Development of Network Platform for Integrated Information Exchange on Shipboard

Hyu-Chan Park[1], Jang-Se Lee[1], Kil-Woong Jang[1], Jeong-Woo Lee[2], Joong-Hyun Park[3], Soon-Yul Kang[4]

[1]*Korea Maritime University, Korea,*
[2]*Daeyang Co., Korea,*
[3]*ForceTEC Co., Korea,*
[4]*E-Biz Information Technology Valley Co., Korea,*

Summary
Several kinds of devices and systems are installed on shipboard, and may generate huge amount of data and information. Such information should be exchanged reliably to aid system integration and safe navigation. This paper proposes a network platform for the integrated information exchange. To verify the functionality of the platform, a prototype system has been implemented and tested. It is expected to be a component of e-Navigation implementation in the future.
Key words:
Shipboard, Network Platform, Information Exchange, IEC 61162-4.

1. Introduction

Ships are equipped with several kinds of devices and systems. Integration of such systems, collection of data, and sharing of information are crucial for safe and efficient ship navigation. Each of them requires standardized data communications with several capabilities. To meet this requirement, International Electrotechnical Commission(IEC) has announced a series of standards for data communications. Fig. 1 depicts the network hierarchy with the standards.

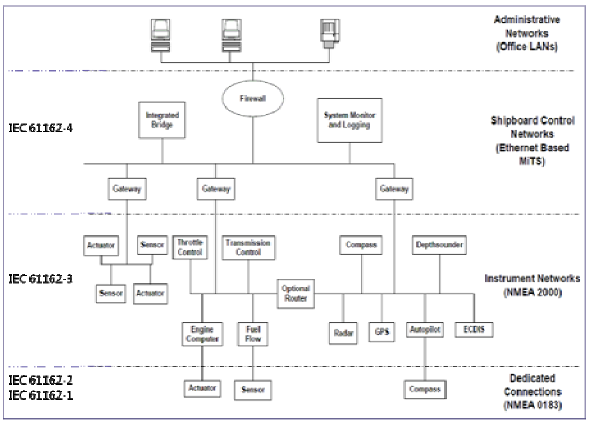


Fig. 1 Network hierarchy on shipboard

Among the standards, IEC61162-4 specifies a communication protocol for use in interconnected maritime systems [1]. It also specifies an interface description language for use together with the protocol, and a set of data format and interfaces described in the language. The protocol is intended for use in the system level of interconnected maritime systems, and designed to integrate various relatively large functional components. It complements other protocols on the instrument level, for example NMEA2000 [2], and on the administrative level.

This paper adopts the standard as the basis of a network platform for the integrated information exchange. In the platform, data from NMEA2000 networks are transformed and collected in a database. The database, in turn, can provide integrated information and several services to the applications. All the information is exchanged through the communication protocol [3]. To verify the functionality of the platform, a prototype system has been implemented and tested. It is expected to be a component of e-Navigation implementation in the future.

This paper first reviews network protocol and data architectures, and then presents design and implementation of the platform.

2. Preliminary

2.1 Communication Protocol

The international standard, IEC61162-4, specifies the communication protocol. The protocol is mainly composed of T-Profile(Transport Profile) and A-Profile(Application Profile) as shown in Fig. 2. T-Profile is defined on the Ethernet, IP, and TCP/UDP. It provides TLI(Transport Layer Interface) to the A-Profile. A-Profile is composed LNA(Local Network Administrator) and MAU(MiTS Application Unit). LNA is the network interface module, and MAU is the container for application programs. On top of the MAU, user applications may be developed to provide several services.

Manuscript received January 5, 2010
Manuscript revised January 20, 2010

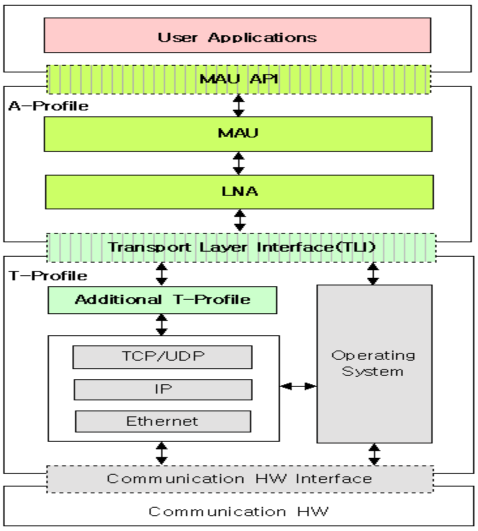


Fig. 2 Protocol stack

2.2 State and Sequence Diagram

The functionalities of LNA and MAU are defined as state diagrams and sequence diagrams. MAU state diagram, as an example of the state diagrams, is depicted in Fig. 3 [1]. A call to the service MauInitialize defines the MAU's context, and then a connection attempt to the LNA is initiated by a call to MauOpen. When the connection to the LNA is established, the MAU enters MAU_OPEN state.

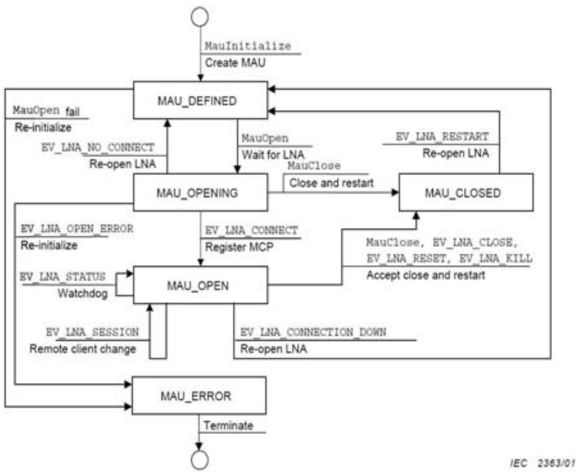


Fig. 3 MAU state diagram

The sequence diagram for normal data transfer, as an example of the sequence diagrams, is shown in Fig. 4 [1]. The sequence is initiated by the client MAU and propagated forward to the remote MAU through the LNAs, and then back again through the LNAs.

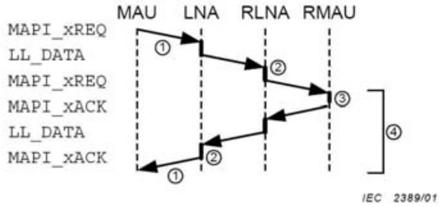


Fig. 4 Normal data transfer sequence

2.3 Message Format

The protocol uses thirty message types for system management and data transfers. Fig. 5 depicts two message types used to transfer data. MAPI_xREQ is an MAU-LNA message, and LL_DATA is an LNA-LNA message .

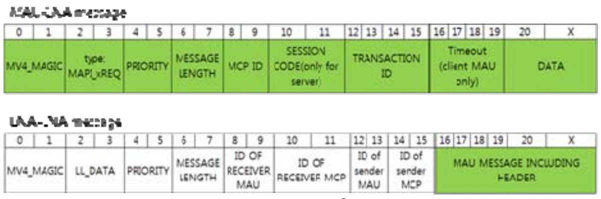


Fig. 5 Message format

2.4 Data Format

The standard also specifies a set of application data formats. The tag structure is the most valuable one, and defined as the following table. TagId is the identification of a Tag, TagInfo is static properties of a Tag, and TagValue contains values for the Tags.

TagId	TagInfo	TagValue
TagNumber number TagName name	TagNumber tagNumber [48]char8_m description TagType tagType EngineeringUnit engUnit [8]char8_m euText TagSemantics tagSemantics TagAuthentication authentication float64_m precision word32_m sampleTime int16_m tagAttributes [10]FieldOk fieldFlags	TagNumber tagNumber StateCode state AlarmState alarm GlobalTime time float64_m value

The tag name is formatted as described in the following table. The maximum length of 24 characters allows main group with three digits instance, sub group with three digits instance, data type, and eight digits serial number.

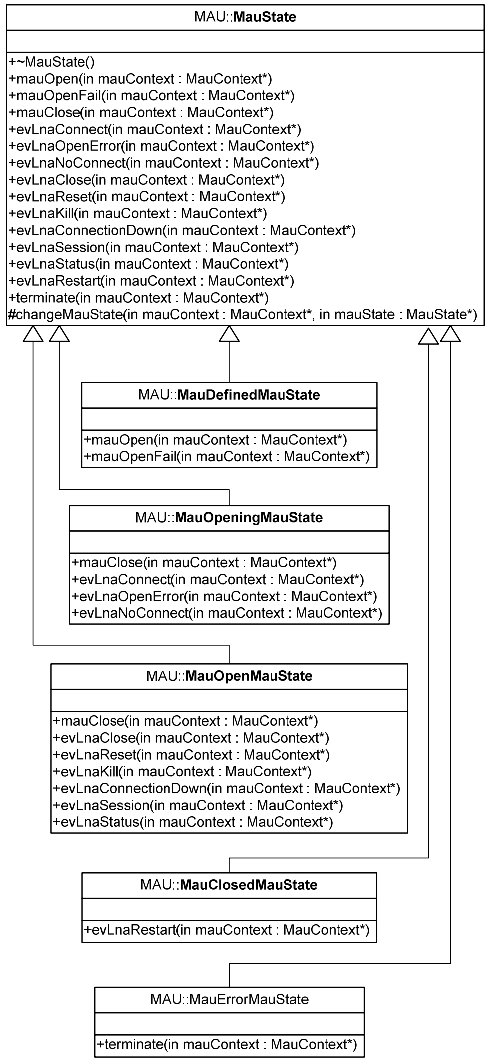
pMGnnn.SGnnn.TC.nnnnnnnn	
p	- Identify p Class
MG	- Main group (two letters)
nnn	- Main group instance number
SG	- Sub group (two letters)
nnn	- Sub group instance number
TC	- Data type code (two letters)
nnnnnnnn	- Unique serial number

3. Design of Network Platform

3.1 Class Diagram

The protocol's state transition can be implemented as objected-oriented classes [4]. Fig. 6 depicts the class diagram corresponding to the MAU state diagram in Fig. 3. MauContext defines methods for every event and action, and maintains a state object (an instance of a subclass of MauState) that represents the current state. It receives events from other objects, and passes the events to the state object. MauState defines methods for every event as default behavior when a specific event is invalid for the current state. Each subclass, for example MauDefinedMauState, defines methods to handle each of the valid events that can be received in that state. These methods simply execute the actions defined in MauContext.

(a) MAUContext class



(b) MAUState class and subclasses

Fig. 6 Class diagram corresponding to the MAU state diagram

3.2 Tag Naming

Each data item of NMEA2000 can be identified by the identity of a device and its PGN(Parameter Group Number) field. The following table shows a Tag name mapping from NMEA 2000 data item.

NMEA2000		Tag Name	
Field		Field	Description
Device	industry group	p	identify p class
	device class	MG	main group
	system instance	nnn	main group instance number
	device function	SG	sub group

	device instance	nnn	sub group instance number
PGN	PGN field data type	TC	data type code (type/role)
	PGN + PGN field number	nnnnnnnn	unique serial number

3.3 Tag Database

Tag database is mainly composed of TagId, TagInfo, and TagValue tables. TagId table simply contains every Tag name with the corresponding Tag number. TagInfo table contains every Tag's properties which can be obtained from NMEA2000 PGN field definitions. TagValue table stores all the data collected from NMEA 2000 devices.

[TagId]

number	name
1	*pNA000.GP000.AX.12902604*
2	*pNA000.GP000.SX.12902605*
...	

[TagInfo]

tag Number	description	tag Type	eng Unit	eu Text	tag Semantics
1	*course over ground*	*0*	*20*	*radians*	*2*
2	*speed over ground*	*0*	*30*	*m/s*	*2*
...					

authentication	precision	sample Time	tag Attributes	field Flags
1000	*1.0E-04*	*250*	*0*	*3FF*
1000	*1.0E-02*	*250*	*0*	*3FF*
…				

[TagValue]

tagNumber	state	alarm	time(sec)	time(usec)	value
1	*0*	*0*	*s1*	*u1*	*2.8001*
2	*0*	*0*	*s1*	*u1*	*12.01*
1	*0*	*0*	*s2*	*u2*	*2.9001*
2	*0*	*0*	*s2*	*u2*	*13.01*
...					

4. Prototype Implementation

The prototype system depicted in Fig. 7 is mainly composed of three components, i.e., Gateway(G/W), Middleware Server(M/W), and Application(App). All the data exchanges are performed through the MAU and the LNA modules.

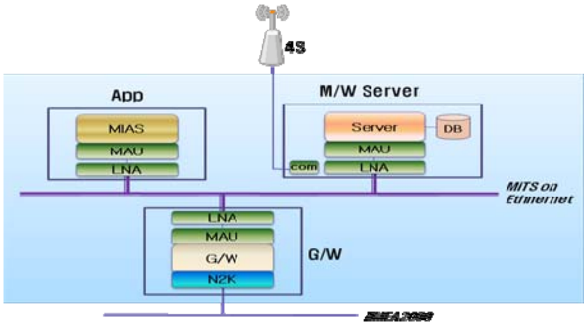


Fig. 7 Prototype system architecture

The G/W receives data from an NMEA2000 network, transforms into the Tag format, and transfers them to the M/W Server. Fig. 8 shows a snapshot of the G/W.

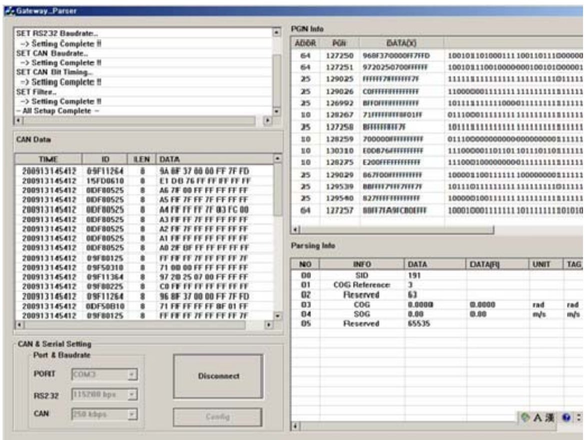


Fig. 8 Snapshot of Gateway

The M/W Server collects Tag data in its database, and then provides services to the App. The Tag data may also be transferred via the 4S(ship-to-ship, ship-to-shore) communications. Fig. 9 shows a snapshot of the M/W Server.

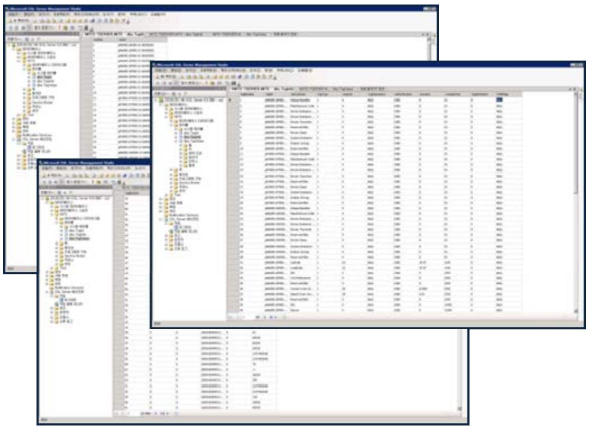
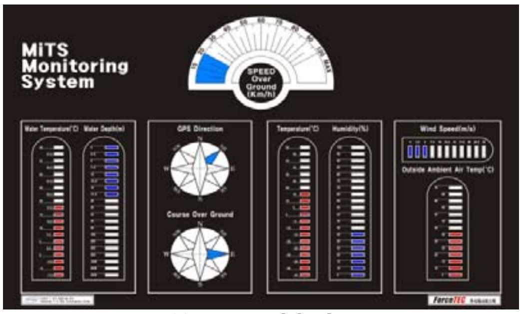
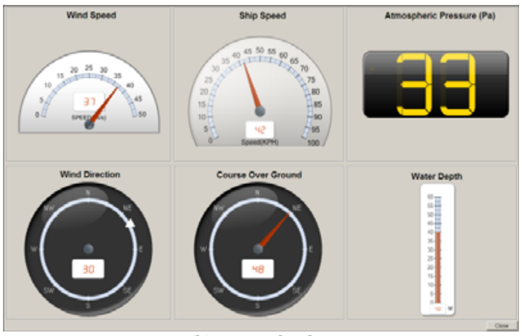


Fig. 9 Snapshot of Middleware Server

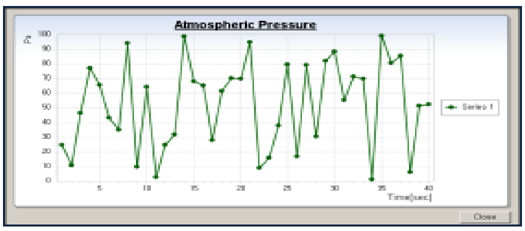
The App requests Tag data from the M/W Server, and then deploys received data to users. Fig. 10 shows a snapshot of the App.



(a) LED panel display



(b) gauge display



(c) chart display

Fig. 10 Snapshot of Application

5. Conclusions

This paper proposed a network platform for the integrated information exchange on shipboard. It collects data from NMEA2000 devices, and transforms them into the Tag format. For the integrated management of the Tag data, a database was designed. In the platform, all the information is exchanged through the communication modules. The functionality of the platform has been verified with the prototype system.

Although this paper has focused on the collection of NMEA2000 data, other data can be collected with an appropriate transformation. The performance evaluation will be conducted in the near future.

Acknowledgement

This work was supported by the IT R&D program of MKE/NIPA. [2008-F-046-01, Development of Core Technology of E-Navigation for IT Ship]

References

[1] *IEC61162-4: Maritime Navigation and Radiocommunication Equipment and Systems - Digital Interfaces - Multiple Talkers and Multiple Listeners - Ship Systems Interconnection*, 2001.

[2] *NMEA2000: Standard for Serial-Data Networking of Marine Electronic Devices*, 2004.

[3] H. C. Park, J. S. Lee, K.W. Jang, *et.al.*, "Network Platform for Integrated Information Exchange on Ship", *Proc. on ISME2009*, 2009.

[4] H. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Objected-Oriented Software*, Addison-Wesley, 1994.