SUBMARINE AUTOMATION: DEMONSTRATION #5

Hui-Min Huang
Mechanical Engineer
National Institute of Standards and Technology
Gaithersburg, Maryland, 20899
email: huang@cme.nist.gov

Keith Young
Captain, USN (Retired)
PdMA Corporation
Rockville, Maryland

Richard Quintero
Group Leader
National Institute of Standards and Technology
Gaithersburg, Maryland, 20899
email: quintero@cme.nist.gov

ABSTRACT

This paper describes the results obtained in the final performing period of the ARPA sponsored submarine automation project[1]. Efforts on the mapping between the submarine operational environment and the RCS software architecture lead to the result of three watch station graphic user interface panels. The submarine automation model has been expanded to include some engineering systems control capability. On the RCS generic structure, authors have explored methods to reuse existing plans for new task requirements. The authors have also illustrated a main feature in RCS, namely, a smooth transition of level of authority in commands from the higher to the lower control levels.

1.BACKGROUND AND PREVIOUS WORK

An earlier paper [Hu 93-1] describes the following: submarines are complex systems. Navigation, communication, hydrodynamic control, power, etc., are just a few among all the subsystems that need to be coordinated when submarines conduct missions. An enormous amount of information must be fused, organized, and communicated to support decision making in real-time. On today's submarines, most of these functions are performed by crew members in extremely tight space. It is, therefore, very desirable to have submarine operations automated.

The Intelligent Systems Division (ISD) of the National Institute of Standards and Technology (NIST) has been supporting the Advanced Research Projects Agency (ARPA) Maritime Systems Technology Office (MSTO) in investigating submarine automation. Our previous accomplishments include a demonstration of an automated maneuvering system for a 637 class nuclear submarine to perform under-ice transits in the Arctic region. The control system is capable of maneuvering the simulated submarine toward its intended destination while using simulated sonar data to avoid dangerous ice keels and to maintain the submarine's ordered depth. The control system can operate either autonomously or

[1] ARPA Order No. 7829, Amendment No. 02.

under human supervision. An operator is presented with all the important maneuvering data graphically in real-time. This development effort has been using a generic approach, called the NIST hierarchical Real-time Control System (RCS) reference model architecture and methodology, described later. Based on an incremental development approach, our results were presented in a series of demonstrations (Demo#1 through Demo#4) leading to the one reported here, Demo#5.

As stated in the earlier papers [Hu 93-1, -2, -3], the major objectives of this project are to:

* Demonstrate the application of the NIST RCS to submarine automation.
* Refine and document the RCS methodology.

This paper, together with the earlier papers, describe how we have accomplished these objectives.

2.SUMMARY OF RESULTS

This cycle of the submarine automation project emphasizes:

* Continuing investigating and developing the human computer interface (HCI).
* Expanding the submarine control system, the simulator, and the animator to include engineering supporting systems.
* Demonstrating reusing the existent automated maneuvering system software.
* Refining the RCS methodology [Qu 93].

These technical objectives are demonstrated by commanding the control system to perform the mission stated in the scenario descriptions, described in section 3 of this paper. Section 4 provides an overview of our generic hierarchical control software environment. This section describes that RCS facilitates software reuse, as new controller nodes are added to the existent control hierarchy established in the previous development cycles (figure 3). A brief comparison to some other related efforts is also given. Section 5 describes our system design, analysis, and implementation effort. Section 6 describes the execution of the demonstration mission through the use of graphical displays. Section 7 illustrates how RCS solves a complex problem by providing smooth transitions that map a complex and high level problem to physical system behavior. Section 8 discusses how RCS can perform software reconfiguration to meet more sophisticated mission requirements. Section 9 is a summary.

3.PROBLEM DOMAIN AND SCENARIO

The RCS methodology [Qu 92] calls for the development of a set of system operational scenarios based on the project technical objectives. The scenario descriptions are to be used to develop the control systems, operator interface, simulation, and animation, as described in the later sections.

3.1 Submarine Mechanical Systems

Figure 1 shows our submarine model. The propulsion system includes a main propulsion system and an emergency propulsion motor (EPM)[2]. The main propulsion system consists of two throttles, the ahead and the astern. They control the steam to rotate the two sets of propellers at the reversed directions. However, the astern throttle is used only: (1) during emergency deceleration of the forward motion of the ship. A submarine never maneuvers

[2] There is also a secondary propulsion system (SPM), called the outboard motor, which we do not model.

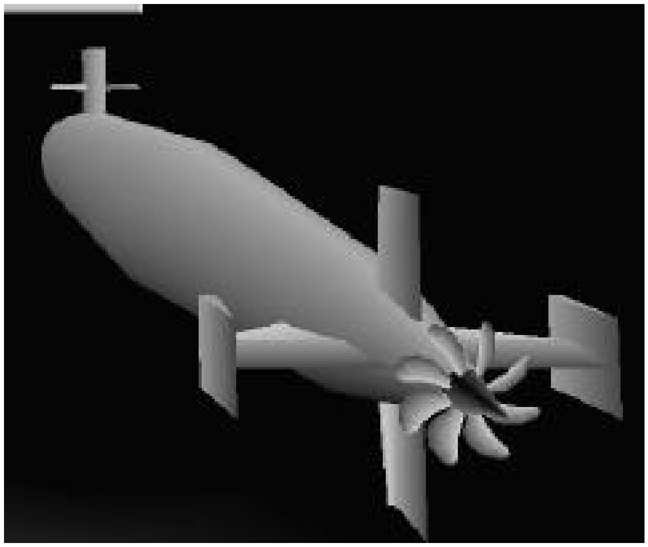


Figure 1: A Computer Model of a 637 Class Submarine

astern submerged; and (2) to help in maneuvering the ship to a dock when the ship is on the surface. The astern propeller set is omitted in our model since the scenarios involve neither of the astern operation conditions.

The sail planes are located on the sail structure at the front of the submarine model. The sail planes are primarily used for depth control. The stern planes are located at the rear of the submarine. They are used for the pitch and depth control. The rudder is located at the rear of the ship and is used for steering the ship left or right. The main and variable ballast tanks are distributed throughout the ship and are used to control the buoyancy of the ship and to adjust its bubble angles (pitch).

Demo#5 expands the previously developed control hierarchy by adding some operations of the ventilation system, which is a part of the engineering support system. The submarine is divided into several major compartments, including the Engine Room, the Auxiliary Machinery Room, the Reactor Compartment, the Operations Compartment, and the Bow Compartment. The ventilation system maintains adequate atmospheric conditions for both the personnel and equipment in all these compartments. Central to the system is a fan room which supplies either fresh or recirculated air through the ducts, hatches, dampers, valves, etc., distributed throughout the ship. The air can be dehumidified, heated, cooled, or purified as required to suit various ship operating conditions. Valves and damper positions can be reconfigured to adjust the air circulation paths and flow rates as the ship's operating condition changes, for example, surfaced, submerged, or a change in the compartments' atmosphere due to the outburst of a fire. Figure 2 shows a simplified ventilation schematic diagram, developed as a part of the operator interface for the demonstration.

3.2 Scenario

The scenarios describe how a submarine operates, currently manually. The objective of automation is to develop an RCS control system to either autonomously or via man-in-the-loop control schemes to operate the ship. The following is the demonstration scenario, described in submarine operational terminology:

A submarine is conducting a submerged transit of the open ocean at its standard speed (15 knots, or 7.7 m/s) and at a keel depth of 120 m. A watchstander[3] informs the Maneuvering Room on the sound powered phone circuit that there is fire in the lower level Engine Room. The fire is reported to be in the vicinity of the main lubrication (lube) oil pumps. The Engineering Officer of the Watch (EOOW) passes the word to the Officer of the Deck (OOD) in the Control Room on both the sound powered phones and the intercom announcing system.

[3] A submarine term, meaning a crew member who is assigned to a designated onboard location, which itself is called a watch station, to perform pre-specified duties.

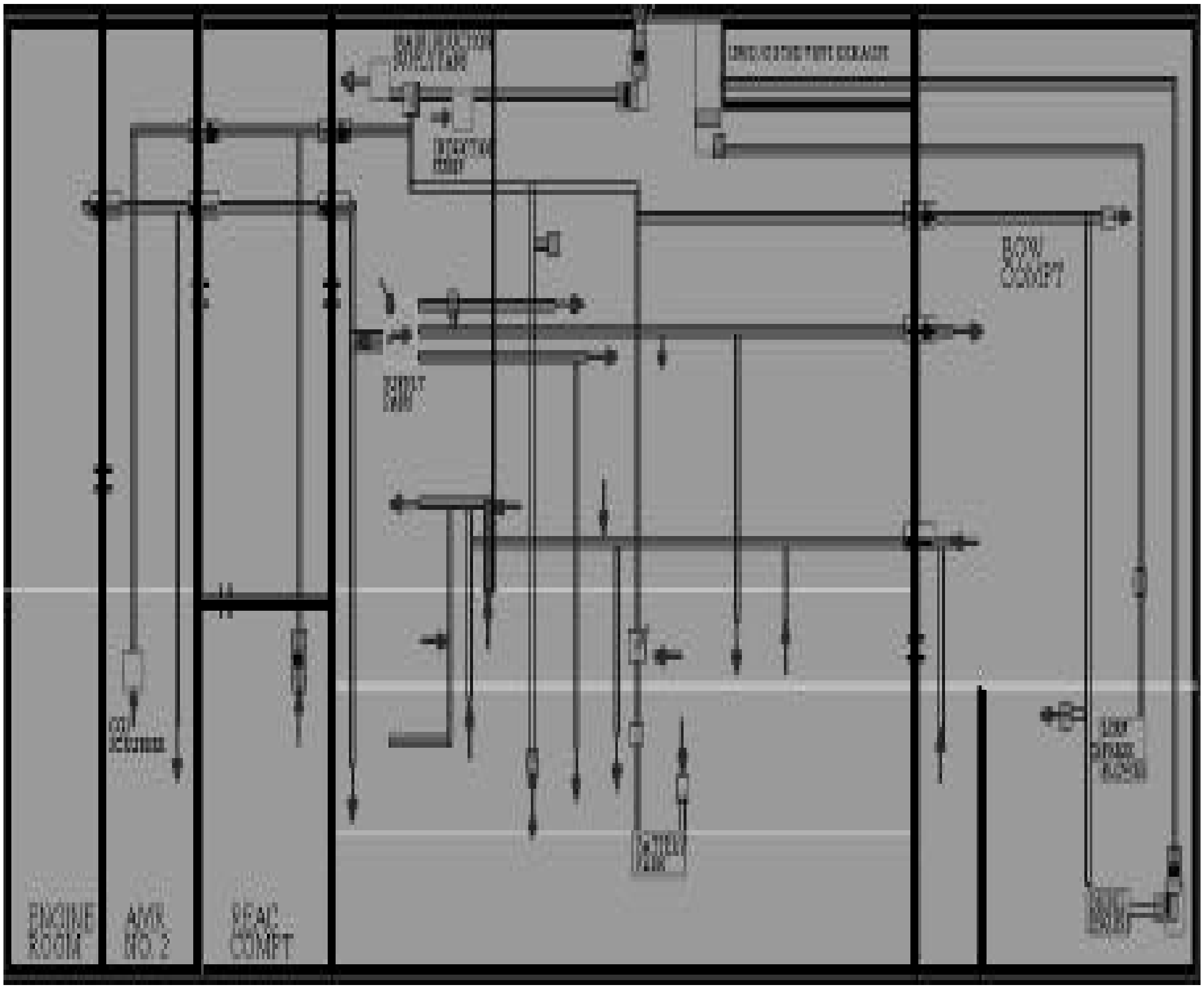


Figure 2: A Computer Model of a Simplified Ventilation Schematic Diagram

The OOD directs the Ballast Control Panel (BCP) operator to pass the word on the general announcing system (1MC) "Fire in the engine room. All hands on EABs (Emergency Air Breathing system masks)," and to sound the general alarm. The OOD completes the actions for coming to periscope depth:

Clearing baffles.
Checking for sonar contacts, close contacts.
Slowing and changing depth (Ahead one-third, keel depth 18 m).
Raising the periscope.

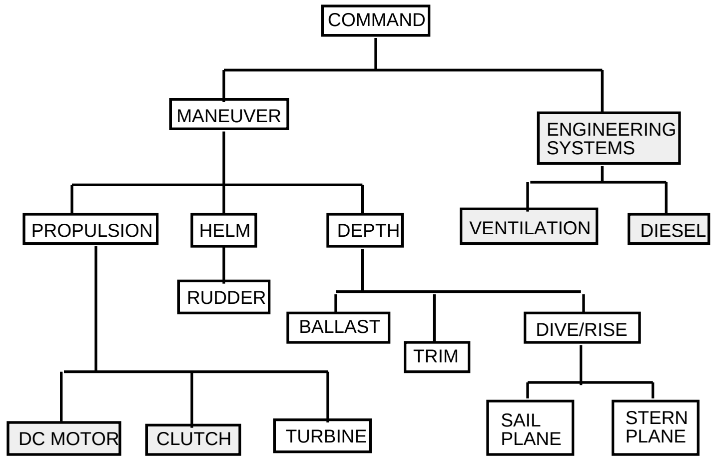
Upon hearing the general alarm the crew proceeds to their assigned general emergency (battle) stations, relief of the section watchstanders occurs.

The damage control party fights the fire in the engine room. On indication of decreasing main lubrication (lube) oil pressure the EOOW recommends to the OOD that propulsion be shifted to the EPM in order to secure the main lube oil to the propulsion turbines.

The Officer of the Deck (OOD) orders "All stop, shift propulsion to the EPM (emergency propulsion motor)." The shaft rotation is stopped and the clutch is used to disengage the shaft from the turbines and the EPM circuit breaker is closed. The Engineering Officer of the Watch (EOOW) reports to the OOD that he is prepared to answer bells on the EPM. The OOD orders "Ahead two thirds" which maintains enough speed for depth and steering control. The EPM operator operates the hand wheel to control the EPM and increase the motor speed to ahead two-thirds.

The damage control party reports to the Engineering Officer of the Watch (EOOW) that "The fire is out, the reflash watch is stationed." The EOOW relays this word to the Officer of the Deck (OOD). The OOD directs the word to be passed "Prepare to emergency ventilate the engine room with the diesel." The BCP selects the ventilation lineup setting it to emergency ventilate the engine room using the diesel engine. When the lineup is proper, the BCP operator reports to the OOD "Prepared to emergency ventilate the engine room with the diesel." The OOD directs "Commence snorkeling." The diesel engine is started and emergency ventilation of the engine room is commenced to remove the smoke and noxious gases from the engine room. The OOD directs that the atmosphere analyzer be used to sample the engine room atmosphere. The atmosphere sample shows that the carbon monoxide level in the engine room is 800 ppm. The Ballast Control Panel (BCP) operator uses the ventilation control panel to determine that with this level of carbon monoxide and ventilation configuration, it will take 80 minutes to reduce the CO level to an acceptable 5 ppm.

As the emergency ventilation of the engine room with the diesel continues, the atmosphere throughout the ship is checked in several locations. In areas where the atmosphere analyzer shows normal conditions, the Officer of the Deck (OOD) grants permission for the removal of Emergency Air Breathing system masks (EABs). When the atmosphere in the engine room reaches acceptable conditions the OOD will order "Secure emergency ventilation of the engine room with the diesel, recirculate." The BCP operator will use the ventilation control panel to line up for normal submerged ventilation. The OOD will order "Secure from General Emergency. Secure from fire in the engine room." The normal underway watch section will resume the watch. The diesel engine and generator will continue to be used to supply power for the emergency propulsion motor (EPM) until the main lube oil system is again ready to supply lubrication to the turbine bearings. When the main lube oil system is restored, the turbines are warmed up with steam, the EPM is ordered to "All stop" and then the clutch re-engages the turbine and the shaft. The EPM circuit breaker is



Note: The controllers in the shaded boxes are developed for the Demo#5 purposes.

Figure 3: Submarine Demo#5 Control Hierarchy

opened. Propulsion orders are again answered using the main engines and propulsion turbines.

4.ARCHITECTURE

The infrastructure that enables the development of the submarine automation model includes a generic reference model, methodology, and software structure. The advantages of using this infrastructure are: expediting system development, facilitating software reuse, and enhancing system integration. We also introduce some other efforts in this area in section 4.3.

4.1 Reference Model and Methodology

The control hierarchy of the submarine automation system is shown in figure 3. The command controller handles the highest level control, namely, the execution of the mission. Such control is achieved by assigning tasks to and coordinating the behavior of the two subordinates, the Maneuver and the Engineering Systems controllers. The tasks that these two subordinates execute are at a lower level of abstraction, at higher resolution, and at a higher level of detail. Similarly, these two controllers complete their tasks by:

* decomposing their tasks and assigning the resulting sub tasks to their subordinate controllers, propulsion, helm, and depth, and ventilation and diesel, respectively.

* coordinating the execution of the subordinate controllers.

As shown in figure 3, there are even lower level controllers. The lowest level contains actuator controllers. All the controllers perform under the same principle as described above. Functionally, each controller contains sensory processing (SP), world modeling (WM), value judgment (VJ), and behavior generation (BG) functions (figure 4). The SP function performs sensory data filtering and fusion. The WM function maintains the knowledge base. The VJ function computes scores and costs to facilitate planning and execution. The BG function contains a job assignor, a planner, and an executor. They plan and execute actions. These functions form a closed-loop for each controller and enable the controllers to act intelligently. In addition, these functions provide a systematic mechanism for the coordination among all the controllers within a hierarchy to achieve system goals. What has been briefly described here is the NIST hierarchical Real-time Control System (RCS) reference model architecture and methodology, which has been documented in many other papers, including [Al 92, Sz 92, Qu 92, Hu 91, Jo 91, Al 89].

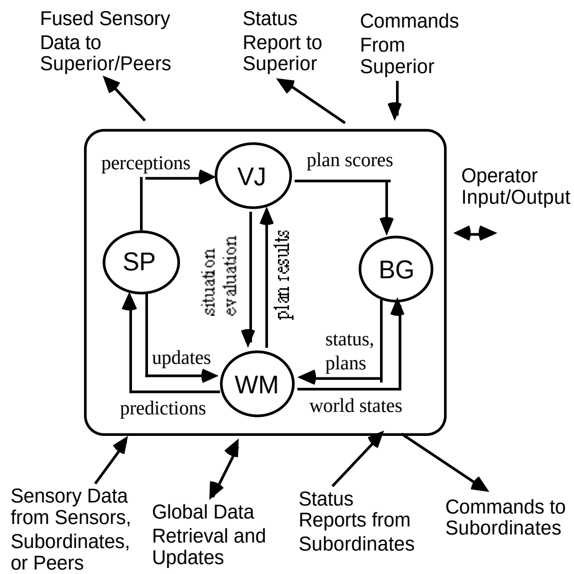


Figure 4: The Functional Model of an RCS Intelligent Controller Node

Some other efforts in the areas of control architectures and methodologies are introduced here as comparisons to RCS. Antsaklis [An 93] and Saridis [Sa 85] stated that intelligent system hierarchies typically consist of three levels: the execution level (EL), the coordination level (CL), and the management and organization level (MOL). EL is responsible for executing control functions. CL is responsible for short range decision making and learning. MOL is responsible for long range planning, decision making, and information management and handling. This concept is completely consistent with the level of abstraction concept in RCS. In general, MOL corresponds to the group level and up in RCS. CL corresponds to the task level through the prim or emove levels. EL corresponds to the servo level or up to the prim level in RCS. RCS, in addition, provides a rigorous method of partitioning the internal functions of intelligent systems into logical and computationally efficient modules.

The Air Force Program for Integrated Computer Aided Manufacturing (ICAM) developed IDEF[4] starting at the 1970s. IDEF is a method for functional modeling of systems [ID 93]. A particular subset, IDEF0, is becoming a part of the government standard known as the Federal Information Processing Standard (FIPS).

IDEF0 defines a set of symbols used for describing the functional models of subject systems or areas. Therefore, IDEF0 may be used to perform some functional analysis during RCS development. RCS, in its entirety, entails a much more complete and specific methodology for real-time embedded system development.

In figure 3, the controller nodes represented in the shaded boxes are those added for this scenario, mission #5. The rest of the controller nodes are completed in the previous development cycles (see previous papers [Hu 93-1, -2, -3, Hu 92]). The capability of the

[4]IDEF stands for ICAM Definition or Integrated Definition for function modeling [ID 93].

submarine RCS control system advances each cycle as we add more controller nodes to the existing hierarchy. This demonstrates that RCS facilitates software reuse.

4.2 Submarine Automation Overall Software Architecture

The generic RCS software architecture includes the following components: RCS controller hierarchy and its operator interface, simulation and its operator interface, and animation. Current initial research results, such as the descriptions given in sections 6 and 7, indicate that these components may all be represented as hierarchies with similar structures, as shown in figure 5. Further investigations are required to answer questions such as whether the task based hierarchical relationships exist in the operator interface hierarchies.

Human interface is allowed for all the modules in the control and simulation hierarchies. Such a setup allows the interjection of various environmental conditions. For example, in the demo series our implementations allow a sudden change in the sea water density to simulate a situation that the submarine runs into a fresh water column. Our implementations also allow activating a lube oil fire in the main shaft area. The control system operators need to intervene in the automatic control when situations like these become severe. Thus, they can be trained to be able to respond to anomalies such as these in a simulated environment before being assigned to a real submarine operating environment.

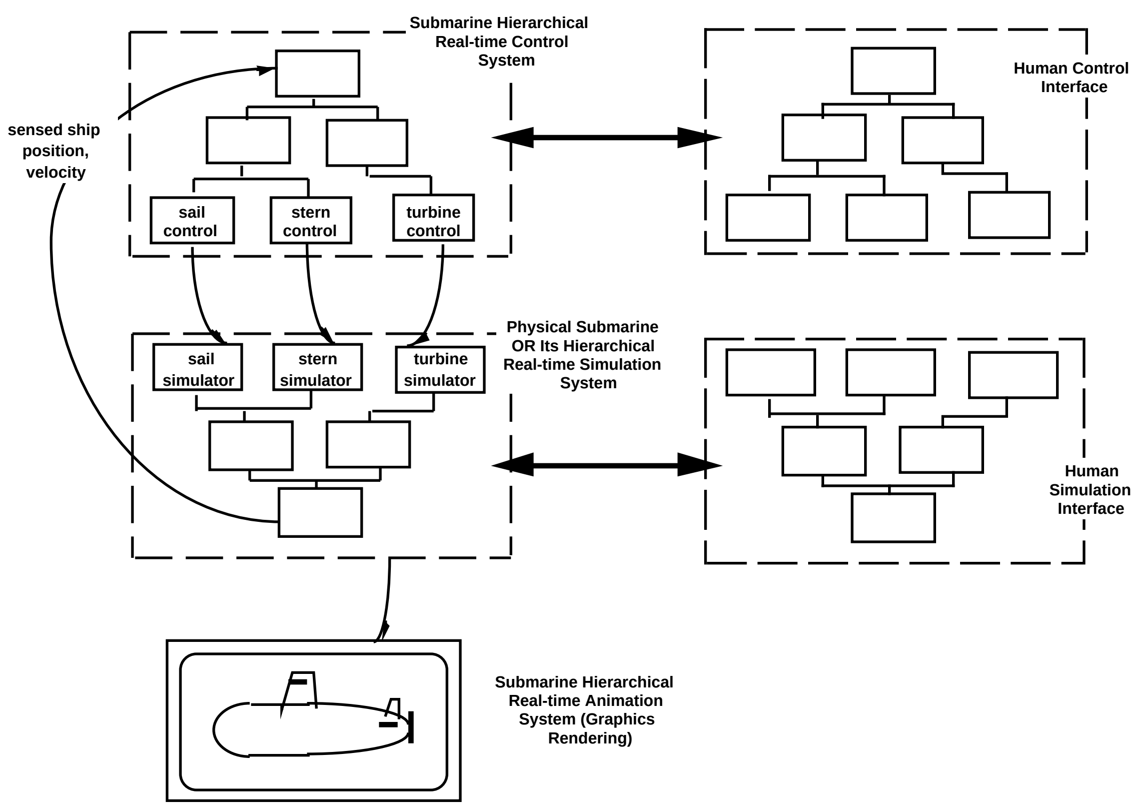


Figure 5: The Software Structure

This software architecture calls for explicit interfaces to be established between the control system and the simulation. These allow the simulator be replaced by a real submarine. While our implementation is a feasibility model, the control system, when fully explored and developed, is expected to be able to control the submarine and perform automatic operations while allowing operator monitor and override.

In this implementation, we utilized a 386 PC[5] compatible for control and simulation and a Silicon Graphics Incorporated workstation for animation and operator interface. Later sections describe how the interaction among different components occurs.

5 TASK ANALYSIS AND IMPLEMENTATION

RCS applies to intelligent systems that perform physical work. Therefore, we maintain that task decomposition, describing actions, as opposed to data models, is the most critical aspect in the control system software. The analysis of tasks should drive the system development effort. The analysis of data and functions are used, in limited context, to support task analysis. Task Decomposition is the fundamental principle in the proposed RCS methodology used to develop the submarine automation system.

5.1 Task Tree--an Outcome of Task Analysis

The “action verbs” in the scenario descriptions were identified as tasks or commands. The tasks were structured as task trees and then mapped onto the controller hierarchy according to their level of abstraction. These tasks then provide the vocabulary to model the intelligent behavior for the control system. Figure 6 shows the partial task tree that was extracted from the Demo#5 scenario. This tree was integrated with the existing task tree obtained in the previous implementation cycles (Demo#1 through #4) [Hu 93-2]. For example, the new Slow_and_change_depth task utilizes the Up_Bubble, Down_Bubble, and Maintain_Depth tasks that were implemented in the previous demos.

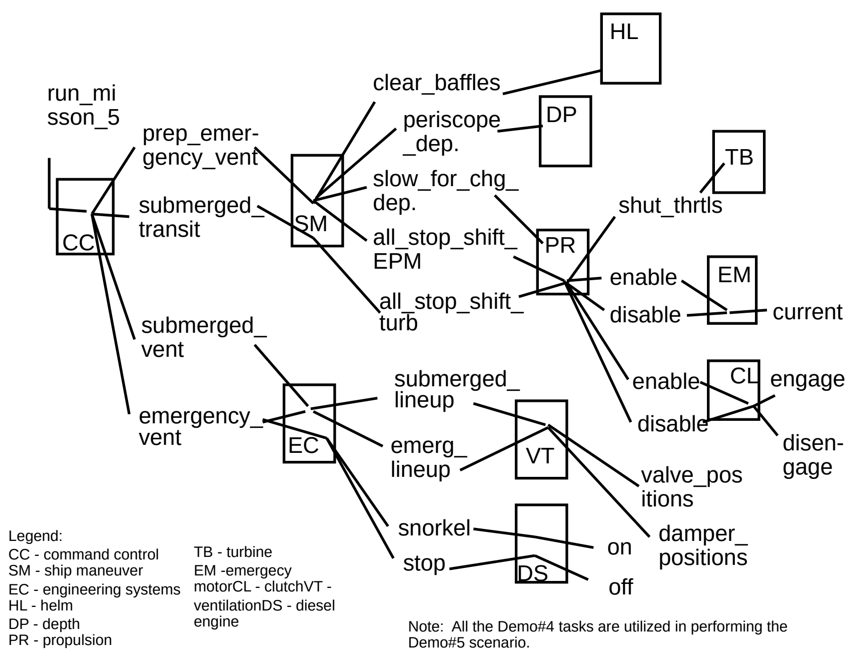


Figure 6: The Added Task thread based on the Demo#5 Scenario

[5]References to product or company names are for identification only and do not imply Government endorsement.

5.2 RCS Plans

From figure 6, the Run_Mission_5 command is decomposed into Prep_emergency_Vent, Submerged_Transit, Submerged_Vent, and Emergency_Vent commands. The exact controller behavior involving these four commands is shown in figure 7. When the mission command is received, the command controller (CC) enters the state (S1). The submarine transits toward the next waypoint with CC ordering the ship maneuver (SM) controller to execute the Submerged_Transit command and the Engineering system Controller (EC) to execute the Submerged_Vent command for normal open sea operations. CC is in the state (S2) waiting for the execution status coming back from SM and EC. (S1) and (S2) describe a feedback control loop for CC under normal conditions. Once all the waypoints are reached, the submarine completes its mission and CC would be in the (done) state.

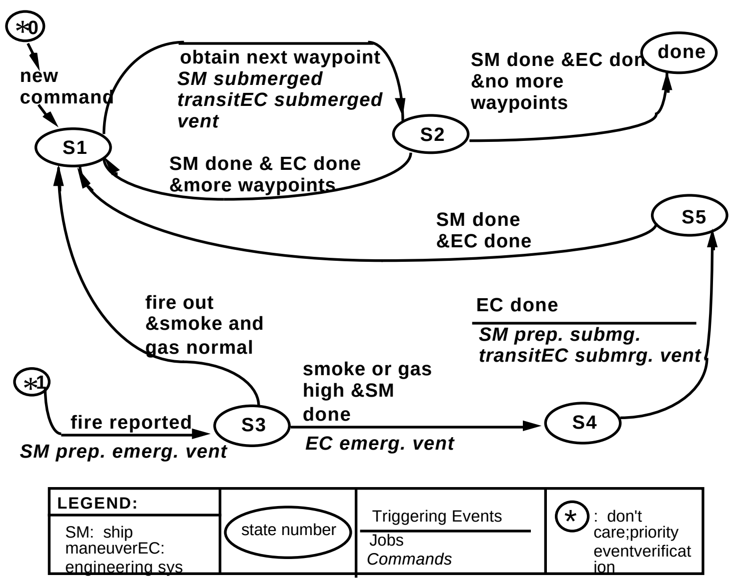


Figure 7: The Demo#5 Mission Plan for the Command Controller

When a fire is reported, CC changes the normal behavior by ordering SM to Prep_emergency_Vent, which includes activities such as Clear_baffles by SM, as seen in figure 6. Once SM is done, CC enters (S3) and orders EC to perform emergency ventilation. EC reconfigures the ventilation system (called line-up in submarine terms) and use the diesel engine to snorkel. Once the contaminants are vented and the atmosphere is safe for breathing, CC enters (S4) and orders SM and EC to prepare to resume the normal open sea transit, (S1) and (S2).

The Run_Mission_5 task has been explicitly decomposed and described using the state diagram, shown in figure 7. Such a description is called an RCS plan. This plan defines the initial state, the goal state, and all the intermediate states for the command controller during the execution of this command. In addition, all the required data, computation jobs, operator input, and subordinate status requirements are identified. This information enables the development of the associated sensory processing, world models, and human interface processes.

Using the same process, we have described each command on the task tree using an RCS plan. In this sense, a task tree provides a structure for organizing the behavioral descriptions. Multiple higher plans may require a same set of lower level plans. In these cases, the capability to avoid resource contention problems should be carefully built into the appropriate plans.

5.3 Programming and Execution

A generic controller template [Hu 93-3] is used to implement all the controllers. During execution, controllers read input data from their superiors, subordinates, and the global memory, they select plans and make decisions according to these plans, then they command their subordinates. This single template approach results in a simple and unified software execution pattern across the entire hierarchy, which facilitates the predictability of the software execution.

6.MISSION EXECUTION AND WATCH STATION ACTIVITIES

Watch station (WS, see footnote #3 for a definition) graphic panels have been developed to demonstrate the execution of the mission in an automated system. During real-time control, the WSs also serve as the human computer interface (HCI) of their corresponding controllers. The following three watch stations have been developed:

* The Officer of the Deck watch station (OOD WS), which serves as the HCI of the command and maneuvering controllers.
* The Ballast Control Panel watch station (BCP WS), which serves as the HCI of the engineering systems, ventilation, and diesel controllers.
* The Engineering Officer of the Watch watch station (EOOW WS), which serves as the HCI of the propulsion controller and all its subordinates.

The human computer interface (HCI) must display the necessary information for all the controllers in order to enable the interaction between the control hierarchy and the submarine operators. Note that the objective of the HCI is not to mimic the current submarine operating environment faithfully. In other words, we do not expect to model an OOD, diving officer, helmsman, etc., as designated on a submarine. Neither is it required to have an individual HCI panel for each controller. Instead, the following three factors are combined in determining the number and types of WS displays: the operator workload [Hu 91], understandability and acceptability by the current submarine operation community, as well as the efficiency of hierarchical system control.

These watch station panels include graphic data displays, control device buttons, and text-message displays. Colors are used in the text displays to distinguish different types of messages: normal operational status, errors, operator input requests, etc. The watch station displays should be installed in the locations where the corresponding manual operations are currently performed, namely: Officer of the Deck and Ballast Control Panel watch stations in the Operational Compartment and the Engineering Officer of the Watch watch station in the Engine Room, as seen in figure 2. This guideline facilitates the integration of automated subsystems into current operating environment.

The Officer of the Deck watch station, shown in figure 8, displays the crucial maneuvering data, including (from left to right) the bubble angles, the heading and speed, and the depth. It also includes two text-message areas for the command that the command controller is outputting (for maneuver) and the announcement that it is making.

The Engineering Officer of the Watch watch station, shown in figure 9, has buttons for engaging or disengaging the main shaft clutch and has a speed control knob for the Emergency Propulsion Motor (EPM). This WS also has two text-message windows. The command text window normally displays the command that the propulsion controller is executing. The window turns yellow when the propulsion controller requests the operator to perform the displayed command. The REPORT message window displays useful messages for the Engineering Officer of the Watch operator.

Figure 8: The Officer of the Deck Watch Station Display

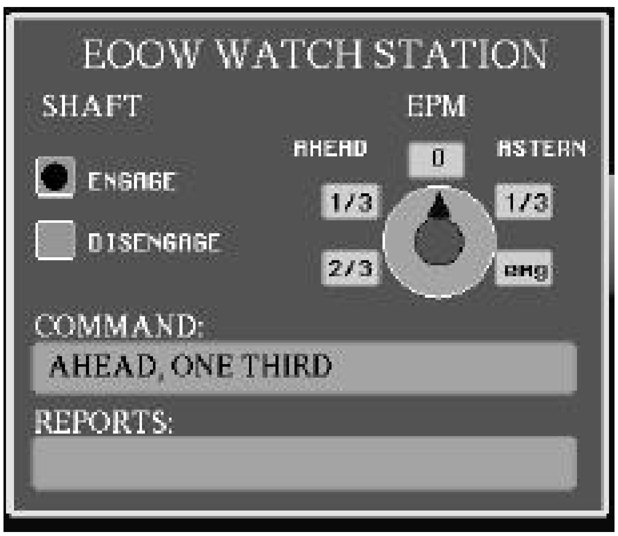


Figure 9: The Engineering Officer of the Watch Watch Station Display

The Ballast Control Panel watch station, shown in figure 10, contains the same types of text-message areas as in the Engineering Officer of the Watch watch station. The Ballast Control Panel watch station also includes: four atmospheric analyzer displays, the main ballast tank control buttons, and a ventilation line-up display (figure 2).

At the beginning of the operation, the submarine is conducting an open sea transit. The Officer of the Deck watch station displays a nominal zero degree bubble angle, a standard speed (15 knots, or 7.7 m/s), and a nominal 60 m keel depth. The ANNOUNCEMENT message window is blank. At the Engineering Officer of the Watch watch station, the COMMAND window displays a standard speed. Neither the SHAFT nor the EPM (Emergency Propulsion Motor) buttons are activated. The atmospheric analyzers in the Ballast Control Panel watch station display normal levels of oxygen, carbon dioxide, smoke, and carbon monoxide. The ventilation diagram displays normal air circulation.

A lube oil fire (see the scenario) is reported through the sensors in both the propulsion and the ventilation control systems. The REPORTS text window in figure 9 displays the fire message. The command controller immediately announces the message of “ENG RM FIRE, ALL HANDS ON EABs” through the Officer of the Deck watch station display.

Meanwhile, the COMMAND window starts displaying "PREP FOR EMER VENT," meaning that the command controller is ordering the maneuver controller to execute the displayed command. Maneuver decomposes this command into three commands: Clear_baffles, Slow_and_Change_Depth, and Shift_To_EPM for its subordinates, as seen in figure 6. This task decomposition activity is displayed in the COMMAND window in real-time. In other words, the displayed commands correspond to the actual states that the Maneuver controller is in. Meanwhile, the ventilation controller SP and WM algorithms update the abnormal concentrations of the modeled air constituents, namely, oxygen, carbon dioxide, smoke, and carbon monoxide. These data are displayed, in real-time, in the Ballast Control Panel watch station atmospheric analyzer displays (figure 10).

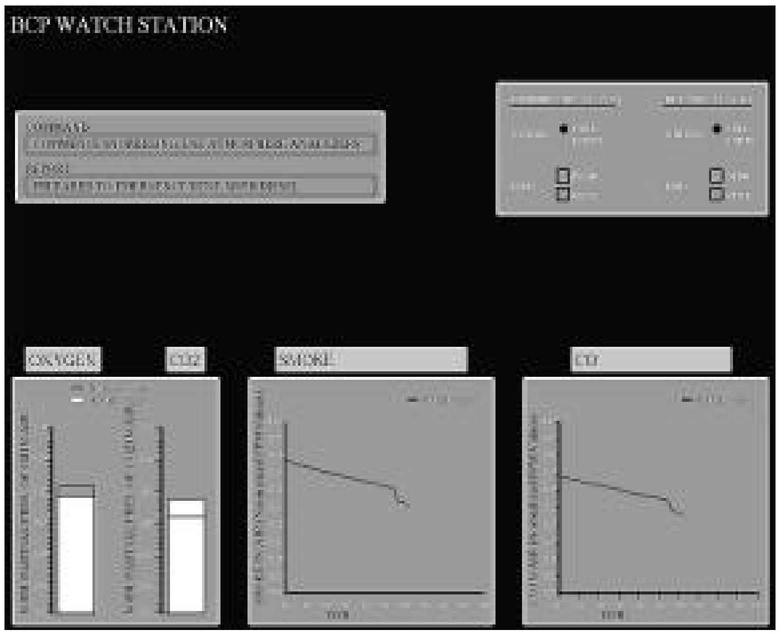


Figure 10: The Ballast Control Panel Watch Station Display

The ventilation system is reconfigured automatically to prepare for emergency ventilation once the submarine is at the periscope depth. Once this command is completed, the Ballast Control Panel watch station ventilation display (figure 2) shows the new paths of air flow. This command completion status also prompts a message stating "Prepared to emergency ventilate with diesel" on the REPORT window (figure 10). The Engineering Systems controller then receives a "Commence Snorkeling, Using Atmospheric Analyzers" command, as shown in figure 10. The diesel engine extracts and exhausts the contaminated air and takes in the fresh air through the mast extending above the level of the water. This command completes when the atmosphere becomes safe to breathe again. At such point the Command controller orders the submarine to resume the open sea transit.

7.Hierarchical Depth Control and Simulation

As described earlier, the RCS methodology provides a behavior oriented analysis method that allows designers to model the internal structure of a system to a sufficient level of detail. This analysis produces a representation consisting of an organization hierarchy, a task tree, and behavior diagrams, as described in section 4. Once the structure is in place, the necessary supporting data, algorithms, simulation, sensors, and operator interface can be identified. The same concept is extended to the development of the simulation structure, which results in a hierarchical simulator. Such a simulator structure facilitates sensory data analysis for the RCS controller units. It also enables incremental testing of the control hierarchy.

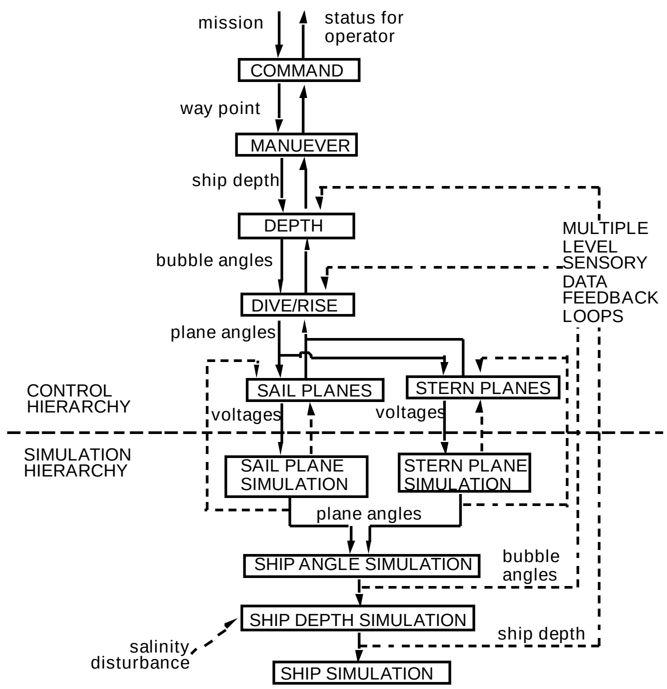


Figure 11: Nested Depth Control and Simulation in Submarine Automation

A mission command given to the command controller covers many aspects, including: the goal, the sizes of the moving haven, and the depth requirements throughout the transit. The depth control requirements must be converted to electrical signals for the sail and stern planes at the lowest level of the control hierarchy. A series of intermediate representations is needed to provide smooth transitions between these two extremes. These intermediate representations can not be chosen arbitrarily. Instead, they should be specified to facilitate human understanding, computation efficiency, and control stability. The command controller (CC) decomposes the mission goal into a series of intermediate goals, or waypoints, and passes them down to the Maneuver controller. Based on the waypoints and the pre stored map data, Maneuver computes the required ship depths and passes them down to the Depth controller. Depth computes a series of bubble angles required to achieve the required depth. The Dive/Rise controller computes required plane angle moves for the Sail and Stern Plane controllers to achieve the required bubble angles. The plane controllers generate electrical signals for the control valves to move the planes to the commanded angles. This decomposition provides a smooth transition of a control variable from a global and abstractive perspective covering a large spatial span to a local and machine executable perspective covering a short spatial span. This facilitates efficient and stable execution. The repetitive structure and limited complexity of each node also facilitate software maintenance.

The submarine depth simulator is developed as an inverted control hierarchy, as seen in the lower portion of figure 11. The only input that the simulator receives from the controllers is the commanded electrical signals. The hydrodynamic model for the submarine is decomposed and distributed in the simulator hierarchy. At the “lowest” level (shown at the top of the simulator hierarchy), the electrical signals are used to compute the simulated plane angles, which are integrated at the next level to form simulated ship bubble angles.

At the next level, the dynamic model uses ship angles to compute the ship depth. All these intermediate results may be used as sensory data feedback to the appropriate controllers. This process demonstrated a similar smooth transition for a submarine state variable. This process also facilitates stable software execution and efficient sensory data analysis for the control system.

8.Reconfiguring Plans and Control Hierarchy to Expand System Capability

In the previous demonstrations, we developed a Come_to_Course plan for the Helm controller, see part A of figure 12. The scenario for Demo#5 requires the addition of a Clear_baffle task, which can be treated as a series of Come_to_Course tasks. There are several approaches to take advantage of the existing Come_to_Course plan, including:

* Write an independent Clear_baffle plan which is composed of a series of subprograms performing the Come_to_Course operation repetitively. This is illustrated in part B of figure 12. This approach suffers from the disadvantage of having a large plan with duplicate software. Its advantage is being straightforward.

* Employ a new controller, denoted SuperHL in part C of figure 12, between SM and HL, to decompose Clear_baffle to a series of come to course operations. This option causes two superiors for HL which is irregular and might cause a resource contention problem once the controllers become complex.

* In part D of figure 12, the irregularity is alleviated by having SHL “decompose” all the commands. This alternative seems acceptable, although a disadvantage is that it causes trivial decomposition of all the other commands (Come_to_Course and Stop).

* Maintain the original controller hierarchy (as shown in part A of figure 12) and expand the functionality of the “planner” (see section 4.1) within the HL controller. This option is shown in part E of figure 12. The planner is to allow intelligent reconfiguration of existing plans to perform more complex tasks. We have selected this approach as an experiment. As a first version, such a planner was implemented in the format of state tables. It plans the operation of clearing baffles by applying a series of Come_to_Course plans: to swing the submarine heading to the left by 30º, followed by steering to the right for 30º, and then a third Come_to_Course to swing the heading to the original course. Detection of external objects may cause additional operations. Some advantages of this approach are that it facilitates a canonical model of planning and that it facilitates real-time reconfiguration of existing software. The trade-off is that it blurs the simplicity and high replicability nature of the original software structure.

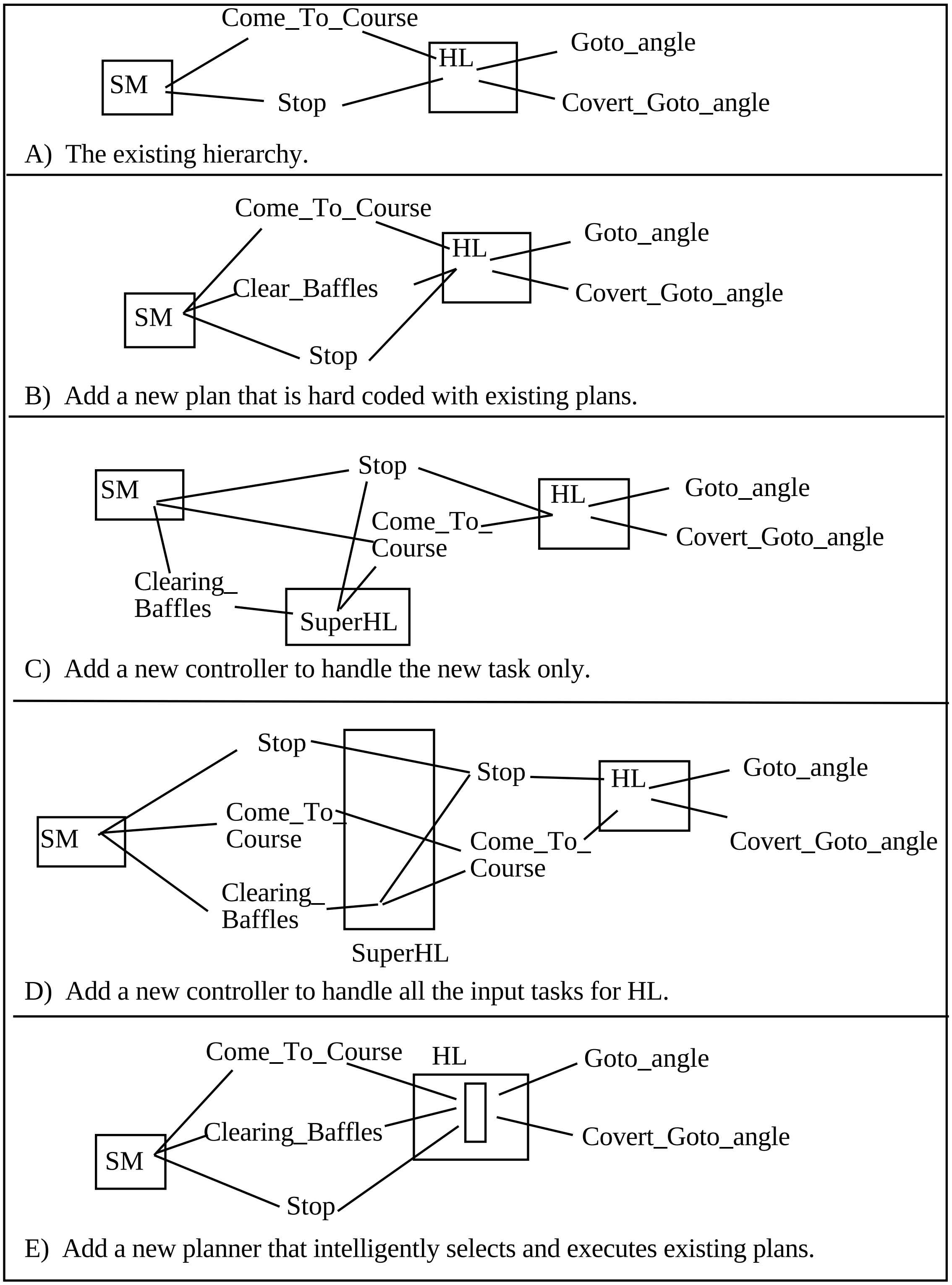


Figure 12: Alternatives for Expanding System Capability

9. SUMMARY:

We have completed the demonstrations of using the RCS methodology to develop a multiple-level hierarchical real-time control system for a submarine. The characteristics of the demonstration include:

* A Behavior Oriented Development Method.

* A High Degree of Operator Interface.

* A Deterministic Execution and Known Performance.

* Single Building Block and Well Defined Interfaces. The benefits include:
  - reducing software complexity.
  - improving human understanding.
  - employing highly replicable controller units.
  - producing flexible control structure.
  - facilitating system extensibility and reusability.

* Cost Effectiveness:
  - hardware: using PC based controllers.
  - development: applying a rigorous methodology.
  - testing: emphasizing using simulation and animation.
  - operation: achieving automation while allowing real-time operator interface.
  - maintenance: requiring only basic system support.
  - upgrade: producing easily portable and reusable code.

We have demonstrated that a system development methodology such as RCS is very effective in handling the problem of submarine automation.

ACKNOWLEDGMENTS

Captain Robert Lowell and Dr. Ed Carapezza of ARPA have been the sponsors of this project at various stages. They have provided valuable technical insights to the problem. Dr. James Albus, the division chief of the Intelligent Systems Division of NIST, and Dr. Anthony Barbera of the Advanced Technology and Research (ATR) Corporation have been leading the research of RCS since they originated RCS two decades ago in NIST. M. L. Fitzgerald, Nat Frampton, Philip Feldman, and Clyde Findley of ATR and Ron Hira, Ross Tabachow, and Will Shackleford of NIST participated in various stages of this project.

REFERENCES

[Al 92] Albus, J.S., Juberts, M., Szabo, S., "RCS: A Reference Model Architecture for Intelligent Vehicle and Highway Systems," ISATA 92, Florence, Italy, June 1992.

[Al 89] Albus, J.S., McCain, H.G., and Lumia, R., "NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)," NBS Technical Note 1235, National Bureau of Standards, U. S. Department of Commerce, April, 1989.

[An 93] Antsaklis, P.J., Lemmon, M., Stiver, J.A., Hybrid System Modeling and Event Identification, Technical Report of the ISIS Group, ISIS-93-002, University of Notre Dame, Notre Dame, IN, January, 1993.

[Hu 93-1] Huang, H., Hira, R., and Quintero, R., “Automation of Submarine Maneuvering Systems: A Model and an Approach,” Approved by WERB and submitted to the IEEE Transaction on Control Systems Technology, December, 1993.

[Hu 93-2] Huang, H., Hira, R., and Quintero, R., "A Submarine Maneuvering System Demonstration Based On The NIST Real-Time Control System Reference Model," Proceedings of The 8th IEEE International Symposium on Intelligent Control, Chicago, Illinois, 1993.

[Hu 93-3] Huang, H., Hira, R., Quintero, Q., and Barbera A., ″Applying the NIST Real-time Control System Reference Model to Submarine Automation: a Maneuvering System Demonstration,″ NISTIR 5126, 1993.

[Hu 92] Huang, H., Hira, R., and Feldman, F. "A Submarine Simulator Driven by A Hierarchical Real-Time Control System Architecture," NISTIR 4875, 1992.

[Hu 91] Huang, H., Quintero, R., and Albus, J.S., "A Reference Model, Design Approach, and Development Illustration toward Hierarchical Real-Time System Control for Coal Mining Operations," in Control and Dynamic Systems, Advances in Theory and Applications, Volume 46, Academic Press, 1991.

[ID 93] Federal Information Processing Standards (FIPS) Publication 183: Specification for Integration Definition for Function Modeling (IDEF0), National Institute of Standards and Technology, U. S. Department of Commerce, to be printed, December 1993.

[Jo 91] Johnson, D.W., Szabo, S., McClellan, H.W., DeBellis, W.B., "Towards an Autonomous Heavy Lift Robot for Field Applications,” 8th International Symposium on Automation and Robotics in Construction, 3-5 June 1991, Stuttgart Germany.

[Qu 92] Quintero, R. and Barbera, A.J., A Real-Time Control System Methodology for Developing Intelligent Control Systems, NISTIR 4936, October, 1992.

[Sa 85] Saridis, G. N., “Foundations of the Theory of Intelligent Controls,” Proceedings of the IEEE Workshop on Intelligent Control 1985, Troy, New York, August 1985.

[Sz 92] Szabo, S., Scott, H.A., Murphy, K.N., Legowik, S.A., Bostelman, R.V., "High-Level Mobility Controller for a Remotely Operated Unmanned Land Vehicle," Journal of Intelligent and Robotic Systems, 5: 63-77, 1992.