Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Message Oriented Management and Analysis Tool for Naval Combat Systems

Yang Wei * **Kyoung Sub Song** ** **Dong-Seong Kim** ***

* *Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, Korea (e-mail: yangwei@kumoh.ac.kr)*
** *Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, Korea (e-mail: kssong86@kumoh.ac.kr)*
*** *Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, Korea (e-mail: dskim@kumoh.ac.kr)*

Abstract: This paper proposes a design of message oriented modeling and analysis tool (MOMAT) for naval combat systems (NCS). NCS is composed of heterogeneous and large-scale components such as communication services and data distribution services (DDS). Each component consists of massive and heterogeneous interfaces and modules that are designed by many developers. Furthermore, MOMAT has to meet the requirements of NCS as follows. First, MOMAT has to be able to control multiple parts such as components, interfaces, messages and publisher/subscriber. Secondly, it should improve the application performance in terms of response time and execution time. Thirdly, the scalability of application functions has to be guaranteed for system integration of NCS with heterogeneous sub-parts. Implementation results show that the MOMAT is suitable for naval combat systems regarding task processing time and database access time.

Keywords: Naval Combat Systems, Message Oriented Modeling and Analysis Tool, Performance Analysis.

1. INTRODUCTION

Influence of modern weapon systems technology has led to changes in naval combat systems (NCS). A concept of Battlefield has been changed by engagement of heterogeneous vehicles such as fleet, submarines, and aircraft. Therefore, NCS provides information of all the situations based on reliability against unpredictable enemy attacks. To analyze the rapidly changing battlefield circumstances, a lot of sensor nodes of NCS renew the data in short period for real-time sharing of information like convention networked control systems D. S. Kim and Park (2003).

Middleware is one of the ways to manage a lot of heterogeneous and multiple data as real-time. Especially, data distribution service (DDS) is usually used in message oriented system environments. In DDS, the data organization of sensor nodes among components is fundamental to publish-subscribe systems. To develop the large-scale systems, number of developers participate in projects and cooperate with each other. Large-scale systems are composed of numerous components which depend on configuration of NCS H. Arciszewski and Delft (2009), Kim and Lee (2011), W. R. Otte and Willemsen (2011), H.-S. Park (2002).

A component of large-scale systems, for example, NCS is implemented by multiple developers. In the procedure of implementation, if change is not properly applied to component, it is suboptimal to maintain the version of each component. Therefore, it will lead to decrease the development efficiency. To solve this problem, various applications have been developed such as message definition & management system (MDMS), real-time data distribution service (RDDS), and etc Yu and Park (2008), W. Kang and Son (2012), N. Wang and Corsaro (2008), Beckmann and Thoss (2012), M. Xiong and Schmidt (2006), G. Deng and Edwards (2007).

However, these applications have a couple of problems such as server overload, low scalability and limitation of web-based application. To cover these problems, message oriented management and analysis tool (MOMAT) is designed to satisfy the following requirements. First, the version control can be divided into component, interface, messages, and publisher / subscriber. Secondly, it should improve the application performance in terms of response time and execution time. Thirdly, the scalability of application function has to be guaranteed for system integration (SI) with heterogeneous sub-part of NCS.

This paper shows a design and an implementation of MOMAT to evaluate a range of publisher / subscriber platforms. The proposed systems is operated on heterogeneous system environments to evaluate how MOMAT architecture affects the performance. Furthermore, this paper describes the design and core features of MOMAT to demonstrate how MOMAT satisfies the requirements of NCS.

The remainder of this paper is organized as follows. Section 2 explains the background such as NCS and DDS. In section 3, the overview of the MOMAT and design considerations are discussed. Implementation results and performance analysis are shown in Section 4. Finally, Section 5 presents the conclusion and future works.

2. BACKGROUND

2.1 NCS architecture

The basic systems of NCS are affiliated large-scale systems, which are composed of a variety of weapon systems such as monitoring and reconnaissance. Each weapon system can be regarded as a component. In large-scale system, it needs to handle more data messages than medium-scale systems. In other words, the number of data messages can be controlled through a message management tool. Table 1 presents the characteristics of the large-scale systems and the medium-scale systems.

Table 1. System characteristics

	Medium-scale systems	Large-scale systems
Total Endpoints	13019	24911
Publisher / Subscriber	5413 / 7606	10284 / 14627
Total Nodes	48	140
The largest topic size	8K	1M
High Frequent data	20Hz	200Hz
Bursty traffic	3000 at the same time per 1 sec	3500 at the same time per 1 sec
Total Domains	6	5

Total endpoints denote all of the publisher / subscriber in the NCS. The total nodes are devices such as radar, weapons and naval control devices. The largest topic size is the largest message size in NCS. High frequent data is an attribute of quality of service (QoS) in DDS. The bursty traffic means the number of publisher or subscriber messages sent at the same time. At last, total domains mean the number of global data space, which is the area for topics to communicate with DDS. That is to say, the NCS is required to have a communication middleware such as DDS to manage a large number of publisher, subscriber, and topics.

2.2 Communication middleware

The most widely used communication middleware can be classified into two categories: object oriented middleware (OOM) and message oriented middleware (MOM). In the case of using the OOM in the NCS, the communication with components is accomplished by changing the properties of the other components or running it. Typical middleware of OOM is common object request broker architecture (CORBA) and distributed component object model (DCOM). Another case of using the MOM in the NCS, the communication is performed through the method of message transfer among the components. OMG DDS, RTI DDS, and STC DDS are typical middlewares using this method in MOM D. S. Kim and Park (2003), H. Arciszewski and Delft (2009), Kim and Lee (2011), W. R. Otte and Willemsen (2011).

DDS is a communication middleware that is widely used in the NCS. An application of NCS can be ran on the NCS, which compose heterogeneous devices and platforms to publisher / subscriber data through a global data space in the NCS. Figure 1 shows the DDS communication structure in NCS. The MOMAT is developed considering environments of DDS communication structure.

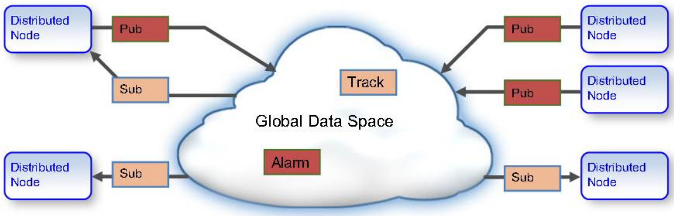

Fig. 1. DDS communication structure in NCS

3. DESIGN OF MOMAT

3.1 Core features and benefits of MOMAT

The MOMAT is developed by Samsung Thales for managing the messages of each component, which provides a data-centric communication based on DDS specification. The main feature of MOMAT is that all messages and components can be controlled by the version such as Subversion and concurrent version system (CVS). In addition, it provides a code generation function using interface definition language (IDL) from object management group (OMG). Using these benefits of MOMAT, message communication and large-scale systems design can be simplified. Scalability can be maximized by minimizing the influence of the existing systems H. Arciszewski and Delft (2009). The MOMAT consists of 4 parts: the database and C# language for operating application, the components, messages and publication / subscription, which is described in Figure 2.

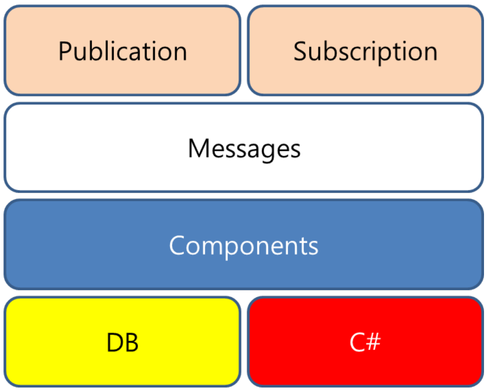

Fig. 2. Architecture of MOMAT

Significant parts of the architecture of MOMAT can be focused on components, messages and publication / subscription, details of these parts are as follows.

(1) **Components** The MOMAT can manage all the messages through components. The components are divided into 3 layers as computer software configuration item (CSCI), computer software components (CSC), and computer software item (CSU). Each component can communicate with another component when they get the data from heterogeneous sensor nodes.

(2) **Messages** A message can be registered to publisher / subscriber. Furthermore, it works as both user defined data structure for project and topic in DDS. Data transmission via messages in project is based on components.

(3) **Publication / Subscription** The publication / subscriber is one of the main features of MOMAT that communicates with each other using global data space.

3.2 Data procedure of MOMAT

MOMAT is designed and implemented using client / server topology. Figure 3 shows the application process of initialization of MOMAT. The information can be got from a database through the global value for decreasing the processing time of data exchange.

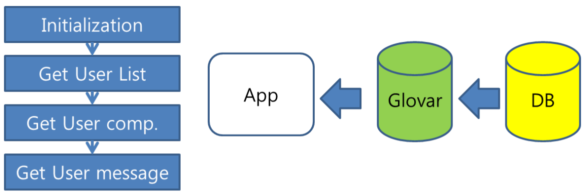

Fig. 3. Initialization procedure of MOMAT

The performance of application can be increased by using the global value. Detailed database process of MOMAT is shown in Figure 4. The update command uses a global value. Add command and delete command are directly connected to database.

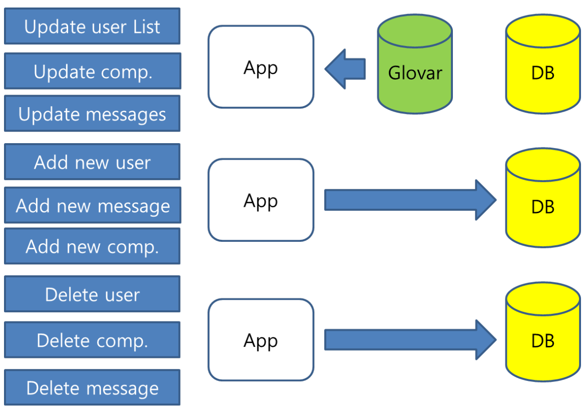

Fig. 4. Diagram of database process on MOMAT

4. IMPLEMENTATION AND PERFORMANCE ANALYSIS

This section describes that how the MOMAT implements and provides performance analysis results. This paper analyzes the degree of performance improvement through performance comparison between the MDMS and MOMAT.

4.1 Implementation

The MOMAT is developed by C# language to provide scalability. Figure 5 shows the layout and work session of MOMAT. Publication / subscription and components are in the left side of the application. The work session in the right side can carry out functions such as add new message or user define data type, modify the message or topic, check the history of message and delete the message.

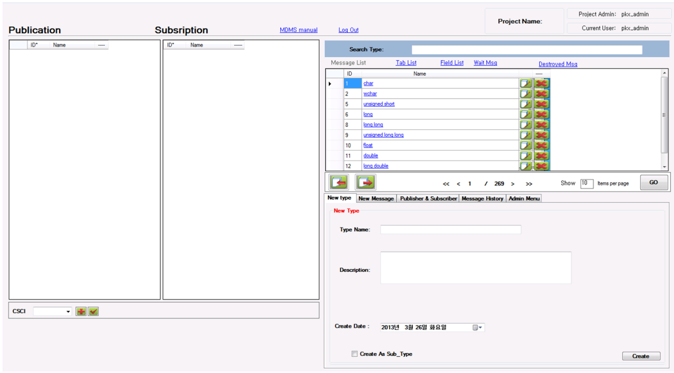

Fig. 5. Screenshot of work session

4.2 Experiment Environments

For enhancement of MOMAT efficiency, this paper focuses on the primary difference between MDMS and MOMAT regarding task processing time of server database and database access time of task. Experiment of MDMS and MOMAT is progressed under centralized server / client model. The database server and client desktop computer are consisted of similarity model based on intel dual core cpu, 2GB memory, windows operating systems and 100 Mbps. Figure 6 shows the centralized MOMAT network model for experiment.

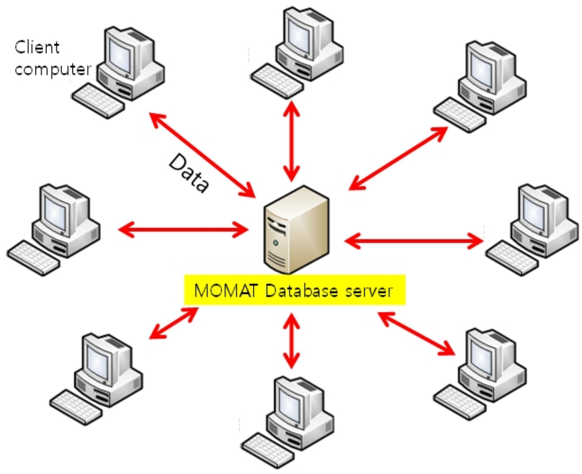

Fig. 6. Network model of centralized MOMAT

4.3 Experiment Analysis

To demonstrate a comparative study by performance analysis, task processing time of server database and average server load are used to compare MDMS and MOMAT. Server access packets are generated by developers when the messages are added, modified, and deleted, It can lead to an overload of server database. To solve this problem and provide a convenience, handiness, and scalability of application, MOMAT is developed by C# language. Through the measurement, this paper estimates that server database task processing time and average server load are more enhanced in the MOMAT than the MDMS in terms of performance analysis. Simulation environments are carried out in the similar condition. That is to say, the MOMAT can be helpful to increase the efficiency of development progress.

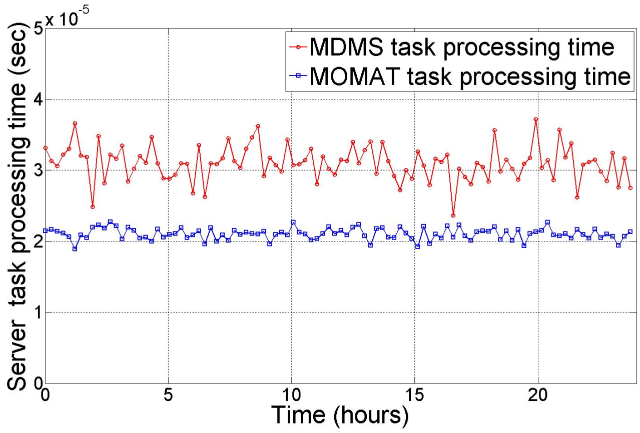

Fig. 7. Task processing time of server database

The experiment results are summarized in the figures in terms of server database task processing time and average server load. Figure 7 shows the task processing time of server database is compared between MDMS and MOMAT, it can be seen task processing time of MOMAT has the lowest value, with at least 30 % reduction and the highest value, with at least 25 % reduction as compared with MDMS. The reduction in task processing time of server database for this experiment describes that MOMAT is more suitable for large-scale system in NCS because MDMS could lead to server load in large-scale system of NCS.

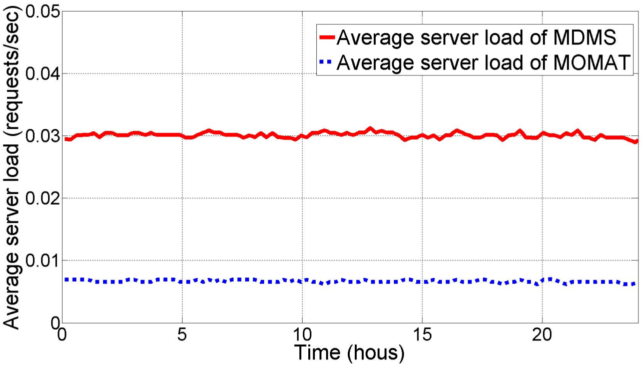

Fig. 8. Comparative results of average server loads

Figure 8 shows that an average server load of MDMS and MOMAT. The server load of MDMS located on 0.03 requests / sec and MOMAT located on 0.15 requests / sec during one day. It means that MOMAT has lower server load compared with MDMS, with at least 85 % reduction. The MOMAT can be more suitable for NCS which consists of number of messages than MDMS in terms of server load and task processing time.

5. CONCLUSION AND FUTURE WORKS

This paper describes a design of MOMAT for NCS. The MOMAT is designed and implemented considering the following requirements. First, the MOMAT has to support efficient management for components, interfaces, messages, publisher/subscriber and so on. Secondly, it should consider application performance in terms of response time and execution time. Thirdly, a scalability of application functions should be guaranteed for naval combat systems with heterogeneous sub-parts. For practical concern, the proposed method can be easily applied to existing devices without major changes. The MOMAT can be an efficient tool for the NCS design and systems integration.

Despite various advantages of the proposed MOMAT, MOMAT includes the limitations due to network congestion by a centralized network model. The centralized networks model can be helpful to control all the information in one server machine, store the information and manage the clients. However, limitation of the centralized network model can lead to a bottleneck when loaded in high traffic or access the server simultaneously. Accordingly, a research is needed to deal with the problems of the centralized network model.

As a future work, the message modeling of MOMAT will consider application execution time and maintenance. To enhance the performance of MOMAT, database optimization and software reconfiguration will also be considered.

ACKNOWLEDGMENTS

This research was financially supported by National Research Foundation of Korea (NRF) through the Human Resource Training Project for Regional Innovation 2014 and Basic Science Research Program (NO. 2011-0025409).

REFERENCES

Beckmann, K. and Thoss, M. (2012). A wireless sensor network protocol for the omg data distribution service. 45-50. IEEE 10th International Workshop on Intelligent Solutions in Embedded Systems.

D. S. Kim, Y. S. Lee, W.H.K. and Park, H.S. (2003). Maximum allowable delay bounds of networked control systems. *Control Engineering Practice*, 11(11), 1301–1313.

G. Deng, M. Xiong, A.G. and Edwards, G. (2007). Evaluating real-time publish/subscribe service integration approaches in qos-enabled component middleware. 222-227. Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing.

H. Arciszewski, T.E.G. and Delft, J.H. (2009). Adaptive automation in a naval combat management system. *IEEE Transactions on Systems, Man, and Cybernetics*, 39(6), 1188–1199.

H.-S. Park, D.-S. Kim, W.H.K. (2002). A scheduling method for network-based control systems. *IEEE Transactions on Control System Technology*, 10(3), 318–330.

Kim, D.S. and Lee, S.K. (2011). Feasibility analysis of hybrid control networks based on common industrial protocol. *Computer Standards & Interfaces*, 33(4), 357–366.

M. Xiong, J. Parsons, J.E.H.N. and Schmidt, D.C. (2006). Evaluating the performance of publish/subscribe platforms for information management in distributed real-time and embedded systems. Vanderbilt University, Nashville TN, USA.

N. Wang, D. C. Schmidt, H.v.H. and Corsaro, A. (2008). Toward an adaptive data distribution service for dynamic large-scale network-centric operation and warfare (ncow) systems. 1-7. IEEE Military Communications Conference.

W. Kang, K.K. and Son, S.H. (2012). Rdds: A real-time data distribution service for cyber-physical systems. *IEEE Transactions on Industrial Informatics*, 8(2), 393–405.

W. R. Otte, A. Gokhale, D.C.S. and Willemsen, J. (2011). Infrastructure for component-based dds application development. 53-62. Proceedings of the 10th ACM International Conference on Generative Programming and Component Engineering.

Yu, J.Y. and Park, J.H. (2008). A message management for cooperative message-based interface development. *Journal of Korean Institute of Information Scientists and Engineers*, 14, 609–613.