

Shipboard Controls of the Future

ABSTRACT: Recent advances in computer networking and control system technologies present an opportunity to improve the capability of naval shipboard control systems. Most existing digital machinery control systems merely replace one-for-one their analog predecessors. These recent advances motivate rethinking the basic role and architecture of shipboard controls. Traditional machinery system control has remained largely separate from combat systems and other ship information systems. Existing machinery control systems have concentrated on four functions: machinery status, control, system stability, and fault response. To implement these functions, custom systems have been designed, built and debugged for each class of ship. This lack of commonality has been expensive in terms of development costs, maintenance costs over the lifetime of the ship, and also the unrealized benefits stemming from prohibitive costs of adapting machinery controls to take advantage of emerging technologies.

This paper proposes a new paradigm for developing a shipboard control system based upon a functional decomposition of ships' missions that leads to defining technology independent interface standards. Multiple vendors may be able to independently develop control system hardware and software elements adhering to such interface standards without a priori knowledge of a particular ship application, leading to the ability to develop a total ship control system with low risk by integrating proven hardware and software elements to meet specific ship design requirements. With this new concept, other functions not normally associated with machinery controls are feasible: spontaneous reconfiguration after a damage event, integrated training, condition based maintenance planning, data archiving, operator assistance, and configuration management. This new approach may also allow for the integration of machinery controls into a total ship control system with seamless support for combat systems. This new shipboard control paradigm promises to speed control system development, improve performance, facilitate maintenance and modernization, and lead to lower life cycle costs.

Introduction

With the introduction of digital computer networks, naval machinery control systems are undergoing tremendous change. However, most of the new computerized machinery control systems are merely a one-for-one replacement of their analog predecessors. Machinery control systems are a barometer for the shipwide control system climate. Little effort has been expended to rethink the basic architecture of machinery control, much less the broader scope of shipwide control. Tremendous gains can be realized if we examine closely the true functionality of shipboard control systems and develop a control architecture that best implements those functions. This paper explores the issues involved with developing a shipwide control system architecture and proposes one approach for meeting shipboard requirements.

The premise of this paper is that, perhaps, before embarking on the design of specific naval ships' shipboard controls, a functional decomposition of naval ships in general will lead to an open, modular architecture which is compatible with changing mission requirements.

The present approach to shipboard controls is:

- Once a set of ship requirements is established, conduct a design which discerns between alternatives to yield a specific solution that fulfills the stated requirements. The design is optimized on acquisition cost, assuming that the stated ship requirements are met. Sister ships are to be as identical as possible.
- Throughout the life of the ship, convert and/or modernize to the degree possible or affordable.

The proposed approach is:

- Conduct a functional decomposition of naval ships in general. This will identify common functions, alternatively referred to as modules or even objects.
- Develop a shipwide architecture which is built upon functional modules and consistent interfaces.
- When a specific ship design is sought (that is, specific ship requirements are established), aggregate selected modules within the framework of the shipwide architecture, with its interfaces. Verify that ship requirements are met. Iterate. A small number of non-common (unique) modules may be identified. They will, however, be formulated with interfaces common to the other modules.
- Throughout the life of the ship, accommodate changes to the ship's mission by module insertion; upgrade modules when justified; scrap the ship when adjusting its mission becomes uneconomical.

This proposed approach to shipboard controls of the future is offered with some underlying assumptions. First, operators should occupy roles which capitalize on human strengths and mitigate weaknesses. Second, it is likely that hierarchical shipwide control systems are eminently more tractable than a monolithic control system. Third, only configuration data and a very few unique modules are necessary to accommodate differences between ships' missions. Last, missions, requirements, resources, and technologies change over the life of a naval ship.

This paper's intent is not to offer any answers; rather, an approach to pursuing some important ones is suggested. This paper is not based upon any specific ongoing research; rather, it is a statement prompted by recent experiences with Standard Monitoring and Control System (SMCS) development, Integrated Power System (IPS) development, and naval ship overhaul. Using machinery control systems as a point of departure, the discussion will provide the motivation/vision for an open, modular architecture. This is followed by a treatment of the intent of functional decomposition. Subsequently, the implications of functional decomposition for shipboard control hierarchy are explored, as are procurement implications, albeit summarily. The role of developing technologies is then touched upon briefly. The assertions made in this paper are supported by analogy with the highly successful personal computer (PC) industry, personal observations, or heuristically; significant research is required for quantitative support. Numerous examples of the proposed approach are provided to provoke thought and dialogue; the examples are not offered as "The solution."

Existing Systems

To date, most machinery control systems on U.S. Navy ships have been custom designed, point-to-point, hardwired systems dedicated to control of specific portions of the propulsion, auxiliary and electrical portions of the ship's support infrastructure. Even when control displays and remote actuation are brought into "integrated" consoles, the controllers themselves are independent. As a result of this design approach, a new controls development must be undertaken for each new ship class, with the associated costs and development risks.

The most modern existing systems are software-based, using digital computers. However, their architecture does not take full advantage of these technology advances. Existing modern control systems merely replace their older analog electrical predecessors on a one-for-one basis; see Figure 1.

Existing control systems suffer from a host of problems. The hardware encounters block obsolescence before a ship reaches the midpoint of its life. The software is difficult to maintain since it is ship-specific and requires significant rewrites whenever hardware upgrades are implemented. System and sensor calibration is difficult and laborious (i.e., expensive). Further, there is no embedded training capability, as exists for many combat systems. As a result, the ship's crew is forced to use "Post-it" notes to simulate a machinery casualty. Although newer systems use computers with limited built-in-test (BIT) capability, there is no system-wide BIT. The ship's crew is forced to do manual troubleshooting, which is very manpower intensive and time consuming for systems distributed throughout the ship, such as a machinery control system.

Outside of the gas turbine engine controller, which is distinct from the machinery control system, little or no automation is utilized. Existing systems simply report

conditions to the human operator and warn of out-of-range parameters, leaving equipment protection and control actions to the human operator to initiate. However, this is not an indictment of the designers of those systems because the design philosophies of the past placed little or no emphasis on automation. On the other hand, today's emphasis on reduced manning requires significant use of reliable automation, leaving humans to set priorities and conduct the warfighting tasks for which they are best suited. The more mundane tasks of system alignment, machinery protection and casualty response must be transferred to the control system if significant manning reductions are to be realized.

The developmental SMCS program has begun to address some of these issues, see Figure 2. It incorporates embedded training, a common man-machine interface at all operator stations and a network-based hardware architecture. However, SMCS was designed for forward-fit and back-fit into existing ships. Consequently, it does not address many of the issues cited above, such as common mode failures, improved sensors or the greater shipboard control problem.

The remainder of this paper intends to apply modern systems engineering principles to the shipboard machinery control problem and highlight desired capabilities that are either available today or in the very near future.

Towards an Open, Modular Architecture

SYSTEM DESIGN

In designing complex systems, determining the boundaries of different subsystems is very important. These boundaries and their interfaces play an important role in determining how the system is designed, constructed, tested, and maintained. Often, however, the boundaries are established with little thought as to the implications on the lifecycle cost of the system. This lack of forethought is probably a result of the manner in which systems engineering is practiced in the United States. Systems are designed to meet specific sets of requirements with little consideration for commonality with other systems or for robustness with respect to changing design requirements. A requirements driven design process often results in interfaces and boundaries that are optimized for a given set of requirements, but are not robust to handle changing requirements typical of large, long-lived systems. A careful design of the boundaries and their corresponding interface standards is critical to the development of robust systems that can handle a wide range of applications.

In the design of naval ships, one should recognize that firm design requirements will rarely ever exist. First, ship

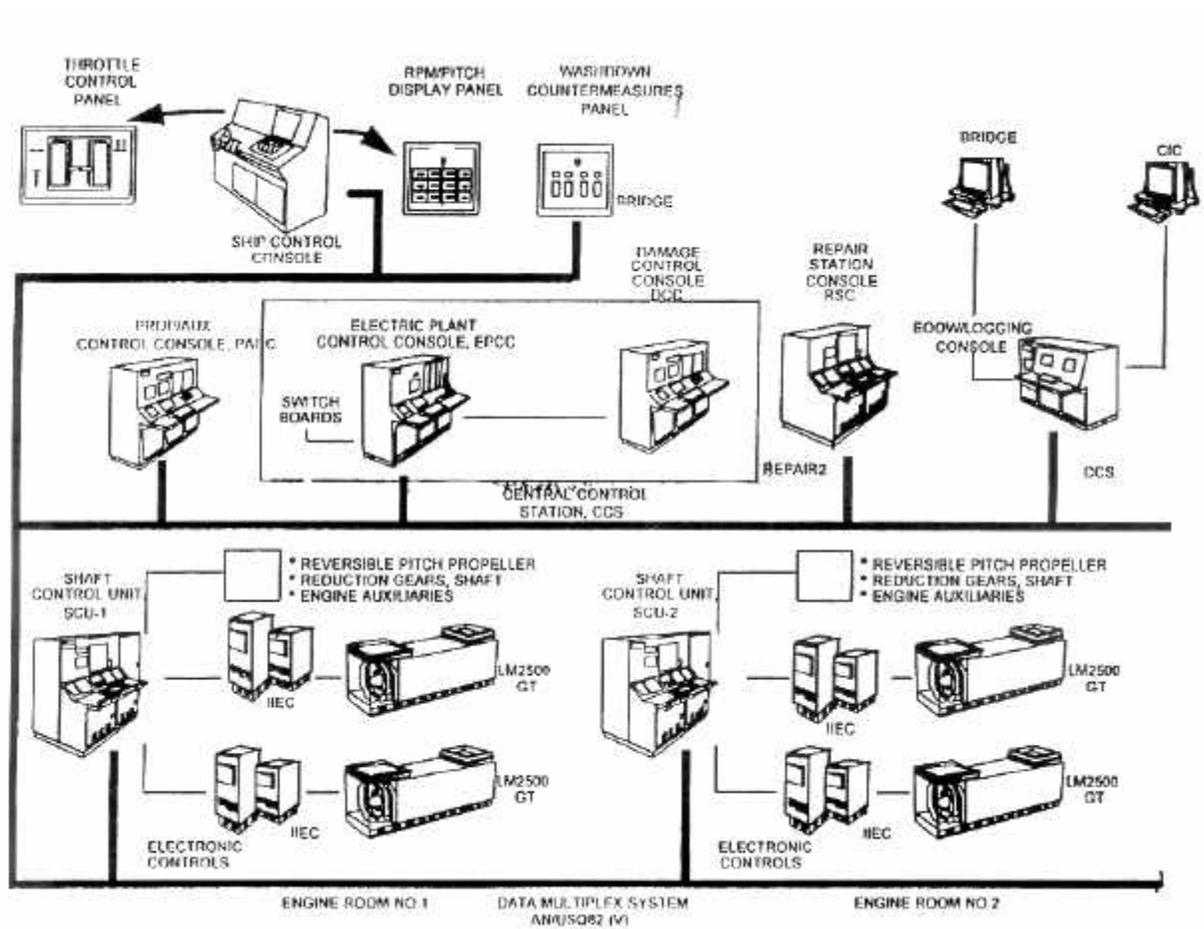


FIGURE 1. MCS Configuration for Flight I & II

requirements are dependent on world conditions that can change tremendously during the roughly fifty years from the first stages of design to ship disposal. One cannot even expect requirements to remain constant during the various stages of ship design. Second, the industrial base changes during the fifty year life which means products and processes available now may not be available in the future. Third, technology will change such that new products and processes will exist that will perform better and at lower cost than current systems. Finally, continuous process improvement necessitates changing requirements.

What does this mean? The classical requirements driven design process does not work for naval ship system design because the requirements cannot be "fixed" until late in the design process, if at all. Instead, one must concentrate on robust design such that a wide range of possible design requirements can be quickly addressed and such that change can easily be implemented. The ability to affordably change the system to meet new design requirements must be a design feature during both the period of pre-production ship design and throughout the ships' lives via ship

alterations (ShipAlts).

ADAPTING TO CHANGING REQUIREMENTS THROUGHOUT A LIFE CYCLE

To achieve a robust design, subsystems should be chosen to minimize the propagation of change across multiple subsystems due to changes in requirements or upgrades of equipment in one subsystem. Furthermore, the propagation of change should not cross multiple organizational boundaries with respect to alteration development. For shipboard control systems, this means that a particular ShipAlt package should be developed by a single organization and include all of the necessary hardware and software needed to implement the improvement. As a consequence,

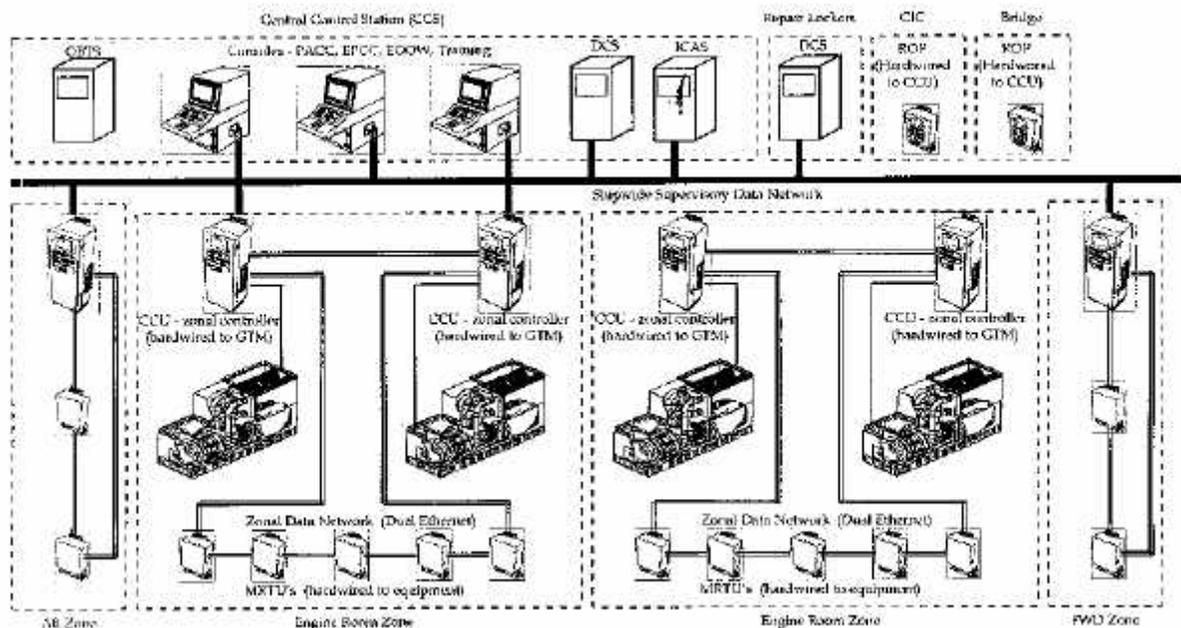


FIGURE 2. SMCS Open Architecture

multiple organizations will be involved with developing software elements for a shipboard control system given that numerous ShipAlts occur through the life of a ship. Hence, when one establishes a boundary in the software system, its interfaces must be specified completely enough to be able to write a contract with other vendors for ShipAlt packages.

In a PC analogy when you purchase a new printer, you also receive a disk containing the printer driver. In this way, the developer of the computer operating system does not have to get involved in adding the new capability to your PC. Of course the operating system vendor must have developed detailed interface requirements that the printer vendor must adhere to in order for the printer to work properly with the system. Furthermore, the operating system vendor provides the necessary tools to install the new device to the operating system configuration.

Currently, machinery control systems are developed by a single manufacturer at the time of ship design. If a ShipAlt is developed later, separate contracts are awarded for the machinery control system changes and for the equipment changes. Coordinating these two efforts is expensive and not always successful. In the PC analogy, this would mean that the operating system vendor would be required to develop and distribute new printer drivers every time a printer vendor developed a new product. The ability of the printer vendor to rapidly bring out new products and to optimize the printer driver would be greatly hindered. Likewise, the ability of the operating system vendor to keep up with the new

products of hundreds of printer vendors is impossible.

FUNCTIONAL DECOMPOSITION IMPLICATIONS FOR CONTROL SOFTWARE

Software engineers often address the partitioning of software by establishing protocols at different hierarchical levels. The lowest level is generally reserved for hardware specific software that provides a consistent software interface to a wide range of hardware. The middle layers are typically associated with communicating the information between multiple computers and different elements of a computer system. The highest layers deal with accomplishing the actual function of the software system using generalized software. A PC once again provides a good example. At the lowest level, the BIOS software provides a consistent software interface of the system hardware to the operating system. At the middle level, the operating system, such as Microsoft Windows, communicates information between the devices of the system and the application software. At the highest level, the application software, such as Microsoft Word, accomplishes the requirements of the user.

Naval machinery control systems such as the SMCS have also relied on multiple layers of protocols. However, the machinery control system for each particular project has been viewed as the application software and has been

custom designed for each ship class. Newer systems, including SMCS, have been developed to re-use as much software as possible between ship classes, but have not gone so far as to establish a control system protocol between the core machinery control system software common to all ships and the ship specific application software to control the equipment on a particular ship. The authors believe that establishing such a protocol is in the interest of the Navy and the general machinery control system community and is at the heart of this paper.

The question becomes how to partition the functions performed by a shipboard control systems into the core shipboard control system software and the application control software. A functional decomposition of the generalized requirements of a shipboard control system is one method.

AN EXAMPLE OF FUNCTIONAL DECOMPOSITION/MODULAR ARCHITECTURE

Traditionally, machinery control systems have implemented four functions directly: machinery status, control, system stability, and fault response. Other computer based functions, also important to the successful operation of the machinery plant, have not traditionally been incorporated in machinery control systems. These functions include integrated training, condition based maintenance, data archiving, operator assistance, and configuration management. Often, a different machinery control system is developed for each machinery system even though control displays and remote actuation is brought into an "integrated" console. The propulsion plant will have a separate control system from the electric plant which, in turn, will have a separate control system from the various auxiliary control systems. However, if one examines these multiple systems, many have the same basic functions: produce a commodity, store the commodity, transport the commodity around the ship, and distribute the commodity to users.

Equipment on a ship can be classified by relationships to the missions of the ship. Certain equipment, such as the propulsion system and combat systems are directly related to achieving the mission of the ship. These systems, however, require services and commodities to function. These services and commodities are supplied by support systems such as the electric plant, chill water system, firemain, hydraulic plant, and so on. These support systems, in turn, may require other support systems to operate. The firemain system for example requires the electric plant to operate. If we functionally decompose a ship in this manner, we come up with the following structure.

Mission Systems (what the ship is supposed to do)

- Mobility (propulsion plus steering)
- Combat Systems
- Command and Control
- Cargo Handling and Storage

Resource Systems (needed to support mission systems and other resource systems)

- Electric Plant
- Auxiliaries
- Messing and Berthing
- Firemain
- Fuel Storage and Transfer
- Data Network
- HVAC
- Crew
- Hull
- Training
- Integrated Logistics Support
- Maintenance Support

Note that the crew of the ship is a resource and not a mission of the ship. The crew is on the ship to provide context and priorities to the mission systems and the other resource systems. The crew is one of the more expensive resources because of the requirements it levies on other systems. The crew is, however, the most affordable means to add intelligence to the ship. This means that, optimally, the functions assigned to the crew should be those which require human intelligence. In a shipboard controls environment, the crew's role should be to prioritize the actions of the ship's systems. Routine functions such as log-taking, setting standard system line-ups, and so on, should be left for the control system to perform since little actual intelligence is required to perform them. The bottom line is that the crew's purpose is to establish the context under which the ship's systems are to operate. The shipboard control system then operates the ship's systems within the bounds of the context established by the operator.

Functional Decomposition

Functional decomposition applies to mission systems and resource systems including their control systems. A functional decomposition forms the basis of modularity. A module is defined to be a specific, assigned function at a chosen, hierarchical level of decomposition. Boundaries of a module (function) enclose all the necessary pieces for performing the assigned function. Modules' functional and physical boundaries should be coincident.

The boundaries between modules constitute the systems' interfaces and are defined/described by interface standards. Interface standards characterize the interface, permitting compatible module design. Interface standards also constrain, place limits on, the behavior of contiguous modules. Adherence to interface standards is a necessary, although not sufficient, condition for system viability.

A general, consistent functional decomposition across fleet applications will identify a minimal, yet broad-scope, set of common modules and common interfaces. Such common interfaces permit utilization of identical modules in multiple fleet applications. Performance requirements

for the modules' function and adherence to common interface standards permit competitive procurement of interchangeable modules.

Functional decomposition and its concomitant modularity enables decentralized, composite systems (decentralized, composite control). Each module within a system, be it a resource system or mission system is envisioned to possess its own intelligence. Module behaviors are actuated through indigenous control; this "local" control seeks to match products with orders for products.

The module's control conducts dialogue with supervisory control, and only supervisory control. Only the interface between the module's indigenous control with supervisory control is specified. This allows wide latitude in physical module and module control implementation.

Immediate module control actions, necessary for safety or to protect equipment, are made by the module's indigenous controller in the event of perturbation. Immediate module responses to contingencies may be specified by supervisory control. Module configuration and condition data is collected and saved by indigenous module intelligence.

Although decentralized and composite in nature, orchestration of modules is achieved by supervisory control modules. Supervisory control modules' decision-making, control-order time scale should roughly be on the order of an alerted human. This level of control would be the point of interface with operators. This level of control would enable the execution of the operator's mission priorities.

EXAMPLES OF FUNCTIONAL DECOMPOSITION

To this point, a specific mission or resource system has not been discussed. The foregoing discussion is general but provides basic examples of functional decomposition.

A functional decomposition could resolve a "command and control" mission system into the following: data collection modules, data analysis modules, data/command communication modules, data presentation modules and data storage modules.

Consider a heating, ventilation, and air conditioning (HVAC) resource system. A functional decomposition could break this system into the following: circulant production modules, circulant transport modules, circulant energy transfer modules and circulant/energy control modules.

Consider a maintenance support resource system. A functional decomposition could break this system into the following: equipment performance specification library module, equipment condition data collection modules, communication modules, data analysis modules, maintenance resource data collection modules, and maintenance action control modules.

Consider a supervisory control system. A functional decomposition could break this system into the following:

operator interface module, software support module, hardware support module, resource manager/mission controller communications module, and so on.

IPS MODULAR ARCHITECTURE

The IPS program uses a functional decomposition of shipboard propulsion and electric power systems as the basis of its open, modular architecture, reference [1]. Here a shipboard propulsion/electric power system has been resolved into six modules, power generation, power distribution, power conversion, energy storage, power loads and power control. In addition to the IPS program developing the performance requirements of such modules, the interface standards associated with a shipboard power system are being developed. The modules' performance requirements and interface standards comprise the means to procure an Integrated Power System. Additionally, the control structure within the IPS architecture is hierarchical. Each module, except power control, has a local controller. The modules' local controllers interface with supervisory level controllers only, the power control modules. The modules' local controllers do not interface directly with any other type of module.

Uniting Shipboard Controls of the Future

CONTROL HIERARCHY

One can visualize a shipboard control system based on the separation of systems between mission controllers and resource managers. Human operators establish the relative priorities of the various shipboard missions. Multiple mission controllers (one for each mission) would translate these mission priorities into prioritized requirements for resources from the resource systems. These resource systems in turn, would have resource managers that would either meet all the resource requirements made on it, or allocate the available resources to the highest priority users. Each resource system can also levy prioritized requirements on resources from other resource systems. For example, the combat systems may require 500 kW of electric power from the electric plant with a high priority. To meet this requirement, the electric plant resource manager would place a high priority requirement on the seawater cooling system for the required flow rate needed to cool the generator sets. The seawater cooling system resource manager would then flow a requirement back to the electric plant resource controller for the number of kW required to supply the seawater.

The scheme described above has some very advantageous architectural features. First, each of the mission controllers and resource managers can perform its jobs with a very simple interface with other resource managers. Consequently, the task of integrating the software for the multiple mission controllers and resource managers is greatly simplified. This facilitates multiple organizations

to develop the different elements of the machinery control system and be assured that the system as a whole will work. The system is also scaleable in the sense that initially, not all resource systems need to have the resource manager software. While the full benefit of this scheme is not realized if some resource systems are not integrated, it can be compensated for with extra redundancy, as in systems of today. The scheme is also robust because changes to equipment on a given ship only require a modification to the appropriate resource manager or mission controller.

For normal operation, this scheme may seem somewhat complex. Today, resource systems are generally designed to meet all demands, regardless of priority. If however, demand exceeds capacity, such as when damaged in battle, the resource system will either fail completely, or the human operator is called on to isolate the damage and determine which loads should be allocated the resources. Unfortunately, for even a moderately sized ship, few human operators can determine precisely which loads should be supported for meeting the operational requirements of the ship. As described above, the flow of resource requirements from one resource system to another can result in resource priorities that are not obvious at first. Consequently, the human operator is not best suited for rapidly establishing priorities for given loads. The human operator is, however, ideally suited for establishing priorities for the ship's missions. In terms of enabling the human operator to establish mission priorities and flow these down to load priorities, the scheme described is relatively simple.

MISSION CONTROLLERS AND RESOURCE MANAGERS

With this scheme, the shipboard control system is partitioned into a number of resource managers and mission controllers. A proposed listing of, the mission controllers and resource managers which are anticipated to be common to all ships follows.

Core Mission Controllers

- Command
- Damage Control
- Mobility Controller

Core Resource Managers

- Ship Properties Information Manager
- Ship Compartment Information Manager
- Human Resource Manager
- Computational Resource Manager
- External Communications Resource Manager
- Internal Communications Resource Manager
- Electric Power Resource Manager
- Fuel System Resource Manager
- Firemain Resource Manager
- Potable Water Resource Manager
- Sewage Resource Manager
- HVAC Resource Manager
- Cooling Water Resource Manager

In addition to these core elements, one can imagine a number of other possible controllers and resource managers specific to certain types of ships.

Optional Mission Controllers

- Antisubmarine Warfare Mission Controller
- Antisurface Warfare Mission Controller
- Antiair Warfare Mission Controller
- Command and Control Mission Controller
- Electronic Warfare Mission Controller
- Information Warfare Mission Controller
- Cargo Handling Controller

Optional Resource Controllers

- Freshwater Cooling Resource Manager
- Aqueous Film Forming Foam Resource Manager
- Hydraulics Resource Manager
- Maintenance Resource Manager
- Training Resource Manager

EXAMPLES OF CONTROL AND RESOURCE MANAGERS

This scheme also offers the ability to define resources in ways which facilitate the minimization of ship specific software. Information, for example, can be classified as a resource. As such, information about a ship's compartment could be controlled by a ship compartment information manager. This information would include the compartment geometry, the list of equipment in the space, the compartment physical properties such as flooding status, fire status, smoke status, compartment temperature, etc. While the data which is handled by this information manager would be very ship specific, the resource managers for the other commodities, such as electric power, could be developed to work with arbitrary compartmentalization schemes. For a particular ship, compartmentalization information is obtained from the ship compartment information manager during program execution. In this manner, the electric power resource manager could automatically reroute power around a flooded compartment without having to determine itself if the compartment is flooded. The ship compartment information manager can also be the repository for configuration control data concerning the equipment and layout of that equipment within each compartment. In fact, the ship compartment information manager could contain the master CAD product model from which all of the ship's drawings concerning that compartment are derived.

A ship properties information manager could also prove useful. It would manage information about the entire ship to include the overall compartment layout, the zone boundaries, ship naval architectural properties such as draft, displacement, trim, stability, etc., environmental conditions such as sea state, seawater temperature, air temperature,

wind speed and direction, water depth, seawater salinity, etc., and navigational information such as ship's position, heading, and speed.

A human resource manager could also provide a means of integrating a standard method for the other resource managers to present information to the operators and to receive commands from the operator. The human resource manager would include the software for user authentication, and the algorithms for translating the resource manager information into display pages for the operator. A means of registering/deregistering new equipment with the various resource managers and mission controllers would also be desirable.

A computational resource manager could provide the network and network operating system resources necessary for the operation of all the other resource managers and their communication with each other. It would include all the networking hardware and software, the application servers with their associated server software, and the data servers with their associated server software. The computational resource manager could determine which of the other resource managers is disabled (based on priorities established by the mission controllers) in the event that severe damage disables a large fraction of the computational infrastructure; this would enable other resource managers to continue functioning.

THE HIGHEST LEVEL OF CONTROL

The highest level of shipboard control is command, which intelligence resides in the captain. Many of the proposed resource managers support this highest level of control directly, many indirectly. This implies that the command module has an interface with the resource managers.

The command module, the highest level of control, achieves actuation of its commands by relaying control instructions to the core and optional mission controllers. This implies that the command module has an interface with the core and optional mission controllers. Furthermore, the command module does not necessarily have a prescribed location or a specific algorithmic process. It must reflect the humanity of the intelligence behind this module.

A rigorous functional decomposition should be conducted to verify the command module as a valid function. Then, defining the implied common interfaces with the resource managers and core and optional mission controllers would serve as a starting point for the development of a total shipboard control system. The second necessity in the development of a command module, given the interfaces are defined, is a performance requirement. In terms of the command module, this performance requirement is inextricably tied to human analytic capability. This implies that the goal of such shipboard controls of the future is to enable the total ship system to support the captain and crew in mission accomplishment. Shipboard controls should not require the captain and

crew to support the ship in mission accomplishment.

Procurement Implications of Functional Decomposition

Government acquisition policy seeks to position procurement approaches on the spectrum between "pure performance" specifications and "non-deviation" drawing specifications, see Figure 3.

"Pure performance" specifications allow great latitude to vendors in implementation; herein the vendor assumes risk. This type of specification complicates configuration control and logistics support, but encourages vendors to compete, particularly if a vendor perceives that their product or process has the requisite qualities for a competitive bid.

"Non-deviation" drawing specifications place the responsibility for performance entirely on the government. Configuration control and logistics support are simplified. Interchangeability is inherent. The fields in which vendors can compete are very limited, possibly leading to sole source situations.

Contemporary defense acquisition policy favors "pure performance" type specifications. This approach allows vendor latitude in meeting specifications which favors diverse competition. Configuration control and logistics support can be accommodated to some degree by adding a requirement of adherence to common interface standards to "pure performance" specifications. This leads to a "coupled" performance/interface specification.

As discussed previously; modularity arising from functional decomposition and the functional performance requirements form the basis of a "coupled" performance/interface specification. This allows improved acquisition and life cycle support of modules.

Successive new applications (new ship classes), having undergone functional decompositions, can utilize those already developed, procurable, supportable modules. The research and development effort traditionally encountered in new ship classes is reduced by introducing fewer new systems and components (including software). This approach, in turn, reduces the introduction of new logistics support requirements and reduces system/ship integration efforts.

A module, which is foreseen to be widely applicable, could be procured in quantities (at a production rate) sufficient to reduce per-unit costs. Vendors won't/don't need to know the final application.

Future Technologies

There are numerous emerging technologies that can be applied to shipboard controls. Many of these support development of the modular architecture described above.

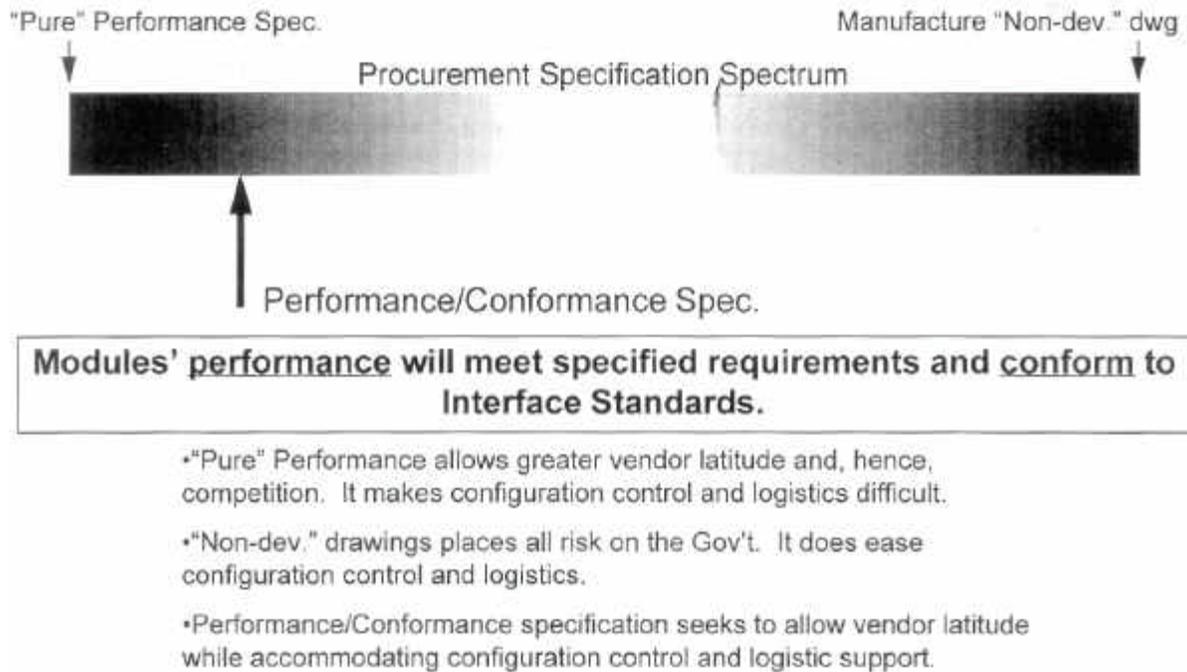


FIGURE 3. Procuring Modular Systems

These technologies can be grouped as either hardware or software. Hardware improvements include wireless and fiber-optic sensors, self-calibrating sensors, commercial and non-developmental computing resources, modern CRT, LCD and other color displays and virtual environment technologies.

There are many software improvements that can be included in new control systems. Analytic redundancy is a technique that allows for duplicate control algorithms to be run at the same time. One control algorithm host is primary and one (or more) are running in hot standby on separate computers to take over if the primary fails. This technique can be used to eliminate common mode failures in the software algorithms and has already been deployed in commercial and military aircraft. Graceful degradation of software and communications links will allow the ship to fight hurt and maintain the vital systems needed to support the combat system. Electrical signature analysis may be used to diagnose the health of many mechanical equipment such as pumps, fans and valves by analyzing the current and voltage waveforms of the electric motor driving the device. This would be a significant improvement to the condition-based maintenance efforts that are presently in their infancy by allowing the more than 5000 electric motors installed on a Navy ship to be monitored.

A true plug and play software system could

eliminate the need for expensive control system software upgrades every time a new piece of hardware is installed aboard ship. Application software that is hardware and operating system independent would allow a single set of application software to be used across multiple ship classes. The SMCS program has partially demonstrated this in that the software has been configured to operate on Microsoft Windows-NT, UNIX and on embedded computers with minimal changes; however, true hardware independence has not yet been achieved.

Conclusion

There are numerous research projects underway that address pieces of the shipboard control problem, many of which are making significant progress in their respective areas of interest. There are programs attempting to add automation to ships, develop a more reliable LAN and computing infrastructure, develop more reliable and self-calibrating sensors, develop a common operating environment for shipboard computing systems, automate the damage control process and implement condition-based maintenance, just to name a few. Table 1 lists some of the U.S. Navy research programs presently underway or proposed. However, there is no overarching systems engineering approach being applied to the "big picture" problem of ensuring all the ships' systems are utilized in an optimal fashion to support the immediate mission of the ship.

TABLE 1

Partial listing of Control and Automation R&D Efforts

Program	Funding
Smart Ship	Various
RSVP (proposed ATD)	ATD
Intelligent Ship Control (ISC)	6.2 ONR
Standard Monitoring & Control System (SMCS)	6.3 N86
Surface Ship Automation (SSA)	ARPA
Advanced Control (ADCON-21)	SC-21
Hyper-D (a computer node network concept)	??
Damage Control System (DCS)	Tycom
Integrated Condition Assessment System (ICAS)	Tycom
Damage Control Automation for Reduced Manning (DC-ARM)	6.3 ONR
Enhanced Machinery Control System (EMCS)	PMS-400D

This paper has attempted to point out that shipboard controls must be viewed from an entirely different standpoint than in the past. Revolutionizing the shipboard control system is one of the key changes that must be made in the way we design and build ships if the manning reduction goals of future ships are to be achieved.

Control system design and engineering follows directly the overall ship system engineering approach. In this context, "overall ship system engineering" refers not to a specific ship program but, rather, to how the U.S. Navy engineers its ships. The proposed approach follows:

- Conduct a functional decomposition of naval ships in general. This will identify common modules.
- Develop a shipwide architecture which is built upon functional modules and consistent interfaces.
- When a specific ship design is sought (that is, ship requirements are articulated), aggregate selected modules within the framework of: the shipwide architecture with its interfaces. Verify that ship requirements are met. Iterate. A small number of non-common (unique) modules may be identified. They will, however, be formulated with interfaces common to the other modules.
- Throughout the life of the ship, accommodate changes to the ship's mission by module insertion (removal); upgrade modules when justified; scrap the ship when adjusting its mission becomes uneconomical.

Shipboard controls have been developed for each ship program. Perhaps the best method for the future is to determine, through research and development, an architecture for shipboard controls for naval ships in general and then adapt it to the specific ship programs of the future. Hence, the approach articulated in this paper is for no particular ship program but for all programs in the future. We believe that this approach is best (most farsighted) for the future operators and fleet staffs who must operate and maintain these ships in a changing world, not

necessarily for the organizations whose interests are parochially one ship class.

Before a "control system hardware backbone," or an intelligence hardware infrastructure," or "software libraries," or even a "common control language" are designed, what must be done on a ship, what must do it, and how well it has to be done, have to be defined. Following a functional decomposition, systems can be developed and acquired in a way which is beneficial to the Navy. It is not likely that such an approach will lead to monolithic structures, quite the contrary. To think of a total shipboard control system as a set of hardware and software which "integrates" the other control systems on a ship is to miss the point entirely. To think of a total shipboard control system as that from which specific controllers are designed is to embrace an approach which may make the U.S. Navy's ships more procurable.

REFERENCES

- [1] Doerry, N. H., J. V Amy Jr., C. Petry and H. Robey. "Powering the Future with IPS," *Naval Engineers Journal*, May 1995.

LCdr. John V. Amy Jr., USN graduated from the United States Naval Academy with a B. S.E.E. degree in 1983, and then served as antisubmarine warfare officer on USS Boone (FFG 28). He then reported to M.I.T where he earned an S.M.E.E.C.S. degree, Naval Engineer degree, and a Ph.D. in naval electric power systems. An engineering duty officer, he was assistant protect officer for aircraft carrier overhaul at Supervisor of Shipbuilding, Conversion and Repair USN Newport News, Virginia assigned to the refueling complex overhaul of USS Enterprise (CVN 65) and her subsequent post-shakedown availability. Presently, he is the deputy program manager for the Integrated Power System (IPS) Program at the Naval Sea Systems Command.

LCdr. Norbert H. Doerry, USN graduated from the United States Naval Academy with a B.S E.E. degree in 1983, and then served as gunnery and fire control officer on USS Deyo (DD 989). He then reported to M.I.T. where he earned a S.M.E.E.C.S. degree, Naval Engineer degree, and a Ph.D. in naval electrical power systems. An engineering duty officer he was assigned to the Advanced Surface Machinery Programs for the development of the Integrated Power System from 1992 to 1995. Presently, he is assistant project officer for aircraft carrier construction at Supervisor of Shipbuilding, Conversion and Repair USN Newport News, Virginia.

LCdr. Timothy J. McCoy, USN graduated from the University of Illinois with a B.S. degree in marine engineering in 1982. He was commissioned in 1985 and served as first lieutenant and communications officer on USS John Young (DD 973). He then was assigned to Supervisor of Shipbuilding, Conversion and Repair USN San Diego, California as repair ship superintendent, repair project manager and new construction ship superintendent. Subsequently, he attended M.I.T., earning a S.M.E.E.C.S. degree, Naval Engineer degree, and a Ph.D. in naval engineering. Presently; he is assigned to Naval Surface Warfare Center Carderock Division, Annapolis Detachment where he works as the technical manager for the Integrated Power System Program.

IMPORTANT NOTE:

This Document was created by scanning the original manuscript and converting the scanned image into a Microsoft Word Document using OCR technology. Although effort was expended to ensure the text of this copy matches the original, there is no guarantee that this document exactly matches the content of the original. Furthermore, no attempt was made to exactly match the page layout or fonts of the original.

This document originally appeared in the *Naval Engineers Journal*, VOL 109, No. 3, published by the American Society of Naval Engineers (ASNE) in May 1997.