

## 선박 내 통합 정보 서비스를 위한 미들웨어 서버의 서비스 모듈 설계 및 구현

황훈규<sup>1</sup> · 김태종<sup>1</sup> · 윤진식<sup>1</sup> · 서정민<sup>1</sup> · 박휴찬<sup>2</sup> · 장길웅<sup>3</sup> · 이장세<sup>†</sup>

(원고접수일 : 2009년 12월 4일, 원고수정일 : 2009년 12월 27일, 심사완료일 : 2009년 12월 29일)

### A Design and Implementation of Service Modules in Middleware Server for Integrated Information Services on Shipboard

Hun-Gyu Hwang<sup>1</sup> · Tae-Jong Kim<sup>1</sup> · Jin-Sik Yoon<sup>1</sup> · Jung-Min Seo<sup>1</sup> · Hyu-Chan Park<sup>2</sup> ·  
Kil-Woong Jang<sup>3</sup> · Jang-Se Lee<sup>†</sup>

**요 약 :** 선박 내의 통합 정보 관리를 위한 MiTS 아키텍처는 게이트웨이, 미들웨어 서버 그리고 어플리케이션과 같은 세 개의 구성 요소로 이루어져 있다. 게이트웨이는 선박 내의 여러 장비에서 발생하는 정보를 수집하여 가공하는 기능을 가지고, 미들웨어 서버는 가공된 정보를 통합하여 데이터베이스에 저장하는 기능을 가지며, 어플리케이션은 수집되어 저장된 정보를 사용자에게 시각적으로 표시하는 기능을 가진다. 본 논문에서는 미들웨어 서버의 서비스를 게이트웨이와 어플리케이션이 제공받을 수 있도록 해주는 국제 표준 IEC61162-4 series 기반의 미들웨어 서버의 서비스 모듈을 제안하고 구현 및 검증한다.

**주제어 :** e-Navigation, MiTS, MiTS 미들웨어, 선박 통합 정보 서비스, 미들웨어 서비스

**Abstract:** In this paper, we design and implement middleware service modules of MiTS system architecture which is proposed for integrated shipboard information management. The MiTS system is divided to three layers that are gateway, middleware server and application. The gateway collects and parses information and data, the middleware server integrates the data and stores to a database, and the application displays visual data to users. The service modules provide MiTS middleware services to gateways and applications. We implement the service modules based on international standard IEC61162-4 series and test those.

**Key words:** e-Navigation, MiTS, MiTS middleware, Shipboard Integrated Information Service, Middleware service module

### 1. 서 론

e-Navigation은 해상 안전·보안과 해양 환경 보호를 위해 전자적 방법으로 정박지에서 정박지까지 항해와 관련 서비스들을 개선하여 선박과 육상

의 해양 정보 수집, 통합, 교환, 표현 그리고 분석이 잘 조화될 수 있도록 하는 것이다[1]. 이는 해상에서 발생할 수 있는 모든 일들의 종합적인 상황 인식을 위해 2006년 5월 국제해사기구(IMO) 해

<sup>†</sup> 교신저자(한국해양대학교 IT공학부, E-mail:jslee@hhu.ac.kr, Tel: 051-410-4577)

1 한국해양대학교 컴퓨터공학과 대학원

2 한국해양대학교 IT공학부

3 한국해양대학교 데이터정보학과

사안전위원회(MSC)의 81차 회의에서 제안되어 활발하게 관련된 연구가 진행되고 있다[2].

이러한 e-Navigation의 구현을 위한 여러 요소들이 필요하며, 그 핵심 요소 중의 하나로 MiTS(Maritime information Technology Standard)를 들 수 있다. [3]에서 제안된 MiTS 시스템 구조는 게이트웨이, 미들웨어 서버 그리고 어플리케이션 세 계층으로 나누어지며, 미들웨어 서버는 다른 구성 요소들에게 통합 정보 서비스를 제공할 수 있도록 해주는 기능이 요구된다. 따라서 본 논문에서는 선박 내 통합 정보 서비스를 위한 미들웨어 서버의 서비스 모듈을 제안한다.

다음 장에서 MiTS 시스템 구조 및 게이트웨이와 어플리케이션 측의 요구사항에 관하여 분석하고 3장에서는 분석된 내용을 바탕으로 미들웨어 서버 및 서비스 모듈의 설계를 다루며 4장에서 구현 및 검증 과정에 관해 다룬 후 5장의 결론 및 향후 연구로 끝을 맺는다.

## 2. 연구 분석

### 2.1 MiTS 시스템 구조

[3]에서 제안된 MiTS 시스템 구조는 **그림 1**과 같으며 크게 세 가지 부분으로 나눌 수 있는데, G/W는 게이트웨이, M/W Server는 미들웨어 서버, APP는 어플리케이션을 뜻한다. 나누어진 세 부분은 데이터 및 정보의 교환을 위하여 IEC61162-4 series에 정의되어있는 통신 프로토콜인 LNA (Local Network Administrator)와 MAU(MiTS Application Unit)를 가진다. 본

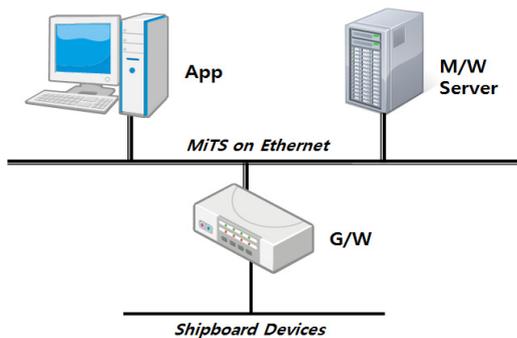


그림 1: MiTS 시스템 개념도

논문에서는 미들웨어 서버 내에 포함되어 게이트웨이와 어플리케이션에 서비스를 제공하는 미들웨어 서버 서비스 모듈의 구현에 관해 다루며 통신 프로토콜, 즉 LNA와 MAU에 대해서는 자세히 다루지 않는다.

MiTS 시스템은 기본적으로 이더넷(ethernet) 기반의 TCP/IP 프로토콜을 사용하고 있으며 MiTS 시스템을 구성하고 있는 각 부분의 기능은 아래와 같다.

첫째, 게이트웨이는 선박의 각종 장비들로부터 생성되는 정보나 데이터를 수집하여 가공하는 기능을 가진다. 선박에 있는 NMEA 2000 장비에서 실시간으로 생성되는 여러 종류의 정보나 데이터를 수집하여 MiTS 시스템에서 사용할 수 있는 형태로 가공해주며 이를 미들웨어 서버로 전송해 준다.

둘째, 미들웨어 서버는 선박에서 생성되는 정보나 데이터를 게이트웨이로부터 수신하여 통합적인 관리를 할 수 있도록 저장하는 기능을 가진다. 데이터베이스에 저장된 정보나 데이터는 어플리케이션의 요청 및 여러 가지 용도로 사용되어 질 수 있다.

셋째, 어플리케이션은 선박의 정보를 사용자가 볼 수 있도록 시각적으로 표시해주는 기능을 한다. 사용자가 원하는 정보나 데이터를 요청하게 되면, 어플리케이션은 미들웨어 서버로 해당 정보를 수신하여 계기판, 게이지, 차트 등의 형태로 표시해준다.

### 2.2 정보 서비스 요구사항 분석

#### 2.2.1 게이트웨이 요구사항

게이트웨이는 선박 내의 장비들로부터 수집하여 가공한 정보나 데이터를 미들웨어 서버의 데이터베이스에 통합·저장하여야 하기 때문에 저장과 관련한 서비스를 요구한다.

수집된 정보를 가공하여 미들웨어 서버의 데이터베이스에 저장할 때, 게이트웨이는 네트워크 상태 및 시스템 상태에 따라 두 가지의 서비스 중에 하나를 선택하여 이용할 수 있어야 한다. 첫째, 네트워크나 시스템의 상태가 비교적 양호하여 네트워크나 시스템의 부하가 걸리지 않을 것이라고 판단되는 경우, 각 정보나 데이터의 수집 시점마다 서비

스 모듈을 호출하여 미들웨어 서버 데이터베이스에 저장할 수 있어야 하며, 둘째, 시스템의 부하가 많을 경우, 정보나 데이터를 일정 크기만큼 누적시킨 후 한 번에 미들웨어 서버 데이터베이스에 저장할 수 있어야 한다.

tag Number	description	tag Type	eng Unit	eu Text	tag Semantics	authentication	precision	sample Time	tag Attributes	field Flags
1	course over ground	0	20	radians	2	1000	1.0E-04	250	0	3FF
2	speed over ground	0	30	m/s	2	1000	1.0E-02	250	0	3FF
...										

그림 3: TagInfo 테이블

### 2.2.2 어플리케이션 요구사항

어플리케이션은 사용자에게 선박 정보들을 시각적으로 표시를 해주며 미들웨어 서버의 데이터베이스에서 필요한 정보 및 데이터를 가져오기 위한 서비스를 요구한다.

tagNumber	state	alarm	time(sec)	time(usec)	value
1	0	0	20090805090701	2500000	2.8001
2	0	0	20090805090701	2500000	12.01
1	0	0	20090805090702	5000000	2.9001
2	0	0	20090805090702	5000000	13.01
...					

그림 4: TagValue 테이블

각 장비마다 표시되는 형식과 기준이 다르고 정보나 데이터가 수집되지 않는 장비에 대해서는 표시해주지 않아도 되기 때문에 선박 정보의 시각적인 표시를 위해서 어플리케이션은 정보나 데이터가 선박의 어떠한 장비들로부터 수집되어 미들웨어 서버의 데이터베이스에 저장되는지 알 수 있어야 한다. 정보나 데이터가 어떤 장비로부터 저장되는지 판단했다면 시각적으로 표시할 정보나 데이터를 미들웨어 서버의 데이터베이스로부터 가져올 수 있어야 한다. 필요에 따라 한 장비의 한 정보나 데이터, 한 장비의 여러 정보나 데이터, 여러 장비의 한 정보나 데이터를 가져오는 서비스를 선택적으로 이용할 수 있어야 한다. 또한 어떤 특정 기간의 정보나 데이터의 표시를 위해서 기간을 지정하여 그 기간의 데이터를 가져올 수 있도록 하는 서비스 모듈도 필요하다.

TagID 테이블은 number와 name의 속성들로 이루어져 있고 TagInfo 테이블은 tagNumber, tagID, description, tagType, engUnit, euText, tagSemantics, authentication, precision, sampleTime, tagAttributes, fieldFlags의 속성들로 구성되어 있으며 TagValue 테이블은 tagNumber, state, alarm, time\_sec, time\_usec, tagValue의 속성으로 이루어져 있다.

## 3. 설 계

### 3.2 서비스 모듈 설계

#### 3.1 미들웨어 서버 데이터베이스 구조

미들웨어 서버의 데이터베이스는 그림 2 ~ 4와 같이 TagID, TagInfo, TagValue의 세 개의 테이블로 이루어져 있다(4).

IEC61162-420의 companion standards 및 앞에서 도출된 요구사항을 바탕으로 게이트웨이 및 어플리케이션에 미들웨어 서버 서비스가 제공될 수 있도록 해주는 핵심 모듈을 설계하였다. 또한 MiTS 시스템 내에서의 최대 전송 사이즈는 그림 5와 같은 형식의 TagValue 구조체를 기준으로 최대 40개이다.

Tag Number	Tag Name
0	-
1	pNA000.GP000.AX.12902604
2	pNA000.GP000.SX.12902605
...	...

그림 2: TagID 테이블

```

struct TagValue // TagValue (data type: struct)
{
    int TagNumber;
    int State;
    short Alarm;
    char Time_sec[21];
    int Time_usec;
    float Value;
};
    
```

그림 5: TagValue 구조체 형식

### 3.2.1 게이트웨이를 위한 서비스 모듈

게이트웨이를 위한 서비스 모듈(함수)은 아래와 같다.

#### - setTagValue

input : struct TagValue  
output: none

게이트웨이가 setTagValue 함수를 호출하면, TagValue 구조체 형식을 가진 하나의 데이터를 미들웨어 서버의 TagValue 테이블에 저장할 수 있다.

#### - setTagValueList

input : struct[] TagValue  
output : none

setTagValueList 함수는 TagValue 구조체 형식을 가진 데이터를 최대 40개까지 한 번에 저장할 수 있도록 해주는 서비스를 제공한다.

### 3.2.2 어플리케이션 위한 서비스 모듈

어플리케이션을 위한 서비스 모듈(함수)은 아래와 같다.

#### - getTagNameList

input : none  
output : struct[] TagName

어플리케이션이 getTagNameList 함수를 호출하면, 미들웨어 서버 데이터베이스의 TagID 테이블에 있는 모든 TagName을 읽어 오는 서비스를 제공한다. 이는 어떤 장비로부터 정보나 데이터가 게이트웨이에 의해 수집되는지를 알 수 있게 해준다.

#### - getTagValue

input : int tagNumber  
output : struct[] TagValue //current value  
미들웨어 서버의 데이터베이스로부터 정보나 데이터를 가져오기 위한 함수들 중 getTagValue 함수는 tagNumber를 입력 값으로 전달받아 입력 값에 해당하는 tagNumber를 가진 모든 열에서 40개의 TagValue 구조체 형식의 값을 선택하여 그 값을 어플리케이션으로 전달한다.

#### - getTagOneValue

input : int tagNumber  
output : struct TagValue // current value

getTagOneValue 함수는 해당하는 tagNumber의 가장 최근의 한 TagValue 구조체 형식의 값을 어플리케이션으로 전달해주는 기능을 한다.

#### - getTagValueList

input : int[] tagNumber  
output : struct[] TagValue

//matching with each tag number

getTagValueList 함수는 입력 값으로 최대 40개까지의 tagNumber를 전달받아 각 입력 값에 해당하는 가장 최근의 TagValue 구조체 형식의 값을 최대 40개까지 출력 값으로 전달해준다.

#### - getTagValueInterval

input : int tagNumber, char[] s\_time\_sec, char[] e\_time\_sec

//2: start time, 3:end time

output : int[] count, struct[] TagValue

//1: counting of selected TagValue

getTagValueInterval 함수는 어떤 장비의 특정한 기간 사이의 정보나 데이터를 읽어오는 서비스를 제공한다. tagNumber와 시작시간, 종료시간을 입력으로 받고 시작시간과 종료시간 사이의 TagValue를 카운트와 함께 출력 값으로 전달해 준다.

### 3.3 미들웨어 서버 시스템 구조

그림 6은 앞에서 설계한 내용을 바탕으로 "M/W Server(미들웨어 서버)"의 내부 동작 과정을 나타낸 것이며 그 설명은 아래와 같다.

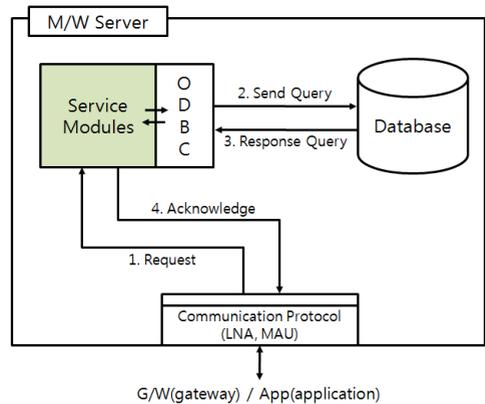


그림 6: 미들웨어 서버 내부 동작 과정

- ① 게이트웨이 혹은 어플리케이션으로부터 서비스 요청을 받는다.
- ② 미들웨어 서버 서비스 모듈은 ODBC(open database connectivity)를 통해 데이터베이스로 질의(query)문을 보낸다.
- ③ 데이터베이스로부터 질의에 대한 응답을 받는다.
- ④ 게이트웨이 혹은 어플리케이션에게 요청받은 서비스에 대한 응답을 해준다.

부이며 미들웨어 서버 데이터베이스의 TagID 테이블에서 TagName을 선택하여 가져오는 것을 확인할 수 있다.

### 4. 구현 및 검증

제안한 서비스 모듈의 동작을 검증하기 위하여 그림 7과 그림 8과 같이 C++ 및 C# 언어를 이용하여 미들웨어 서버의 서비스 모듈을 구현하였다.

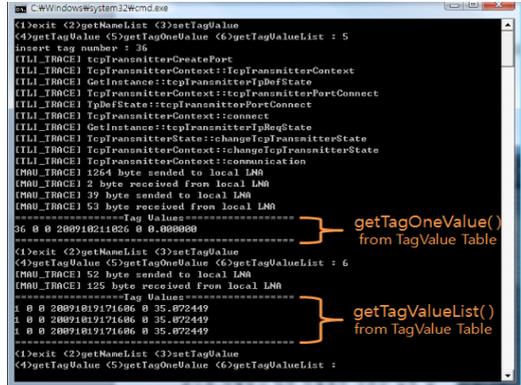


그림 9: getTagOneValue 및 getTagValueList 실행 결과

```
#include "MW_Functions.h"

#include "Lengths.h"
void ProcessLogMessages(HEWY pin_henv, HDBC pin_hdbc, HSTNT pin_hstnt, char *logstring)
bool MSSQLConnect(...)
// 연결해제

void SetTagValue(TagVal _TagData)
void SetTagValueList(int cnt, TagVal* _TagData)
void GetTagNameList(int* pt_cnt, TagName* nameList)
void GetTagValue(int _TagNumber, TagVal* _tagvalue)
TagVal* GetTagValueList(int _TagNumber, int* pt_cnt)
TagVal* GetTagValueInterval(int _TagNumber, int* pt_cnt, char* startime, char* endTime;
```

그림 7: C++로 구현한 서비스 모듈 코드 일부

```
namespace MiTS_MW_Func
{
class DB
{
public struct TagValue // tagValue 구조체 데이터
public struct TagName // tagName 구조체 데이터

string SQLconn;
SqlConnection Conn = new SqlConnection();

public void Database_Init(...) // DB Initialization Function

public TagName[] GetTagNameList(...)

public TagValue[] GetTagValue(int _TagNumber...)

public TagValue[] GetTagOneValue(int _TagNumber...)

public void setTagValue(TagValue _inData)

public void setTagValueList(TagValue[] _inData)

}

class Program
{
static void Main(...) // Test codes

}
}
```

그림 8: C#으로 구현한 서비스 모듈 코드 일부

그림 9는 본 논문에서 구현한 서비스 모듈 중 getTagOneValue 함수와 getTagValueList 함수를 호출한 결과 화면이며 미들웨어 서버 데이터베이스의 TagValue 테이블에서 데이터를 선택하여 가져오는 것을 확인할 수 있다. 또한 그림 10은 getTagNameList 함수를 호출한 결과 화면의 일

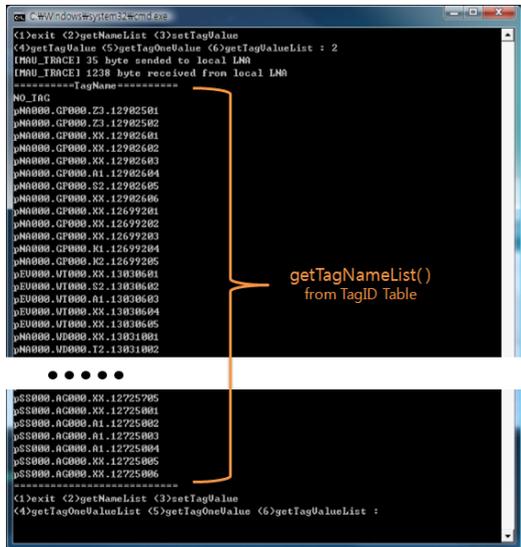


그림 10: getTagNameList 실행 결과의 일부

### 5. 결론 및 향후 연구

본 논문에서는 선박 내의 통합 정보를 제공하기 위한 MiTS 미들웨어 서버의 서비스 모듈을 제안하고 구현하였다. 구현된 미들웨어 서버 서비스 모듈을 통해 게이트웨이와 어플리케이션이 MiTS 미들웨어 서버의 서비스를 제공받을 수 있음을 확인하였다. 또한 C++ 및 C# 언어를 사용한 구현을

제안하여 개발환경에 따라 서비스 모듈을 선택적으로 이용할 수 있도록 하였다.

향후 연구로는 본 논문에서 제안한 6가지 핵심 서비스 모듈 이외에 IEC61162-420 annex E에서 요구하는 부가적인 서비스 모듈 및 기능에 대한 구현이 필요하며 새롭게 대두되고 있는 IEC61162-450에 관한 연구 또한 필요하다.

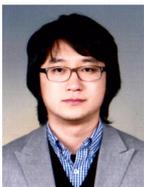
후 기

본 연구는 지식경제부 및 정보통신산업진흥원의 IT핵심기술개발사업의 일환으로 수행하였음[2008-F-046-01, E-Navigation 대응 IT-선박 융합 핵심기술 개발].

참고문헌

- [1] 해양수산부, 국제해사기구(IMO) 제53차 항해 안전전문위원회(NAV) 회의 결과, 해양수산부, 2007
- [2] 정중식, 김선영, "E-Navigation 서비스 제공을 위한 통신망 아키텍처의 설계", 한국항해항만학회지, 제32권, 제1호, pp. 37-45, 2008
- [3] 박휴찬, 이장세, 장길웅, 이정우, 정희섭, 박중현, 강순열, "선박에서의 통합 정보처리를 위한 시스템 아키텍처", 2009년도 전기학술대회논문집, 한국마린엔지니어링학회, 2009
- [4] Hyu-Chan Park, Jang-Se Lee, Kil-Woong Jang, Jeong-Woo Lee, Joong-Hyun Park, Soon-Yul Kang, "Network platform for integrated information exchange on ship", Proceedings of the ISME 2009, 2009

저 자 소 개



황훈규(黃勳圭)

2009년 한국해양대학교 IT공학부 컴퓨터정보공학전공(공학사), 2009년-현재 한국해양대학교 대학원 컴퓨터공학과 석사과정. 관심분야: 정보보안, 시뮬레이션, 네트워크



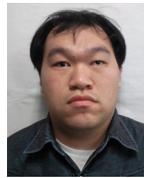
김태중(金泰鍾)

2008년 한국해양대학교 기계·정보공학부 제어·컴퓨터정보공학전공(공학사), 2008년-현재 한국해양대학교 대학원 컴퓨터공학과 석사과정. 관심분야: 데이터베이스



윤진식(尹珍植)

2009년 한국해양대학교 IT공학부 컴퓨터정보공학전공(공학사), 2009년-현재 한국해양대학교 대학원 컴퓨터공학과 석사과정. 관심분야: 정보보안, 네트워크, 포렌식



서정민(徐正民)

2009년 한국해양대학교 IT공학부 컴퓨터정보공학전공(공학사), 2009년-현재 한국해양대학교 대학원 컴퓨터공학과 석사과정. 관심분야: 데이터베이스



박휴찬(朴休讚)

1985년 서울대학교 전자공학과(공학사), 1987년 한국과학기술원 전기및전자공학과(공학석사), 1995년 한국과학기술원 전기및전자공학과(공학박사), 1987년 - 1990년 금성반도체, 1997년 - 현재 한국해양대학교 IT공학부(교수). 관심분야: 데이터베이스, 데이터마이닝, 해양정보시스템



장길웅(張吉雄)

1997년 경북대학교 컴퓨터공학과(공학사), 1999년 경북대학교 컴퓨터공학과(공학석사), 2002년 경북대학교 컴퓨터공학과(공학박사), 2003년 - 현재 한국해양대학교 데이터정보학과(부교수). 관심분야: 네트워크 프로토콜, 유비쿼터스 네트워크



이장세(李章世)

1997년 한국항공대학교 컴퓨터공학과(공학사), 1999년 한국항공대학교 컴퓨터공학과(공학석사), 2003년 한국항공대학교 컴퓨터공학과(공학박사), 2004년 - 현재 한국해양대학교 IT공학부(조교수). 관심분야: 컴퓨터보안, 지능시스템, 모델링 및 시뮬레이션