Component: NAVY

Topic #: N201-036

Title: Dynamic Loadable Module Architecture and Applications Program Interface for a Distributed Common Operational Picture Subsystem

Technology Areas: Info Systems

Acquisition Program: PEO IWS 1.0, AEGIS Combat System Program Office

ITAR: The technology within this topic is restricted under the International Traffic in Arms Regulation (ITAR), 22 CFR Parts 120-130, which controls the export and import of defense-related material and services, including export of sensitive technical data, or the Export Administration Regulation (EAR), 15 CFR Parts 730-774, which controls dual use items. Offerors must disclose any proposed use of foreign nationals (FNs), their country(ies) of origin, the type of visa or work permit possessed, and the statement of work (SOW) tasks intended for accomplishment by the FN(s) in accordance with section 3.5 of the Announcement. Offerors are advised foreign nationals proposed to perform on this topic may be restricted due to the technical data under US Export Control Laws.

OBJECTIVE: Develop a real-time extensible and evolvable architectural model, software framework, and Applications Program Interface (API) for a modular software execution environment capable of supporting dynamic run-time installation and control of new capabilities via the use of dynamically installable and reconfigurable software modules.

DESCRIPTION: The Navy has a requirement to expand its sea-based advantage through increased capability. This need can be addressed by providing technology that has the potential to improve ship combat effectiveness and efficiency by significantly improving battlespace situational awareness, thus reducing the management complexity of the overall battlespace. This may allow for a reduction in the number of platforms needed in a specific tactical arena to provide an equivalent track engagement capability, and for reduced staffing or increased duty time. By reducing the potential stress and fatigue levels experienced by the operator while monitoring tracks in a sensor- or communications-compromised or denied environment, the Navy can potentially reduce shipboard manning requirements, and subsequently improve affordability.

The current AEGIS combat system implementation does not include a comprehensive distributed (i.e., multi-platform) capability for capturing the complete battlespace operational, environmental, and tactical picture in a coherent, integrated manner. Currently available commercial systems and software, which might be considered for adaptation to our needs (e.g., the FAA Air Traffic Control System hardware and software), are dated in their designs, and lack the flexibility and track capacity required to adequately address Navy tactical needs. Specifically, currently available commercial technology is limited in that it lacks the capability to track, identify, and manage complex air, surface and subsurface entities and threats present in the DoD environment. The Navy needs a modular software execution environment and API intended for integration into a Distributed Common Operational Picture (DCOP) software subsystem. This modular execution environment will provide the DCOP software subsystem with the ability to dynamically (on-the-fly) install, remove, or modify DCOP capabilities without disrupting the ongoing real-time performance of the DCOP subsystem, other currently executing combat systems applications, or the host combat systems performance as a whole. The capability is needed within AEGIS to present a common operational picture (COP) to the combat systems watch stander. The subsystem must be modular in nature, and support the sharing of COP data across all participating platforms within the battlegroup in a manner, which ensures the real-time multi-platform coherence and synchronization of COP data on every platform to the greatest extent possible.

The DCOP architectural model, software framework, and API should be considered in context with an appropriate DCOP data model (DM), DM markup language, and multi-platform data coherency and synchronization algorithm set. These components, when considered as a whole, should be capable of supporting the functional capabilities and requirements needed to provide a comprehensive real-time battlespace DCOP to each Navy or allied warfighting platform capable of hosting a DCOP subsystem.

The DCOP architectural model and software framework must be capable of providing both combat systems operators and combat systems software applications with real-time access to a distributed (i.e., multi-platform) COP. This COP represents a complete tactical
view of the battlespace as well as all tactical and non-combatant entities present within the ship’s Area of Responsibility (AOR) and is characterized by a set of quantized parameters associated with each “entity” resident within the battlespace. The actual parameters for each entity will be defined by a DCOP real-time extensible and evolvable battlespace data model (and its associated markup language) which will constitute an associated modular component of the overall DCOP subsystem.

The DCOP architectural model and software framework must be capable of supporting “on-the-fly” addition and/or deletion of DCOP capabilities, with each capability implemented via a loadable software module. The installation, removal, activation, and deactivation of software modules within an executing DCOP implementation should have no adverse effect on the real-time performance of the DCOP system and/or the services it provides to the host platform and operator at the time those changes are implemented. An exemplar of this type of low/no impact behavior during runtime installation and removal of capabilities within an executing system can be observed in the kernel module control facilities of the Linux operating system (kernel 4.4 and above), such as the insmod, rmmod, depmod, ismod, modinfo and modprobe commands [Refs. 2, 3]. The process of installing, removing, or otherwise controlling DCOP services and capabilities within an executing DCOP installation should be easily executed by combat systems watch personnel without the need to stand down, halt, or reload the currently running combat systems software instance and without the attention of specially trained software maintenance personnel.

The DCOP subsystem architecture and software framework must be capable of supporting battlespace common operational picture data access control and multi-platform DCOP data coherency and synchronization mechanisms as a modular replaceable or upgradable component of the overall DCOP architecture. To this end, the DCOP subsystem architecture will utilize a modular multi-platform high-reliability communications services capability provided by the host combat system, in conjunction with its own resident modular multi-platform data coherency/synchronization algorithm. This will ensure that the DCOP battlespace data model reflects the current real-time state of the battlespace, notwithstanding data update related multi-platform communications issues due to enemy electronic countermeasure action and problematic atmospheric radio frequency environment issues.

The DCOP API should provide a DCOP data access and subsystem control interface capable of supporting two major categories of DCOP-related software applications or modules. First, it must support the DCOP subsystem capabilities implementation modules, which add, remove, or control organic capabilities and services within the DCOP subsystem itself, with the purpose of enhancing the overall suite of capabilities and services, which DCOP provides to the platform’s combat system. Second, it must support the COP data access to combat systems hosted and console operator-initiated client applications or other software entities (e.g., software agents) resident within the combat systems suite, which require DCOP API-based real-time access to common operational picture data. The API architecture and software framework will provide an accessible data abstraction layer between any combat systems client software application and the actual DCOP data structure implementation maintaining common operational picture data, insuring that any combat systems applications remain independent of any implementation-specific changes, enhancements, etc. made to the DCOP data model and supporting data structures.

Both the DCOP architectural model and its API shall be well documented and conform to open systems architectural principals and standards [Ref. 4]. Implementation attributes should include scalability and the ability to run within the computing resources available within the AEGIS combat systems BL9 or later environment.

The software implementation of the DCOP prototype subsystem should be compatible with the C++ programming language and capable of running in a Linux (Redhat RHEL 7.5/Fedora 29/Ubuntu 18.4.1 or later) processing environment as a standalone application (i.e., no critical dependencies on network-based remotely hosted resources, save for sensor data emulators). The prototype DCOP subsystem implementation will demonstrate the following: (i) the ability to install, remove, and control various DCOP capability software modules in an executing system with minimal/no impact on system performance; (ii) the ability to support third party sourced C++ and Java applications requiring real-time access through the DCOP API to common operational picture data resident within the DCOP data model; and (iii) the ability to demonstrate real-time multi-platform sensor data updates, synchronization, and data coherency across multiple executing instances of the DCOP subsystem.

Work produced in Phase II may become classified. Note: The prospective contractor(s) must be U.S. Owned and Operated with no Foreign Influence as defined by DOD 5220.22-M, National Industrial Security Program Operating Manual, unless acceptable mitigating procedures can and have been be implemented and approved by the Defense Security Service (DSS). The selected contractor and/or subcontractor must be able to acquire and maintain a secret level facility and Personnel Security Clearances, in order to perform on advanced phases of this contract as set forth by DSS and NAVSEA in order to gain access to classified information pertaining to the national defense of the United States and its allies; this will be an inherent requirement. The selected company will be required to safeguard classified material IAW DoD 5220.22-M during the advance phases of this contract.

PHASE I: Design, develop, and deliver a concept for an architectural model and software framework of a DCOP modular software
execution environment and API capable of meeting the subsystem and API requirements and capabilities outlined in the Description. Establish the feasibility of the concept through evaluation of the ability of the proposed model to successfully capture all tactical and operational battlespace parameters as detailed in the Description. The Phase I Option, if exercised, will include the initial design specifications and capabilities description to build a prototype solution in Phase II.

PHASE II: Ensure that the software implementation of the DCOP prototype subsystem is compatible with the C++ programming language and capable of running in a Linux (Redhat RHEL 7.5/Fedora 29/Ubuntu 18.4.1 or later) processing environment as a standalone application (i.e., no critical dependencies on network-based remotely hosted resources, save for sensor data emulators). Ensure that the prototype DCOP subsystem implementation will demonstrate the following: (i) the ability to install, remove, and control various DCOP capability software modules in an executing system with minimal/no impact on system performance; (ii) the ability to support third party sourced C++ and Java applications requiring real-time access through the DCOP API to common operational picture data resident within the DCOP data model; and (iii) the ability to demonstrate real-time multi-platform sensor data updates, synchronization, and data coherency across multiple executing instances of the DCOP subsystem. It is probable that the work under this effort will be classified under Phase II (see Description section for details).

PHASE III DUAL-USE APPLICATIONS: Support the Navy in transitioning the DCOP subsystem software for Navy use. Support implementation that will include the integration of the DCOP subsystem software into a prototype combat system implementation, consisting of one or more of the following: AEGIS BL9 or greater or Common Core Combat System (CCCS) experimental prototype and implemented on a virtualized hardware environment within an AEGIS compliant land-based testbed. This capability has potential for use within the commercial Air Traffic Control system in future development of an air traffic common operational picture, which would be capable of handling complex traffic control patterns.

REFERENCES:


KEYWORDS: Real-time Extensible and Evolvable Battlespace Data Model; Resident Modular Multi-platform Data Coherency/Synchronization Algorithms; Current Real-time State of the Battlespace; Loadable Software Module; Run-time Installation and Removal of Capabilities; Common Operational Picture Data