", SOUV-Technical Report, International Laboratory of Marine Biology, Ile D'Yeu, France, March 1993.

[34] J. Pignon and C. Bizingre, "Mission Management System Specification and Validation", THOMSON SINTRA-ASM, MAST-2-CT92-0021 Technical Report, October 1993.

[35] J. Pignon and C. Bizingre, "Spécification et Conception Préliminaires du Système de Contrôle du Robot MARIUS", THOMSON SINTRA-ASM, Working Document, March 1994.

[36] G. Saridis and J. Graham, "Linguistic decision schemata for intelligent robots," Automatica, vol. 20. 20, n° 1, pp. 12-126, 1984.

[37] G. Saridis and K. Valavanis, "Analytic design of intelligent machines", Automatica, Vol. 24, 1988, pp. 123-133.

[38] G. Saridis, "Analytic Formulation of the Principle of Increasing Intelligent with Decreasing Precision for Intelligent Machines," Automatica, Vol. 25, n°. 3, 1989, pp.461-467.

[39] G. Saridis, "Architecture for Intelligent Machines", Technical Report CIRSSE-58, Center for Intelligent Robotic Systems for Space Exploration (CIRSSE), Rensselaer Polytechnic Institute, 1991.

[40] C. Silvestre, P. Oliveira and A. Pascoal, "The Hardware and Software Architectures for the MARIUS Vehicle", SOUV-Technical Report, Institute for Systems and Robotics, October 1993.

[41] F. Wang, K. Kyriakopoulos, A. Tsolkas and G. Saridis, "A Petri-net for an Intelligent Mobile Robot," IEEE Trans. on Systems, Man and Cybernetics, Vol. 21, No. 4, July/August 1991, pp. 777-789.