

함정 전투체계의 도메인 컴포넌트 참조 아키텍처

이선열¹ · 채홍석^{2*}

Domain Component Reference Architecture within Naval Combat System

Seonyeol Lee¹ · Heung-Seok Chae^{2*}

¹Graduate Student, School of Computer Science and Engineering, Pusan National University, Busan, 46241 Korea

^{2*}Professor, School of Computer Science and Engineering, Pusan National University, Busan, 46241 Korea

요 약

함정 전투체계는 기술의 발전을 적극적으로 수용하여 점진적으로 진화하고 있다. 함정 전투체계의 도메인 컴포넌트들 또한 함정 전투체계의 점진적 진화에 맞추어 변경되어야 한다. 즉, 도메인 컴포넌트는 변경용이성을 가져야 한다. 그러나 기존 도메인 컴포넌트는 변경용이성을 충분히 고려하지 않는다. 특히, 미들웨어로부터 계획된 것보다 더 많은 메시지를 수신할 경우, 이에 대응하기 위한 방안을 고려하지 않는다. 우리는 증가된 메시지에 신속하게 대응할 수 있는 도메인 컴포넌트 참조 아키텍처를 제안한다. 제안된 아키텍처는 비즈니스 로직이 독립적으로 동작할 수 있도록 도메인 컴포넌트를 분리하고, 다수의 비즈니스 로직이 동시에 동작하도록 설정하여 증가된 메시지를 처리하고자 한다. 제안된 참조 아키텍처를 이용하여 표적관리 도메인 컴포넌트를 설계하여 참조 아키텍처의 적용 가능성을 확인하였다.

ABSTRACT

The naval combat system is gradually evolving, actively accepting technological advances. The domain components of the naval combat system must also be adapted to the gradual evolution of the naval battle system. In other words, domain components should be easy to change. However, existing domain components do not fully consider changeability. In particular, when receiving more messages than planned from the middleware, it does not consider a way to process them in a timely manner. We propose a domain component reference architecture for timely processing of increased messages. The proposed architecture separates domain components so that business logic can operate independently, and sets up multiple business logic to operate simultaneously to handle increased messages. The track management domain component is designed using the proposed reference architecture to confirm the applicability.

키워드 : 함정 전투체계, 참조 아키텍처, 도메인 컴포넌트, 변경용이성, 확장성

Keywords : Naval Combat System, Reference Architecture, Domain Component, Modifiability, Scalability

Received 16 August 2019, Revised 26 August 2019, Accepted 14 October 2019

* Corresponding Author Heung-Seok Chae(E-mail:hschae@pusan.ac.kr, Tel:+82-51-510-3517)

Professor, School of Computer Science and Engineering, Pusan National University, Busan, 46241 Korea

Open Access <http://doi.org/10.6109/jkiice.2019.23.11.1443>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

우리 해군의 함정 전투체계는 미래 전장 환경에 부합하고 향후 국내개발 무기체계와의 상호운용성을 보장하기 위해 공동운영환경 기반의 개방형 아키텍처를 기반으로 설계되고 있다[1]. 이는 임무 지향적인 폐쇄형 아키텍처와는 다르게, 정보/과학 기술의 발전을 적극적으로 수용하여, 임무 변화에 대한 적응과 정보처리능력의 향상, 장기적 기술 발전능력 확보 등의 많은 기술적, 경제적 이점을 제공한다[2~4]. 그림 1은 함정 전투체계 개방형 아키텍처 구조를 나타낸다[1].

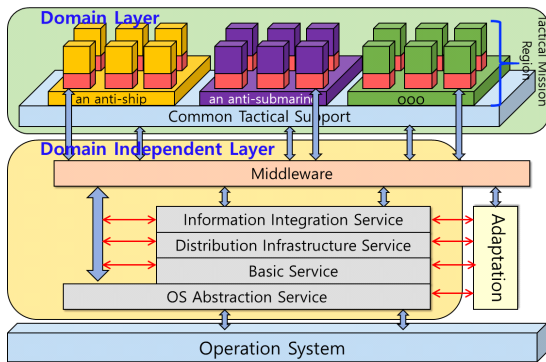


Fig. 1 An naval combat system open architecture

그림 1의 도메인 계층(Domain Layer)에는 전술지원과 전술운용을 위한 도메인 컴포넌트들이 존재한다. 전술지원 도메인 컴포넌트들은 표적 관리, 위협 평가 등의 함정 전투체계에서 공통적으로 이용되는 기능들을 제공한다. 반면, 전술운용 도메인 컴포넌트들은 대잠전, 대함전 등과 같이 함정 유형별로 고유한 임무 수행 및 전술 운용을 위한 기능을 제공한다[1].

도메인 컴포넌트들은 개방형 아키텍처 기반 함정 전투체계가 진화함에 따라서 적절히 변경되어야 한다. 네트워크 기술의 발전은 함정에 더 많은 표적탐지 센서의 탑재를 가능하게 한다. 그리고 탐지 기술의 발전은 더 넓은 범위를 커버하는 새로운 센서의 탑재를 가능하게 한다. 이는 표적관리 도메인 컴포넌트가 더 많은 표적을 신속하게 관리할 수 있도록 변경되어야 함을 의미한다. 즉, 도메인 컴포넌트는 변경용이성(modifiability)을 가져야 한다.

도메인 컴포넌트의 변경용이성은 외부의 변경에 대

비한 설계에 의해 달성될 수 있다. 그림 1과 같이, 도메인 컴포넌트는 미들웨어를 이용하여 타 도메인 컴포넌트와 메시지를 송수신하기 때문에, 메시지 송수신 API 및 메시지 형식의 변경 그리고 발신 메시지의 증가 등이 도메인 컴포넌트의 변경을 유발하는 요인이 된다. 이들에 의한 도메인 컴포넌트의 변경이 용이하게 수행될 수 있도록 도메인 컴포넌트가 설계되어야 한다.

특히, 발신 메시지 증가에 대비하여 도메인 컴포넌트의 확장성(scalability)은 반드시 고려되어야 한다. 확장성을 고려하지 않은 도메인 컴포넌트는 수신 메시지 증가에 신속하게 대응하기 어렵다. 이를 적시에 처리하려면, 도메인 컴포넌트의 많은 부분이 변경되어야 한다. 심지어, 도메인 컴포넌트가 재개발되어야 할지도 모른다. 함정 전투체계의 진화에 따라, 도메인 컴포넌트의 수신 메시지가 필연적으로 증가하기 때문에, 확장성을 고려하지 않은 컴포넌트의 유지보수 비용은 매우 높을 것이다.

그러나 기존 도메인 컴포넌트는 확장성을 고려하여 설계하고 있지는 않은 것으로 보인다[5~7]. 이들은 미들웨어로부터 메시지를 송수신하는 컴포넌트와 메시지 변환 컴포넌트를 이용하여 미들웨어의 API 및 메시지 형식의 변경이 비즈니스 로직을 수정하는 경우를 방지하고자 하였지만, 도메인 컴포넌트의 확장성이 문제가 될 수 있음을 언급하지는 않는다. 그리고 이 문제에 대한 대응 방법도 기술하고 있지 않다.

본 논문은 미들웨어로부터의 변경에 신속하게 대응할 수 있는 도메인 컴포넌트의 참조 아키텍처를 제안한다. 특히, 도메인 컴포넌트의 확장성 문제에 신속하게 대응하고자 한다. 제안된 아키텍처는 비즈니스 로직이 독립적으로 동작할 수 있도록 도메인 컴포넌트를 분리하고, 독립된 다수의 비즈니스 로직이 동시에 병렬적으로 동작할 수 있도록 설계하였다. 수신 메시지의 증가에 따라 신속하게 비즈니스 로직을 추가할 수 있기 때문에, 미들웨어의 메시지 발신 속도 증가에 대한 도메인 컴포넌트의 변경을 용이하게 수행할 수 있다.

제안된 참조 아키텍처를 이용하여 표적관리 도메인 컴포넌트를 설계하였다. 확장성을 고려하여 다수의 비즈니스 로직이 동시에 표적을 처리할 수 있도록 설계하였다. 이를 위해, 컴포넌트의 각 구성요소가 표적의 동시 처리 중 발생할 수 있는 문제를 다루는 방법도 설명하였다. 그리고 설계된 표적관리 컴포넌트가 함정의 임

무와 규모에 따라 처리 표적의 수가 달라질 수 있는 함정 전투체계에 적합함을 설명하였다.

본 논문의 구성은 아래와 같다. 2장은 제안한 참조 아키텍처를 소개하고, 3장은 표적관리 도메인 컴포넌트에 제안한 참조 아키텍처를 적용한 결과를 소개한다. 4장은 함정 전투체계 아키텍처와 관련된 연구를 소개한다. 5장은 본 연구의 결론을 소개한다.

II. 도메인 컴포넌트 참조 아키텍처 제안

2.1. 함정 전투체계 도메인 컴포넌트 설계 고려사항

개방형 아키텍처 기반 함정 전투체계가 점진적으로 진화하기 때문에, 함정 전투체계의 도메인 컴포넌트 또한 전투체계의 진화 속도에 맞추어 변경되어야 한다. 즉, 도메인 컴포넌트는 변경용이성을 가져야 한다.

도메인 컴포넌트가 변경용이성을 가지려면, 외부의 변경에 대비한 설계가 수행되어야 한다. 도메인 컴포넌트는 미들웨어를 통해 타 도메인 컴포넌트와 상호작용하기 때문에, 메시지 송수신 API 및 메시지 형식의 변경 그리고 발신 메시지의 증가와 같은 미들웨어의 변경이 도메인 컴포넌트의 수정을 유발할 수 있다. 이들의 변경에 대비한 설계가 필요하다. 특히, 증가된 발신 메시지를 적시에 처리할 수 없다면, 해당 컴포넌트의 재개발이 필요할 만큼 심각한 문제가 될 수 있기 때문에, 도메인 컴포넌트의 확장성은 반드시 고려되어야 한다.

2.2. 함정 전투체계 도메인 컴포넌트 참조 아키텍처

본 논문은 도메인 컴포넌트의 확장성을 고려한 참조 아키텍처를 제안한다. 함정 전투체계의 도메인 컴포넌트는 미들웨어로부터 수신한 메시지를 순차적인 절차에 따라 처리하기 때문에 Pipes and Filter 아키텍처 패턴 [8]을 기반으로 한다. 그림 2는 함정 전투체계 도메인 컴포넌트 참조 아키텍처를 나타낸다.

메시지 수신자(Message Receiver)는 미들웨어로부터 메시지를 수신하고, 수신된 메시지를 비즈니스 로직이 처리할 수 있는 도메인 데이터로 변환한 후, 도메인 데이터 큐에 데이터를 인큐(enqueue)한다. 메시지 수신자가 도메인 컴포넌트와 미들웨어 간 인터페이스를 캡슐화함으로써, 비즈니스 로직은 미들웨어에서 발신된 메시지 형식에 의존하지 않고, 자신의 핵심 기능을 구현할

수 있다.

도메인 데이터 큐(Domain Object Queue)는 비즈니스 로직에 의해 처리될 도메인 데이터가 대기하는 저장소이다. 비즈니스 로직에 처리할 수 있는 것보다 더 많은 데이터가 입력되면, 비즈니스 로직은 과부하에 의해 중단될 수 있다. 이 큐는 비즈니스 로직이 처리해야 할 데이터를 순서대로 저장하고 있기 때문에, 비즈니스 로직은 이 큐를 이용하여 자신의 처리 속도로 도메인 데이터를 처리할 수 있다. 도메인 컴포넌트가 복수의 비즈니스 로직을 사용한다면, 큐의 데이터가 복수의 비즈니스 로직에 의해 동시에 처리될 수 있다. 동시에 처리되는 데이터가 의존 관계를 가지면, 비즈니스 로직이 잘못된 결과를 출력할 수 있기 때문에, 의존 관계를 가지는 도메인 데이터들이 인큐되지 않도록 도메인 데이터 큐가 설계되어야 한다.

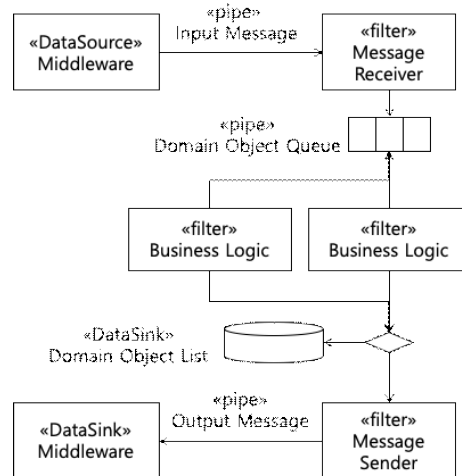


Fig. 2 A domain component reference architecture within naval combat system

비즈니스 로직(Business Logic)은 도메인 컴포넌트의 핵심 기능을 구현한다. 비즈니스 로직의 입력은 도메인 데이터 큐에 디큐(dequeue) 연산을 수행한 결과로 얻은 도메인 데이터이다. 경우에 따라서, 도메인 데이터 목록에 저장되어 있는 기존 도메인 데이터 처리 결과를 필요로 할 수도 있다. 비즈니스 로직은 데이터를 처리할 수 있을 때, 도메인 데이터 큐를 디큐하므로, 자신의 처리 속도에 따라 안정적으로 데이터를 처리할 수 있다. 비즈니스 로직이 처리한 결과는 도메인 데이터 목록에 저장되고, 메시지 발신자에 전달된다. 도메인 컴포넌트의 특

성에 따라 복수의 비즈니스 로직이 사용될 수도 있다. 복수의 비즈니스 로직은 도메인 데이터 큐 및 도메인 데이터 목록에 동시에 접근할 수 있기 때문에, 이들에 대한 접근 통제가 필요하다.

도메인 데이터 목록(Domain Object List)은 비즈니스 로직의 도메인 데이터 처리 결과를 저장한다. 언급한 것처럼, 비즈니스 로직은 기존의 결과를 필요로 하기도 하기 때문에, 도메인 데이터 목록은 비즈니스 로직에 의해 읽혀지고, 쓰여진다. 복수의 비즈니스 로직이 사용될 경우, 도메인 데이터 목록에 대한 읽기와 쓰기가 모두 통제되어야 한다. 이는 도메인 데이터 목록의 사용 효율을 급감시킬 수 있기 때문에, 이를 고려한 설계가 필요하다.

메시지 발신자(Message Sender)는 비즈니스 로직으로부터 전달받은 데이터를 메시지로 변환하여 미들웨어에 발신한다. 메시지 발신자는 도메인 컴포넌트와 미들웨어 간 인터페이스를 캡슐화한다.

2.3. 참조 아키텍처의 미들웨어 변경에 대한 대응

참조 아키텍처는 미들웨어의 메시지 송수신 API 및 송수신 메시지 형식의 변경에 따른 영향력을 메시지 수신자와 메시지 발신자에 국한시켰다. 이로 인해 미들웨어와의 인터페이스 변경이 도메인 컴포넌트의 핵심 기능을 구현한 비즈니스 로직의 변경을 유발하지 않는다.

그리고 함정 전투체계에 타 도메인 컴포넌트의 추가, 변경 및 확장으로 인해 미들웨어가 더 많은 메시지를 발신하는 경우에도 참조 아키텍처는 확장성을 가진다. 도메인 컴포넌트의 수신 메시지가 증가하면, 비즈니스 로직이 처리할 데이터가 도메인 데이터 큐에 쌓이게 된다. 제안 아키텍처는 쌓인 데이터를 동시에 소비할 수 있는 복수의 비즈니스 로직이 독립적으로 동작할 수 있도록 설계하였다. 이를 위해 도메인 컴포넌트 내부의 다른 구성요소들이 변경될 필요는 없다. 단, 도메인 데이터 큐 및 도메인 데이터 목록이 복수의 비즈니스 로직에 의해 처리될 수 있도록 설계되어 있어야 한다.

III. 표적관리 도메인 컴포넌트 아키텍처

3.1. 표적관리 도메인 컴포넌트

표적관리 도메인 컴포넌트는 함정에 탑재된 다종의 센서 및 데이터링크로부터 수신한 표적(원시표적)정보

를 적절한 방법으로 처리(동일표적 판단, 결정, 추정 등)하여 전술대응이 가능한 표적(전술표적)정보를 생성하고 목록으로 관리한다. 관리된 표적은 실시간 전술 대응을 위해 전투체계 내의 타 도메인 컴포넌트에 전달된다. 또한, 전술데이터링크 연동 기능을 이용하여 타 함정과 표적정보를 공유한다. 그림 3은 표적관리 도메인 컴포넌트의 컨텍스트 다이어그램을 나타낸다.

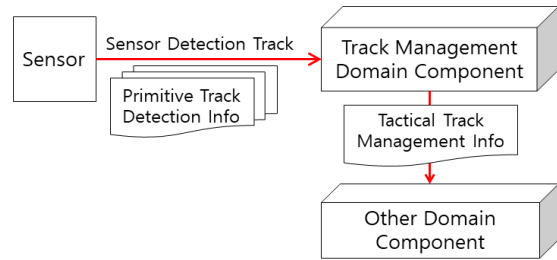


Fig. 3 The reference architecture for a domain component within naval combat system

표적관리 도메인 컴포넌트는 센서로부터 여러 센서 탐지표적 메시지를 수신한다. 하나의 센서가 한 주기 동안 다수의 표적을 탐지하기 때문에, 센서탐지표적 메시지는 다수의 원시표적 탐지정보가 포함된다. 메시지를 수신받은 표적관리 도메인 컴포넌트는 메시지의 각 원시표적 탐지정보를 반영하여 신규 전술표적을 생성하거나, 기존의 전술표적을 갱신 및 삭제한다. 이와 같은 전술표적 관리정보 메시지는 교전과 같은 타 도메인 컴포넌트로 발신된다. 그림 4는 표적관리 도메인 컴포넌트의 유스케이스 다이어그램을 나타낸다.

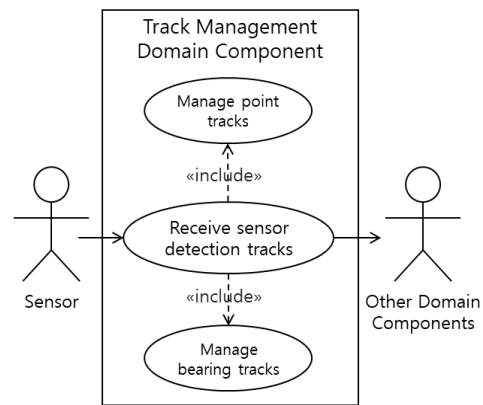


Fig. 4 A track management use case diagram

표적관리 도메인 컴포넌트의 액터는 센서(Sensor)와

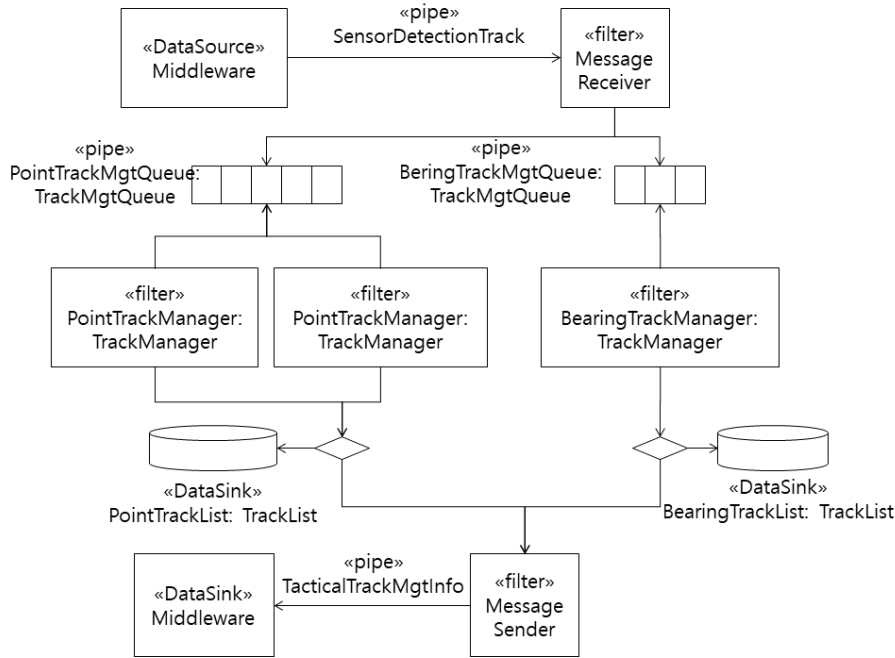


Fig. 5 A track management domain component architecture based on the proposed architecture

타 도메인 컴포넌트(Other Domain Components)이며, 센서탐지표적수신(Receive sensor detection tracks) 유스케이스, 점 표적 관리(Manage point tracks) 유스케이스 그리고 방위 표적 관리(Manage bearing tracks) 유스케이스가 있다. 센서탐지표적 메시지 수신 유스케이스는 점 표적 관리 유스케이스와 방위 표적 관리 유스케이스를 포함한다. 이는 센서로부터 탐지된 원시표적 탐지 정보들이 점 표적 또는 방위 표적으로 구분되어, 전술표적 관리에 반영됨을 의미한다.

원시점표적 탐지정보(Primitive Point Track Detection Info) 또는 원시방위표적 탐지정보(Primitive Bearing Track Detection Info)로 변환한다. 변환된 탐지 정보들은 동일한 센서에 의해 탐지된 표적정보임을 나타내는 원시점표적 탐지목록(Primitive Point Track Detection List)과 원시방위표적 탐지목록(Primitive Bearing Track Detection List)으로 그룹핑되어, 그림 5의 표적관리큐인 점표적관리큐(PointTrackMgtQueue)와 방위표적관리큐(BearingTrackMgtQueue)로 전달된다.

3.2. 표적관리 도메인 컴포넌트 아키텍처

그림 5는 제안한 참조 아키텍처를 기반으로 설계한 표적관리 도메인 컴포넌트 아키텍처를 나타낸다. 아래에서 아키텍처를 상세히 설명한다.

3.2.1. 메시지 수신자

센서에서 발신된 센서탐지표적 메시지는 미들웨어를 거쳐 표적관리 도메인 컴포넌트에 수신된다. 메시지 수신자는 이 센서탐지표적(Sensor Detection Track) 메시지를 수신한다. 메시지 수신자는 수신 메시지 내의 각 원시표적 탐지정보들(Primitive Track Detection Info)을

3.2.2. 표적관리큐

표적관리큐(TrackMgtQueue)는 원시표적 탐지목록이 그림 5의 표적관리자(TrackManager)에 의해 처리될 때까지 대기하는 큐이다. 점표적관리큐(PointTrackMgtQueue)는 원시점표적 탐지목록이, 방위표적관리큐(BearingTrackMgtQueue)는 원시방위표적 탐지목록이 대기한다. 이 두 큐는 저장하는 표적의 종류가 다른 표적관리큐의 인스턴스들이다.

표적관리큐는 대기큐(Waiting Queue)와 처리큐(Handling Queue)를 가진다. 대기큐는 표적관리자에 의해 처리를 대기하는 원시표적 탐지목록들을 저장하고, 처리큐는

표적관리자들이 현재 처리중인 원시표적 탐지목록의 원시표적 탐지정보들을 저장한다. 그림 6은 표적관리큐를 나타낸다.

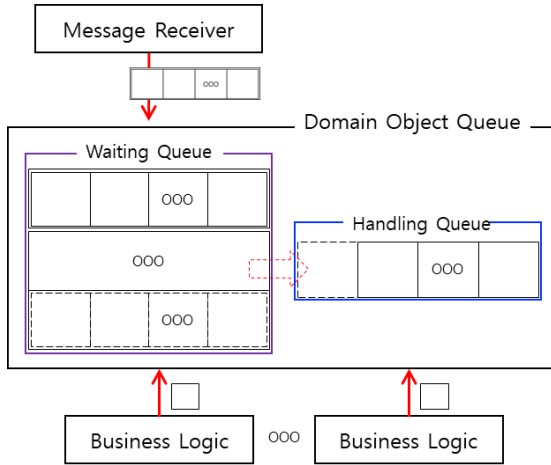


Fig. 6 The structure of track management queue

그림 6과 같이, 메시지 수신자로부터 입력된 원시표적 탐지목록은 대기큐에 삽입된다. 그리고 처리큐는 그림 5의 표적목록이 처리큐의 원시표적 탐지정보 삽입 함수를 콜백으로 호출하면, 대기큐의 탐지목록 중 가장 먼저 삽입된 탐지목록을 대기큐에서 제거하고, 제거된 목록의 탐지정보들을 처리큐에 삽입한다. 콜백 함수가 호출되는 경우는 아래 표적목록을 설명하는 절에서 자세하게 설명한다.

표적관리큐가 그림 6의 구조를 가지는 이유는 다른 센서에 의해 탐지된 원시표적 탐지정보들은 동시에 처리될 수 없지만, 하나의 센서에 의해 탐지된 원시표적 탐지정보들은 동시에 처리될 수 있기 때문이다. 센서들은 많은 표적을 탐지한다. 탐지된 표적 중 일부는 다수의 센서에 의해 동시에 처리될 수 있다. 만약, 다른 센서에 의한 원시표적 탐지정보들이 표적관리자들에 의해 동시에 처리된다면, 표적관리자는 동일한 전술표적 정보를 동시에 편집할 것이다. 즉, 동시성 문제가 발생할 수 있다. 그러나 하나의 센서에 의해 탐지된 원시표적 탐지정보는 각자 다른 전술표적 정보를 수정하기 때문에 동시성 문제가 발생하지 않는다.

3.2.3. 표적관리자

표적관리자(TrackManager)는 표적관리큐의 처리큐

를 폴링하여 얻은 원시표적 탐지정보를 이용하여, 신규 전술표적을 생성하거나, 기존의 전술표적을 갱신 및 삭제한다. 이렇게 관리된 전술표적 정보는 그림 5의 표적 목록에 저장된다.

점표적과 방위표적은 점표적관리자와 방위표적관리자에 의해 분리되어 관리된다. 두 표적이 서로에 대해 독립적으로 관리되기 때문이다. 그리고 두 표적의 관리는 두 표적의 동일표적 판단 알고리즘을 제외한 나머지 절차가 모두 동일하기 때문에, 알고리즘 변경에 대한 확장성을 가진 전략(strategy) 패턴과 널 객체(null object) 패턴을 적용하여 구현 코드를 완전히 재사용할 수 있다. 그림 5에서 점표적관리자와 방위표적관리자가 표적관리자의 인스턴스로 표현된 이유이다. 표적관리를 위한 자세한 절차는 이호철 외[9]을 참조한다.

그림 5에는 2개의 점표적관리자와 1개의 방위표적관리자가 존재한다. 즉, 점표적을 동시에 처리할 수 있도록 표적관리 도메인 컴포넌트를 설계하였다. 이는 함정 전투체계에 매우 많은 수의 점표적 탐지 센서가 존재하기 때문이다. 다수의 점표적관리자가 점표적관리큐를 경쟁적으로 소비하여, 원시점표적이 신속하게 처리될 수 있도록 하였다. 동시 처리에 의한 동시성 문제는 그림 6의 구조와 같은 표적관리큐와 후술할 표적목록을 통해 다루어진다. 상대적으로 적은 수인 방위표적은 하나의 방위표적관리자에 의해 처리된다.

3.2.4. 표적목록

표적목록(TrackList)은 표적관리자에 의해 처리된 전술표적 정보를 저장한다. 표적처리 과정에서 점표적과 방위표적이 서로 의존하지 않기 때문에 별도의 목록으로 존재한다.

점표적목록은 점표적관리자의 동시 처리를 지원하기 위해 읽기 목록과 쓰기 목록을 가진다. 점표적관리자들은 읽기 목록의 표적정보를 얻어 원시표적 탐지정보를 처리하고, 처리 결과를 쓰기 목록에 갱신한다. 이와 같은 방법으로 점표적관리자들이 서로 영향을 받지 않고 원시표적 탐지정보를 동시에 처리할 수 있다.

표적관리자가 그림 6의 처리큐에 있는 모든 원시표적 탐지정보를 소비하면, 표적목록에 모든 데이터가 소비되었음을 알린다. 표적 목록은 이 알림을 받으면, 쓰기 목록의 내용을 읽기 목록으로 갱신하여 동기화된다. 읽기 목록과 쓰기 목록이 동기화된 후, 앞서 언급하였

던, 처리큐의 콜백 함수를 호출하여, 처리큐에 원시표적 탐지정보를 다시 채운다.

3.2.5. 메시지 발신자

메시지 발신자는 표적관리자로부터 처리된 전술표적 관리정보를 얻어서, 전술표적 관리정보를 전술표적 관리정보 메시지로 변환된다. 변환된 메시지는 미들웨어를 통해 타 도메인 컴포넌트로 전달된다.

3.3. 표적관리 도메인 컴포넌트의 변경 요구사항 수용

함정 전투체계에는 표적을 탐지하는 센서가 탑재된다. 탑재되는 센서의 수와 종류는 함정의 크기나 임무에 따라 달라질 수 있으며, 탑재된 센서의 수와 종류에 따라 미들웨어의 센서탐지표적 메시지 발신 속도는 달라진다.

표적관리 도메인 컴포넌트는 미들웨어의 센서탐지표적 메시지 발신 속도에 의존하지 않고 이들을 모두 적시에 처리할 수 있어야 한다. 이들 중 일부라도 적시에 처리할 수 없다면, 함정 전투체계의 임무는 실패할 수 있다.

그림 5와 같은 표적관리 도메인 컴포넌트는 이와 같은 상황에 신속하게 대응할 수 있다. 함정의 필요에 따라 센서탐지표적 메시지의 처리 속도를 조절할 수 있기 때문이다. 그림 5와 같이, 점표적 탐지 비율이 높은 함정은 다수의 점표적관리자를 동작하도록 설정하여, 더 많은 수의 점표적을 처리할 수 있다. 방위표적 탐지 비율이 높다면, 다수의 방위표적관리자를 사용할 수 있다. 이를 위해 다른 컴포넌트가 변경될 필요는 없다. 단지, 새로운 표적관리자가 표적관리큐의 데이터를 소비하고, 표적목록에 처리된 결과를 저장하도록 설정하기만 하면된다.

IV. 관련 연구

심준용 외[5]는 다 표적 동시 교전 기능을 위한 교전 관리 도메인 컴포넌트를 설계하였다. 교전관리 도메인 컴포넌트는 전투체계 관리자와 무장 인터페이스로 구성된다. 무장 인터페이스는 외부와의 인터페이스를 구현하고, 전투체계 관리자는 도메인 컴포넌트의 비즈니스 로직을 구현한다. 이들은 다 표적 동시 교전 처리 시 표적 정보를 동기화하는 방법에 대해 자세히 설명한다.

박규진 외[6]는 DDS를 이용한 표적관리 도메인 컴포넌트에 대해 소개하였다. 표적관리 도메인 컴포넌트는 3계층으로 구분된다. Operating System 계층은 센서로부터 데이터를 입력받고, Platform 계층은 센서의 데이터를 도메인 데이터로 변화하고, 표적을 처리한다. 그리고 Application 계층은 표적을 디스플레이한다. 또한, 표적 처리를 위한 흐름을 소개한다.

정영란 외[7]는 대함유도탄 방어유도탄(SAAM) 도메인 컴포넌트에 대한 설계를 소개한다. 이들은 외부 컴포넌트와 통신을 위한 컴포넌트와 메시지 변환 컴포넌트를 통해 미들웨어의 인터페이스 변경이 비즈니스 로직에 영향을 미치지 않도록 설계하였다.

소개한 연구들은 공통적으로 미들웨어와의 인터페이스를 위한 컴포넌트를 이용하여 비즈니스 로직이 미들웨어와의 인터페이스 변경에 의해 변경되지 않도록 하였다. 그러나 이들은 도메인 컴포넌트의 확장성이 문제가 될 수 있음을 언급하고 있지 않고, 그에 대한 대응 방법도 기술하고 있지 않다.

V. 결론

함정 전투체계의 도메인 컴포넌트는 미들웨어로부터 메시지를 수신하고, 이를 적시에 처리하여 그 임무를 수행한다. 함정 전투체계가 진화함에 따라, 도메인 컴포넌트는 계획된 것보다 더 많은 메시지를 수신할 수 있다. 이들 또한 적시에 처리해야 한다. 만약, 이들을 적시에 처리할 수 없다면, 도메인 컴포넌트는 재개발되어야 한다.

본 논문은 도메인 컴포넌트의 확장성을 고려한 참조 아키텍처를 제안한다. 제안된 아키텍처는 비즈니스 로직이 독립적으로 동작할 수 있도록 도메인 컴포넌트를 분리하고, 다수의 비즈니스 로직이 동시에 동작하도록 설정하여 증가된 메시지를 처리하고자 한다. 비즈니스 로직 추가 시, 기존 컴포넌트의 수정이 필요하지 않으므로, 메시지 발신 속도 증가에 신속하게 대응할 수 있다.

제안된 참조 아키텍처를 이용하여 표적관리 도메인 컴포넌트를 설계하였다. 표적관리 도메인 컴포넌트는 표적을 동시에 처리할 수 있기 때문에, 컴포넌트의 각 구성요소에서 동시성이 어떻게 다루어지고 있는지도 함께 설명하였다. 그리고 설계된 아키텍처가 함정의 크기와 임무에 따라 수신 메시지 속도가 달라질 수 있는

표적관리 도메인 컴포넌트에 적합함을 설명하였다.

ACKNOWLEDGEMENT

This work was supported by Agency for Defense Development(ADD) of Republic of Korea under Project No.UD170004DD. The authors are very grateful for this support.

REFERENCES

- [1] S. H. Baek, "Proposal of Open Architecture Reference Model for Naval Combat System," *Marine Strategy*, vol. 144, pp.105-136, Dec. 2009.
- [2] T. Strei, "Overview of naval open systems: the plans, the approach, the promise," in *Proceedings of the sixteenth annual systems & software technology conference*, Salt Lake City, Utah. pp. 2-7, 2004.
- [3] Object Management Group std, *Data Distribution Service (DDS) Version 1.4*, OMG, 2015.
- [4] C. S. Kim, H. M. Chang, J. H. Joo, and G. J. Lee, "A Study on Real-Time Operating Systems for Architectural Improvement of Naval Combat Systems," *Journal of the Korea Institute of Military Science and Technology*, vol. 16, no. 3, pp. 260-267, Jun. 2013.
- [5] J. Y. Shim, W. S. Lee, S. K. Lee, and S. H. Wi, "Design of Engagement Management Module for Multi Target Simultaneous Engagement Function of Naval Combat System Simulator," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 43, no. 05, pp.863-865, May. 2018.
- [6] K. J. Park, S. Y. Choi, B. D. Bae, and Y. S. Seo, "The study of Track management through DDS," in *KSAS 2017 Fall Conference*, pp.288-289, 2017.
- [7] Y. R.Jung, C. H. Kim, W. G. Han, J. I. Kim, and H. S. Kim, "A Study on the Design and Verification-Validation of the Supportive Equipment for Shipyard Test of Naval Combat System," *Journal of the Korea Institute of Military Science and Technology*, vol. 17, no. 3, pp. 318-326, Jun. 2014.
- [8] H. Subramanian, A. Raman, and P. Raj, *Architectural Patterns*, Packt Publishing, 2017.
- [9] H. C. Lee, T. S. Kim, and H. J. Shin, "Multi Sources Track Management Method for Naval Combat Systems," *Journal of Institute of Control, Robotics and Systems*, vol. 20, no. 2, pp. 126-131, Feb. 2014.



이선열(Seonyeol Lee)

부산대학교 정보컴퓨터공학부 졸업(학사)
부산대학교 전자저기컴퓨터공학부 졸업(석사)
부산대학교 전자전기컴퓨터공학부 박사과정
※관심분야 : 객체지향 방법론, 소프트웨어 테스트, 소프트웨어 아키텍처



채흥석(Heung-Seok Chae)

서울대학교 원자핵공학과 졸업(학사)
한국과학기술원 전산학과 졸업(석사)
한국과학기술원 전산학과 졸업(박사)
2000년~2003년 동양시스템즈 기술연구소 선임연구원
2003년~2004년 한국과학기술원 전산학과 초빙교수
2004년~현재 부산대학교 컴퓨터공학과 교수
※관심분야 : 객체지향 방법론, 소프트웨어 테스트, 소프트웨어 검증