Information systems leveraging legacy systems and developments in infrastructure technologies

Kurt Veum, Hilde Hafnor and Ole Martin Mevassvik

Norwegian Defence Research Establishment (FFI)
P. O. Box 25, NO-2027 Kjeller, Norway
Phone: +47 63 80 70 00
E-Mail:{kurt.veum|hilde.hafnor|ole-martin.mevassvik@ffi.no}

Abstract

A study of information systems employed in the domain of maritime C2 and the requirements derived from the characteristics of the domain has been carried out at the Norwegian Defense Research Establishment (FFI). Based on this study, a set of properties that future systems should possess, and which the underlying infrastructure technology must support, has been identified. Examples of such properties are adaptability, affordability, interoperability and composability. This paper discusses the rationale for and the means by which these properties may be realized in future maritime C2 information systems. For instance, current efforts for achieving interoperability include architecture standardization efforts and middleware technologies that support dynamic interoperability. As a case study the maritime C2 project is developing a Recognized Maritime Picture (RMP) production demonstrator. Functionality in two operational systems for RMP production is integrated using Common Object Request Broker Architecture (CORBA), Jini and Control of Agent Based Systems (CoABS). Furthermore, the experiences using middleware technologies for integration of legacy systems and realization of the properties listed above are reported.

1. Introduction

The work reported in this paper has been conducted as a part of a project carried out at FFI. The project is developing a future command and control (C2) system concept for maritime operations and a derived C2 system plan. The objective is to recommend guiding principles and ideas for the evolution of the maritime C2 system in the years to come and based on these recommend a maritime C2 system plan. The recommendations are based on cost effectiveness analysis of alternative C2 systems [1].

A study of information systems employed in the domain of maritime C2 and the requirements derived from the characteristics of the domain has been performed. Examples of domain characteristics are increasing diversity in operations and dynamic and uncertain environments. Based on this study a set of properties that future C2 information systems (C2IS) should possess has been identified by the project. Examples of such properties are adaptability, composability and scalability in order to handle requirements such as affordability, interoperability and heterogeneity in future C2IS. Affordability refers to cost-effectiveness regarding acquisition, development and maintenance costs. Interoperability concerns integration and information exchange aspects, and heterogeneity refers to heterogeneity in programming languages, operating systems, hardware platforms, network protocols, etc.

This paper discusses the rationale for and the means by which these properties may be realized in future maritime C2IS, and the experiences gained from using middleware technologies for integration of legacy systems and realizing some of the properties in a Recognised Maritime Picture (RMP) production demonstrator built by the project.

The paper is organized as follows. Section two outlines the C2 domain characteristics and the status for C2IS in the maritime domain. Section three presents a selection of the derived C2IS properties, and section four discusses the means by which these properties may be realized in future maritime C2IS. Section five gives a presentation of the RMP production demonstrator. Section six describes the demonstrator architecture, and in section seven some of the experiences using middleware technologies are given. Finally, a summary is given in section eight .

2. C2 domain characteristics and the status for C2ISs

FFI has conducted a study of selected information systems employed in the domain of maritime C2. Furthermore, requirements derived from characteristics of the domain have been identified**.** These are briefly discussed below.

Information systems in the maritime C2 domain have to support diverse operations such as intelligence and surveillance, exercising sovereignty, crisis management, territorial defense and securing vital civilian functions. Participation in international operations has recently been a prioritized area. As the diversity of operations increase and the technology opens up new possibilities, the change in requirements continues to increase. Furthermore, operations are performed in dynamic and uncertain environments involving forces from multiple services and multiple, and to a certain degree, unanticipated coalition partners.

Depending on the situation and the scale of conflict the manner C2 is exercised varies. In a crisis situation the political/strategic military level may desire detailed information concerning a limited area and have direct control over own units in this area. In war a chain of command is established dynamically and C2 is exercised at several levels in the organization. In this case the higher levels of command is provided aggregated information covering larger areas and issue intentions and coarse plans rather than exercising direct control. Thus, the information system must allow for flexible flow of information and being able to support dynamically changing organizations.

The geographical distribution of forces and assets forces the communications to be made over the horizon. This may lead to scenarios with unreliable communication with continuously changing bandwidth, e.g. use of HF radios at northern latitudes. In general the quality of the services provided by C2ISs varies a lot depending on the scenario in which they are being used.

Traditionally systems have been built for supporting individual C2 tasks such as production and maintenance of recognized picture and plans, which results in low interoperability. For a commander, judging the impact changes in the situation have on plans is often a complex task and automated support is crucial. It is therefore essential that the current situation, active plans and the relationships between these have representational counterparts in the C2IS supporting the C2 process. This is often not the case. Furthermore, the C2IS containing the C2 process artifacts are often not interoperable.

Up until now C2IS have been developed in isolation, with proprietary technical infrastructures and no concerns for interoperability with other systems. Systems often emerged as an answer to acute needs for solving specific local problems within the organization. The result is a conglomerate of standalone systems that do not exchange information with each other. The systems overlap with respect to functionality and information content, and are costly to maintain. Today, the relation between organization processes and the supporting information technology has been recognized, and the need of a more holistic perspective on how system acquisitions and system development processes within the organization are conducted has emerged. To meet this challenge NATO is using architectures as an approach (2).

3. **C2IS properties**

In the study of information systems employed in the domain of maritime C2 and the requirements derived from the characteristics of the domain, a number of desired properties of future systems were revealed. Some of these properties are not yet well defined, and they can sometimes conflict with one another. They are also difficult to attribute to specific parts of a system. They may not all be equally realizable in a future C2IS dependant of which properties that will be prioritized. Some of the properties are observed during operational use of the system since they relate to the dynamic behavior of the system. Others are observed during development or maintenance activities since they reflect the design itself. Prioritized properties have an impact on which overall architecture is chosen for a system. A selection of the properties is listed below.

Adaptability: Is to which degree a C2IS, despite increased heterogeneity in information content and realization, is able to cope with changes in the command and control organization.

Composability: Is the ability to offer new services by assembling additional components. This contributes to increased system flexibility, since a specific configuration of components, which in all offers the necessary and desired services, can be assembled and run.

Scalability: Denotes the C2IS ability to accommodate growing computational load. This requires scalable architectures.

Modifiability: Refers to how amenable the system's architecture is to effective modifications. The architecture needs to be stable even in the presence of changing functional requirements.

Reusability: An important aspect of cost-effective systems is the ability to use previously developed components as well as design and implementation patterns.

Availability: Availability is related to the flexible flow of information and the continuous availability of necessary services.

Timeliness: The system has to be able to respond in real time, i.e. process and provide adequate information within a specific time frame. In time critical applications the system may have to make trade offs between time used and the quality of the information it provides.

Robustness: A C2IS will have to continue to operate, even in the presence of faults. If parts of the system are unavailable, other parts must still be operational in order to maintain the availability of necessary information and services to support C2 , although perhaps with a degraded quality.

4. **Architectures and infrastructure standardization**

The strategy proposed by our project for future development of national C2 information systems relies on a vision that the present ”system of systems” in time will migrate and transform into "a integrated whole" - a distributed and component based/service integrated information system. The recommended strategy for achieving this vision is to develop a joint national reference architecture. A reference architecture is an architecture which provides the context, guidelines and standards for future system development. Using a reference architecture a system is viewed from several separated but interrelated perspectives and the description of these perspectives form the complete description of a system. Each perspective is an abstraction of the whole system but with focus on a specific area such as the enterprise (operational). The reference architecture serves as glue between the organization and the supporting system. The reference architecture will be based on an architecture framework. Examples of architecture frameworks are C4ISR-AF [4], which focuses on military operations, and ISO RM-ODP [5], which focuses on distributed processing. A third example is Minimal Architecture for Command and Control Information Systems (MACCIS) [7]. MACCIS defines a framework for describing architectures for C2IS within the Norwegian Army. The main concepts of this description framework are based on the ISO RM-ODP and C4ISR-AF. An architecture framework provides a norm for describing systems and their architectures so that they are comparable and thus support interoperability and enable identification of reusable components. A component based approach to systems development [6] fits naturally with a reference architecture since many standards can be articulated by interface based specifications.

The overall motivation for recommending an architectural based C2IS strategy is to contribute to a more cost-effective development process through coordination of the development of distributed systems, and as a contribution to support ongoing operational and systems interoperability efforts, both national [7] and international [3]. Furthermore, the technological development and the requirements for closer integration of sensors, communication systems and information systems will lead to a significant increase in the complexity of the C2IS in the future. Separation of concerns is an approach to cope with increased complexity and can be achieved through the use of several perspectives. By using several perspectives to isolate the more durable parts of a system such as the enterprise aspects from the less durable parts, such as the technical aspects, the system will also be more amenable to change.

Middleware technology is used to develop, integrate and distribute software components in an environment of heterogeneous computers, operating systems, network protocols and programming languages. Middleware offers encapsulated and generic infrastructure services such as transaction-, database- and fault management services, which are important to C2ISs. The infrastructure services and distribution transparencies offered by middleware allows the focus in system development to be shifted from infrastructure development to the essential domain problem at hand. Common Object Request Broker Architecture (CORBA) [10], Distributed Common Object Model (DCOM) and Java Remote Method Invocation (RMI) are examples of such middleware. CORBA is an open industrial standard specified by the Object Management Group (OMG), a non-profit organization that has more than 800 members. DCOM is Microsoft's component object model, and Java RMI is introduced by SUN and is solely based on the Java language.

Connection or networking technologies such as Jini [11], which is also regarded as middleware, enables creation of dynamic self-healing network of software components and devices. Originally, Jini was promoted as a technology for dynamic interconnection of embedded devices. In recent applications, this technology has demonstrated its strength in dynamic integration of software components [8]. The Jini framework allows a component joining a federation to advertise its services and to start using the existing services offered by other components in the federation. Components may join federations of components and offer their services in an unplanned and opportunistic way. Jini is promoted as a framework for building robust, scalable, distributed systems and is as such a highly relevant technology for the C2 domain, which is characterized as dynamic and uncertain. In contrast to other middleware technologies services are explicitly represented entities in the Jini framework. Properties of a service such as its quality may be represented as attributes and support for quality of service aware components may thus be provided.

Furthermore the protocol for using a service is stored in a single place and dynamically loaded by clients of the service. Thus a component may use a service without prior knowledge about the protocol for using it. Thus, in contrast to other middleware, no installation is needed on the client side. In this way, Jini makes a significant contribution to *dynamic interoperability*.

With open standards full interoperability is not guaranteed due to poor adherence to specifications or ambiguous specifications.

Since it will take some time to establish a joint national reference architecture for C2IS immediate measures should be taken in order to coordinate the development of existing systems. These are as follows:

- Introduce and standardize on commercially available middleware technology. This will reduce the significant costs associated with the maintenance of proprietary infrastructures of current systems.
- The proposed criteria according to which middleware technology should be selected are open technology standard, performance, flexibility and provision of available infrastructure services. CORBA conforms to some of these criteria. It is an open industrialized standard, which is operating system and programming language neutral, and offers support for real time applications and a number of infrastructure services relevant for C2IS. If explicit representation and dynamic integration of services is needed, a connection technology such as Jini is recommended as a supplement.
- Wrap and offer existing C2IS functionality and services as components.
- Aim for a reduction of the number of C2ISs in service.

5. The RMP production demonstrator

The purpose of developing a RMP production demonstrator is to demonstrate key aspects of a future C2IS. RMP production has been selected as the functional area since a consistent RMP is a prerequisite for performing C2 activities. One of the objectives of the RMP production demonstrator is to demonstrate:

- advantages of modern software architectures with respect to assuring key properties as discussed above, supporting the production and distribution of the situation picture, and integration of functionality from existing C2IS.

RMP is not a well-defined concept. It can however be stated informally that the RMP consists of the whereabouts and identity of objects in an area of interest of a commander. It consists primarily of surface and subsurface objects, but may also contain information on air and space objects. Its purpose is to support a maritime commander in his/her decision-making. RMPs are built and maintained at different levels in the organization, and is a key decision aid for maritime commanders.

6. The RMP production demonstrator architecture

The architecture for the RMP production demonstrator fulfils the following requirements. It facilitates integration of selected functionality from legacy systems, supports distribution of software components on heterogeneous platforms localized in one C2 node and finally supports message based information exchange between geographically distributed C2 nodes. To support separation of concerns with respect to the presentation, the service and the data/integration layer, a three-tier architecture (Figure 1) was chosen as the architecture to be used for each RMP production site.

The functionality developed in the demonstrator is encapsulated in components, each with well-defined interfaces/services according to an interface-based specification. Using this approach it is possible to develop the demonstrator in an incremental and iterative manner.

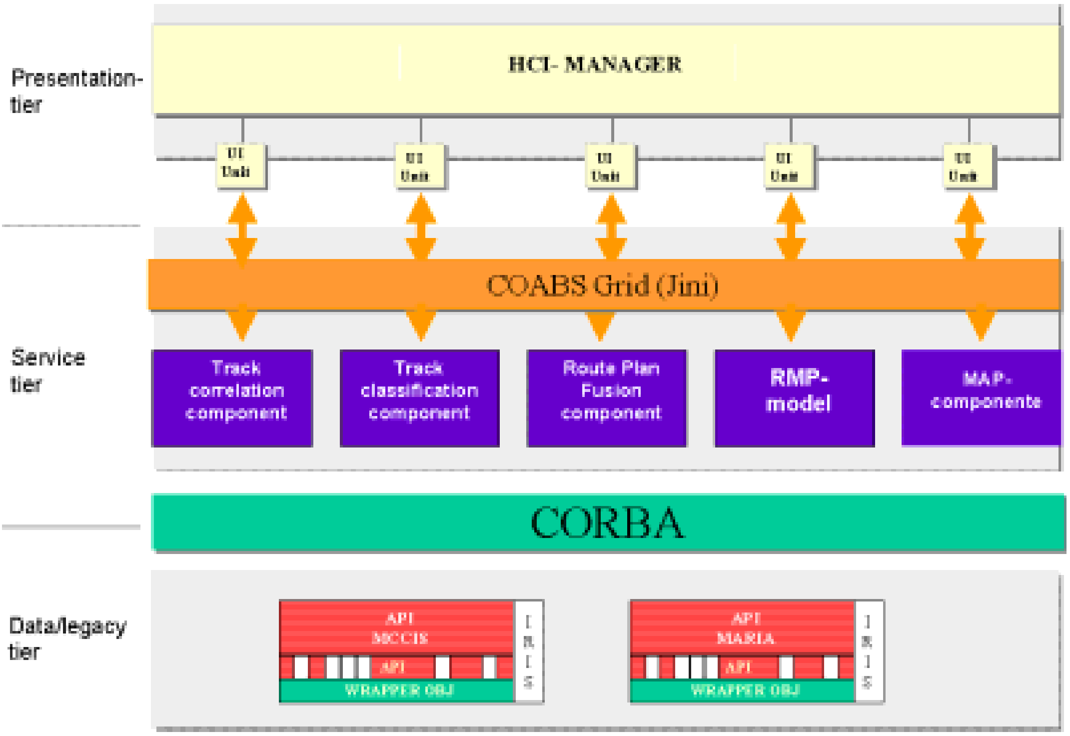

Figure 1 RMP architecture

As illustrated in the figure, two infrastructure technologies namely CORBA and Control of Agent Based Systems (CoABS) Grid [9] have been employed. CORBA is used for legacy system integration and CoABS Grid is used as the infrastructure for integrating the service layer components within a single picture compilation node and for the information exchange among distributed nodes. The CoABS Grid enables dynamic, runtime integration of agent, object and legacy SW components into applications. The C2 domain is characterized as a dynamic and uncertain domain, and thus the support of dynamic integration of services is a key feature of a middleware for C2ISs. Furthermore, the CoABS Grid supports invocation of remote services as well as information exchange using messages. It is implemented in Jini and owes many of its properties to Jini. Openwings [12] is an open specification sponsored by US Army, which resembles the CoABS Grid. It is a framework that enables development of highly available, secure distributed systems for mission critical applications. Its first implementation is likely to use Jini.

The two existing C2IS utilized in the demonstrator is the Maritime Command and Control Information System (MCCIS), which is a NATO C2IS, and the Navy tactical workstation (SjøTAS), which is a national system for presenting tactical RMPs. Since CORBA is platform, operating system, network protocol and programming language neutral the functionality made externally available in these systems can be accessed, not only from the RMP production demonstrator components, but also from any external component.

The functionality selected from MCCIS is track correlation. As well as accessing MCCIS's functionality remotely it is possible to exchange data by sending messages to MCCIS's network socket interface.

In the case of SjøTAS the map component (MARIA) is extracted and made available externally.

Figure 2 illustrates the part of the demonstrator's architecture, which is supporting distributed picture production. The CoABS Grid is used for information exchange between nodes

("Sensor Grid", "Information Grid") using HF communication and simulated satellite communication. Sensor platforms and RMP sites announce their services on the grid making them available for other RMP sites. Figure 2 also illustrates the two environment simulators and their relationship to the demonstrator architecture.

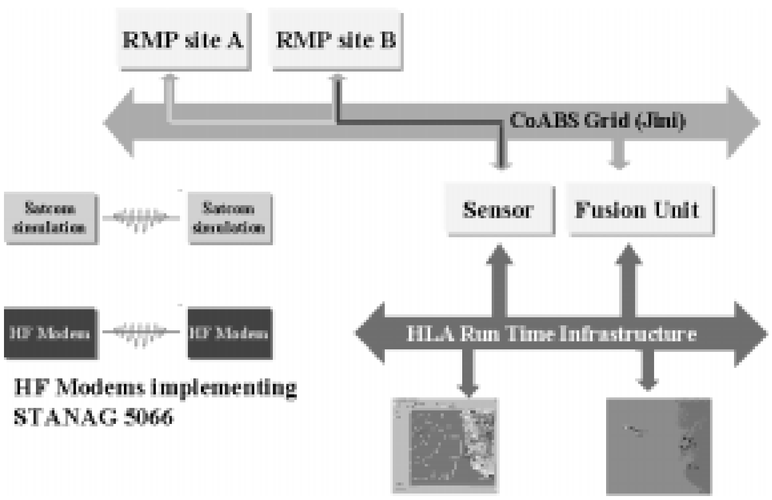

Figure 2 Architecture for distributed picture compilation

The two simulators are integrated using High Level Architecture (HLA), which is an architecture for distributed simulation. They simulate movements of vessels and aircrafts, sensors (radar, SAR, ISAR, ESM, IFF, EO, sonar), picture production, communication architecture and message exchange protocols. The behavior of the entities in a scenario is simulated according to a scenario script. The simulation can be run interactively and a user can introduce events during a simulation.

7. Experiences

Through the development process the project has gained experience with use of CORBA and CoABS. Functionality from two existing operational C2IS has been isolated and made available for external components. Components at the service layer have been integrated using CoABS. Although the integration of functionality from existing systems was successful the threshold for productive use of CORBA was significant. A lot of time was spent on gaining sufficient insight into the standard and the vendor's product supporting the standard.

The use of the Jini based CoABS Grid for integration of service layer components on the other hand, gave a different experience. The time invested in understanding the use of the Grid resulted in a much higher return. Although the CoABS Grid abstracts many details from Jini, it is believed that using Jini alone is also much simpler than using CORBA. Furthermore, as described in this paper, Jini and CoABS provide a flexibility, which is not provided by middleware such as CORBA or DCOM.

In the demonstrator CORBA is used solely for the integration of the legacy systems. Another approach to the integration would have been to Jini enable these systems. However, by using CORBA the legacy systems can now be accessed from any programming language, operating system and computer platform.

8. Summary

This paper has discussed the rationale for a set of essential properties of future maritime C2ISs and the means by which these properties may be realized. The paper has argued that architectural standardization efforts and middleware technologies that support dynamic interoperability are both important means for realizing the properties. As a case study on realizing a selection of the properties, a RMP production demonstrator with an open architecture, facilitating integration of two existing military operational systems using CORBA, Jini and CoABS Grid, has been built.

References

[1] Malerud, S., Helene Feet, E.F., Enemo, G., Bråthen, K. (2000), Assessing the Effectiveness of Maritime C2 Systems - Measures of Merit, in Proc. of the 2000 Command and Control Research and Technology Symposium, Monterey, California.

[2] NC3B (2000): NATO C3 System Architecture Framework, AC/322-WP/0125, 2000 (NATO Unclassified)

[3] NATO (1999): NATO Policy for C3 Interoperability, AC/322-WP/72, 1999 (NATO Unclassified).

[4] C4ISR AWG (1997): C4ISR Architecture Framework Version 2.0, C4ISR Architecture Working Group.

[5] ISO/IEC (1995): ITU-T X.901 | ISO/IEC 10746-1 ODP Reference Model Part. 1 Overview.

[6] Butler Group (1998): Component-Based Development - Application Delivery and Integration Using Componentised Software, Management Guide, UK.

[7] Aagedal J, Berre A-J, Bjanger B W, Neple T, Roark C B (1999): ODP-based Improvements of C4ISR-AF, Command and Control Research and Technology Symposium, Newport, US.

[8] Jai B, Ogg Michael, Ricciardi A (2000): Effortless Software Interoperability with Jini Connection Technology, Bell Labs Technical Journal, April-June 2000.

[9] Kahn M, Cicalese C (2001): CoABS Grid Scalability Experiments, 5th International Conference on Autonomous Agents. Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS 2001.

[10] Siegel J (2000): CORBA 3 Fundamentals and programming, Second edition, Object Management Group.

[11] Arnold K, et al (2000): The Jini Specifications, Second Edition, Addison-Wesley Pub Co; ISBN: 0201726173

[12] http://www.openwings.org