

개방형 아키텍처 컴퓨팅 환경의 기술 아키텍처 및 구성요소 분석에 관한 연구

최동진*, 김민규*, 윤희병*, 이일로**

*국방대학교 전산정보학과, **국방기술품질원 SW팀

The Technology Architecture and Main Component Analysis of Open Architecture Computing Environment

Dongjin Choi, Minkyu Kim, Heebyung Yoon, Illo Lee

Korea National Defense University, Defense Agency for Technology and Quality

E-mail : pooh50@empal.com, kmk0117@hanmail.net, hbyoon37@hanmail.net, 215-b@hanmail.net

요 약

본 논문에서는 NCW 환경에서 요구되는 무기체계 성능을 만족시키기 위해 미 해군에서 추진 중인 개방형 아키텍처 컴퓨팅 환경(OACE)의 기술 아키텍처 및 구성요소에 대해 분석한다. 먼저 개방형 아키텍처와 개방형 아키텍처 컴퓨팅 환경에 대한 개념 및 특징을 살펴보고, OACE 기술 아키텍처를 분석한다. 그리고 OACE를 구성하는 핵심요소인 운영체제, 미들웨어, 정보관리, 정보보호, 프로그래밍 언어 등에 대한 연구를 수행한다. 향후 본 논문에서 수행한 연구는 무기체계 구축이 OACE 기반으로 변화될 경우 기술적 측면에서 많은 기여가 될 것으로 기대한다.

1. 서론

현대 무기체계는 소프트웨어 집약 시스템으로 발전하고 있다. 기존의 국방 표준 기반 소프트웨어 아키텍처는 네트워크 중심전(NCW: Network Centric Warfare) 환경에서 요구되는 무기체계의 성능을 만족시킬 수 없기 때문에 미국을 비롯한 선진국에서는 개방형 아키텍처 컴퓨팅 환경(OACE: Open Architecture Computing Environment)과 같은 개방형 표준 기반의 무기체계 소프트웨어 아키텍처를 구축하는 전략으로 변화하고 있다.

특히 미 해군은 Aegis 전투체계 및 E-2 Hawkeye 시스템 개선에 OACE를 적용하는 등 활발한 연구가 진행 중에 있으나, 한국군의 무기체계 분야에서는 이에 대한 연구가 미흡한 실정이다.

따라서 본 논문에서는 미 해군의 OACE 분석을 통한 기술 아키텍처 및 구성요소 분석에 관한 연구를 수행한다. 본 논문의 구성은 다음과 같다. 먼저 2장에서 OA(Open Architecture)와 OACE의 개념 및 특성에

대해 알아보고 3장에서는 OACE 기술 아키텍처 및 OACE의 주요 구성요소에 대한 연구를 수행하며 마지막으로 4장에서 결론을 맺는다.

2. 관련 연구

2.1 개방형 아키텍처(OA)

OA란 시스템 내의 핵심 인터페이스에 대해 개방형 표준을 적용한 아키텍처로써[1] 독자적인 특정 밴더에 의존하는 것이 아니라, 제한사항 없이 이용 가능한 명세서에 기반을 두고 있는 아키텍처를 의미한다.

시스템 개발에 OA를 도입하게 되면 광범위한 산업적 기반을 통한 기술 경쟁력 강화로 신기술 도입 및 운용에 효과적으로 대처할 수 있으며, 시스템에 대한 표준 인터페이스의 운용을 통해 시스템 효율을 최적화 할 수 있다. 또한 컴포넌트에 대한 재사용성을 증대하여 비용 절감에도 효과적이다[2].

미 해군의 OA 기반 무기체계의 가장 큰 특징은 개방형 표준의 사용과 OACE의 구축에 있다. 개방형 표준을 사용함으로써 시스템 개발 시 성능 향상 및 비용 절감을 이룰 수 있으며, OA 기반 무기

본 연구는 국방기술품질원의 무기체계 내장형 소프트웨어 기술정보 통합관리체계 개념연구 등(과제번호:제 2008_235호)의 연구비지원에 의하여 연구되었음

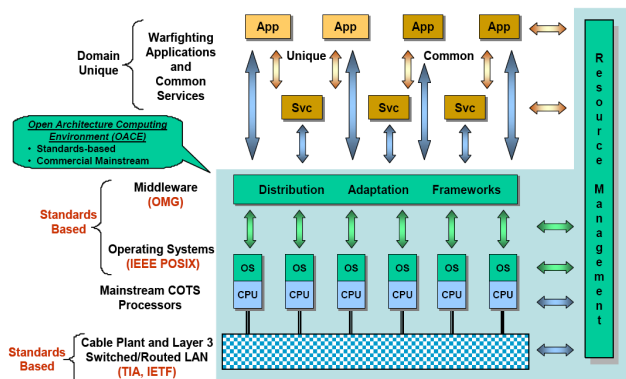
체계에 공통적으로 적용되는 컴퓨팅 기반환경인 OACE 구축을 통해 상호운용성 및 소프트웨어 컴포넌트에 대한 재사용성 등을 최대화시킬 수 있다.

2.2 개방형 아키텍처 컴퓨팅 환경(OACE)

OACE란 OA 기반의 전투 시스템 내에서 사용되는 통합 표준 기반의 컴퓨팅 자원의 집합과 기반 구조로써 이를 구축하기 위한 기술과 표준을 포함하고 있다[3].

주요 특징은 개방형 표준의 사용과 계층화된 모듈화 구조에 있다. 특히 계층화 구조를 통해 소프트웨어 컴포넌트들은 각 계층으로 그룹화되며, 각 계층은 표준화된 인터페이스를 사용하여 계층 간의 분리를 구현한다.

OACE는 일반적인 시스템과 전투체계 및 지원 시스템 모두에서 공통적으로 사용할 수 있는 컴퓨팅 하드웨어와 운영체제, 미들웨어, 자원관리 등을 그 범위로 하고 있으며 특정 도메인에 특화된 애플리케이션 및 서비스는 해당 범위에서 제외된다. OA 기반 시스템을 도식화하면 <그림 1>과 같으며, 음영으로 처리된 부분이 OACE의 범위에 해당한다.



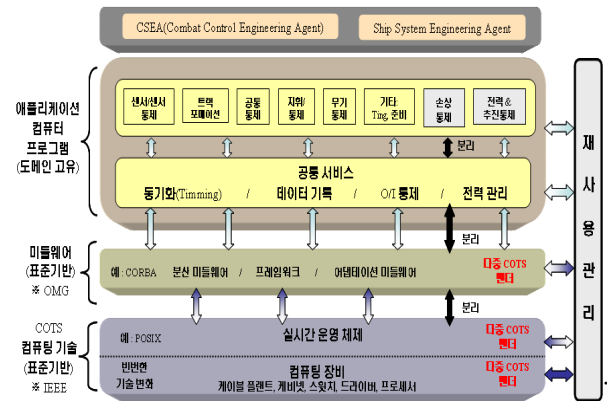
<그림 1> OA기반 시스템의 OACE의 범위

3. OACE 기술 아키텍처 분석

OACE 기술 아키텍처는 컴퓨팅 하드웨어와 운영체제, 미들웨어, 공통서비스 및 임무에 특화된 애플리케이션이 계층화 구조로 이루어져 있으며, OACE 기술 아키텍처에서 각 계층은 정의된 인터페이스를 이용한 느슨한 연결을 통해 계층별로 분리된다. 이를 통해 계층 간 독립적인 관계를 유지함으로써 각 계층의 변경에 대한 타 계층의 영향성 최소화가 가능하다.

또한 재사용 관리를 통해 데이터를 체계적으로 정리하여 축적된 데이터에 대한 중복과 이로 인한 자원의 낭비를 관리한다. 이러한 OACE 기술 아키텍처

계층구조는 <그림 2>와 같다[4].



<그림 2> OACE 기술 아키텍처

기술 아키텍처의 각 계층은 다수의 소프트웨어 컴포넌트로 구현된다. 사용자가 특정 기능을 요구하게 되면 이는 QoS 컴포넌트와 리소스/애플리케이션 관리자를 거쳐 실제 사용자 애플리케이션 컴포넌트와 연결된다. 이후 사용자 애플리케이션은 필요에 따라 분산 미들웨어를 통해 다른 컴포넌트와 연결되며 운영체제 서비스가 필요한 경우 POSIX 표준 API를 이용하여 운영체제와 직접 연결되고, 운영체제와의 직접 연결이 불가능한 경우에는 적응 레이어를 통해 서비스를 받게 된다.

이러한 OACE 환경에서 사용되는 소프트웨어 컴포넌트 설계 시에는 컴포넌트의 결합도와 입도를 고려하여 설계해야 한다[1]. 결합도는 컴포넌트 사이의 종속도(Dependency)와 관련된 사항으로 OACE 환경에서는 프로그램 간의 명확한 메시지 전달이나 분산 객체 호출 기법을 통하여 컴포넌트 간의 종속성을 최소화시킨다. 입도(Granularity), 즉 입자의 크기란 컴포넌트의 상대적인 크기나 규모, 디테일 정도를 의미하는 것으로써[5], 입자의 크기가 중요한 이유는 컴포넌트의 재사용성 및 성능의 상관관계 때문이다. 이러한 결합도와 입도를 고려한 적절한 컴포넌트 설계를 통하여 소스코드의 재사용성을 높일 수 있으며, 안정적인 컴포넌트 인터페이스를 제공함으로써 시스템의 전문성, 확정성에 대한 증진 및 컴포넌트 버전의 최적화를 가능하게 한다.

4. OACE 구성요소 분석

4.1 OACE 핵심 구성요소

OACE는 물리 매체와 컴퓨팅 자원, 운영체제, 미들웨어 등 총 13가지의 요소로 구성되어 있으며, 이 중 가장 중요한 역할을 수행하는 것은 운영체

제, 미들웨어, 정보관리, 정보보호 및 프로그래밍 언어 등 5가지이다. 이들 핵심 구성요소에 대한 개념도를 <그림 3>과 같이 제시할 수 있다.



<그림 3> OACE 주요 구성요소

사용자 애플리케이션은 적용/분산 미들웨어를 통해 데이터 교환 및 운영체제와 연결되며, 정보관리 서비스는 컴퓨팅 작업 전반에 대한 데이터 제공 및 관리 역할을 수행한다. 이러한 모든 작업에 대한 보안성은 정보보호 컴포넌트에 의해 제어되며, OACE 기반의 모든 소프트웨어는 C++과 자바 프로그래밍 언어로 구현된다.

4.2 운영체제

일반 임베디드 시스템을 위한 운영체제의 특성과 비교해서 OACE에 적용하기 위한 운영체제의 가장 큰 특징은 스케줄러의 목표에 있다. 일반 임베디드 시스템은 프로세스에 대한 공정한 컴퓨팅 자원 할당과 CPU 액세스 저하 방지에 주안점을 두지만, OACE 운영체제에서는 처리 마감 시간에 대한 준수가 가장 중요한 사항이다. 또한 mission-critical 임무의 정확한 수행을 위해 실시간 성능의 구현이 매우 중요하다. 이는 실시간 스케줄링과 프로세스 및 스레드에 대한 우선순위 지정 알고리즘을 통해 구현되어야 한다.

OACE 운영체제는 POSIX 표준의 준수를 통해 애플리케이션에 대한 플랫폼 간 재사용을 극대화하고, COTS 소프트웨어 제품의 도입을 최대화한다. POSIX 표준을 적용하지 않은 운영체제를 사용한 레거시 시스템의 경우 적용 미들웨어를 이용하여 POSIX 표준 API를 제공하고 있다.

그리고 커널 수준 스레드의 지원을 통해 프로세스의 작업대기로 인한 스레드 전체의 블로킹(Blocking)이나 멀티프로세서 환경에서의 성능 저하를 최소화하고 있다.

4.3 미들웨어

OACE에서는 운영체제 추상화를 위한 적용 미들웨어와 소프트웨어 컴포넌트 간의 연결 및 커뮤니케이션을 위한 분산 미들웨어가 사용된다.

이전을 위한 분산 미들웨어가 사용된다.

적용 미들웨어는 운영체제의 상부에 위치하며 운영체제에 대한 추상화를 통해 각기 다른 운영체제를 동일한 인터페이스나 API를 제공하는 것처럼 만든다. 이를 통해 애플리케이션을 운용시스템으로부터 분리시키며 응용 애플리케이션에 대한 이식성을 증가시킨다. 이를 위해서는 미들웨어가 표준 기반의 운영체제 API에 문법적, 의미적으로 유사한 인터페이스를 제공해야 하며, 다양한 종류의 운영체제 및 컴퓨팅 자원 환경에서 사용될 수 있어야 한다.

현재 적용 미들웨어에 대한 표준은 지정되어 있지 않으며 운영체제에 대한 POSIX 표준 API를 제공할 것을 명시하고 있다.

분산 미들웨어는 두 개 이상의 소프트웨어 컴포넌트나 시스템 사이에 위치하여 컴포넌트 간 연결 및 통신을 담당하는 소프트웨어 유형으로 높은 수준의 시스템 추상화를 통해 시스템 설계를 단순화하고 프로그래밍 에러를 감소시킨다. OACE에서는 분산 객체, Pub/Sub, 병렬 데이터 프로세싱 등 세 가지 종류의 분산 미들웨어 기술이 사용되며, 이에 대한 세부적인 분석내용은 <표 1>과 같다.

<표 1> OACE 적용 분산 미들웨어

구분	내용
분산 객체	-클라이언트 객체는 ORB(Object Request Broker)를 이용하여 분산환경 내 다른 객체의 동작을 호출하고, 서버는 이를 처리 후 결과를 돌려주는 방식 -네트워크 내의 다른 장소에 위치하는 객체의 메소드 호출을 통한 정보 교환을 허용
Pub/Sub	-특정 이벤트에 대한 공지를 통한 상호작용 방식 · 클라이언트: 관심 이벤트/메시지를 시스템에 등록 · 서비스: 주어진 타입의 메시지 및 이벤트를 발생 · 이벤트 발생 시 시스템은 등록 테이블을 검색 후 해당 이벤트에 관심 있는 클라이언트에 이를 전송
Data Parallel	-다수의 병렬 프로세서 간 백본을 통한 통신 방식 -신호처리와 같은 병렬처리 애플리케이션에서 사용

현재 업계에서 가장 많이 사용하고 있는 분산 객체 표준으로는 OMG 그룹의 CORBA와 마이크로소프트의 DCOM, 자바 RMI가 있으며 Pub/Sub에서는 DDS, 병렬 데이터 프로세싱에서는 MPI가 있다. 이 중 DCOM은 가장 높은 시장 점유율을 보이고 있으나[3] 높은 벤더 종속성과 낮은 실시간 성능으로 인하여 OACE 표준에는 채택되어 있지 않으며 다른 표준들은 OACE에 적용된다.

4.4 정보관리

정보관리는 OACE에서 서비스로 제공되고 있으며 ①데이터에 대한 생성, 읽기, 갱신 및 삭제

포함한 객체 및 데이터 수명주기 관리 ②트랜잭션 및 버전 관리 ③다수 애플리케이션에 대한 데이터 동시 접근 관리 등을 수행한다.

주요 OACE 적용 정보관리 표준으로는 ISO SQL과 JDO(Java Data Object) 및 JDBC(Java Database Connectivity)의 세 가지가 있으며 이에 대한 분석 내용은 <표 2>와 같다.

<표 2> OACE 적용 DBMS 분석

구분	내용
ISO SQL	-데이터베이스에서 정보를 얻거나 갱신하기 위한 표준화된 언어 -대화형, 또는 프로그램 내에 삽입하여 사용 가능
JDO	-자바 커뮤니티 프로세스에서 개발한 표준 인터페이스 기반의 자바 모델 추상화 기술 -동일한 애플리케이션 프로그램 인터페이스를 통해 객체지향/SQL 기반 데이터 저장소 제공
JDBC	-자바로 작성된 프로그램을 일반 데이터베이스에 연결하기 위한 응용프로그램 인터페이스 규격 -자바 환경에서 관계형 DB에 대한 표준 인터페이스를 제공하며 일반 SQL 데이터 형식을 대부분 지원

4.5 정보보호

OACE에서 정보보호의 목표는 시스템 구축에 사용되는 각종 애플리케이션으로부터 데이터 및 컴퓨팅 자원을 보호함과 동시에 하위 레이어의 보안 기술을 숨기는 것에 있다[3]. 이를 위해 우수한 민간 기술의 적용과 데이터의 분할 관리를 수행하고 있다. 민간 기술의 경우 방화벽, 침입 탐지 등의 자산 보호와 관련된 기술이 사용되며, 데이터 분할 기술은 여러 데이터에 대한 다른 보안 수준으로의 분리를 위해 사용한다.

정보보호와 관련된 표준은 ISO와 IETF(Internet Engineering Task Force), NIST(Nation Institute of Standards and Technology) 등의 표준을 따르고 있으며, 이에 대한 세부 분석 내용은 <표 3>과 같다.

<표 3> 정보보호 관련 표준 분석

구분	적용 분야
ISO	- IT 시스템의 정보보호에 대한 전반적 표준
IETF	- 네트워크 장비에 대한 접속 제어 - 보안 서비스에 대한 프로그래밍 인터페이스 - 원격 접속 사용자에게 대한 접속 제어 - IP 패킷에 대한 보안 서비스 분야 - TCP 기반 애플리케이션에 대한 보안 메커니즘
NIST	- 암호화 및 인증 관련분야 · 데이터 모듈 암호화 · 메시지 인증 · 디지털 서명 표준

4.6 프로그래밍 언어

OACE 구축을 위해 사용되는 프로그래밍 언어는 Java와 C++ 및 Ada95의 세 가지 언어를 사용한다. Java는 JVM(Java Virtual Machine)에 근거하고 있는 프로그래밍 언어이자 플랫폼으로써, Java로 작성된 모든 프로그램은 JVM이 설치되어 있는 모든 시스템에서 실행이 가능하므로 플랫폼 및 운영체제에 대한 높은 이식성을 가지게 된다.

C++은 C언어의 확장판으로 만들어진 객체지향 언어로써 대규모 응용프로그램 작성에 있어 최적의 프로그래밍 언어로 간주되고 있다. OACE에 C++을 적용할 경우 컴파일러 및 라이브러리는 ANSI 및 ISO/IEC 표준에 순응해야 한다.

Ada95는 현재 운용 중인 레거시 시스템에 대한 애플리케이션 수정 등에만 사용할 수 있으며 이의 경우 컴파일러/라이브러리 등은 ISO/IEC 표준과 호환되어야 한다.

5. 결론

본 논문에서는 미 해군이 운영 중인 OACE의 기술 아키텍처 및 핵심 구성요소에 대한 세부적인 기술과 표준에 대해 분석하였다. 분석결과 OACE는 느슨한 결합을 통해 각 계층이 독립적으로 운영되고 있으며, 공개 표준을 도입함으로써 소프트웨어 컴포넌트의 유지보수성과 재사용성을 극대화하고 있음을 알 수 있다. 또한 OACE 핵심 구성요소들은 실시간 처리 성능과 공개 표준의 사용을 강조하고 있다.

향후 본 논문을 보다 발전시켜 한국군의 실정에 맞는 OA 및 OACE 구축의 기반연구로 활용될 수 있을 것이다. 또한 이를 통해 미래 전장환경에 부합하는 무기체계의 합동 전투능력 향상과 무기체계 도입에 따른 비용문제를 해결하는데 기여할 것으로 기대한다.

참고문헌

- [1] PEO IWS, *Open Architecture Computing Environment Design Guide Ver. 1.0*, USN, 2004.
- [2] James Shannon, *Naval Open Architecture Overview on OA*, PEO IWS, 2006.
- [3] PEO IWS, *Open Architecture Computing Environment Technologies and Standards Ver. 1.0*, USN, 2004.
- [4] Tom Strei, *Open Architecture*, PEO IWS, 2004.
- [5] Wayne Wolf, *High-Performance Embedded Computing: Architecture, Application, and Methodologies*, Elsevier, 2007.