Ryan Downs, MPR Associates

Design, Development and Deployment of Automated Distributed Control Systems on Active Navy Surface Combatants

ABSTRACT

As the Navy builds ships with more complex, distributed systems, centralized control of distributed systems becomes a single point of failure. This is particularly important for recovering from damage, when the rapid execution of complex tasks often is necessary to effectively recover from damage. A control system that is vulnerable to damage when it is needed most, when recovering from damage, presents a significant survivability concern.

The Navy Research Laboratory's Damage Control Automation for Reduced Manning (DC-ARM) program demonstrated that automated "distributed control" results in more a survivable control system than the centralized control architecture typical aboard today's Navy ships. The Chilled Water Automation System being installed aboard new construction DDG 51 Class destroyers is the first installation of the "distributed control" technology developed by the DC-ARM Program. Live fire weapon effects testing for the DD (X) program demonstrated the superior survivability of this "distributed control" technology for fluid systems.

As used in this paper, "distributed control" is the distribution of control capabilities throughout a system as opposed to "central control" in which control functions for a system are performed by a central computer (or by redundant, yet still centralized control computers). A key feature of properly designed distributed control is that there is less dependence on long communication paths between sensors, the control processor, and the control actuators. This reduced dependence on vulnerable communications improves the survivability of the controls.

In a properly designed distributed control system, the control decisions at each distributed component are likely to be simpler than a total system control logic in a centralized processor. The resulting simplicity makes the development and maintenance of a distributed system easier, and it typically leads to a more robust, more reliable system.

To achieve the benefits of a distributed control system, it must be properly designed.

This paper highlights a successful control system design approach, how that approach was applied in a complete prototype control system and how aspects of that control system were transitioned for shipboard application.

INTRODUCTION

Automating the control of distributed systems, such as the firemain and chilled water systems, is an enabler of optimized shipboard manning and of improved damage control performance. To achieve these objectives, the automation must function after the ship, and the associated distributed systems, take damage. The survivability of the controls for distributed systems, therefore, is essential. The centralized control typical aboard Navy ships today is a single point of failure; as such, it is not adequately survivable to meet the objectives of effective, or improved, damage recovery with a reduced crew.

Proper design of the control system is imperative for the successful execution of complex tasks, particularly in post-damage situations. As the complexity of systems and system functionality increases, central control of all system processes becomes so complex that reliable design and performance become exceedingly difficult to achieve. Much like the shipboard organization

where responsibility of decision-making is delegated to the lowest practical level, so too does a well-designed automated distributed control system delegate control to individual devices. A centralized control system has survivability risks and may be too complicated to successfully develop all of the necessary control algorithms. Similarly a system using only device level controls (i.e. logic only at each device) may be inappropriate and too complicated to develop effectively. The objective of the design, therefore, is to achieve an optimum balance for distributing controls throughout a system which provides the most efficient, easy to manage, cost effective control system with sufficient survivability.

A BRIEF OVERVIEW OF CONTROL SYSTEM ARCHITECTURE

Centralized Control

First generation control systems are governed from a primary (central) location. All information is collected from remote locations, returned to the central processor where decisions are made. The commands based on these decisions are communicated from the central processor back to the remote locations where the actions are executed. These systems have the ability to access a tremendous amount of data from remote locations to make a complete decision. A centralized control system is applicable where:

- Probability of system damage is low (including damage to communication paths)
- Consequence of system damage is low, and
- Response time of the system is not important.

Centralized control, therefore, is appropriate for non-essential applications such as office building ventilation. Figure 1 shows a notional centralized control system where all sensors report to a single location and all actions are controlled from the same location.

The central processor concept is not suited for situations where the probability of system damage and/or the consequences of system damage are high or where the response time of the system is critical due to communication lag between sensors, the central processor and end effectors as well as excessive processing loads required of a central processor. A single point of failure in the central processor can prevent an entire system from performing its intended function.

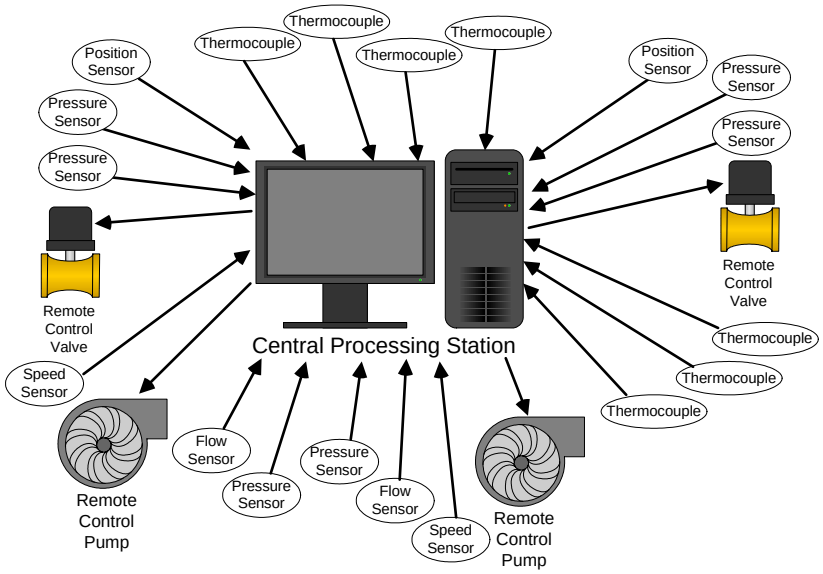
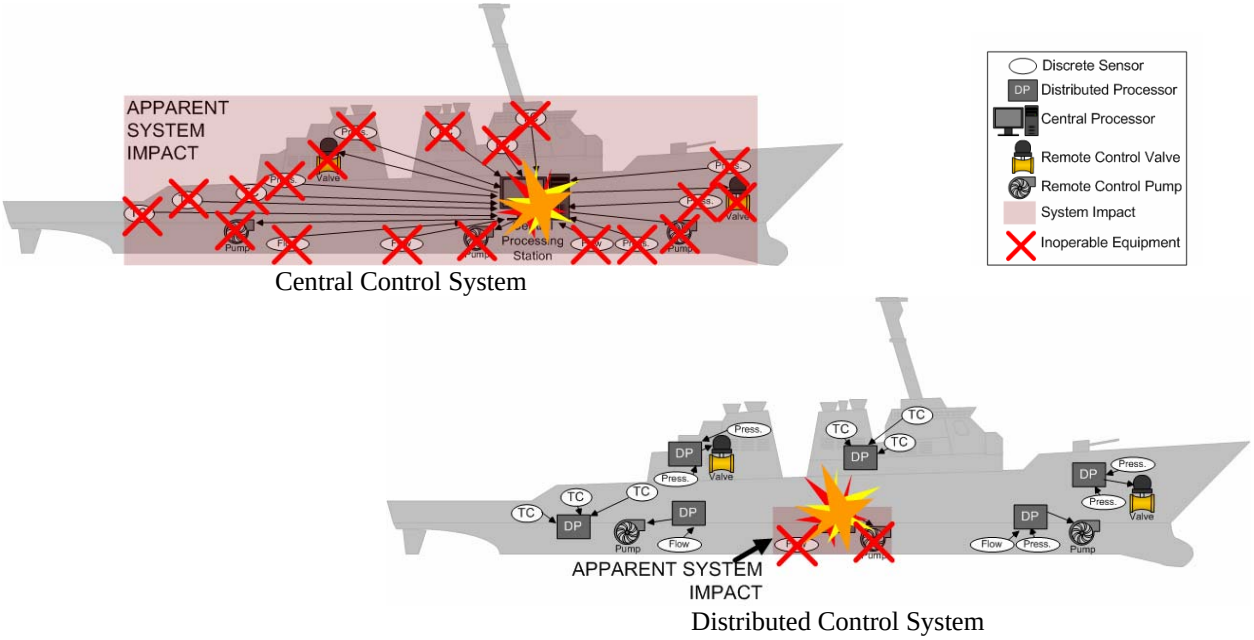


FIGURE 1 Notional Centralized Control System

A central control system is also vulnerable to damage to any communication path to and from remote locations where data may be collected or actions may be executed. Long communication paths increase the target area (damage location which impact system performance) which can render a centralized system inoperable. A hit anywhere in the target area can make the system inoperable. A small target area, therefore, is more survivable because it is less likely to get hit. Even if redundant central processors are installed to improve the survivability of the controls, the system still is vulnerable to the loss of communications paths. The converse of this example is also true, where damage to the central processing station can render all of the system components throughout the ship inoperable even though the individual pieces of equipment did not sustain any damage. A distributed system can sustain damage to any distributed processing location but will only lose the functionality of the equipment in the immediate vicinity of the distributed processor, controlled by the distributed processor, which also was likely damaged in the event. Other independent distributed stations elsewhere in the ship will continue to operate providing some level of functionality in all damage scenarios. Figure 2 shows the difference in apparent system impact for a central control system with remote equipment, where damage to the central process can result in total system loss, and a distributed control system with localized processing. Damage to the central processor can render the entire system inoperable for centralized control while damage is localized and system impacts are contained to area controlled the single distributed processor resulting in increased system availability.



Central Control System

Distributed Control System

FIGURE 2 Apparent System Impact to a Small Damage Event

Distributed Control

Distributed control is a relatively new concept in control system design. Microprocessors have become smaller and less expensive, allowing them to be installed in smaller devices and distributed throughout a system. The control logic executed on each distributed processor must be properly designed to benefit from the distribution of the control processors. An improperly designed distributed control system, on the other hand, will have few of the benefits of being distributed and most of the negative attributes of both a central control system and a distributed control system. A properly designed and implemented distributed control system has the following requirements:

- Each distributed processor only requires data available in the immediate vicinity of the distributed processor to perform its function. (The "immediate" vicinity could be the equipment itself, e.g. a valve or pump, the area local to the equipment, a single compartment, or, perhaps, a zone in the ship.
- Each distributed process only controls equipment in the immediate vicinity of the distributed processor.

A common example of a distributed element is an electric circuit breaker. The circuit breaker opens a local switch when the local current exceeds the threshold. A collection of circuit breakers is a basic distributed control system. A distributed control system is beneficial where:

- Probability of system damage is high
- Consequence of system damage is high
- Data and/or actuation locations are physically separated by significant distances from other data and/or actuation locations. (Each remote location should have its own distributed control for local data and/or actuation.)
- System is complex so that developing software for all scenarios and failures is difficult or impossible, and
- Response time of the system is important.

Distributed control systems are effective when quick local reaction time is required such as in the circuit breaker example above. If the circuit breaker was required to transmit the current measurement to a remote location for further processing and wait for instructions from the central processor, the electrical system or downstream equipment could be damaged.

It is very important to note that to achieve the benefits of distributed control, the distributed control processor should require data only from its immediate vicinity to make its control decisions.

Distributed control systems can interface with users at distributed or central locations for the purpose of monitoring and/or human control. A centralized operator interface should not be confused with central processing. The primary functions of the distributed processor are still performed locally even though the operator can observe and control the distributed devices from central or remote locations. The key in a pure distributed system is logic should be executed at the most distributed (lowest) level where all of the data required to execute an action is located.

Distributed systems can be implemented to achieve high survivability. This makes distributed control applicable for shipboard critical systems such as:

- firemain system
- chilled water system
- fuel systems
- electric power distribution, and

non-shipboard applications such as:

- chemical processing and,
- emergency systems.

Hybrid Control

In practice, most properly designed control systems will be a mix of distributed control and central control. Some of the system functions can be executed at the lowest level because all of the data to make a decision is available at the low level. Other system functions that do not have sufficient data locally must be performed at a higher, more centralized level, but not necessarily the highest level. Again, the function should be performed at the lowest level where all of the information for the decision is available. There are several good examples of hybrid control systems.

HUMAN BODY

The oldest, yet most elegant example of a properly designed control system is the central nervous system of the human body. The brain can be considered the central processor. The spinal cord node centers can be considered a mid-tier processor each communicating with many nerves (sensors) and muscles (actuators). All spinal cord node centers interface with the brain.

When the hand touches a hot surface, the hand pulls away before the person realizes any pain or comprehends what occurred. The local spinal cord node centers (distributed processor) located throughout the spinal column receive information from the local nerve endings in the hand, there is an excessive heat source. The local spinal cord node center has all the information required to recognize the problem and send a command to the hand to move away.

The spinal cord node center reacts locally and the upper processing tiers may not need to be aware of what transpired. Distributed processors have the ability to react with considerable speed as it is not necessary to transmit information long distances for processing or to invoke an action. The left side of Figure 3 depicts the simple process of a reflexive action. Central processing is reserved for the most complicated decisions, those requiring data from many distributed devices. Walking is a good example of central processing as it requires multiple sub-systems including the eyes for navigation, the legs for motion and the arms for balance. The brain directs multiple spinal column node centers which invoke action. The right side of Figure 3 depicts the method the human body uses to process and execute simple and complex decisions and actions.

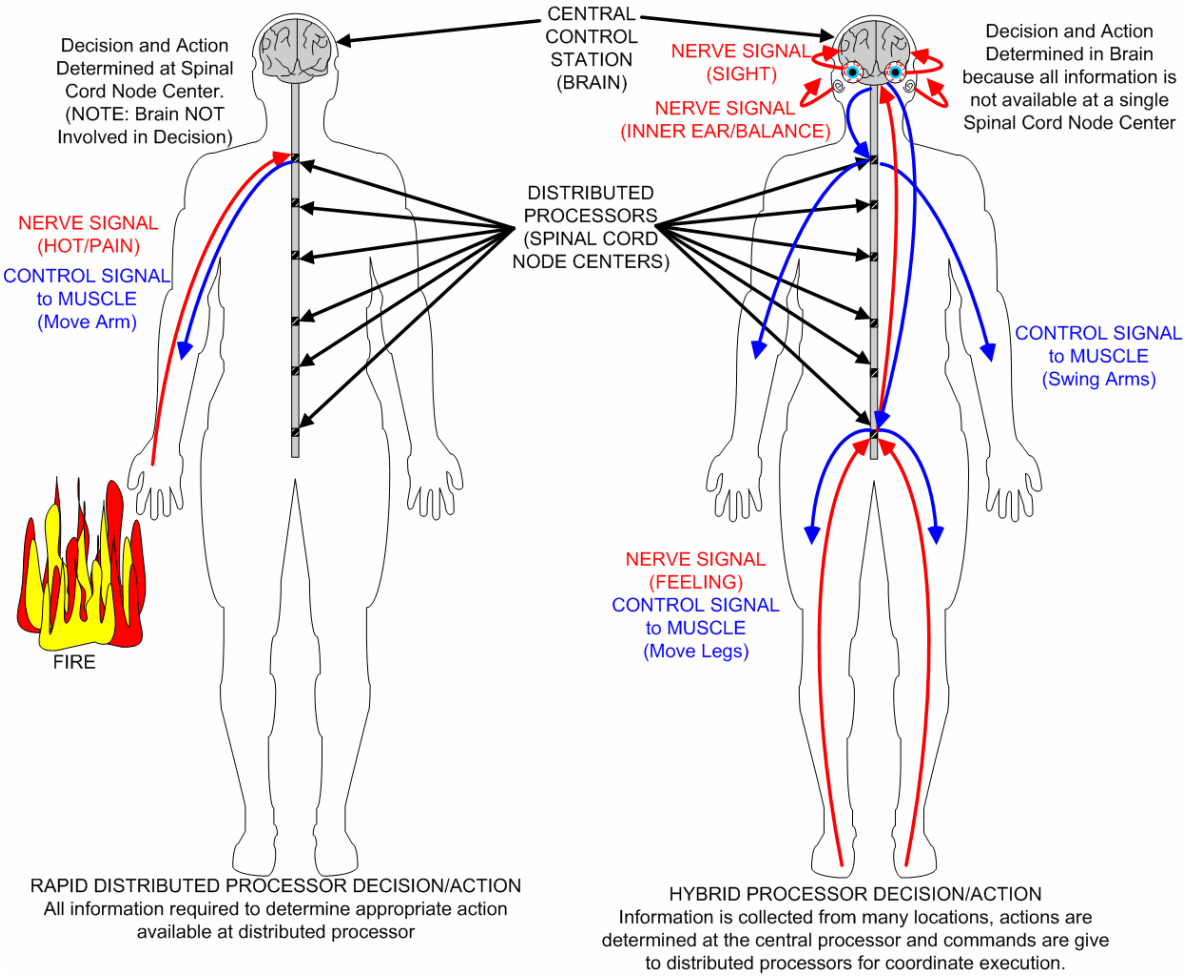


Figure 3 Example of Human Nervous System as a Hybrid Control System

Each level of processor in the body is properly sized for the complexity and speed with which it is required to act. Spinal cord node centers process a limited amount of data and execute a limited number of actions (such as, course hand motion), but these decisions can be made much faster locally at the spinal cord node centers than anywhere else in the body. The brain is the slowest and most complex since it processes the most information and must transmit information and commands the furthest distance.

MILITARY ORGANIZATION

Military organization is another example of hybrid processing. Each tier in an organization has the capability to manage their specific area of authority, while accepting orders from higher tiers who have access to more information or a more complex action plan. The mechanism for a Navy ship's organization is much the same as the human body. The commanding officer is similar to the brain, who receives information from department heads, who receive information from the division officers, who receive information and give orders to those under them. Each organization has a limited amount of autonomy based on what occurs in their sphere of knowledge. Complex decisions are provided from above, where all of the required information is available to make an informed decision. Figure 4 highlights a hybrid ship manning organization, where lower level officers have a certain level of automonomy to accomplish actions within their area of responsibility without asking permission to accomplish the task but rather to inform higher levels the actions were completed. Higher level officers receive general status from lower tiers and issue high level mission objectives to lower tiers for actions. Those lower tiers receive the objectives and determine the necessary actions to accomplish the higher level objectives. The result is less data and fewer commands exchanged between successively higher levels in the organization with more comprehensive information and commands at these levels.

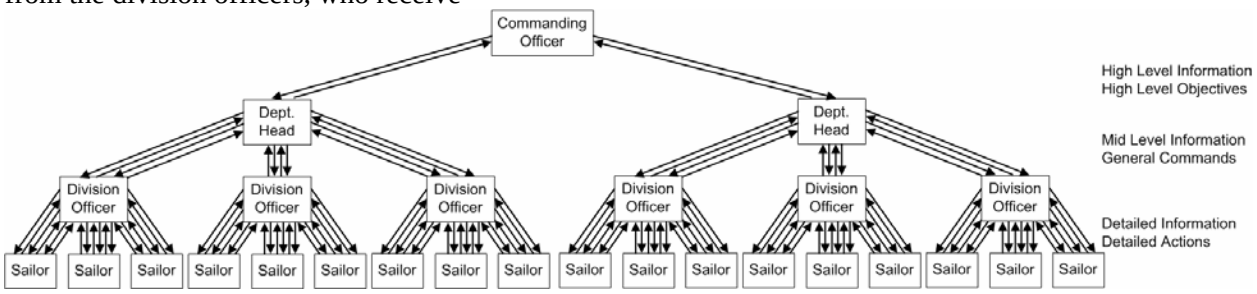


Figure 4 Example of Military Organization Highlighting Quantity of Information Flow between Levels

PRACTICAL CONTROL SYSTEM DESIGN

There are many factors which ultimately affect the design of a control system, beyond the attributes discussed thus far.

The control system architecture typically is based on the equipment in the control system vendor's commercial line. This typically has resulted in central controls or controls distributed to only a limited extent.

A paradigm shift is necessary to fully realize the benefits of distributed control enabled by today's technology.

Engineering the Architecture of Distributed Control Systems

It is not unusual for projects that implement complex control systems to experience cost overruns and schedule delays. Then, even after the added time and expense and substantial changes to the system, the system still does not perform as expected. Such performance problems usually are exacerbated when the system operates in off-design conditions or equipment fails. Such experiences, which are probably representative of the state-of-the-art in industry today, indicate that designing a control system to support damage control (when off-design conditions and equipment failures are to be expected) is at the edge of, or exceeding, the state-of-the-art. The key to successful implementation of distributed control systems is rigorous engineering of the architecture of the distributed control system. As with any design, this critical, first step determines the performance limits that can be achieved with the design.

The basic steps to engineer the architecture of as distributed control system include:

1. Defining the control decisions that will be executed by the system.
2. Developing the control decision logical architecture.
3. Defining candidate hardware and software architectures.
4. Evaluating the candidate hardware and software architectures and selecting the optimum.

Each of these steps is described briefly below.

Defining the Control Decisions that Will Be Executed by the System

The first step in engineering the architecture of the distributed control system is defining and understanding the control decisions that will be executed by the system. In a system where people must interact closely with the systems being controlled, a human-centered (i.e. human systems integration) approach to the design is vital to the effective operation of the system. A functional analysis methodology should be used to define the control decisions that are executed by the system and the information that the system displays to the operators. The functional analysis also provides an integrated definition of functions performed by personnel and by ship systems. The process of developing the functional analysis, and the resulting product, also provide both the user and the developer with a clear, detailed understanding of the performance to be expected from the control system. Figure 5 is an example of part of the functional analysis for firefighting in the DC-ARM program with integrated human and automated system response.

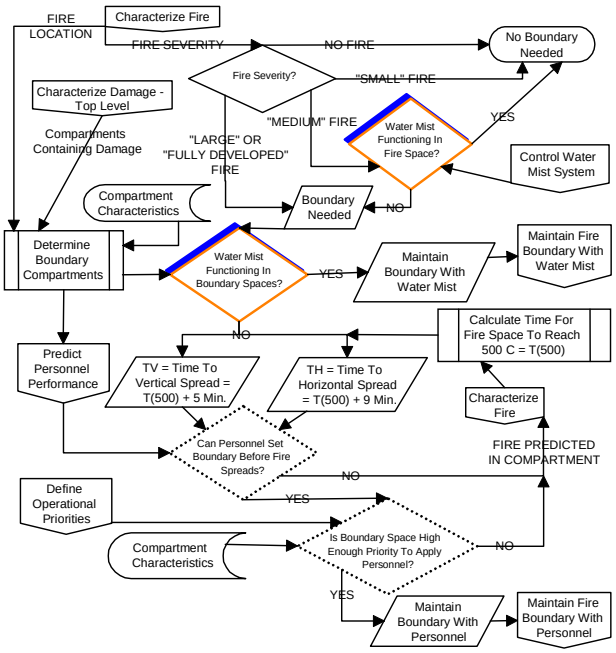


Figure 5 Sample Functional Analysis for Firefighting

Developing the Control Decision Logical Architecture

The second step in engineering the architecture of a distributed control system is understanding the *logical architecture* of the control decisions. This understanding of the logical architecture is essential to making subsequent decisions about the architecture of the hardware and software that will result in an effective, robust, survivable distributed control system that is practical to develop, install, and maintain.

For the DC-ARM distributed control system, three levels were defined for the architecture of the control system logic (other control applications could define different levels for the logic architecture):

1. *Device Level Control Logic* is logic that requires inputs only from the device itself to make the control decision. For example, a Smart Valve requires data only from pressure sensors installed in the valve itself to make the control decision to detect and isolate a rupture. A typical electrical circuit breaker and a conventional pressure regulating valve are other examples of the application of device level logic. Similarly, from a compartment perspective, *compartment level control logic* would require inputs from only one compartment to make the control decisions regarding that single compartment. Automatic actuation of a fire suppression system could be an application of compartment level control logic.

2. *System Level Control Logic* is logic that requires inputs from more than one device in the system to make the control decision. For example, using flow balance among multiple sensing points in a fluid system to identify a rupture is system level logic. Similarly, from a compartment perspective, *zone level control logic* would require inputs from more than one compartment in a zone to determine fire boundary locations, for instance.

3. *Ship Level Control Logic* requires inputs from more than one system, or more than one zone, or a combination of systems and zones to make the control decision. For the DC-ARM distributed control system, any control decision involving actions by people is considered ship level control logic.

It is very important to keep in mind that this is a *logical architecture*, it is not a definition for the architecture of the software or hardware that processes the control decisions. Figure 6 is an example of the logical architecture for firemain control.

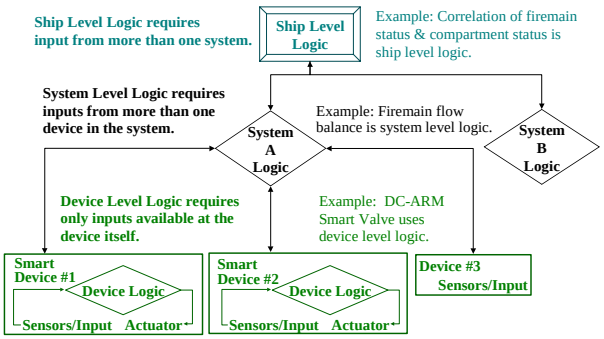


Figure 6 Example Logical Architecture for Firemain Control

Defining Candidate Hardware and Software Architectures

The control decision logical architecture provides the basis for the synthesis of candidate hardware and software architectures that meet the same functional requirements. Factors such as processor load, communications load, survivability requirements, and cost constraints should be considered in synthesizing candidate hardware and software architectures. Experience in developing distributed controls for shipboard fluid systems provides the following lessons:

- Defining boundaries for application software modules consistent with the levels in the control logical architecture, and consistent with the boundaries in the functional analysis, provide a software architecture that is naturally easy to understand. This simplifies the development of the software and will greatly simplify the maintenance of the software.
- Executing device level logic at the device level (sensors, processor, and actuator all part of the device) provides a very robust (system functions with multiple device failures), highly survivable control capability. True device level control (executing device level logic on device level processors) also inherently supports "plug and play" upgrades to the system.
- Executing system level logic or ship level logic on device level processors, on the other hand, would likely result in a control system with a very high communications load (hampering response during a casualty when the data rate is high), and in a control response that would likely be difficult to predict. Such an approach also makes it extremely difficult to achieve a robust capability (i.e. a system that would function in spite of multiple device failures). Consequently, only device level logic should executed on processors at the device level.
- Using a hardware architecture that mirrors the control decision logical architecture leads to the maximum level of survivability that could be achieved for the control decisions that need to be executed. Combining levels of control decisions in higher level processors (i.e. executing system level logic in a ship level computer) is practical and could reduce the cost of the system at the risk of decreased survivability. Therefore, it's acceptable to execute lower level decision logic in a higher level processor; it is not acceptable to execute higher level decision logic in a lower level processor.

Evaluating the Candidate Hardware and Software Architectures and Selecting the Optimum

The candidate hardware and software architectures could be evaluated against program selection criteria (relative priorities for survivability, cost, robustness, maintainability, and other program criteria) to select the architecture that is optimum for the program. Hardware and software selection are secondary compared to designing and implementing the logical architecture correctly. Hardware and software architecture are typically selected based on other requirements such as shock hardness, EMI concerns or equipment reliability.

DISTRIBUTED CONTROL SYSTEM RESEARCH & DEVELOPMENT IN THE NAVY

The Damage Control Automation for Reduced Manning (DC-ARM) Program was started in the 1990's to develop, test and demonstrate technology for enabling reductions in the manning needed for Navy damage control (DC) while, at the same time, improving DC performance. [1] The DC-ARM Program was sponsored by the Office of Naval Research (ONR) and managed by the Naval Research Laboratory (NRL). The DC-ARM program developed and demonstrated several new technologies which improve damage control performance while reducing the number of dedicated damage control personnel. The individual technologies were monitored and controlled by the DC-ARM Supervisory Control System. [2] Figure 7 shows the DC-ARM Supervisory Control System display, developed using the distributed control system design approach discussed above.

Figure 7 DC-ARM Supervisory Control System Damage Control Central aboard ex- USS SHADWELL

The Supervisory Control System was a successful demonstration that a rigorous design process with respect to assigning functional requirements to different levels of software architecture resulted in a control system which was inherently survivable. Survivability was achieved through autonomous capability at several tiers of the control system architecture.

The base of the system was composed of device level controllers. Autonomous Smart Valves, standard valves with pressure sensors, an electric motor actuator and a microprocessor, were installed as the device level controllers in the firemain system. Each Smart Valve was capable of independent firemain piping rupture detection and isolation using only the two pressure transducers installed in the valve body. The valve had the capability to detect and the authority to isolate firemain ruptures if the device was given the authority from a higher logic level in the control system architecture. The smart valves communicated with a firemain system controller which monitored the status of the firemain valves comparing pressure and flow data looking for valve component failures. To provide redundancy and avoid common mode failures, the system controller used a fundamentally different logic for rupture detection than the device level logic in the smart valves. MPR received a US Patent for the Smart Valve distributed rupture detection and response logic, Reference 3.

Each compartment had a compartment logic which monitored all of the fire detection sensors within the space and controlled the fire sprinkling nozzles within the space. The compartment controllers could detect fires and had the authority to actuate the fire sprinklers within its space to suppress the fires. Each compartment controller communicated with a zone controller which was responsible for decisions and actions which involved information from more than one compartment. The zone controllers determine fire boundaries around the compartments on fire. This is a zone level function because the function requires information about more than one compartment.

The ship level controller communicated with each of the fire zone level controllers and each of the system level controllers. The ship controller managed decision and action conflicts between zone objects and conflicts between zones and systems. The result was a control system which could perform essential damage control functions after sustaining various levels of damage. Figure 8 depicts the DC-ARM

Supervisory Control System logical architecture, where logic is distributed between compartment, zone, device, system and ship control modules.

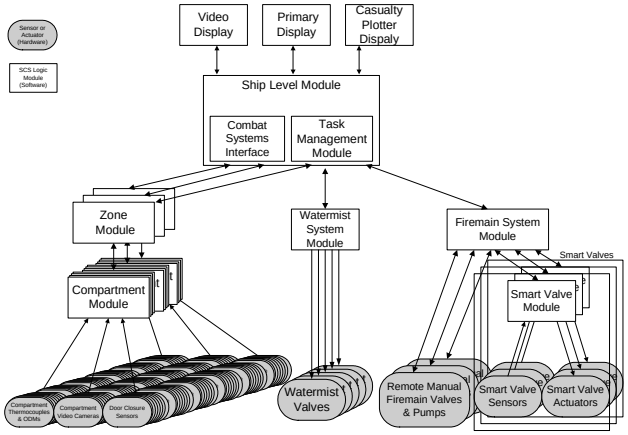


Figure 8 DC-ARM Supervisory Control System Logical Architecture Design

The prototype system was installed aboard the ex-USS SHADWELL. The system installed aboard the ex-USS SHADWELL was designed with the software architecture illustrated in Figure 8. To minimize program costs, some lower level software modules were executed in a higher level processor. The DC-ARM program demonstrated the successful performance of the distributed system designed with modular software that mirrored the logical decision architecture as described above. The survivability of the system could easily have been improved by distributing all of the software consistent with the logical architecture.

The survivability of a Smart Valve based fluid system was tested during the DDG1000 Autonomic Fire Suppression System (AFSS) Engineering Design Model (EDM) weapons effect test aboard the ex-USS PETERSON. Smart Valves installed in the firemain system were subject to a weapon event which severely damaged the ship structure and the firemain system. The Smart Valves successfully detected and isolated the rupture even with unanticipated damage to Smart Valves located near the blast. These Smart Valves used the same logic developed as part of the DC-ARM program and would transition to the first Smart Valves installed on an active ship.

THE FIRST AUTONOMOUS SHIP SYSTEM

The Smart Valve technology from the DC-ARM program was selected by the Navy as the first autonomous device level component for damage recovery to be installed on an active ship fluid system. The Smart Valve system, named the Chilled Water Automation System (CWAS), is currently being installed in new construction DDG 51 Class ships, DDG 107 and follow. Testing and experience have demonstrated that critical shipboard equipment can overheat due to loss of cooling in substantially less time than the crew can locate and isolate damage to the chilled water system. Twenty-two smart valves are being installed to enable the chilled water system to recover automatically from damage and, if needed, automatically switch critical loads to the alternate chilled water loop, before critical equipment overheats.

Several fieldbus subnets connect the 22 Smart Valves to the ship's Fiber Optic Data Multiplexing System (FODMS).

The CWAS includes a monitoring and control software package installed on seven repair station consoles. This application installs, configures and maintains the software on the smart valves and continually monitors all of the networked equipment for problems. The Installation, Maintenance, Diagnostics and Troubleshooting (IMD&T) application also allows remote control of the Smart Valves if the operator does not want the system to respond automatically to ruptures. The IMD&T also displays the critical data available at the valve, thereby enabling substantial situation awareness.

The CWAS is integrated with the legacy Automated Common Diagrams and can easily be integrated with other control systems or HMIs. The Smart Valves do not require network communication with a central processor or communication other smart valves to detect, isolate or recover from a chilled water piping rupture; that is, they apply true distributed device level logic. This means the valves can accomplish their function without

communication between devices or distributed monitoring stations, thereby increasing survivability. Since the Smart Valves do not require communication between valves, if a Smart Valve fails for any reason, the upstream Smart Valves will isolate the rupture. The system fails gracefully and will accomplish its objective as long as any two valves on either side of the rupture can function following system damage.

The system has completed land based testing at NAVSSES Philadelphia in the summer of 2005, Reference 4. The land based testing utilized a full scale, reduced scope mock up of the DDG 51 chilled water system. During the land based testing, the CWAS consistently detected chilled water ruptures in less than one second. Ruptures were isolated within eleven seconds. Flow was restored to critical equipment within 30 seconds and the system achieves optimum availability after a maximum of 90 seconds depending on rupture location.

The CWAS is first distributed control for a fluid system deployed in the active Fleet. The Navy and MPR have been working for close to ten years, developing R&D prototypes, computer simulations (Reference 5 & 6), production prototypes and production units.

THE FUTURE OF DISTRIBUTED PROCESSING

The success of the DDG 51 Chilled Water Automation System has shown that distributed processing and automation can be successful if careful attention is applied during the functional logic design.

The Navy is expanding automated distributed processing on other ships as well. The DDG1000 (formally DD(X)) is employing Smart Valves in the chilled water system, as in DDG 51 CWAS and in the firemain system as in the DC-ARM program. The DDG 1000 is considering the use of smart valves in other critical fluid systems.

The design approach for engineering the architecture of distributed control systems could be applied to a variety of systems, including: compressed air systems, electric power distribution systems, unmanned air, surface or subsurface vehicles, and combat systems that integrate sensors and weapons.

CONCLUSION

As the Navy requires more complex control systems, it will become imperative to properly allocate control functionality throughout a distributed control system rather than the traditional central control system. Dividing the logic into smaller independent functions, based on the control decision logic, will allow complex systems to operate with fewer, simpler functions.

The NRL DC-ARM program showed the successful integration of damage control functions across various distributed control logical tiers.

The AFF EMD weapon effects testing aboard the ex-USS PETERSON demonstrated distributed processing is survivable and provides graceful degradation in unpredictable damage events.

The CWAS program successfully demonstrated the first successful implementation of automated distributed processing and will prove to the example for future automated distributed processing.

The distributed technology and Smart Valves in particular have been rigorously designed, tested in a live fire R&D environment, tested in weapon effects tests, tested in formal Navy qualification tests and are now being installed aboard active ships, introducing the Fleet to the benefits of distributed processing.

This paper explains the design approach that has been successfully applied across several programs all with positive results. This design approach can be applicable to any control system, not only those which involve fluid systems or Navy ships.

REFERENCES

[1] Buchanan, J., Downs, R., Durkin, A., Farley, J., Gottuk, D., Luers, A., Nguyen, X., Pham, H., Runnerstrom, E., Scheffey, J., Tatum, P., Williams, F. W., Wong, J., "FY 2001 DC-ARM Final Demonstration Report," NRL Memorandum Report 8623, May 31, 2002.

[2] Downs, R., Farley, J., Runnerstrom, E., Williams, F.W., "Damage Control Automation for Reduced Manning (DC-ARM) Supervisory Control System Software Summary Final Report," NRL Memorandum Report 8609, March 21, 2002.

[3] Downs, R., Runnerstrom, E., et al., "Method and Apparatus for Detecting and Isolating a Rupture in Fluid Distribution System," US Patent 6,535,827, March 18, 2003.

[4] Runnerstrom, E., et. al., "CWAS Upgrade Project – Land Based Demonstrator Test & Validation Report", MPR Document Number 0552-0005-506, Rev. 0, October 27, 2005.

[5] Downs, R., Runnerstrom, E., Loper. C., "CWAS Upgrade Project – Computer Based Simulator – Technical Report", MPR Document Number 0552-0004-414, Rev. 0, May, 2, 2005.

[6] Runnerstrom, E., et. al., "CWAS Upgrade Project – Computer Based Simulation Software Test & Validation Report", MPR Document Number 0552-0004-407, Rev. 0, April 19, 2005.

ACKNOWLEDGMENTS

MPR Associates would like to acknowledge the Naval Research Laboratory and the Office of Naval Research for the early support for automated device level logic and Smart Valve logic.

MPR Associates would like to acknowledge PMS500 for their effort in executing the DDG1000 AFSS EDM weapon effects test aboard the ex-USS PETERSON.

MPR Associates would like to acknowledge PMS400D8 and the Bath Iron Works for supporting the CWAS program through design and installation on DDG 51s.

MPR would like to acknowledge Tyco, Valves & Controls for supporting the design of the Smart Valve hardware.

Ryan Downs is the principal author and he is an employee of MPR Associates. Since joining MPR in August of 1997, Mr. Downs has been involved in a variety of projects for the U.S. Navy and the nuclear power industry emphasizing hardware and software design, distributed control system design and qualification, equipment qualification testing, fluid system automation, network design, shipboard test plan execution, valve diagnostics, and ship survivability design evaluations. Mr. Downs was the Project Manager and Lead Software Developer for the DC-ARM Supervisory Control System and the DDG 51 Chilled Water Automation System.