Journal of the KIMST, Vol. 23, No. 2, pp. 176-184, 2020
ISSN 1598-9127(Print) · 2636-0640(Online)
DOI https://doi.org/10.9766/KIMST.2020.23.2.176

Research Paper 체계공학 부문

함정 기반 다중 무장 호환 운용을 위한 발사제어기 아키텍처

신진범*,1) · 조길석1) · 유명환1) · 김태현1)
1) 국방과학연구소 제1기술연구본부

Architecture of A Launch Control Unit for the Compatibility of Weapon Systems Based on Shipboard

JinBeom Shin*,1) · KilSeok Cho1) · MyongHwan Yoo1) · TaeHyon Kim1)
1) *The 1st Research and Development Institute, Agency for Defense Development, Korea*

(Received 3 September 2019 / Revised 6 February 2020 / Accepted 20 March 2020)

Abstract

In this paper, we have proposed hardware and software architecture of a launch control unit for the compatibility between air defense weapon systems loaded on shipboard. Until now, there is no compatibility between weapon systems loaded in battleships of korean navy. In the case of HaeGung system recently completed the test and evaluations, although it will be deployed on several kinds of shipboards, it has no compatibility and flexibility with other air defense weapon systems. Recently it reports that a long range air defense weapon system will be carried on future korean destroyer KDDX. Because the HaeGung and a long range air defense system will be operated together in KDDX, it is necessary to provide the compatibility between two weapon systems. So we have proposed architecture to provide the compatibility of the launch control unit that controls the launching system and the missile interface unit, and the missile in each weapon systems.

Key Words : Software Architecture(소프트웨어 아키텍처), Compatibility(호환성), Flexibility(유연성), Air Defense Weapon System(대공무장체계), Launch Control Unit(발사제어기)

1. 서 론

세계적으로 해군 함정들의 전투체계 및 탑재무장들의 다양화에 따라 함정 전투체계들의 범용화 개발[1]이 이루어지고 있고, 전투체계와 무장 간의 연동 프로토콜도 표준화[2]하고 있으며, 순양함과 구축함용 이지스 전투체계(미국)는 현대화 계획에 따라 호환성을 위한 Baseline 개념[3,4]으로 개발되고 있다. 전투체계와 연동하여 운용되어야 하는 국내 개발 대함, 대공 및 대지 무장체계의 경우에도 무장 간의 호환성을 고려한 개발이 필수적으로 요구되며, 따라서 무장 호환 운용을 위한 하드웨어 및 소프트웨어 아키텍처 설계가 필수적이다.

* Corresponding author, E-mail: jinbumshin@add.re.kr
Copyright ⓒ The Korea Institute of Military Science and Technology

함정에 기존에 개발된 다수의 무장체계를 탑재할 경우, Fig. 1과 같이 각 무장체계는 개별적으로 탑재될 수밖에 없다. 이 구조에서는 무장체계 발사제어기(Launch Control Unit)에서 수직발사대(Vertical Launching System, 이하 발사대라고 함) 및 유도탄의 혼합 운용이 불가하므로 전투체계 WCS(Weapon Control System)에서 무장체계 개별로 교전 명령을 송신하고, 무장체계는 각각의 발사제어기에서 유도탄을 발사하게 된다. 이런 구성은 무장체계 탑재 간에 호환 및 확장이 불가하며, 무장 수에 대비 장비 수량이 증가하여 장비 공간이 늘어나고, 운용 및 비용 측면에서도 비효율적이다. 그러므로 향후 개발 함정 무장체계는 무장 상호간에 호환성과 확장 유연성이 필수적으로 요구된다.

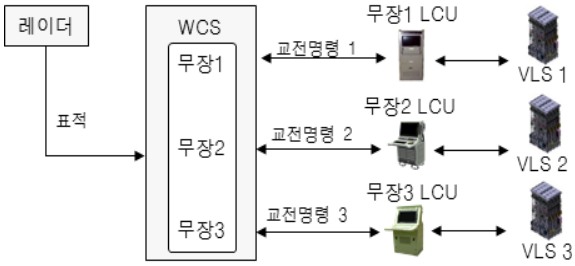

Fig. 1. Legacy weapon system on shipboard

다수 무장을 제어할 수 있는 호환성과 확장성을 가진 발사제어기를 적용한 함정 무장체계의 구성은 Fig. 2와 같으며, 발사제어기를 중심으로 발사대 및 유도탄의 다양한 무장 혼합 운용이 가능하다. 발사제어기는 유도탄 종별 인벤토리를 함정 전투체계 WCS에 보고하고, WCS는 무장(유도탄)별 교전 명령을 발사제어기로 송신하며, 발사제어기는 무장별 발사절차를 수행하여 유도탄을 발사할 수 있다. 이런 구성은 무장 간 변경 및 확장 시에도 큰 변경 없이 적용 가능하다는 큰 장점이 있다. 이질적인 함정용 대공, 대함 및 대지 무장체계 간의 혼합 운용을 위해서는 무장 이더넷 네트워크 구성과 무장 연동 메시지 및 발사제어기 소프트웨어의 표준화 기준이 필요하다. 소프트웨어의 경우 안정성을 위한 모델 기반 객체 구조[5,6] 설계와 객체 컴포넌트들의 연동 호환성[7,8] 설계가 요구되며, 최종적으로 소프트웨어의 안정성 검증[9]이 요구된다.

다수 무장의 혼합 운용이 가능한 발사제어기를 포함한 무장체계 기본 아키텍처는 Fig. 3과 같으며, 발사제어기에는 혼합 운용 소프트웨어가 탑재된다. 발사제어기는 이더넷 기반으로 함정 플랫폼 및 무장체계 장비들과 메시지를 송수신한다. 함정 레이더를 통한 업링크가 요구되는 대공 무장의 경우 발사제어기에서 함정 연동단을 통하여 레이더로 업링크 정보를 전송한다. 본 논문은 대함, 대공 및 대지 무장체계의 호환 운용을 위한 아키텍처를 제안하고, 무장체계 네트워크 구성안과 무장 연동 메시지들의 표준안 및 발사제어기 소프트웨어의 표준안을 제시한다. 또한 제안된 혼합 무장 발사제어기 소프트웨어가 제시된 기법대로 동작하는 지에 대한 시뮬레이션 결과를 제시한다.

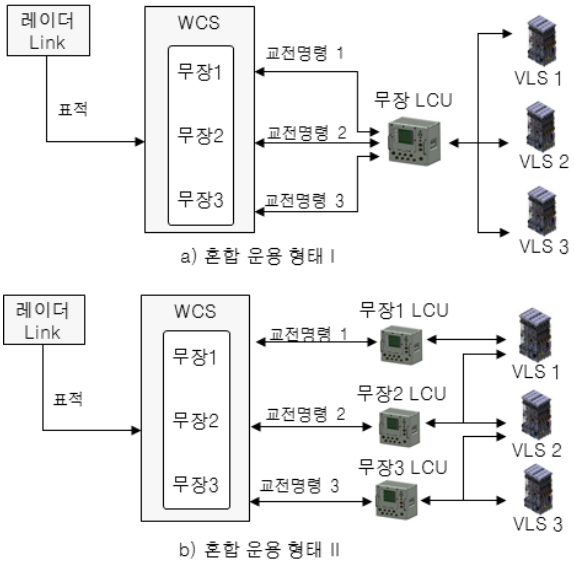

Fig. 2. Configuration of future weapon systems

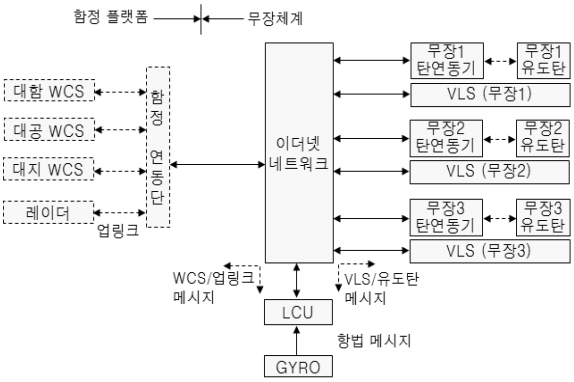

Fig. 3. Architecture of future weapon systems

2. 호환성을 위한 무장체계 하드웨어 및 소프트웨어

2.1 무장 네트워크 구성

발사제어기가 함정 플랫폼과 발사대 및 유도탄을 호환 운용하기 위해서는 Fig. 4와 같은 이더넷 네트워크의 구성이 필요하다. 이더넷을 지원하지 않는 구성 장비(예로 함정 자이로는 RS-422 통신 방식일 수 있음)에는 전단에 이더넷 변환기가 탑재되어야 하며, 변환기에 대해서는 별도 표기하지 않는다. 발사제어기는 함정 연동단을 통하여 함정 전투체계의 WCS 및 레이더와 연동하며, 무장 이더넷 네트워크를 통하여 함정 자이로, 탄연동기 및 발사대와 연동한다. 탄연동기는 대함, 대공 및 대지 유도탄과 무장 개별의 인터페이스로 연동한다. 이런 네트워크 구성은 무장 간의 구성 변경 및 확장 운용에 대하여 유연성을 제공하므로 필수적으로 적용이 고려되어야 한다.

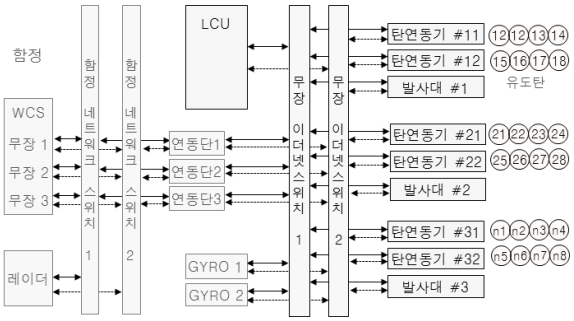

Fig. 4. Network configuration of future weapon systems

2.2 무장 메시지 이중화 송수신 프로토콜

무장 이중화 네트워크에서 메시지 송수신 프로토콜은 Replication 기법을 적용하며, 송신자는 메시지를 네트워크로 이중 송신한다. 수신자는 먼저 도착한 메시지를 사용한다. 이중화 기법의 메시지 처리를 위해서는 메시지 패킷의 헤더에 Sequence Number의 적용이 필요하다.

2.3 발사제어기 소프트웨어

무장 호환성을 위하여 발사제어기에는 Fig. 5와 같은 LCM(Launch Control Module) 소프트웨어가 필요하며, LCM 소프트웨어는 함정 WCS, 발사대(VLS), 탄연동기(MIU) 및 Cell(유도탄) 들의 호환 운용을 위한 연동 Controller Class 들과 발사제어기를 운용하고 발사절차를 생성 통제하는 Module Controller Class로 구성된다.

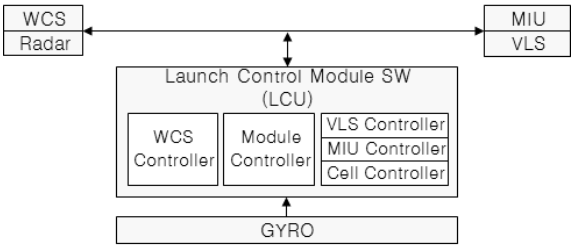

Fig. 5. LCM software configuration embedded in LCU

3. LCM 소프트웨어

3.1 LCM 소프트웨어 Class Diagram

Table 1. Function of LCM software classes

Class	Function
WCS Controller	• 무장체계의 함정 전투체계 WCS 연동 • 함정 레이더 업링크 송신기 연동 • 전투체계 WCS 및 무장체계 간 상태정보 교환 - 운용상태, 교전모드 및 Weapon Status • 전투체계 WCS의 교전명령 수신 처리 - 교전할당, 발사중지, 교전중단/종료, 비상폭파
Module Controller	• Controller Class와 연동/발사 이벤트 Data 송수신 • 연동상태, 장비상태, 운용상태, 교전모드 및 Weapon Status 정보 관리 • WCS의 교전명령에 대한 발사채널 관리, 발사절차 이벤트 수행 및 발사절차 Safety Check
VLS Controller	• 발사대 상태정보 및 구동 명령 Data의 생성 및 발사대 Proxy와 Data 송수신 처리
MIU Controller	• 탄연동기 하드웨어로 수행되는 발사절차 명령 Data 생성 및 탄연동기 Proxy와 Data 송수신 처리
Cell Controller	• MIU 경유 유도탄과 직접 송수신되는 명령 Data 생성 및 MIU Proxy와 Data 송수신 처리
Data Container	• Controller 간 교환되는 Data
무장1 Proxy	• 전투체계 무장1 WCS와 LCM 간 연동 인터페이스
무장2 Proxy	• 전투체계 무장2 WCS 및 레이더(업링크)와 LCM 간의 연동 인터페이스
MIU Proxy	• LCM과 MIU 간의 연동 인터페이스
VLS Proxy	• LCM과 VLS 간의 연동 인터페이스
GYRO Proxy	• LCM과 함정 자이로 간의 연동 인터페이스

무장1과 무장2를 혼합 운용하는 LCM의 Class Diagram은 Fig. 6과 같고, Class별 기능은 Table 1과 같다. 다수의 무장으로 확장이 가능하나, 2 개의 무장 만을 표현하였다. Data Container Class는 Controller 간에 교환되는 무장 정보이다.

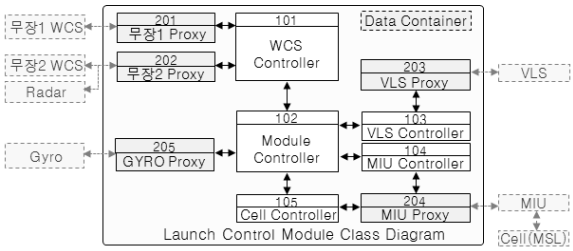

Fig. 6. Class diagram of LCM software

3.2 LCM 소프트웨어 Data Flow

LCM Object들 간의 Data Flow는 Fig. 7과 같으며, 전투체계 WCS, 레이더, 함정 자이로, 발사대 그리고 탄연동기 인터페이스를 포함한다. 유도탄(Cell) Controller의 개수는 유도탄 탄종 수만큼 생성되고, 발사대(VLS) Controller 및 탄연동기(MIU) Controller는 발사대 개수와 탄연동기 개수만큼 생성된다. 이러한 기법은 이기종 유도탄, 탄연동기 및 발사대에 대한 연동 호환성 및 유연성을 제공한다.

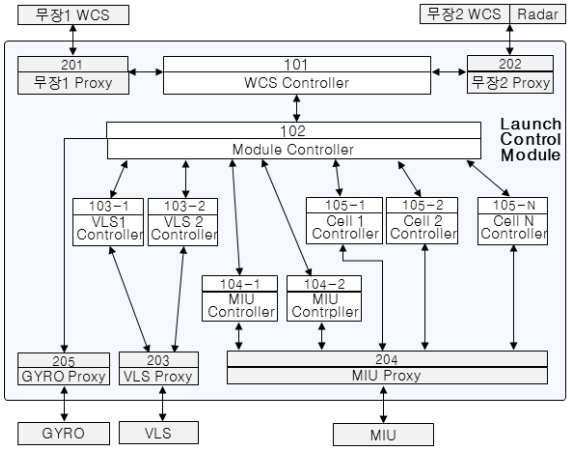

Fig. 7. Data flow of LCM objects

3.3 LCM 소프트웨어 연동 메시지 그룹

혼합 무장의 운용 및 통제 호환성을 위해서는 LCM 내부의 Controller Object 간에 교환되는 Data의 전송 절차 및 외부 연동 메시지의 생성과 송수신 절차의 호환성이 매우 중요하다. 따라서 무장체계의 연동 메시지들에 대한 기능별 그룹화, 발사절차 이벤트 Data의 생성 및 송수신, 그리고 Controller 간의 송수신 Data로부터 그룹별 연동 메시지의 생성 및 송수신은 발사제어기 아키텍처 설계에 필수적이다.

Fig. 8은 발사제어기의 LCM 소프트웨어에서 외부로 송수신되는 모든 메시지를 나타내며, 메시지들은 1x 및 1xx 등과 같이 ID 그룹군으로 표현하였다. 예로, 1xx 그룹군에는 링크연동 제어 메시지, 상태정보 송수신 메시지 및 주기성 정보 교환 메시지와 같은 장비간 연동 관련 메시지들이 포함되며, 메시지는 기능별로 11, 12, 13, .. , 14 식별자 ID 메시지로 구성한다.

메시지 ID(Identification)는 LCM의 각 Controller에서 메시지의 차별적 처리를 위한 필요하며, LCM에서 같은 ID 계열 메시지 정보 흐름은 Controller 간에 동일하며, 메시지 정보들은 Class Diagram에서 Controller 간에 송수신되고, Proxy에서 연동 메시지로 생성되어 외부 장비로 송수신된다. 발사제어기는 외부장비 연동 시 기본적으로 Point-to-Point 방식의 메시지 연동을 해야 하지만 불가피하게 다른 Controller를 통하여 외부 장비로 메시지가 전달되어야 하는 경우(아래 Fig. 8에서 발사제어기에서 탄연동기를 거쳐 발사대로 연동 메시지가 전달되어야 하는 경우)는 외부장비들 간에 메시지 송수신 처리가 되어야 한다.

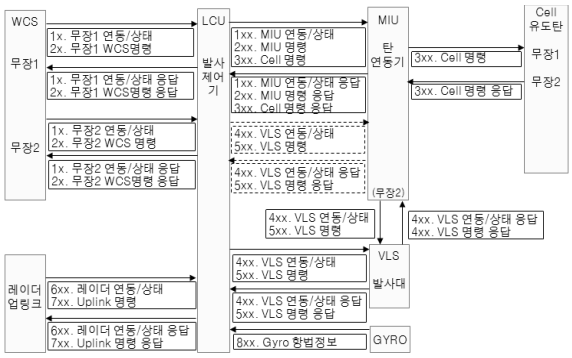

Fig. 8. External interface messages of LCM

4. LCM 소프트웨어 연동 메시지 처리

4.1 WCS 연동 메시지

전투체계 WCS와 무장 발사제어기 간의 송수신 메

시지 그룹은 Table 2와 같다. 1x 메시지 그룹은 장비 간 초기 링크 연결/해제, 상태정보, Weapon Status, 교전상태, 발사상태, 그리고 주기성 교환 메시지(예, 시간 동기)들의 요청과 응답으로 구성된다. 전투체계 무장1 WCS에서 요청된 메시지는 LCM의 무장1 Proxy(201) 및 WCS Controller(101)를 통하여 Module Controller (102)로 전달되며, Module Controller(102)는 처리 결과는 WCS Controller(101) 및 무장1 Proxy(201)를 거쳐 무장1 WCS로 전송된다. WCS와의 연동상태, 운용상태, Weapon Status, 교전상태, 발사상태 및 시간 동기화는 Module Controller에서 관리된다. 발사제어기에서 전투체계 WCS로 전송되는 상태정보에는 WCS가 무장을 통제 및 관리할 수 있는 Weapon Status 정보들, 즉 연동 탄연동기 수량, 탄연동기별 Weapon Type, 탄연동기별 탄종과 탄 인벤토리 정보가 포함되어야 한다. 시간동기는 주기적으로 WCS에 요청되어 시간 동기화를 유지하여야 한다.

교전명령 및 응답 2x 그룹은 교전할당, 발사/발사중지, 교전중단, 교전종료 및 비상폭파와 같은 명령과 응답으로 구성된다. 교전명령 및 응답 정보의 흐름 절차는 1x 그룹과 동일하며, Module Controller는 수신 교전할당에 대한 무장별 발사채널을 생성하고, 채널별 발사절차 이벤트 정보(Data)를 생성하여 무장 Controller로 전송하여 일련의 유도탄 발사절차를 수행하게 된다.

Table 2. Combat system WCS interface messages

메시지 그룹명		메시지 기능
1x	연동/상태정보/주기성 정보요청	• 발사제어기 링크 연결/해제 요청 • 전투체계 WCS 상태정보 송신 (동작상태, 운용상태, 교전모드 등) • 시간동기 요청 등
	응답	• 발사제어기 링크 연결/해제 확인 응답 • 발사제어기 상태정보(동작상태, 운용상태, 교전모드, 무장(교전)상태, 발사상태 등) 응답 • 시간동기 확인 응답 등
2x	교전명령	• 전투체계 WCS 교전명령 교전할당, 발사/발사중지, 교전종료 및 비상폭파
	응답	• 발사제어기 교전명령 확인

4.2 레이더 업링크 연동 메시지

유도탄 업링크가 필요한 무장체계는 레이더 업링크 송신기로 Table 3과 같은 업링크 메시지의 송신이 필요하다. 연동/상태정보 및 응답의 6xx 그룹 메시지는 장비 간의 연동 링크 연결/해제와 상태정보 교환 메시지로 구성되며, 송신 메시지는 Module Controller(102)에서 생성되어 WCS Controller(101)를 거쳐 레이더로 전송되며, 수신 메시지 정보는 역으로 전달된다.

업링크 명령 및 응답의 7xx 그룹 메시지는 업링크 메시지 송신과 수신 응답 메시지로 구성되며, 유도탄 발사후 이벤트 시점에 Module Controller(102)에서 생성되어 WCS Controller(101)를 거쳐 레이더로 송수신된다.

Table 3. Uplink interface messages

메시지 그룹명		메시지 기능
6xx	연동/상태정보 요청	• 레이더 링크 연결/해제 • 무장 동작상태
	응답	• 레이더 링크 연결/해제 확인 • 레이더 동작상태
7xx	업링크 명령	• 업링크 정보
	응답	• 업링크 정보 응답

4.3 탄연동기(MIU) 연동 메시지

발사제어기와 무장 탄연동기 간의 송수신되는 메시지 그룹은 Table 4와 같다. 탄연동기 연동/상태정보 및 응답의 1xx 그룹 메시지는 Module Controller(102)에서 자체 생성되어 MIU Controller(104)와 MIU proxy(204)를 거쳐 탄연동기로 전송/응답되며, 링크 연결/해제와 상태정보 등으로 구성된다. 탄연동기 상태정보에는 무장 관리를 위한 탄연동기 ID, Weapon Type, 탄연동기 연동 탄수, 그리고 유도탄 인벤토리가 포함된다.

탄연동기 명령/응답의 2xx 메시지는 탄연동기 하드웨어에 의해 제어되는 유도탄 신호들의 제어 명령 및 응답으로 구성되며, Module Controller(102)의 발사채널에서 생성된 발사 이벤트 명령/응답 Data(예, 유도탄 전원 ON)는 MIU Controller와 MIU proxy를 거쳐 탄연동기로 송수신된다. 응답은 명령의 처리(예, 전원을 ON 하였음)에 대한 확인이며, 릴레이 및 착화 동작에 대한 응답은 1xx 상태정보 응답으로 확인하여야 한다.

Cell 명령/응답의 3xx 메시지는 탄연동기에서 유도

탄으로 포워딩되는 유도탄의 처리/응답되는 메시지들이며, Module Controller(102)의 발사채널에서 생성된 유도탄 관련 이벤트 명령/응답 Data(예, 항법정보 입력)는 Cell Controller(105)와 MIU proxy(204)를 거쳐 탄연동기로 전송/수신되며, 탄연동기는 유도탄으로 메시지를 포워딩/응답한다.

Table 4. MIU interface messages

메시지 그룹명		메시지 기능
1xx	연동/상태정보 요청	• MIU 링크 연결/해제 요청 • 발사제어기 상태정보(동작상태, 운용상태 등)
	응답	• MIU 연동링크 연결/해제 응답 • MIU 상태정보(MIU ID, Weapon Type, MIU 통제 가능 탄수, 유도탄 인벤토리 및 발사대 구동상태 등)
2xx	MIU 명령	• MIU 하드웨어의 유도탄 제어 및 발사 제어 명령 - 유도탄 전원인가 - 유도탄 발사 시작
	응답	• 명령에 대한 처리 결과
3xx	Cell 명령	• MIU 경유 유도탄으로 포워딩되는 명령 - 유도탄 BIT 요청 - 표적정보 및 항법정보 장입
	응답	• 응답은 명령에 대한 유도탄 ACK 또는 처리 결과

4.4 발사대(VLS) 연동 메시지

발사제어기와 무장 발사대(VLS) 간의 연동 메시지 그룹은 Table 5와 같다. 발사대 연동/상태정보의 4xx 메시지들은 Module Controller(102)에서 자체 생성되어 VLS Controller(103)와 VLS proxy(203)를 거쳐 발사대로 전송/응답되며, 링크 연결/해제와 상태정보 메시지 등으로 구성된다. 발사대 상태정보에는 발사대 관리를 위한 발사대 ID, 발사대 Type, 장비 동작상태, 그리고 발사대 구동상태 정보를 포함한다. 발사대 명령/응답의 5xx 메시지들은 발사대 구동 및 수신 확인 메시지들이며, Module Controller에서 발사채널 이벤트 시점에 생성되어 VLS Controller와 VLS proxy를 통하여 발사대로 전송되며, 발사대는 명령을 처리하였음을 응답한다. 명령에 따른 발사대 구동상태는 발사대 상태정보로 수신된다. 만약 탄연동기를 거쳐 발사대 연동이 필요한 경우 이 메시지들은 탄연동기로 전송되어야 하며, 탄연동기 장비는 이 메시지를 발사대와 송수신 처리를 할 수 있어야 한다.

Table 5. VLS interface messages

메시지 그룹명		메시지 기능
4xx	연동/상태정보 요청	• VLS 링크 연결/해제 요청 • 발사제어기 상태정보(발사제어기 동작상태)
	응답	• VLS 링크 연결/해제 확인 응답 • VLS 상태정보(VLS ID 및 Type, 구동상태 등)
5xx	명령	• VLS 구동 명령
	응답	• VLS 구동 명령 수신 확인(구동상태 및 결과는 VLS 상태정보에 포함)

4.5 함정 자이로 연동 메시지 처리

함정 자이로 메시지는 무장체계로 주기적으로 송신만 되며, Table 6과 같다. Module Controller(102)는 GYRO proxy(205)로부터 주기적으로 수신되는 자이로 항법정보를 관리하며, 항법정보는 탄연동기 연동의 Cell 명령 메시지로 송신된다.

Table 6. Gyro interface messages

메시지 그룹명		메시지 기능
8xx	함정자이로 정보 수신	• 항법정보 및 자이로 상태정보 수신

5. LCM Controller 간 연동 정보 흐름

혼합 무장 운용을 위하여 발사제어기에서 외부 연동 메시지를 처리하고 송수신하는 LCM 내부 Controller 간의 Data(정보) 흐름은 Fig. 9와 같으며, 점선은 링크 연동과 체계장비상태, 무장상태 및 주기성 교환 정보들의 흐름을 나타내며, 실선은 교전/발사 정보들의 흐름을 나타낸다. Controller 간에는 연동 관련 Data가 송수신되며(Data Container로 교환), Controller는 연동 Data

로부터 메시지 Payload를 생성하여 외부 연동 proxy에 전달하며, 각 연동 proxy는 Payload(연동 메시지)가 포함된 이더넷 패킷을 연동 장비로 송수신한다.

LCM의 핵심 소프트웨어 모듈인 Module Controller는 무장체계의 링크 연동 관리, 연동장비 상태관리 및 무장 상태정보를 관리한다. 또한 전투체계 WCS에서 수신된 교전명령에 대한 발사채널을 생성하고, 발사 이벤트 시각에 발사제어 명령 Data를 해당 Controller로 전송하며, 발사 Safety Check와 함께 발사상태를 감시한다.

Fig. 9에서 VLS Controller와 MIU Controller 및 Cell Controller가 두 개씩 생성된 것은 무장1과 무장2의 유도탄과 발사대 및 탄연동기와의 연동 Data 흐름을 보여주는 것이다. 이 아키텍처는 본 논문에서 제안하는 핵심으로 복잡 무장의 호환 무장 운용을 가능하게 하며, 발사제어기의 설계 유연성 및 향후 확장성을 제공한다.

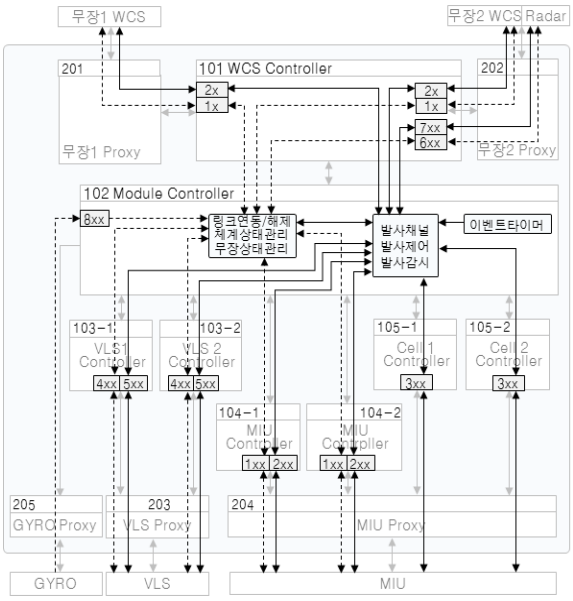

Fig. 9. Data exchange flow of LCM controllers

6. 시뮬레이션 및 고찰

Table 7은 제안된 LCM 소프트웨어의 Controller 및 Proxy 간에 연동 메시지 정보의 송수신 경로 흐름을 보여주는 시뮬레이션 결과이며, 정보의 흐름이 제안된 기법으로 처리되는 것을 보여준다. Table 7에서 무장 연동 메시지 번호에 대하여 경유하는 Controller 번호와 출력 Proxy 번호가 제안된 기법과 일치하는 것을 보여준다. 운용 초기에 발사제어기에서 탄연동기로 연동 시작을 요청하는 101번 메시지(Table 4의 1xx 메시지)는 Module Controller(102)에서 생성되며, MIU Controller (104-1)와 MIU Proxy(204)를 거쳐서 탄연동기(MIU)로 전송되는 것을 보여준다. 전투체계 WCS에서 교전을 지시하는 교전할당 21번 메시지는 WCS Controller(101)를 거쳐서 Module Controller(102)로 전달되며, 역으로 확인 응답을 전투체계 WCS로 송신한 후에 발사채널을 생성하여 일련의 유도탄 발사절차를 시작하도록 한다. 유도탄 발사절차는 유도탄 전원을 인가하고, 유도탄 BIT(Built In Test)를 확인하고, 항법정보 및 표적정보를 장입하고, 최종적으로 발사시작을 탄연동기로 전송한다. 201번 유도탄 전원인가는 Module Controller (102)와 MIU Controller(104-1, 무장1)를 통하여 출력 MIU Proxy(204)로 전송되어 탄연동기로 전송되며, 탄연동기는 해당 유도탄에 전원을 공급한 후에 응답을 역으로 전송되게 한다. 201번 메시지는 탄연동기에서 유도탄 전원을 출력하고 종료되는 메시지이다. 계속하여 LCM은 유도탄 BIT를 요청하는 303번 메시지를 탄연동기로 송신하며, 3xx번 메시지는 탄연동기에서 유도탄으로 포워딩하는 메시지이므로 유도탄으로 송신된다. 유도탄은 BIT 결과를 응답하며, 탄연동기는 303번 응답 메시지로 LCM에 응답한다. 303번 응답 메시지는 시뮬레이션에 포함하지 않았다. 이후 301번 표적정보 장입 메시지와 302번 항법정보 장입 메시지가 유도탄으로 동일한 메카니즘으로 유도탄으로 송신 및 응답되고, 최종적으로 204번 발사시작 메시지가 Module Controller(102) 및 MIU Controller(104-1)를 거쳐서 MIU Proxy(204)를 통하여 탄연동기로 전송된다. 탄 연동기는 하드웨어 유도탄 점화회로를 순차적으로 제어하여 유도탄을 발사하게 된다.

무장1, 무장2 및 무장3에 대한 혼합 연동 시의 연동 메시지들의 송수신 시뮬레이션은 하지 않았지만 무장1 연동 메시지의 시뮬레이션 결과와 동일할 것으로 판단된다. 그러므로 제안된 아키텍처는 혼합 무장 탑재 시에도 표준화 발사제어기를 공용 하나로 사용하거나 앞의 Fig. 2에 도시된 것처럼 혼합 운용이 가능함을 보여주는 것으로, 제안된 발사제어기 아키텍처는 이기종 발사대 및 유도탄의 혼합 운용이 가능하게 된다.

Table 7. Simulation results on data flow between controllers

메시지 구분	메시지 번호	전달되는 Controller 번호		입출력 Proxy 번호
		1st	2nd	
무장1 MIU 연동 요청	101	102	104-1	204
무장1 MIU 연동 요청 응답	101	104-1	102	204
무장1 MIU 상태정보 요청	102	102	104-1	204
무장1 MIU 상태정보 응답	102	104-2	102	204
무장1 VLS 연동 요청	401	102	103-1	203
무장1 VLS 연동 요청 응답	401	103-1	102	203
무장1 VLS 상태정보 요청	402	102	103-1	203
무장1 VLS 상태정보 응답	402	103-1	102	203
무장1 WCS 발사제어기 연동 요청	11	101	102	201
무장1 WCS 발사제어기 연동 요청 응답	11	102	101	201
무장1 WCS WCS 상태정보 송신	12	101	102	201
무장1 WCS 발사제어기 상태 정보 응답	12	102	101	201
무장1 WCS 발사제어기 시간 동기 요청	13	102	101	201
무장1 WCS WCS 시간동기 응답	13	101	102	201
무장1 WCS 교전할당	21	101	102	201
무장1 WCS 발사제어기 교전 할당 확인 응답	21	102	101	201
무장1 유도탄 전원인가	201	102	104-1	204
무장1 유도탄 BIT 요청	303	102	105-1	204
무장1 표적정보 장입	301	102	105-1	204
무장1 항법정보 장입	302	102	105-1	204
무장1 발사시작	204	102	104-1	204

시뮬레이션에 적용된 연동 메시지는 송수신 흐름 검증을 위한 ID 만을 적용하였고, 메시지의 기능적 내용은 포함하지 않았다. 이런 내용들을 향후 무장체계 개발 시에 구현되어야 하며, 개발 시에 추가적인 안정성 시험이 필요할 것으로 사료된다.

7. 결 론

제안된 함정 대함, 대공 및 대지 무장 발사제어기의 하드웨어 및 소프트웨어 아키텍처는 시뮬레이션을 통하여 무장 연동 메시지 정보들이 발사제어기의 LCM 내에서 잘 동작하는 것을 입증하였다. 제안된 발사제어기는 함정에 탑재되는 다수 무장 간의 호환성과 확장성을 제공하여 이기종 발사대 및 유도탄의 혼합 운용이 가능할 것으로 사료된다. 기존 무장체계는 무장별 발사제어기와 개별 할당된 발사대 및 유도탄을 사용해야 하므로 무장간의 변경이나 확장이 불가능하지만, 제안된 방법은 무장 구성 변경이나 확장성에 매우 유리하며, 하나의 발사제어기가 다수의 무장을 혼합 운용하는 것도 가능하다. 따라서 향후 함정 탑재용 무장체계에 제안된 기법을 적극적으로 적용이 필요하다고 사료되며, 표준화로 제시된 네트워크와 연동 메시지 설계 및 LCM 소프트웨어의 활용이 필수적이라고 사료된다. 향후 함정 탑재 무장체계의 개발 시에 제안된 기법을 적용한 발사제어기의 전체 소프트웨어를 개발하여 추가적인 소프트웨어의 안정성과 신뢰성에 대한 검증이 요구된다.

References

[1] Jung Young-ran, Han Woong-gie, Kim Cheol-ho, Kim Jae-ick, “On the Development of the Generic CFCS for Engineering Level Simiulation of the Surface Ship,” The Korea Institute of Military Science and Technology, Vol. 14, No. 3, pp. 380~387, 2011. 6.

[2] Shin Jin-beom, Cho Kil-seok, Yoo Myong-hwan, Kim Tae-hyeon, “A Standardized Design of Interface Design Specification for Weapon Information Exchange with Several Kinds of Command and Control System,” The Korea Institute of Military Science and Technology, Vol. 18, No. 6, pp. 771~

778, 2015. 12.

[3] NAVSWC TR 91-795, “Combat Systems Vision 2030, Combatsystem Architecture : Design Principles and Methodology,” 1991. 12.

[4] Congressional Research Service, “Navy Aegis Cruiser and Destroyer Modernization : Background and Issues for Congress,” 2010. 6.

[5] Yoo Myong-Hwan, Bae Jung-Il, Shin Jin-Hwa, Cho Kil Seok, “Development of the Engagement Control Software Architecture Based on UML 2.0 Model,” The Korea Institute of Military Science and Technology, Vol. 10, No. 4, pp. 20~29, 2007. 12.

[6] Martin Obermeier, STeven Braun, and Birgit Vogel-Heuser, “A Model-Driven Approach on Object-Oriented PLC Programming for Manufacturing Systems with Regard to Usability,” IEEE Transactions on Industrial Informatics, Vol. 11, No. 3, pp. 790~800, 2015. 6.

[7] D. C. Craig and W. M. Zuberek, “Compatibility of Software Components - Modeling and Verification,” IEEE Proceedings of the International Conference on Dependability of Computer Systems, 2006.

[8] Carlos Canal, Ernesto Pimentel, Jose M. Troya, “Compatibility and Inheritance in Software Architecture,” Science of Computer Programming, Vol. 41, pp. 105~138, 2001.

[9] W. M. Zuberek, “Checking Compatibility and Substitutability of Software Components,” Models and Methodology of System Dependability, Chapter 14, pp. 175~184, 2010.