AN OPERATOR EXPERIENCE WITH A HIERARCHICAL REAL-TIME CONTROL SYSTEM (RCS)

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

ABSTRACT

The issue of operator interface in the RCS hierarchical control systems is the focus of this report. In a submarine automation project, we have studied the logical structure of the operator interface function and the way it should be integrated with existing systems. A design issue is to allow operators to be involved in system operations in different degrees. In some situations, the operators are requested to perform certain manual operations and report the status back to the control systems. In some other situations, the operators are required to make some decisions for the controllers. At different control levels, operators require different information. Software servicers and system developers may require totally different types of information.

1. INTRODUCTION

The researchers at the Intelligent Systems Division of the National Institute of Standards and Technology (NIST) supported an ARPA submarine automation project[1]. We used the SSN 637 class nuclear submarine data as our model. With the given level of funding, our study focused on some of the maneuvering and engineering support functions. We developed a series of software systems [1, 2] to demonstrate our results.

The core technology that we apply to the submarine automation problem is the NIST Real-time Control System (RCS) reference model architecture. While numerous papers have been published describing the RCS theory and applications [3, 4, 5, 6], the issue of operator interface requires systematic studies. This paper attempts to focus on the operator interface issues in RCS based control systems. We view that, in the RCS architecture, the operator interface function forms a logically distinct structure parallel to the control hierarchy. In RCS, a central concept is multiple levels with distinct but successive levels of abstraction. We attempt to apply the same concept to the operator interface structure. The goal is to form a unified and comprehensive logical structure for the software development, operation, and service of RCS applications.

1.1 An Illustrative Scenario

In the RCS application design methodology [4, 7], a beginning step is to develop typical operational scenarios, i.e., the actual activities when the physical systems are conducting given missions. These scenarios serve the purposes of highlighting and illustrating the functions and capability of the control systems. The following is an illustrative scenario for our case study:

---

[1] ARPA Order No. 7829; the ARPA Maritime Systems Technology Office.

A submarine is conducting a submerged transit of the open ocean at its standard speed (7.7 m/s, equivalent to 15 knots or nautical miles/hour) and at a keel depth of 200 m. A watchstander (a submarine term, meaning a crew member who is assigned to a designated onboard location, called Watch Station, to perform the pre-specified duties) reports that there is a “lube” (lubrication) oil fire in the lower level Engine Room. The Officer of the Deck (OOD) directs the Ballast Control Panel (BCP) operator to pass the word on the general announcing system. The OOD completes the following actions for coming to periscope depth: Clearing baffles, Checking for sonar contacts and close contacts, Slowing and changing depth, and Raising the periscope.

The damage control party fights the fire in the engine room. On indication of decreasing main lube oil pressure the 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 damage control party reports that the fire is out. The BCP selects the ventilation lineup and sets it to emergency ventilate the engine room using the diesel engine. When the lineup is proper, the OOD directs "Commence snorkeling."

1.2 The Control System

In [2], we gave a detailed description of the submarine control system. We briefly summarize the hierarchy, shown in Figure 1, to facilitate the understanding of this operator interface issue. A box is called a controller within the hierarchy. The command controller handles the highest level control, namely, the execution of the missions. 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, with higher resolution, and containing a higher level of detail. Similarly, these two controllers complete their tasks by:

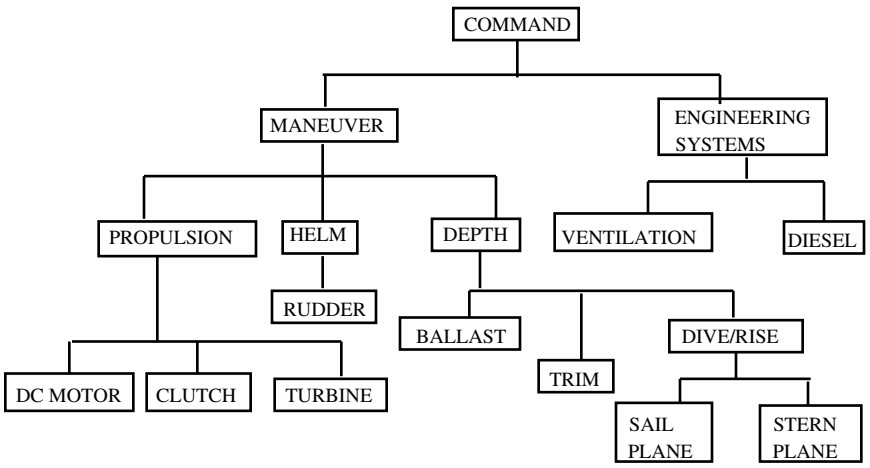

Figure 1: The illustrative submarine automation control hierarchy

* 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.

The same principle applies to further lower level controllers. Maneuver coordinates the propulsion, helm, and depth controllers. Engineering systems coordinates the ventilation and the diesel engine controllers. Propulsion coordinates the DC motor, the clutch, and the turbine controllers. The lowest level contains actuator controllers.

1.3 Basic Principles for Operator Interface

RCS prescribes that humans can interact with any controller node in an RCS control hierarchy. A study conducted by Internal Research Network on Culture and Production (CAPIRN) found out that, to automate manual work is a relatively low priority design criteria to machine tool manufacturers [8]. This is indicative that, operators have been, and are expected to continue to be, involved in automated system operations.

Our basic principles for operator interface design are:

* to have a well-defined overall structure and well-defined roles for individual operators,
* to integrate seamlessly with the system control,
* to integrate seamlessly with the current operating environment,
* to match user requirements and expertise,
* to allow emergency procedures,
* to optimize both real-time performance and operator workload,
* to inform intuitively, in real-time and as much as operators desire, but not to overload, and
* to support system service and development, including testing and simulation.

The remainder of this paper describes these principles in detail.

2. THE ORGANIZATION AND FUNCTIONING OF THE CONTROL OPERATOR INTERFACE

We propose that each of the controller nodes in an RCS hierarchy is logically associated with an operator interface (OI) node. This concept facilitates a well-defined overall structure for OI. All the desired OI functions for the control system can be implemented systematically. In Figure 2, the left-hand side shows the control hierarchy. The right-hand side shows an image hierarchy for OI. Section 2.1 describes the levels in the OI hierarchy. Section 2.2 describes the three shaded areas.

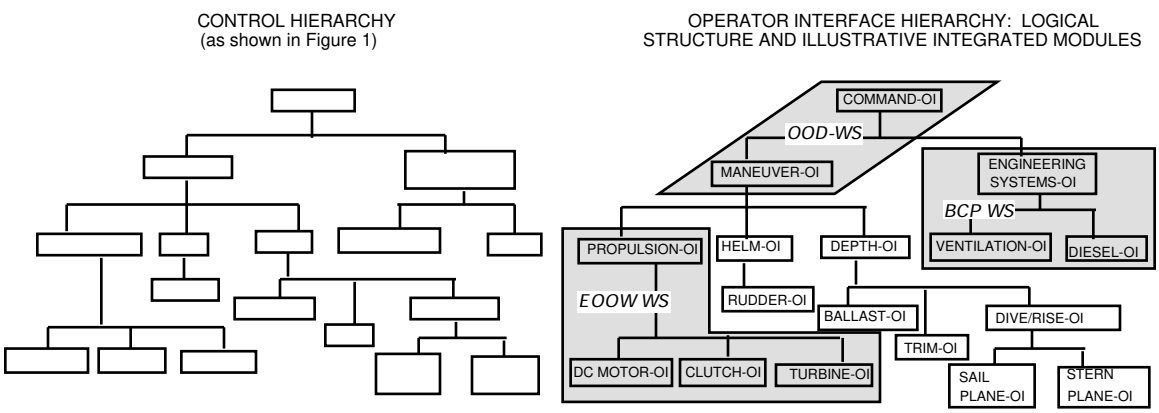

Figure 2: Operator interface logical structure

2.1 Levels of Commanding Authority

In order to support hierarchical system real-time control, in RCS, the operator interface functions are divided into six types of levels of commanding authority. Specific applications may not contain all the levels. The level types, from the highest to the lowest levels of authority, are:

Level 6 -- application domain level, or mission level, operator. This is the highest level. The operator enters and monitors the overall command for the entire control systems. In our case study, the command controller operator belongs to this level.

Level 5 -- group level operators. They handle control activities involving either multiple coordinated control entities or multiple coordinated groups of control entities. Our case does not involve this function.

Level 4 -- equipment or task level operators. These operators deal, through their corresponding controllers, with the control activities of individual major physical entities. In the hierarchy shown in Figure 1, the maneuver and the engineering systems controller operators are of this type.

Level 3 -- elementary move (emove) level operators. These operators deal, through their corresponding controllers, with the kinematics of the control activities. In our case study, the propulsion, helm, and depth controller operators belong to this level. The operators may monitor or command such elementary move control activities as avoiding obstacles, including the Arctic region ice keels, and as avoiding kinematic limits, including the instability of the submarine caused by large bubble (pitch) angles.

Level 2 -- primitive (prim) level operators. These operators deal, through their corresponding controllers, with the dynamics of the control activities. These operators must ensure the dynamic achievability and smoothness of the kinematically sound control activities. The dive/rise, trim, and ventilation operators are of this type.

Level 1 -- actuator level operators. These operators deal, through their corresponding controllers, with direct environmental interaction, such as the electrical and mechanical signals that drive the actuators toward the goals. The DC motor and clutch operators, shown in Figure 2, belong to this level.

This hierarchical organization also allows a problem with a high level of abstraction to be logically and smoothly transitioned to a set of sub problems with low levels of abstraction. In this way, the operator interface for a complex problem can be shared by a set of operators with well defined and limited responsibilities.

2.2 Watch Station Activities

For well-established industry areas or any types of systems in which manual operations dominate, it is beneficial to introduce automation in an evolutionary, as opposed to revolutionary, manner. Automation and intelligent control should be incrementally integrated in. Therefore, we propose that the logical OI structure must be designed to be compatible with the current submarine operating environment.

We illustrate a submarine Operational Compartment in Figure 3. This diagram shows how operators are stationed onboard a submarine. Note that the propulsion control is performed in a separate location called Engine Room. We have developed three Watch Station (WS, see section 1.1 for definition) graphic panels to map the OI hierarchy (Figure 2) to this

Operational Compartment. These WSs provide the input and output capability for operators during real-time control. The three WS displays are:

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

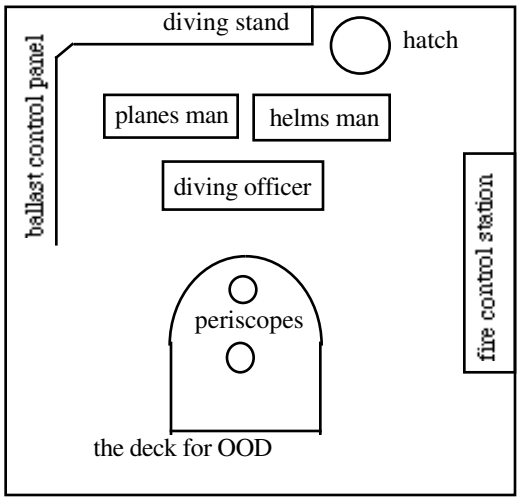

Figure 3: An illustrative submarine operational compartment

The three displays are shown in the three shaded areas in Figure 2.

The operator interface (OI) must display the necessary information for all the controllers, in real-time, in order to enable the interaction between the control hierarchy and the submarine operators. Note that the objective of the OI 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. Instead, the following three factors are combined in determining the number and types of WS displays: the operator workload [5], 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 may be installed at the onboard locations where the corresponding 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.

2.2.1 The Officer of the Deck watch station

The Officer of the Deck watch station, shown in Figure 4, displays the crucial maneuvering data, including (from left to right) the bubble angles, the heading and speed, and the depth. The WS also includes two text-based message areas for the command that the command controller is outputting (for maneuver) and the announcement that the controller is making.

At the beginning of the operation as described in the scenario, 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 (7.7 m/s, or 15 knots), 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 controllers. The REPORTS text window in Figure 5 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 (Figure 4). 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: to prepare for emergency ventilation. Maneuver decomposes this command into subtasks for its subordinates (see the scenario described in section 1.1 and see [2]). This task decomposition activity is displayed in the respective COMMAND windows in real-time. The displayed commands correspond to the actual states that the controllers are in.

2.2.2 The Engineering Officer of the Watch watch station

The Engineering Officer of the Watch watch station, shown in Figure 5, employs two buttons for engaging or disengaging the main shaft clutch and employs 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 shades in 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.

Per the scenario, the "PREP FOR EMER VENT" command at the higher level is decomposed into disengaging the main turbine shaft and adjusting the EPM speed as commanded. The Engineering Officer of the Watch operator uses the watch station to perform these operations and to report the status.

Figure 4: The Officer of the Deck Watch Station Display

Figure 5: The Engineering Officer of the Watch Watch Station Display

2.2.3 The Ballast Control Panel watch station

The Ballast Control Panel watch station, shown in Figure 6, 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 the concentration analyzer displays for the air constituents of oxygen, carbon dioxide, smoke, and carbon monoxide. Per the scenario, after the lube oil fire broke out, the sensory processing and world modeling algorithms of the ventilation controller detected the abnormal concentrations of these air constituents. These data are updated, in real-time, in the atmospheric analyzer displays. Once the submarine is ready for emergency ventilation, the ventilation system is reconfigured automatically to prepare for emergency ventilation. In [2], we described that the watch station employs a ventilation line-up display that shows the current ventilation configuration.

In addition, the Ballast Control Panel watch station contains the main ballast tank control buttons for the emergency surfacing purposes. We had demonstrated the usage of these control buttons in an earlier software system [1].

2.2.4 A summary of the operator interface displays

The following summarizes the data types developed for operator interactions (see [2] for more details):

Output to the operators:
* Continuous operational data, including ship depth displays and paths.
* Digital operational data, including ship speed displays.
* Discrete activities, such as commands, announcements, watch station reports.
* Operator input requests.
* Schematic diagram.
* Errors and recommendations.
* Controller performance, such as execution time.
* Debug data, including command, state, status, world data.

Input from the operators:
* Status.
* Command selections.
* Environmental variable settings.
* Actuator override devices.

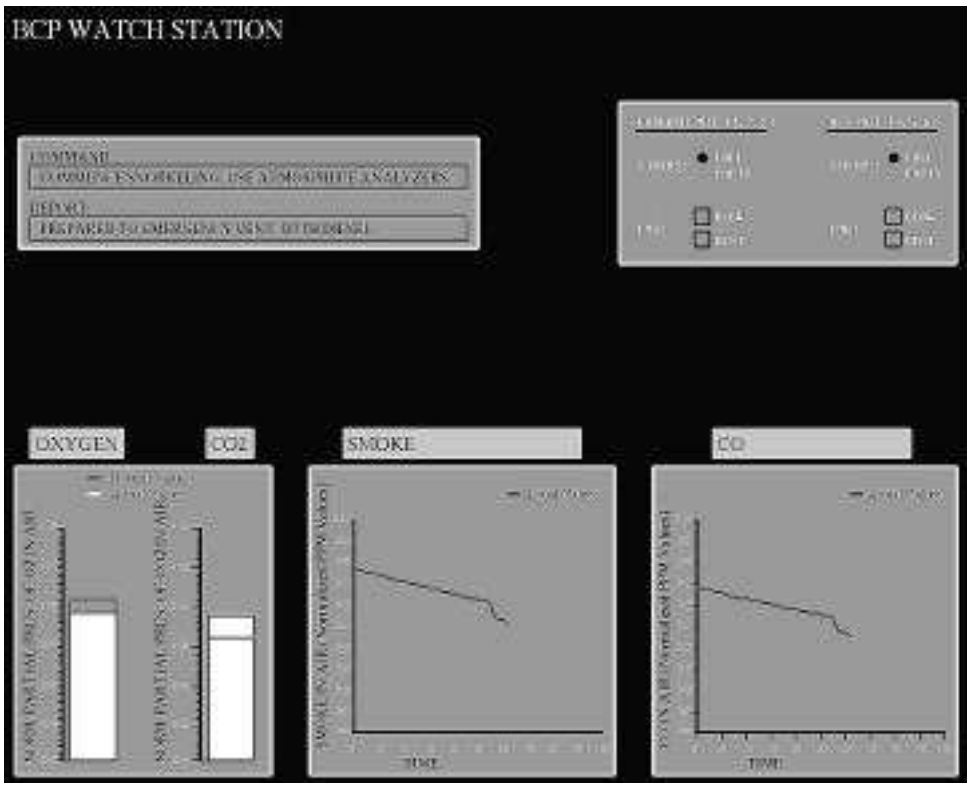

Figure 6: The Ballast Control Panel Watch Station Display

3. LEVELS OF OPERATOR INVOLVEMENT DURING OPERATIONS

RCS allows operators to be involved in system operations at various degrees. In this application, we started experimenting with five degrees of access control, giving the operators from the least to the most amount of authority to interact with individual control entities in the hierarchy:

* Monitor, or to be informed.
* Respond to system requests.
* Alter system behavior by issuing new commands.
* Manual override.
* Modify system, or to reconfigure control hierarchy.

The access control should be applied to both the control nodes and the tasks. The former means that each control node in the hierarchy has a logical operator interaction function with assigned degrees of operator access control. The latter means that each step in a task plan may be assigned certain degrees of operator access control. For example, operator intervention may not be allowed during the execution of a safety related task. Our experiments focused on the operator interaction on a node basis. Further research is required to integrate these two aspects.

We use a two-dimensional matrix to visualize such an access control concept, as shown in Figure 7. Monitoring is the default degree of authority, which all the operators are allowed.

The next unit on the horizontal axis, respond to system requests, does not allow operators to initiate interruptions to system operations. We do not allow the propulsion operator to issue new commands. Rather, he is only authorized to perform requested actions, such as CHANGE_TO_EPM, as described in section 1. Once the operator sees the command displayed on the COMMAND text window (Figure 5), he watches the turbine speed decreasing to zero (the submarine must have and does have a low speed as a result of the inertia). He then, manually, disengages the main turbine shaft, engages the EPM shaft, and clicks the button on the display to report the completion. In this case, the operator serves as the EPM control node. This man-in-the-loop control operation also suggests that our method supports integrating the automation technology to legacy systems with manual operations.

Another example of responding to system requests is for operators to make decisions for the control system. The submarine may run into some salinity disturbances close to coast where the depth controller can not maintain the ship's depth. The error messages and a set of options are displayed for Maneuver. The operator selects a best command for the controller to perform [1].

The next unit on the horizontal axis, alter system behavior, means that the operator initiates graceful changes to system operations. In other words, operator commands are integrated with the on-going system automatic operation. The command controller operator can alter system behavior by issuing a new mission command and having the new and the existing mission commands integrated.

The next unit on the horizontal axis, manual override, means that operator takes over control and the automatic control commands are ignored, suspended, terminated, or aborted. The lowest level operators can block the automatic commands and directly manipulate the actuators [1]. When the submarine is about to hit ice keels, the helm operator must be allowed to enter an emergency turning command.

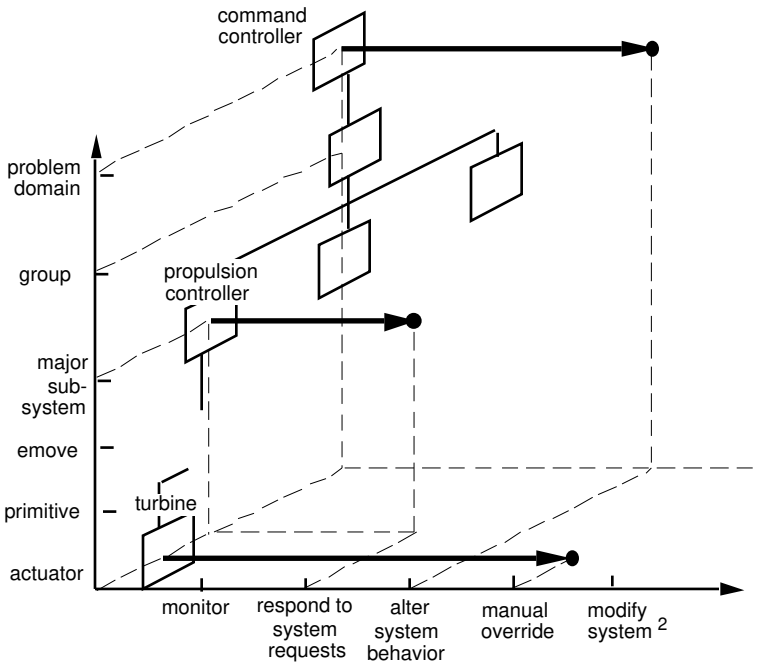

Figure 7: Different degrees of operator involvement, an illustration

Reconfiguring control hierarchy means that the operator can re-align the command chain of the hierarchy or can expand or reduce the control capability of control nodes. In an early version of the submarine control system, a FORTH based programming language was used, which allowed the system operation to be halted, new task plans to be programmed in and incrementally loaded, and the system operation to be resumed[2].

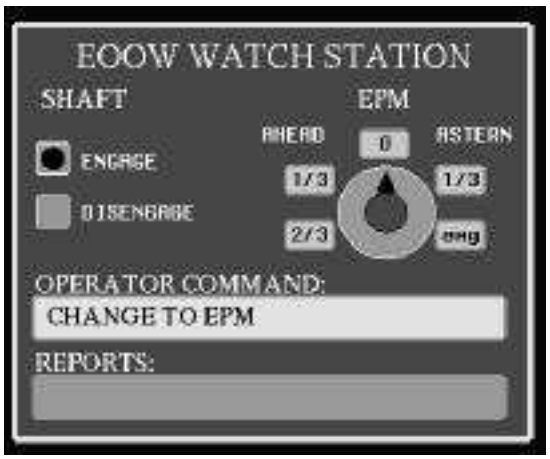

Figure 8: A request for operator action

[2] This capability, enabled by the particular language, was abandoned in favor of the using the C as the implementation language which is widely supported in the industry.

4. User Expertise and Hierarchical Situation Perception

As stated in our basic principles, operator interface information must meet the user requirements and expertise [9]. All users are not alike. They require different types of data. In section 4.1, we describe knowledge processing within a control level. In section 4.2, we describe knowledge processing along the hierarchical levels.

4.1 Knowledge Transformation

The data that are useful to a submarine control operator may or may not be either useful or sufficient to a system developer or servicer. A system developer or servicer may need to access all the available raw data during the debugging periods. Our implementation makes the raw data available.

On the other hand, it is beneficial for a submarine operator to view concise and processed data. In emergency situations, he may not have time to read through all raw data to find out the problems. Therefore, knowledge processing (Figure 9), a part of our RCS world modeling functions, can take two different meanings:

* From raw data to concise data.
* From the raw or concise data to conventional terms that match the operator expertise.

We have used submarine terms in our message sets, including "Fire in Engine Room, all hands on EABs (emergency air breathing units)" and "All stop, shift to EPM (emergency propulsion motor)." A particular message for a control operator may be an inferenced result of many pieces of raw data.

Note that, although we have made a distinction between these two types of data, they are not sorted explicitly by the two types of users respectively. Some concise and processed data may be just as informative to a debugger. Note also that there exist common criteria for the two types of data, such as: they be graphic if possible to help human dissemination, they be displayed in real-time if possible, and they share a common knowledge base.

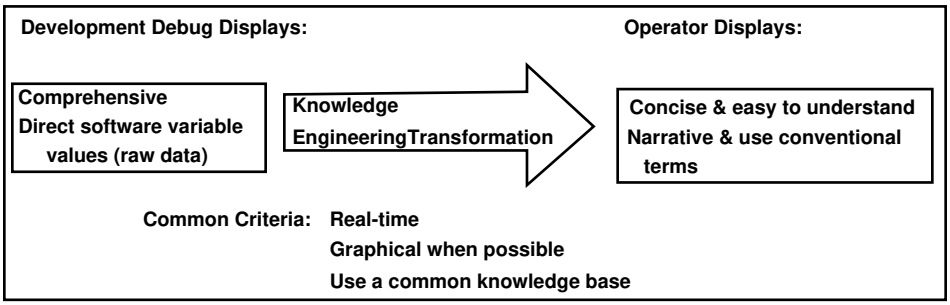

Figure 9: Knowledge processing

4.2 Commanding Levels

In the RCS hierarchical environment, different levels deal with events and commands at different resolutions and time scales. Temporal and spatial integrations must be performed when lower level information, either discrete events or continuous data, is delivered to higher levels. This integration process creates a different perception. An operator only requires to understand the situations at two levels of resolution: his own level and the level below.

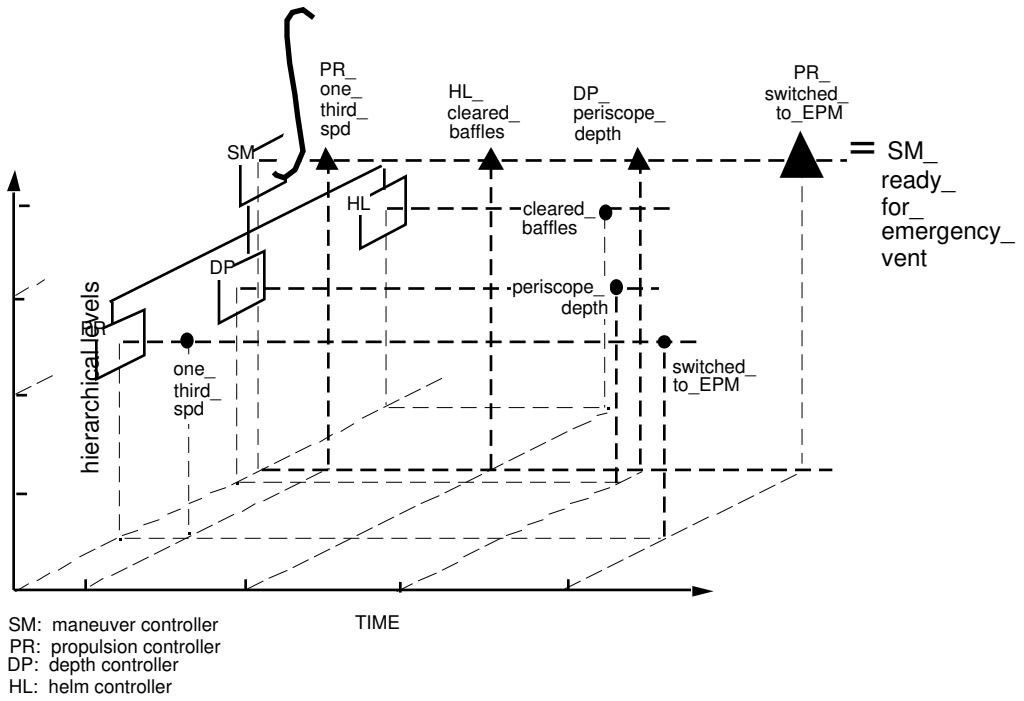

Figure 10: hierarchical situation evaluation

In Figure 10, we illustrate that the high level control level, Maneuver, is integrating the information received from the three spatially distinct subordinates, propulsion, depth, and helm. At the first instance, propulsion reports that the submarine has achieved the required speed. At the second instance, helm reports that the areas of concern have been cleared of hostile objects. At the following instance, depth reports that the submarine has reached the required depth. Next, propulsion reports that the emergency motor has been engaged. Maneuver summarizes all the information and reaches the conclusion that it is ready to perform emergency ventilation.

5. **Supporting Other System Functions**

The earlier sections of this paper focused on the operators' roles in system control and in system development and service. There are, however, additional types of operators that support the control systems, including simulation operators and system trainees. When the control system is used as a trainer, the trainees may assume large parts of the system control, shown in Figure 2.

Figure 11: Simulation operator interface

The system may need to be operated at non or at multiples of real-time for particular simulation, training, or system testing needs. Animation may be beneficial.

5.1 Simulation Operator Interface

The role for simulator operators is primarily to set environmental states for performing tests or training operations. In our case study, the lube oil fire is triggered on and off from a button on the screen, as illustrated in Figure 11. A detailed description of the simulator structure is given in [1].

5.2 Animation

Animation, especially run in real-time, provides the most direct perception of a physical system when the actual system is not available or when it is not feasible for operators to touch and feel the environment. In RCS, we focus on real-time animation. Visualization is a most direct method to conceptualize a new system.

During system development, service, or testing, the simulation data can feed the graphic animation system. During operations, the sensory or world model data can be used.

We illustrate, in Figure 12, the animation of our submarine model transiting under ice. A detailed description of the animator structure is given in [1].

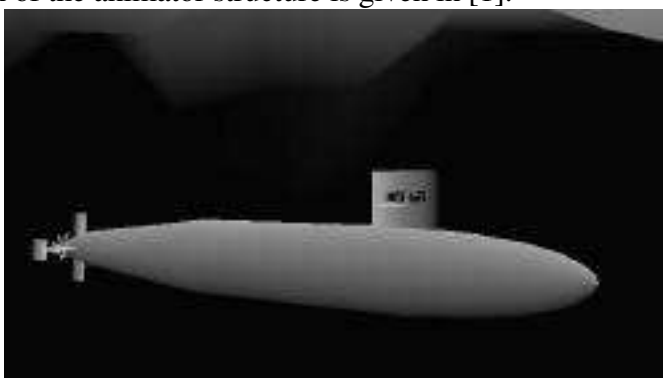

Figure 12: A submarine model

6. SUMMARY

We presented the results that we have obtained for this submarine automation operator interface issue. We proposed a logical structure for operator interface to support hierarchical real-time control system control. The following are some specific achievements:

* Well-defined operator interface logical structure.
* User interface organized as watch stations to suit distributed operational environment.
* Different levels of operator involvement.
* Different emphasis on debug and operational displays.
* Separate operator interface for control system and simulation.

To improve the operator interface system, we must generalize our illustrations. The following are some specific areas for improvement:

* Access control for different watch station displays - authorized operators only.
* Expand simulation station input - multiple casualties.
* Expand operator input capabilities - select any commands or parameters from the system task structure.

Additionally, there is an issue of whether an operator interface function is identified for an RCS control node to support the system operations or an operator her/himself is to serve as a control node. In this paper, we, as an initial effort, reference the latter situation as "man-in-the-loop." However, further study is needed to provide rigorous guidelines and methods to distinguish these two roles.

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. Richard Quintero of NIST managed and established the technical objectives for this project. 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

1 Huang, H., Hira, R., and Quintero, R., "A Submarine Maneuvering System Demonstration Based On The NIST Real-Time Control System Reference Model," in *Proceedings of The 8th IEEE International Symposium on Intelligent Control*, Chicago, Illinois, 1993, pp. 376-381.

2 Huang, H., Young, K., and Quintero, R., "Submarine Automation: Demonstration #5," NIST IR 5676, June, 1995.

3Albus, J.S., "Outline for a Theory of Intelligence," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 3, May/June 1991, pp. 473-509.

4 Huang, H., "An Architecture and a Methodology for Intelligent Control,"*IEEE Expert*, Volume 11, Number 2, April 1996, pp. 46-55, IEEE Computer Society, Washington, D. C.

5 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," Book Chapter in *Control and Dynamic Systems, Advances in Theory and Applications,* Volume 46, Academic Press, 1991, pp.173-254.

6 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, 1992, pp. 63-77.

7 Quintero, R. and Barbera, A.J., "A Software Template Approach to Building Complex Large-Scale Intelligent Control Systems," in *Proceedings of The 8th IEEE International Symposium on Intelligent Control*, Chicago, Illinois, 1993, pp. 58-63.

8 Corbett, M., "The Development of User-Centered Advanced Manufacturing Technology: New Design Practice of New Marketing Rhetoric," The International Journal of Human Factors in Manufacturing, Vol. 6 (2), pp. 79-87, John Wiley & Sons, Inc., 1996.

9 Mayhew, D. J., *Principles and Guidelines in Software User Interface Design*, Prentice Hall, Englewood Cliffs, NJ, 1992.