

# 무기체계 구성장치의 연결성 제어 및 자율 재구성을 위한 플러그앤플레이 프레임워크

장혜민<sup>\*,1)</sup> · 강석종<sup>1)</sup> · 조영걸<sup>1)</sup> · 윤주홍<sup>1)</sup> · 윤지혁<sup>1)</sup>

<sup>1)</sup> 국방과학연구소 지상기술연구원

## Plug-and-Play Framework for Connectivity Control and Self-Reconfiguration of Weapon System Components

HyeMin Chang<sup>\*,1)</sup> · SukJong Kang<sup>1)</sup> · YoungGeol Cho<sup>1)</sup> · JooHong Yoon<sup>1)</sup> · Jihyeok Yun<sup>1)</sup>

<sup>1)</sup> Ground Technology Research Institute, Agency for Defense Development, Korea

(Received 14 January 2021 / Revised 5 March 2021 / Accepted 15 April 2021)

### Abstract

A study on common modular design based on open standards to reduce the life cycle cost of ground weapon system is underway. Since the ground weapon system includes major mission equipment such as fire control system, it is essential to apply the concept of fault tolerance through automatic reconfiguration and blocking unspecified equipment through connectivity control. However, it is difficult to generalize due to the difference in operating characteristics for each system. In this paper, we propose a plug-and-play framework, which includes plug-and-play architecture and mechanism. The proposed method can be used in common by the application of each component as it is divided into a common service layer. In addition, the proposed connectivity control and autonomous reconfiguration method facilitates reflection of operating characteristics for each system. We constructed a verification environment that can simulate ground weapon systems and components, and verified that the proposed framework works through scenario-based functional tests.

Key Words : Plug-and-Play(플러그앤플레이), Connectivity Control(연결성 제어), Self-Reconfiguration(자율 재구성),  
Weapon System(무기체계), Ground Weapon System(지상무기체계)

### 1. 서론

급속한 기술발전 속에서 기능의 고도화 및 시스템

복잡도 증가에 따라 상호운용성, 유지보수성, 재사용성, 가용성 등의 품질 충족을 위한 요구가 증가하고 있다. 기존 무기체계 장치는 특정 시스템 목적에 따라 최적화 설계됨에 따라 다른 시스템에서 유사 목적으로 활용하게 되더라도, 설계 변경과 수정에 따른 막대한 비용이 발생하는 문제가 있다. 플러그앤플레이 기

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

술<sup>[1]</sup>은 네트워크 상에 연결한 장치 및 제어기 간의 상호 운용을 위한 필수 기술 중 하나로서, 운용자 개입이나 설정 없이, 연결하면 자동으로 동작하도록 하기 위한 공용 아키텍처 및 동작방법이다. 향후 개발되는 무기체계 장비에 네트워크 기반의 플러그애플레이 기술 적용시 개발 단계, 운용 및 유지보수 단계를 포함한 전생애주기 비용 절감 뿐 아니라, 임무 맞춤형 무기체계 개발을 용이하게 할 수 있는 장점이 있다.

지상무기체계 분야에서는 개방형 아키텍처 설계<sup>[11,12]</sup> 및 미들웨어 기반의 공통 데이터 운용을 통하여 플러그애플레이 개념을 적용하고 있다<sup>[4,5]</sup>. 이를 통해 지상 무기체계의 각 구성장치는 서로 다른 시스템에서 미들웨어 기반의 자동 발견 방법<sup>[2,3]</sup>과 시스템 공통화 된 데이터 규격에 따라 상호운용이 가능하다. 하지만, 무기체계별 임무 특성을 반영한 맞춤형 설계를 위해서는 무기체계별 서로 다른 장치 구성 및 재구성이 가능해야 한다. 특히, 무기체계는 사격통제장치 등 주요 임무장치를 포함하고 있어 불특정 장비 또는 서비스에 대한 차단과 같은 상위수준의 접근통제가 가능해야 하며, 주요 장치의 고장시 대체운용 규칙에 따라 임무를 지속 가능해야 한다. 이러한 무기체계별 서로 다른 운용 특성 차이는 공용화 설계 및 재사용성 향상을 어렵게 하는 주요 요인이다.

본 논문은 서로 다른 시스템에서 서비스 수준의 연결성 제어 및 자율 재구성이 가능한 플러그애플레이 프레임워크를 제안한다. 플러그애플레이를 위한 주요 모듈들을 발견/구독 방식의 표준 미들웨어 기반의 공통 서비스 계층 상에서 동작하도록 설계함에 따라, 기존의 무기체계 아키텍처를 재사용성 측면에서 개선하였다. 또한, 서로 다른 운용 특성을 갖는 각 무기체계 구성장치에 대하여 공용화 설계를 따르면서도 시스템별 운용 특성차이를 손쉽게 반영 가능하도록, 시스템 명세서 기반의 서비스 연결성 제어 방법과 대체운용 규칙에 기반한 자율재구성 방법을 제안하였다. 제안 내용을 무기체계에 적용시 아키텍처 개선에 따라 재사용성이 향상되고, 플러그애플레이에 의한 각 구성장치의 상호운용성, 자율 재구성에 의한 가용성과 유지보수성이 향상된다.

본 논문의 구성은 다음과 같다. 2장에서는 플러그애플레이 및 공용화 아키텍처 설계 분야의 관련 연구와 차별성을 기술한다. 3장에서는 제안하는 플러그애플레이 프레임워크를 제안한다. 4장에서는 제안하는 방법을 검증하고, 5장에서는 결론을 기술한다.

## 2. 관련 연구 및 차별성

본 장에서는 플러그애플레이 관련 연구 및 차별성에 대하여 기술한다. 2.1절에서는 대표적인 플러그애플레이 기술 관련 표준에 대한 무기체계 적용 가능성을 검토하고, 2.2절에서는 지상무기체계 분야의 기존 유사 연구에 대한 본 연구와의 차별성을 기술한다.

### 2.1 관련 표준

플러그애플레이 기술은 마이크로소프트 사의 Plug and Play(PnP)와 Universal Plug and Play(UPnP)<sup>[1]</sup>가 가장 대표적이며, 장치의 탈부착시 물리적인 장치 설정 또는 사람의 개입 없이 자동적으로 발견하고 동작하도록 정의된 표준 아키텍처 및 규격이다. UPnP는 네트워크 상의 IP 기반의 장치를 연결할 경우 주소화, 자동 발견, 서비스 기술 단계를 거쳐 장치 제어, 이벤트링, 프리젠테이션이 가능한 6단계의 표준화된 절차를 제공한다. UPnP를 사용하는 모든 장치는 UPnP 규격이 구현된 모듈을 내장하고 있어야 하며, IP 및 http를 이용한 웹 기반의 프로토콜을 사용한다. UPnP의 광범위한 활용성에 불구하고 발견/구독 방식의 미들웨어를 적용하는 해외 국방 시스템 발전 추세를 고려할 때, UPnP의 직접적 적용은 적합하지 않다.

IoT(Internet of Things) 기술 발전에 따라 단순한 제어기-장치 간의 1:1 원격제어 기술을 넘어 다양한 기기, 시스템, 서비스 간의 연결성(connectivity)을 제공하는 것을 목표로 다양한 공용 플랫폼 기술을 개발하고 있으며, 표준화가 진행 중에 있다<sup>[6-8]</sup>. 단순한 네트워크 연결, 장치 간 연결에서 나아가 서비스 단위의 고유 ID를 기반으로 해당 서비스에 대한 접근성을 다수의 통신 기술을 기반으로 정의함으로써, 기기 간의 호환성 및 접근성을 확보할 수 있도록 구성되는 것이 특징이나, 무기체계 개방형 구조와 부합되는 표준이 제공되지 않고 있다. OCF<sup>[7]</sup>는 IP 기반의 다양한 저전력 통신 프로토콜(Zigbee, Bluetooth 등)과의 연동 기술 제공을 주요 목적으로 하고 있다. oneM2M<sup>[8]</sup>은 스마트 홈, 스마트카 등의 도메인 간의 상호연동을 위한 공통 플랫폼 제공을 목표로 하고 있으나, 웹 서비스 기반의 응용소프트웨어 개발을 주 목적으로 하고 있어, 발견/구독 방식의 표준 미들웨어 기반의 무기체계 환경에 직접 적용하기에 어려움이 있다.

### 2.2 지상무기체계 분야 관련 연구

선진국에서는 효율적인 장비운용, 향후 장비 업데이트 및 부품단종 대비 등이 가능하도록 네트워크기반 개방형 아키텍처 연구를 수행하고 있다. 미국은 2012년 이후 개발 무기체계에 대하여 사격통제, 지휘통제 및 차량플랫폼을 포함한 VICTORY<sup>[11]</sup> 규격을 적용하도록 하고 있다. NATO는 NGVA(STANG 4754)<sup>[12]</sup>를 군용 차량 아키텍처 기준으로 제공하고 있으며 DDS 기반의 모듈화 및 플러그인 개념을 적용하고 있다. 국내에서는 지상무기체계 분야의 아키텍처 관련 연구로 지상전투차량을 위한 개방형 표준 기반의 공용화 설계 연구가 수행되었다<sup>[5]</sup>. 표준 미들웨어를 기반으로 각 어플리케이션 간의 데이터 인터페이스와 동작 절차를 공용화 설계하고 있다는 점에서 유사한 구조를 갖는다.

이러한 미들웨어 기반의 공통 데이터 운용을 통하여 하나의 장치를 유사한 다른 시스템에서 공통적으로 사용 가능함에 따라 상호운용성이 향상된다. 하지만, 기존<sup>[5]</sup>의 시스템 아키텍처는 장치 구성 및 재구성 방법 등 서로 다른 무기체계 운용 특성을 공용화 설계에 반영하고 있지 못하다. 장치 구성과 관련하여, 시스템 구성장비 중 특정 시스템 전용 장치는 해당 시스템에서만 연결되어야 한다. 특히, 사격통제장치 등을 포함하고 있는 무기체계와 같이 상위수준의 접근통제가 요구되는 시스템에서는 의도치 않은 임의의 장치 연결로 인하여 치명적 사고를 유발할 수 있다. 또한, 재구성 방법과 관련하여, 시스템 운용 전 또는 운용 중 예기치 못한 고장 발생시 상태정보를 기반으로 대체 운용 등을 통하여 시스템 운용을 지속할 수 있어야 한다.

본 논문은 이러한 시스템별 임무 특성을 반영한 맞춤형 설계를 위하여 시스템 명세서 기반의 연결제어와 대체 운용 규칙 기반의 자율재구성 방법을 포함하는 서비스 수준에서의 플러그엔플레이 프레임워크를 제안하였다. 기존 유사 연구<sup>[9]</sup>에서는 시스템 유형에 따른 서비스 인가 절차를 통한 서비스 연결제어 방법을 제안하였다. 본 논문은 [9]의 후속 연구로서, 공용화 설계 관점에서 플러그엔플레이 주요 모듈을 공통 서비스 계층 상에서 동작하도록 아키텍처를 개선하였고, 시스템별 운용 특성을 일반화된 방식으로 반영하도록 연결 제어 방법을 개선하였다. 다른 유사 연구<sup>[10]</sup>에서는 지상무기체계를 위한 고장배제 설계 방법을 제안하였다. 본 논문은 플러그엔플레이의 기존 개념을

확장하는 관점에서 고장배제 절차를 플러그엔플레이 프레임워크로 통합하고, 무기체계별 대체운용 규칙을 기반으로 자율재구성 가능한 방법을 제안하였다. 또한, 무기체계 모의환경을 구성하여 제안한 방법을 검증하였다.

### 3. 제안하는 플러그엔플레이 프레임워크

본 장에서는 무기체계 구성장치의 연결 제어 및 자율 재구성을 위한 플러그엔플레이 프레임워크에 대하여 기술한다. 제안하는 시스템의 운용 개념도는 Fig. 1과 같다. 모든 장비는 최초 연결시 각 장비의 네트워크 연결상태 및 운용 하드웨어 및 소프트웨어 상태 정보를 제공하며, 제어기는 신규 장비 연결시 연결 상태 정보를 수신하여, 시스템 전체 상태를 확인 및 배포해야 한다. 또한, 시스템 유형에 따라 연결 인가 또는 비인가 처리 할 수 있어야 하며, 운용 중 연결 해제 또는 추가 장착시 상황에 따라 자율 재구성 가능하여야 한다.

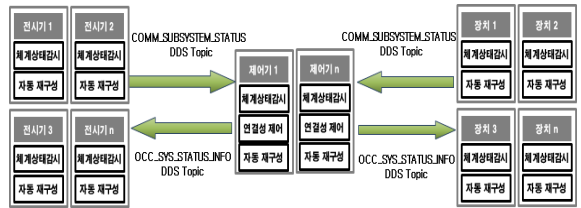


Fig. 1. Plug-and-play conceptual diagram

3.1절에서는 공용화 설계 관점의 시스템 아키텍처 설계 및 플러그엔플레이를 위한 주요 모듈 구성을 기술하고, 3.2절에서는 플러그엔플레이 동작 방법, 3.3절에서는 시스템 특성을 반영한 연결제어 및 자율 재구성 방법을 각각 기술한다.

#### 3.1 시스템 아키텍처

제안하는 플러그엔플레이(PnP)를 위한 시스템 아키텍처는 Fig. 2와 같다. 공용화 설계 관점에서 플러그엔플레이 프레임워크를 미들웨어와 응용 계층 사이의 공통 서비스 계층에 위치하도록 구분하였다. 플러그엔플레이 프레임워크는 ‘서비스 연결성 제어’, ‘자율 재구성’, ‘체계/서비스 상태감시’의 3가지 모듈로 구성되며, 공통 서비스 계층에서 동작한다. 이러한 공통 아

키텍처를 적용하여 개발된 무기체계 각 장비는 서로 다른 시스템에서 손쉽게 재사용 가능하다는 이점이 있다. 또한 제안한 플러그앤플레이 프레임워크는 DDS 미들웨어 상에서 동작하도록 설계함에 따라서 DDS 표준을 적용하는 국내외 무기체계에 적용 가능하다.

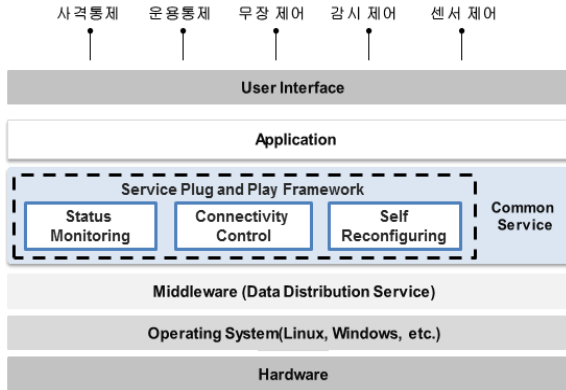


Fig. 2. Architecture diagram

모든 연결 장비는 최초 구동시 자동적으로 발견 및 연결이 수행되어야 하며, 운용 중에도 지속적 모니터링을 통하여 연결 상태를 업데이트 하고, 필요할 경우 자율 재구성을 통하여 운용을 유지할 수 있어야 한다. 이를 위해 플러그앤플레이 주요 모듈들은 주기적으로 실행되도록 설계되어야 한다. Fig. 3은 플러그앤플레이 주요 모듈의 주기적 실행 개념과 개략적인 실행 흐름을 나타낸다. 시스템 시작 및 전처리가 완료되면 연결 상태 모니터링, 연결성 제어 및 자율 재구성 순으로 플러그앤플레이 프레임워크를 실행하며, 운용 중 발생하는 변경 상황에 따른 업데이트를 위해 주기적으로 동작한다. 주기 설정에 있어서, 플러그앤플레이의 실행주기를 짧게 설정할 경우, 연결상태 및 장치의 상태변화에 보다 신속한 모니터링 및 대처가 가능하지만, 시스템 및 장치의 주요 임무 처리(비주기 또는 10~200 Hz 이상) 성능에 영향을 줄 수 있다. 본 연구에서는 각 장치별 고유 임무처리 모듈에 실행 시간을 우선적으로 분배하기 위하여 플러그앤플레이를 위해서는 1 Hz 주기로 실행하도록 설계하였다.

제안하는 PnP 아키텍처의 구성요소는 전시기, 제어기, 장치의 3가지 그룹으로 구분하였다. UPnP의 경우 제어기와 장치로 구분하고 있으나, 지상무기체계 구성품 특성을 고려하여 제어기를 전시기와 제어기로 세분화하여 구분하였다. 전시기는 운용자 인터페이스를

통한 제어명령 송신 및 전시 기능을 하며, 제어기는 시스템 통합 관리 및 장치 제어 기능을 수행한다. 장치는 무장, 감시, 센서 등 실제 시스템 본연의 임무수행을 위해 요구되는 제어의 대상으로 제어명령에 따라 상태를 변경하거나, 현재 상태를 통합제어기 및 전시기에 제공한다.

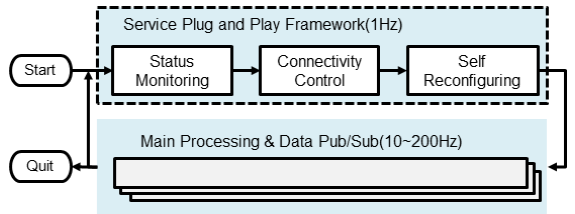


Fig. 3. Execution concept of the main PnP modules

### 3.2 플러그앤플레이 동작 방법

제안하는 플러그앤플레이를 위한 전시기, 제어기, 장치로 구성되는 각 장비 유형별 동작 방법은 Fig. 4와 같다.

#### Step 0: 초기 시작 조건 및 전처리

플러그앤플레이의 시작 이벤트는 하드웨어 전원을 켜고 동시에 발생된다. 단, 전시기의 경우 운용자 로그인에 의하여 시작 이벤트가 발생되며, 운용자 로그인 단계에서 입력 받은 운용자 역할, 시스템 유형 정보는 시스템 연결성 제어 단계에서 활용된다. 시작 이벤트 발생 후 발간-구독 미들웨어는 자동으로 디스커버리를 수행하여 이더넷 네트워크 상의 연결을 수행한다. 디스커버리가 완료되면, DDS 토픽 메시지를 교환을 통하여 플러그앤플레이를 위한 각 단계 1~3을 수행한다.

#### Step 1: 시스템 상태 감시

모든 구성장비(전시기, 제어기, 장치)는 자체진단을 통한 상태보고를 수행하고, 연결 상태와 운용 상태를 제어기로 보고한다. 제어기(운용통제컴퓨터)는 각 구성장비의 상태정보를 수신하고 종합하여 시스템 상태를 판단한다.

#### Step 2: 서비스 연결 제어

제어기는 시스템 동작을 위해 필요한 각 장치 및 각 장치가 제공하는 서비스가 연결될 수 있도록 인가 또는 비인가 처리한다. 이를 위해 장치/서비스 구성정

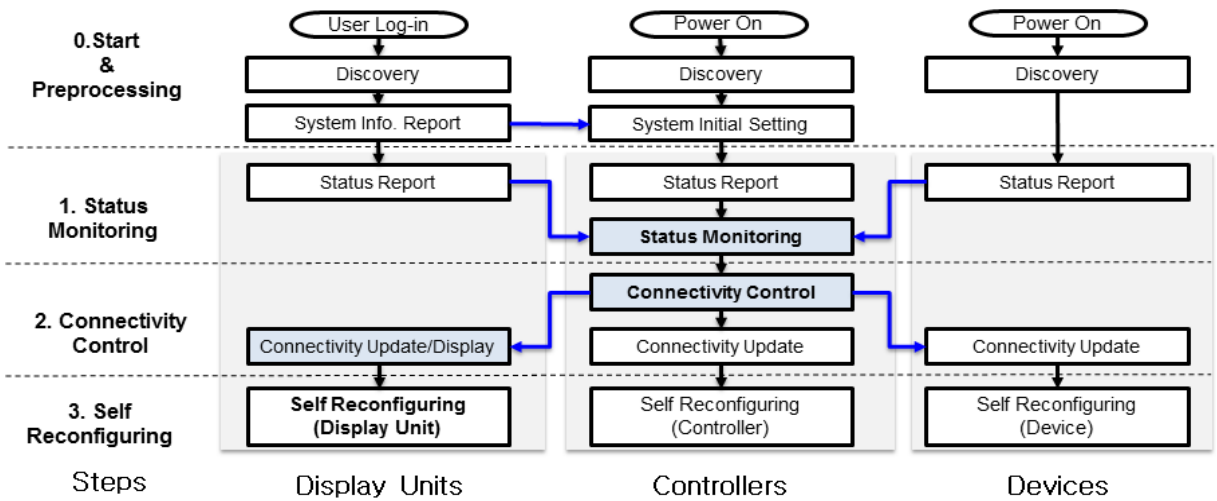


Fig. 4. Service plug-and-play operation flow

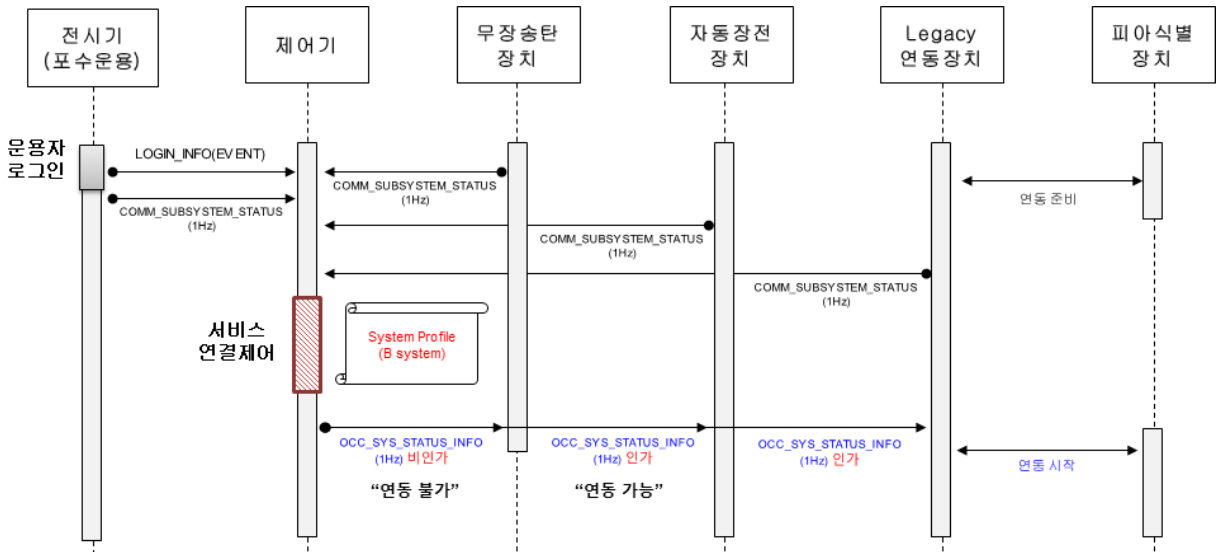


Fig. 5. Sequence diagram of connectivity control for system configuration

보 등을 포함하는 시스템 명세서(3.3.1절)와 전시기로부터 수신된 운용자 역할 정보(전차장, 포수, 운전병 등), 시스템 유형, 그리고 각 장치로부터 수신된 구성 장비/시스템의 현재 상태정보를 활용한다. 제어기는 수신한 각 장치 상태정보와 시스템 명세서 상의 구성이 일치할 경우 인가 메시지를, 불일치하는 경우 비인가 메시지를 보낸다<sup>9)</sup>. Fig. 5는 연결 제어 시나리오에 대한 시퀀스 다이어그램으로, 장치가 운용자, 유지보

수 등의 이유로 잘못 연결된 상황을 보여준다. 다소 일반적인 상황은 아니지만, 네트워크 상에 서로 다른 시스템의 모든 장비가 물리적으로 연결된 상황에서 서비스 수준의 연결 인가 및 비인가 처리 결과를 확인 가능하도록 설계하였다. 인가된 각 장치는 동작을 개시하고, 전시기는 운용자 로그인시 설정된 역할 (Table 1)에 따라 장치 서비스 접근을 위한 전시 UI 및 제어 UI 기능을 활성화 처리한다.





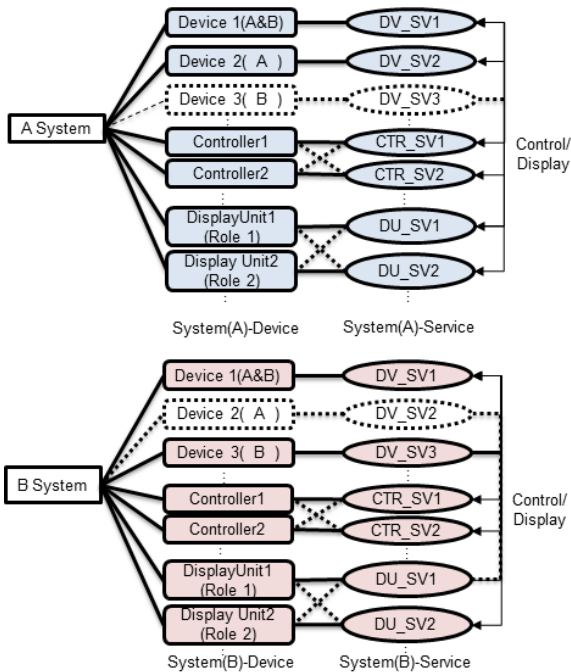


Fig. 7. Conceptual diagram of the weapon system components(system, device and service)

```
<?xml version="1.0"?>
- <System_Profile>
  <name> a_system </name>
  + <system_configuration>
  - <hw_list>
    <item name="device_1" hw_ref_id="101"/>
    <sub_item name="sub_device_1"/>
    <sub_item name="sub_device_2"/>
    <sub_item name="sub_device_3"/>
    <sub_item name="sub_device_4"/>
    <sub_item name="sub_device_5"/>
    <item name="device_2" hw_ref_id="102"/>
    <item name="device_3" hw_ref_id="103"/>
    ...
  </hw_list>
  - <svc_list>
    <item name="service_1" svc_ref_id="10001"/>
    <item name="service_2" svc_ref_id="10002"/>
    <item name="service_3" svc_ref_id="10003"/>
    ...
  </svc_list>
  </system_configuration>
  + <fault_tolerance_rules>
  + <hw_dict>
  + <svc_dict>
</System_Profile>
```

Fig. 8. An example of system profile

수 있다. 시스템 명세서에 구성장치 정보 이외에 추가적으로 각 장치/서비스 세부정보를 정의함으로써 서비스를 인가 또는 비인가 처리할 수 있도록 설계하였다.

```
<?xml version="1.0"?>
- <System_Profile>
  <name> a_system </name>
  + <system_configuration>
  + <fault_tolerance_rules>
  - <hw_dict>
    - <hw>
      <id> 101 </id>
      <type category="display_unit"/>
      <version> 1.1 </version>
      <manufacturer> company_a </manufacturer>
      <Severity> High </Severity>
    </hw>
    ...
  </hw_dict>
  - <svc_dict>
    - <svc>
      <id> 10001 </id>
      <type category="display"/>
      <version> 1.01 </version>
    </svc>
    + <svc>
    ...
  </svc_dict>
</System_Profile>
```

Fig. 9. An example of defining detailed information of hardware devices and services

Fig. 9와 같이 장치 명세서의 세부적인 장비 모델명과 제조사 정보를 포함하도록 명세서를 정의함으로써, 불특정 제조사, 모델일 경우 또는 버전정보가 맞지 않는 경우 비인가 처리할 수 있다.

### 3.3.2 대체운용 규칙 기반의 자율재구성 방법

자율 재구성 방법에 있어서 기본적으로는 기존 연구<sup>[10]</sup>의 이중화를 통한 대체운용 개념을 적용하되, 복수대의 전시기 간의 다중화 대체운용이 가능하도록 재구성 개념을 확장하고, 자율 재구성 규칙 기반으로 처리 가능하도록 일반화하였다. 또한, 전시기 GUI의 역할별 서비스 할당에 따른 고장발생시 다중 전시기 간의 서비스 대체운용 시나리오를 제시하고, 실제 동작 확인을 통하여 검증하였다.

전시기용 GUI는 공용화 설계됨에 따라 동일한 GUI 프로그램을 사용하며, 운용자 역할별 GUI 기능(서비스)의 활성화/비활성화 처리를 통하여 장치 제어/운용 권한을 부여하고, 장치 운용 중 상태변화에 따라 자율 재구성을 통한 대체운용을 수행한다. Table 1은 전시기의 역할별 전시기 서비스 할당 예시이다. 본 예시에서 전시기는 5개의 역할(R1~R5)과 4개의 전시기 서비스(DU\_SV1~4)로 구성된다. 각 전시기 역할별로 전시기 서비스가 할당되어 운용자 로그인시 GUI 상에 해당 서비스가 활성화 또는 비활성화 처리된다.

Table 1. Service allocation for each role of multiple display units

Roles   Services	DU_SV1 (WPN)	DU_SV2 (SA)	DU_SV3 (DEF)	DU_SV4 (SYS)
Role1(전차장표적)	O			
Role2(전차장운용)		O	O	O
Role3(포수표적)	O			
Role4(포수운용)		O	O	
Role5(보병용)	O	O		

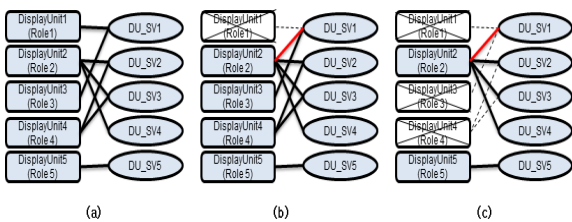


Fig. 10. Alternative operation of display equipment (a) normal operation (b) 1 display unit failure (c) 3 display units failure

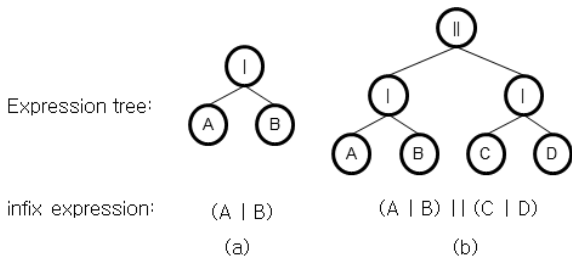


Fig. 11. Rule expressions of alternative operation (a) for controllers (b) for display units

Fig. 10은 Fig. 6의 시나리오에 따라 전시기 1대 고장과 3대 고장 발생시 대체운용 결과를 보여준다. Fig. 10(a)는 Table 1과 같은 전시기 역할-서비스 할당에 따라 정상운용 상태를 보여주며, Fig. 10(b)는 전시기 1대 고장시 DisplayUnit2가 DisplayUnit1의 역할인 DU\_SV1을 대체 운용함을 보여준다. Fig. 10(c)는 전시기 3대 고장시 DisplayUnit2가 DisplayUnit 1,3,4의 역할을 모두 대체 운용하게 됨을 보여준다. 제안하는 대체 운용 방법을 통하여 전시기의 모든 서비스가 고장시 항상 가용 상태를 유지함을 보여준다.

```

1 // Return Pair, parentPair
2 Leafnode Leafnode.pair(Leafnode self);
3 Branchnode Leafnode.parentPair(Leafnode self);
4 // Return true(alive) or false
5 Bool Leafnode.isAlive();
6 // Return true(at least one child is alive) or false
7 Bool Branchnode.isAlive() {
8     if (node.leftchild.isAlive() && (node.rightchild.isAlive()))
9         return true;
10    else return false;
11 }
12 BackupPair(ETree rules, Leafnode self) {
13     if self.pair.isAlive() is False
14         doBackup(self, self.pair);
15 }
16 BackupParentPair(ETree rules, Leafnode self) {
17     if self.parentPair.isAlive() is False {
18         if self.pair.isAlive() is False {
19             doBackup(self, self.parentPair.leftchild);
20             doBackup(self, self.parentPair.rightchild);
21         }
22     }
23     else if (self.parentPair.leftchild == self)
24         doBackup(self, self.parentPair.leftchild);
25     else if (self.parentPair.rightchild == self)
26         doBackup(self, self.parentPair.rightchild);
27 }
28 //Self Reconfigure
29 Bool SelfReconfigure(ETree rules, Leafnode self) {
30     BackupPair(rules, self);
31     if (self.parent != root)
32         BackupParentpair(rules, self);
33     return true;
34 }
    
```

Fig. 12. Autonomous reconfiguration method based on alternative operation rules

본 연구는 자율 재구성을 위한 대체 운용 규칙을 일반화하여 표현하기 위한 다양한 구현 방법 중 하나의 예시로서, expression tree 기반의 대체운용 규칙을 활용하는 방법을 제안한다. 제안하는 방법은 시스템 특성에 따라 각 구성장치 그룹(제어기, 장치, 전시기) 별로 쉽게 정의 가능하며 변경시 주요 로직에 대한 영향성이 최소화됨에 따라 재사용률이 향상되는 장점이 있다. Fig. 11은 구성장치 그룹별 대체운용 규칙을 부호 ‘|’와 ‘||’로 정의하고, 문자열(infix expression)과 expression tree 형태로 표현한 예시이다. 여기서 L1|L2는 Leaf node인 L1과 L2가 동일 parent를 갖는 sibling 관계임을 의미하고, 이를 pair 관계로 정의하였다. B1||B2는 Branch node인 B1과 B2가 sibling 관계임을 의미하고, 이를 ParentPair 관계로 정의하였다.

Fig. 12는 Fig. 10(b)의 expression tree 형태의 전시기 대체운용 규칙을 활용하여 현재 노드(예:B)로부터 Pair와 Parentspair에 대한 상태정보 기반의 대체운용 처리 방법이다. 먼저, 각 전시기 노드인 A~D에 관한 대체운용 규칙 (A|B)|(C|D)을 읽어들이어 ExpressionTree형태로 구성한다. 구성된 대체운용 규칙 ETree rules을 기반으로 BackupPair()와 BackupParentpair() 함수를 통하여 현재노드로부터 pair와 ParentPair에 대한 상태정보 기반



의 대체운용 처리를 수행한다(Line 28-34). isAlive() 함수는 취합된 시스템 상태정보를 참조하여 현재노드의 가용상태를 확인 후 반환한다(Line 5-10). doBackup() 함수는 첫 번째 인자의 노드가 두 번째 인자의 노드 역할을 대체운용하도록 처리한다. 여기서, 각 전시기 노드 A~D의 역할은 Table 1의 Role 1~4과 대응되며, 할당된 전시기 서비스인 DU\_SV1~4와 매핑된다. 따라서, doBackup() 함수는 전시기 pair 노드가 비가용시 해당 노드의 서비스를 활성화하게 된다.

#### 4. 실험 및 검증

##### 4.1 검증 방법 및 환경 구성

제안한 방법을 검증하기 위한 검증 방안은 Table 2와 같으며, 이를 위해 개방형 구조 설계 개념에 따라 Table 3과 같이 상용 컴퓨터 하드웨어와 범용 OS, DDS 미들웨어를 적용한 실험 환경을 구성하였다.

실험 환경 세부 사양은 아래와 같다. 분산된 각 노드 간의 로깅 시간 측정을 위하여 네트워크 기반 PTP (Precision Time Protocol) 방식의 시간 동기화를 적용하였다.

- 네트워크 : 이더넷 1G 스위치
- 운영체제 : Linux RT Patch
- 미들웨어 : 상용 DDS 미들웨어(SmartDDS<sup>[3]</sup>)
- 시간동기화 : PTP(Precision Time Protocol)

Table 2. Verification method of main PnP modules

Items	Verification methods
Status Monitoring	<ul style="list-style-type: none"> <li>• Check logging messages</li> <li>- Status report(1Hz)</li> <li>- Distribution of system status(1Hz)</li> </ul>
Connectivity Control	<ul style="list-style-type: none"> <li>• Operating scenario of Fig. 5</li> <li>• Checking connectivity control results through the GUI</li> <li>- each common device</li> <li>- non-common devices(ALU, GFC) for specific systems</li> </ul>
Self-Reconfiguration	<ul style="list-style-type: none"> <li>• Operating scenario of Fig. 6</li> <li>• Checking alternative operating results through the GUI</li> <li>- among 4 display units</li> <li>- between 2 controllers</li> <li>- between 2 devices</li> </ul>

Table 3. HW and SW configuration for verification

HW	SW
Controllers(2EA)	• System status management
Display units (5EA)	• Operating common GUI and role-based(Role 1~5) activation of each display services
Device Simulators(8EA)	• Generating simulated data for each component device(8 types)
Analysis unit (1EA)	<ul style="list-style-type: none"> <li>• Message logging</li> <li>• Operation of failure scenarios</li> </ul>

Seq#	SendTime	RecvTime	Length	MsgID	SVID	TOPIC_NAME
795	1607354454.45	1607354454.45	8	6	7	COMM_SUBSYSTEM_STATUS_CPS
857	1607354454.46	1607354454.46	8	6	2	COMM_SUBSYSTEM_STATUS_GTDS
124	1607354454.50	1607354454.50	8	6	3	COMM_SUBSYSTEM_STATUS_ALU
386	1607354454.55	1607354454.55	152	6	9	COMM_SUBSYSTEM_STATUS_COD
124	1607354454.73	1607354454.73	8	6	4	COMM_SUBSYSTEM_STATUS_GFC
858	1607354454.78	1607354454.78	8	6	5	COMM_SUBSYSTEM_STATUS_DCC
128	1607354454.85	1607354454.85	8	6	8	COMM_SUBSYSTEM_STATUS_GPS
858	1607354454.86	1607354454.86	152	6	10	COMM_SUBSYSTEM_STATUS_CTD
884	1607354454.89	1607354454.90	152	6	11	COMM_SUBSYSTEM_STATUS_GOD
827	1607354454.94	1607354454.94	152	6	13	COMM_SUBSYSTEM_STATUS_MSD
855	1607354454.97	1607354454.97	8	7	0	OCC_SYS_STATUS_INFO
858	1607354455.12	1607354455.12	8	6	16	COMM_SUBSYSTEM_STATUS_APS
596	1607354455.15	1607354455.15	8	6	6	COMM_SUBSYSTEM_STATUS_INS
818	1607354455.32	1607354455.32	8	6	0	COMM_SUBSYSTEM_STATUS_FCC
837	1607354455.37	1607354455.37	152	6	12	COMM_SUBSYSTEM_STATUS_GTD
796	1607354455.45	1607354455.45	8	6	7	COMM_SUBSYSTEM_STATUS_CPS
858	1607354455.46	1607354455.46	8	6	2	COMM_SUBSYSTEM_STATUS_GTDS
125	1607354455.50	1607354455.50	8	6	3	COMM_SUBSYSTEM_STATUS_ALU
387	1607354455.55	1607354455.55	152	6	9	COMM_SUBSYSTEM_STATUS_COD
859	1607354455.78	1607354455.78	8	6	5	COMM_SUBSYSTEM_STATUS_DCC
129	1607354455.85	1607354455.85	8	6	8	COMM_SUBSYSTEM_STATUS_GPS
859	1607354455.87	1607354455.87	152	6	10	COMM_SUBSYSTEM_STATUS_CTD
885	1607354455.89	1607354455.90	152	6	11	COMM_SUBSYSTEM_STATUS_GOD
828	1607354455.94	1607354455.94	152	6	13	COMM_SUBSYSTEM_STATUS_MSD
856	1607354455.97	1607354455.97	8	7	0	OCC_SYS_STATUS_INFO
859	1607354456.12	1607354456.12	8	6	16	COMM_SUBSYSTEM_STATUS_APS
597	1607354456.15	1607354456.15	8	6	6	COMM_SUBSYSTEM_STATUS_INS
819	1607354456.32	1607354456.32	8	6	0	COMM_SUBSYSTEM_STATUS_FCC
838	1607354456.37	1607354456.37	152	6	12	COMM_SUBSYSTEM_STATUS_GTD
797	1607354456.45	1607354456.45	8	6	7	COMM_SUBSYSTEM_STATUS_CPS
859	1607354456.46	1607354456.46	8	6	2	COMM_SUBSYSTEM_STATUS_GTDS
126	1607354456.50	1607354456.50	8	6	3	COMM_SUBSYSTEM_STATUS_ALU
388	1607354456.55	1607354456.55	152	6	9	COMM_SUBSYSTEM_STATUS_COD
860	1607354456.78	1607354456.78	8	6	5	COMM_SUBSYSTEM_STATUS_DCC
130	1607354456.85	1607354456.85	8	6	8	COMM_SUBSYSTEM_STATUS_GPS
860	1607354456.86	1607354456.87	152	6	10	COMM_SUBSYSTEM_STATUS_CTD
886	1607354456.89	1607354456.90	152	6	11	COMM_SUBSYSTEM_STATUS_GOD
829	1607354456.94	1607354456.94	152	6	13	COMM_SUBSYSTEM_STATUS_MSD
857	1607354456.97	1607354456.97	8	7	0	OCC_SYS_STATUS_INFO
860	1607354457.12	1607354457.12	8	6	16	COMM_SUBSYSTEM_STATUS_APS
598	1607354457.15	1607354457.15	8	6	6	COMM_SUBSYSTEM_STATUS_INS
820	1607354457.32	1607354457.32	8	6	0	COMM_SUBSYSTEM_STATUS_FCC
839	1607354457.37	1607354457.37	152	6	12	COMM_SUBSYSTEM_STATUS_GTD

Fig. 13. Message logging of status report and connectivity control

##### 4.2 로깅 메시지 확인

각 장치별 공통 운용 데이터 검증(120분 소요) 시 분석장치의 송수신 메시지 로깅을 통하여 정상 운용됨을 확인하였다. Fig. 13은 Fig. 5의 시나리오 수행 중 상태

보고 메시지의 정상 동작을 확인하기 위하여 송수신 메시지를 로깅한 결과이다. 제어기 2대, 전시기 5대, 장치모의기 8대로 구성되는 환경에서 각 장치는 상태 보고 토픽을, 제어기는 통합 시스템 상태 정보 토픽을 주기적(1 Hz)으로 발간하고 있음을 확인하였다. 또한 각 메시지의 수신시간(RecvTime) 기록을 통하여 상태 보고 및 시스템 상태보고 메시지가 1초 주기로 정상 수신되고 있음을 확인 가능하였다. A 시스템 연결 시나리오에서 자동장전장치(ALU,녹색) 및 전체 장비는 연결 인가됨에 따라 메시지를 계속 송신하고, 무장송탄장치(GFC, 적색)는 연결 비인가 처리되어 더 이상 상태보고 메시지를 송신하지 않음을 확인 가능하였다.

### 4.3 시나리오 기반 기능 검증

#### 4.3.1 서로 다른 시스템 플러그애플레이 기능 확인

Fig. 5의 시나리오에 따라 A시스템과 B시스템에서의 연결 인가 및 비인가 처리가 정상적으로 이루어지는지 전시기 GUI를 통하여 확인하였다(Fig. 14, 15). A시스템의 경우 무장송탄장치가 비인가됨에 따라 사각박스 안에 흰색 배경으로 표시됨을 확인하였다(Fig. 14). B 시스템에서는 자동장전장치가 비인가됨에 따라 사각박스 안에 흰색 배경으로 표시됨을 확인하였다(Fig. 15).

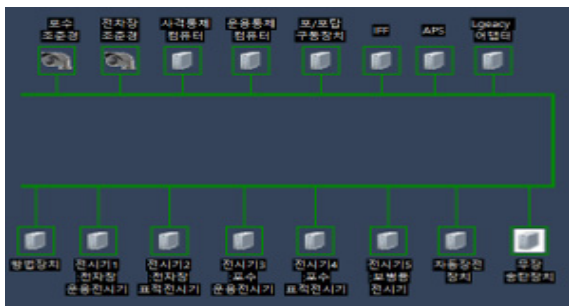


Fig. 14. Connectivity control result for A system

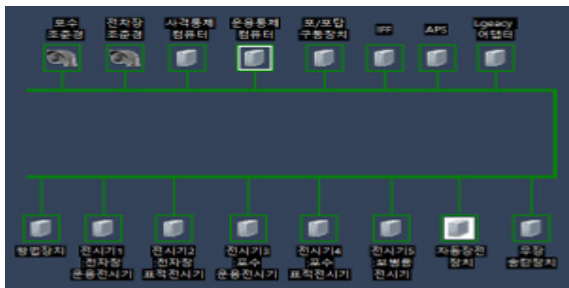


Fig. 15. Connectivity control result for B system

#### 4.3.2 전시기 자율 재구성 기능 확인

Fig. 6의 시나리오에 따라 4대의 전시기 중 1대 고장시, 3대 고장시 대체운용이 정상적으로 이루어지는지 전시기 GUI를 통하여 결과를 확인하였다. Fig. 16은 전시기 3대 고장시 전시기 2가 백업운용을 하고 있음을 보여준다. 제어기 2대 상호 간의 대체운용 시험 결과는 Fig. 17과 같다.

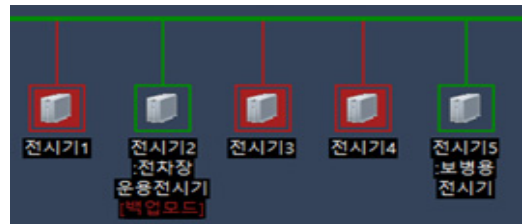


Fig. 16. Self-reconfiguration results of display units



Fig. 17. Self-reconfiguration results of controller-failure scenarios (a) failure of controller (Case 1) (b) failure of controller(Case 2)

## 4. 결론

본 논문은 서비스 연결제어 및 자율재구성을 통한 플러그애플레이 방법을 제안하고, 지상전투차량 유사 환경에서의 검증을 위한 실험 시체를 구성하여 작동 방법을 검증하였다. 제안한 방법은 지상무기체계 아키텍처 상의 공통 서비스 계층에서 표준화된 규격인 발간-구독 미들웨어 기반으로 동작하도록 설계하였다. 또한 무기체계별 임무 특성을 반영한 맞춤형 설계를 고려한 일반된 방식의 연결성 제어 방법과 운용 중 연결상태 및 운용상태 변화에 따른 자율 재구성이 가능한 통합 프레임워크를 제안하였다. 제안된 방법을 통해 아키텍처 관점에서는 무기체계별 특성을 반영하면서도 응용 소프트웨어 간 공용화 설계가 가능함에 따라 재사용성 및 상호운용성이 향상된다. 운용자 관

접에서는 장비 탈착 만으로도 장비 구성 및 재구성이 용이해짐에 따라 운용성 및 유지보수성 향상을 기대할 수 있으며, 궁극적으로는 비용 절감 효과를 기대할 수 있다. 향후 제안된 방법을 기반으로 각 장치 하드웨어 및 소프트웨어 운용 관련 보다 다양한 조건을 기반으로 연결 제어 및 자율재구성을 처리하도록 확장 가능할 것이다. 또한 제안하는 방법의 실제 무기체계 적용을 통한 다양한 운용 및 고장 상황에서의 검증이 수행되어야 할 것이다.

### References

- [1] Effective January 1, 2016, UPnP Forum Assigned Their Assets to the Open Connectivity Foundation (OCF).
- [2] OMG, The Real-Time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol Specification, V2.1, <http://www.omg.org/spec/DDS-RTPS/2.1>, 2010.
- [3] OMG, Data Distribution Service for Real-time Systems Version 1.2. <http://www.omg.org>.
- [4] H. M. Chang, et. al., “An Architectural Improvement of Fire Control Software for Ground Combat Vehicle,” KIMST Annual Conference Proceedings, pp. 1527-1528, 2018.
- [5] S. J. Kang, et. al., “Open Architecture Design for Fire Control System of Combat Vehicle,” KIMST Annual Conference Proceedings, pp. 1342-1343, 2019.
- [6] Soraya Sinche, et. al., “A Survey of IoT Management Protocols and Frameworks,” IEEE Communications Surveys & Tutorials, Vol. 22, No. 2, Second Quarter 2020.
- [7] OCF(2020), ‘OCF Core Specification v2.1.2’.
- [8] oneM2M, “oneM2M TR-0009 v0.4.0 - Technical Report - Protocol Analysis, January 2014.
- [9] H. M. Chang, et. al., “Adapting PnP to Fire Control System with Service Authorization,” KIMST Fall Conference Proceedings, pp. 116-117, 2019.
- [10] Y. J. Lee, et. al., “A Study on the Fault Tolerant Design of Ground Weapons System Based on Distributed System,” KIMST Fall Conference Proceedings, pp. 116-117, 2019.
- [11] Department of Defense(DoD), Vehicular Integration for C4ISR/EW Interoperability(VICTORY) Standards, Victory-standards.org, 2010.
- [12] Manas Pradhan, et. al., “Integrating Automotive Bus-based Networks in the NATO Generic Vehicle Architecture,” Twenty-First International Command and Control Research and Technology Symposium, September 2016.