

CBD를 활용한 웹 기반 분산 시스템 연구¹

유진무* , 조병인**

*엠앤디정보기술(주), **국방과학연구소

A Study On Web Based Distributed System Using Component Based Development

Yoo Jin-Moo* , Cho Byung-In**

*M&D Information Technology Co., **Agency for Defense Development,

E-mail : jayuin@mndinfo.com, chobyun@dreamwiz.com

요 약

CBD(Component Based Development)가 소프트웨어 개발의 새로운 패러다임으로 주목을 받으면서 최근 수년간 CBD를 적용한 소프트웨어 개발이 증가하고 있다. 국방과학연구소에서는 국방 소프트웨어 개발을 위한 표준 방법론으로 제시하고자 '국방 CBD 방법론'을 개발하고 있다. 본 논문에서는 '국방 CBD 방법론'에 대하여 소개한다. 방법론의 개발 프로세스 및 특징을 설명하고, 국방 정보체계 개발에 적용되고 있는 국제 표준과의 연관성을 설명한다. 또한 방법론의 적용 사례로서 시범 개발한 상황보고 저작기를 소개한다. 웹 환경에서 컴포넌트 기반의 분산 시스템으로 구축된 상황보고 저작기의 아키텍처에 대해 설명하고, 웹 서비스 기술의 활용 사례를 설명한다.

1. 서론

CBD(Component Based Development)가 소프트웨어 개발의 새로운 패러다임으로 주목을 받으면서 최근 수년간 해외의 컴포넌트 기반 개발 방법론(이하 CBD 방법론)이 국내에 도입되어 소프트웨어 개발에 적용되고 있으며, 기업체 및 연구소에서 자체적으로 CBD 방법론을 개발하는 사례도 증가하고 있다. 국가 정보화 정책 차원에서 컴포넌트 산업을 육성하고 있으며, 국방 분야에서도 CBD에 대한 요구가 증가하고 있다.

국방부는 국방정보체계의 상호 운용성 증진을 위해 공동운용환경(COE)의 구축을 추진하고 있으며[1], 재사용성 증대를 위해 CBD를 권장하고 있다. 즉 국방정보체계 개발에 있어서의 중복 투자 위험성 감소 및 유지보수성 증대를 통한 비용 감소를 달성하기 위한 방법으로서 공동운용환경 하에서 CBD의 활용을 권장하고 있다. 현재 C4I 체계 등 국방정보체계 구축사업에서 CBD를 활발하게 적용하고 있으나 CBD 방법론은 각 사업별로 선택하여 적용 중이며, 국방부 차원에서 컴포넌트 기반으로 체계를 개발하는 표준 방법론은 정해지

¹ 본 연구는 2003년 국방과학연구소의 '핵심 서비스 컴포넌트 시범개발 용역 사업'의 지원으로 수행됨

지 않은 상황이다. 이에 따라 CBD를 적용하여 개발된 소프트웨어 컴포넌트 및 산출 문서의 재사용성을 극대화하기 위해서는 CBD 방법론의 표준을 제정할 필요성이 부각되었다. 이러한 배경에서 국방과학연구소는 국방 분야의 소프트웨어 개발 사업에 적용될 표준 방법론으로 활용하고자 ‘국방 CBD 방법론’을 개발하였다.

본 논문에서는 ‘국방 CBD 방법론’과 이 방법론을 적용하여 시범 개발된 상황보고 저작기에 대해 소개하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 ‘국방 CBD 방법론’의 특징 및 개발 프로세스에 대해 소개하며, 군에서 보편적으로 사용되고 있는 표준인 ISO 12207과 MIL-STD-498과 ‘국방 CBD 방법론’과의 연관성에 대하여 설명한다. 3장에서는 ‘국방 CBD 방법론’을 적용하여 시범 개발한 상황보고 저작기에 대하여 소개한다. 상황보고 저작기의 기능 및 아키텍처를 설명하고 웹 환경에서 분산 시스템 구축을 위해 적용된 기술에 대하여 설명한다. 끝으로 4장에서 결론을 기술하고 본 논문을 마무리하고자 한다.

2. 국방 CBD 방법론

‘국방 CBD 방법론’이 국방 소프트웨어 개발을 위한 표준 방법론으로 지정되어 국방정보체계 개발 사업에 이 방법론이 적용되면, 향후 국방정보체계들의 재사용성을 최대로 높이고, 기 개발된 컴포넌트를 활용한 조립 생산이 가능하게 될 것으로 생각한다.

국방 CBD 방법론은 2002년도 1년 동안 국방과학연구소를 중심으로 연구과제를 통해 초안이 완료되었고[2], 2003년부터 상황보고 저작기를 시범 개발하면서 초안으로 작성된 국방 CBD 방법론의 개발 프로세스 전반에 대한 보완 및 추가 작업을 수행하였다. 이로써 ‘국방 CBD 방법론’의 실용성을 검증하고, 완성도를 높일 수 있었다. ‘국방 CBD 방법론’은 설명서, 기법서, 산출물 양식, 적용 사례를 포함하도록 개발하였다.

본 논문에서 소개하는 ‘국방 CBD 방법론’은 현재 개발 완료 직전에 있으므로, 최종 ‘국방 CBD 방법론’과 동일할 것으로 예상되지만 일부 내용이 변경될 가능성은 배제할 수 없다. 현재까지 개발된 ‘국방 CBD 방법론’을 소개하면 다음과 같다.

2.1 특징

‘국방 CBD 방법론’을 개발하기 위하여 먼저 국내외의 다양한 CBD 방법론의 장단점을 비교 분석하였다. 위의 분석 결과 및 국방 분야의 특성을 수용하고 최근 방법론의 동향을 반영하여 방법론을 개발하였다. 이를 통해 실용적이고 유연하며, 미래 지향적인 방법론이 되도록 노력하였다. ‘국방 CBD 방법론’의 대표적인 특징은 다음과 같다.

- 국제 표준인 ISO/IEC 12207[3]의 개발자 공정 기반의 프로세스 정의
- 아키텍처 중심의 개발 방법론
- UML 기반의 산출물 작성 표기법 사용
- MDA 개념 도입으로 플랫폼 독립적인 작업과 플랫폼 종속적인 작업의 분리
- XP의 테스트 우선 프로그래밍 기법 적용

2.2 국방 CBD 방법론 개발 프로세스

‘국방 CBD 방법론’의 개발 프로세스는 단계, 활동, 작업으로 구성된다. 그리고 각 작업을 수행하기 위한 세부 절차와 작업의 결과인 산출물을 정의하였다. 본 방법론은 아래 표와 같이 4개의 단계와 12개의 활동 38개의 작업으로 구성된다.

단계 (Phase)	활동 (Activity)	작업 (Task)
분석IR	요구 사항 정의 IR1	상위 요구사항 정의 (IR11)
		도메인 모델링 (IR12)
		현행 시스템 분석 (IR13)
		비즈니스 모델링 (IR14)
		요구사항 명세 (IR15)

	아키텍처 정의 IR2	소프트웨어 아키텍처 정의(IR21)
		시스템 아키텍처 정의 (IR22)
		표준 지침 수립 (IR23)
	요구 사항 분석 IR3	유스케이스 모델링 (IR31)
		사용자 인터페이스 프로토타이핑 (IR32)
		클래스 모델링 (IR33)
설계2D	개략 설계 2D1	요구 사항 테스트 설계 (IR34)
		컴포넌트 식별 (2D11)
		컴포넌트 획득 방법 식별 (2D12)
		인터페이스 상호작용 정의(2D13)
		컴포넌트 명세 (2D14)
		사용자 인터페이스 설계(2D15)
	상세 설계 2D2	데이터 모델링(2D16)
		컴포넌트 내부 설계(2D21)
		컴포넌트 구현 설계(2D22)
		사용자 인터페이스 구현 설계 (2D23)
		데이터 베이스 설계 (2D24)
구현 및 테스트 3T	테스트 준비 3T1	테스트 계획(3T11)
		단위 테스트 설계 (3T12)
	구현 3T2	데이터 베이스 구축(3T21)
		컴포넌트 구현 및 테스트(3T22)
		사용자 인터페이스 구현 (3T23)
		단위 테스트 수행(3T24)
	통합 테스트 3T3	통합 테스트 설계(3T31)
		통합 테스트 수행(3T32)
	시스템 테스트 3T4	시스템 테스트 설계(3T41)
		시스템 테스트 수행(3T42)
	지침서 작성 3T5	사용자 지침서 작성(3T51)
		운용자 지침서 작성(3T52)
인도 4S	시스템 설치 4S1	시스템 설치 계획(4S11)
		시스템 설치(4S12)
	인수 지원 4S2	인수 테스트 지원(4S21)
		사용자 교육(4S22)
총 4단계	총 12 활동	총 38작업

2.3 표준과의 연관성

국방정보체계 개발에 있어서 가장 폭넓게 사용되는 표준이 ISO/IEC 12207과 MIL-STD-498[4]이다. ISO/IEC 12207은 소프트웨어 생명주기 공정에 대한 국제 표준으로서 개발 공정뿐만 아니라 획득 공정, 공급 공정, 운용 공정, 유지보수 공정 등의 기본 공정과 다양한 지원공정, 조직공정을 정의하고 있다. ‘국방 CBD 방법론’은 ISO/IEC 12207의 개발 공정을 기준으로 프로세스를 개발하였다. ‘국방 CBD 방법론’과 ISO/IEC 12207 개발 공정의 관계를 활동 수준에서 비교하면 아래 표와 같다. 활동명과 함께 ()안에 기록된 것은 ISO/IEC 12207 표준 문서의 해당되는 목차이다.

단 계	활 동	ISO/IEC 12207 개발 공정의 활 동
분석	요구사항 정의	시스템요구분석(5.3.2)
	아키텍처 정의	시스템 구조설계(5.3.3)
	요구사항 분석	소프트웨어 요구분석(5.3.4)
설계	개략 설계	소프트웨어 구조설계(5.3.5)
	상세 설계	소프트웨어 상세설계(5.3.6)
구현 및 테스트	테스트 준비	소프트웨어 코딩 및 시험(5.3.7)
	구현	
	통합 테스트	소프트웨어 통합(5.3.8)
		소프트웨어 자격 시험(5.3.9)
	시스템테스트	시스템 통합 (5.3.10)
		시스템 자격 시험(5.3.11)
인도	지침서 작성	소프트웨어 구조설계(5.3.5)
	시스템 설치	소프트웨어 설치(5.3.12)
	인수 지원	소프트웨어 수락 지원(5.3.13)

‘국방 CBD 방법론’은 각 작업의 결과로 생성되는 41개의 산출물을 정의하였다. 다음 표는 ‘국방 CBD 방법론’과 MIL-STD-498 산출물의 관계를 나타낸 표이다. 즉 MIL-STD-498을 기준으로 산출물을 제출하고자 할 경우 ‘국방 CBD 방법론’의 각 산출물들이 MIL-STD-498의 어떤 산출물에 포

함되는지를 나타내고 있다.

작업		
상위 요구사항 정의	상위요구사항 정의서	SRS
도메인 모델링	도메인 명세서	SRS
	용어집	SRS
현행시스템 분석	현행 시스템 분석서	SRS
비즈니스 모델링	비즈니스 정의서	SRS
요구사항 명세	요구사항 명세서	SRS
소프트아키텍처 정의	소프트웨어 아키텍처 정의서	SSDD
시스템 아키텍처 정의	시스템 아키텍처 정의서	SSDD
표준 지침 수립	표준 지침서	SSDD
유스케이스 모델링	유스케이스 명세서	SRS, IRS
사용자 인터페이스 프로토타이핑	초기 사용자 인터페이스 설계서	SRS
클래스 모델링	클래스 명세서	SRS
요구 사항 테스트 설계	요구사항 테스트 설계서	STD
컴포넌트 식별	컴포넌트 목록	SDD, IDD
	컴포넌트 아키텍처 정의서	SDD
컴포넌트 획득 방법 식별	컴포넌트 획득방법 식별서	SDD
인터페이스 상호작용 정의	인터페이스 상호작용 명세서	SDD
		IDD
컴포넌트 명세	인터페이스 명세서	SDD
	컴포넌트 명세서	SDD
사용자 인터페이스 설계	사용자 인터페이스 설계서	SDD
		SDD
데이터 모델링	데이터 설계서	DBDD
컴포넌트 내부 설계	컴포넌트 설계서	SDD
	트랜잭션 정의서	SDD

컴포넌트 구현 설계	컴포넌트 구현 설계서	SDD, IDD
	트랜잭션 정의서(보완)	SDD
사용자 인터페이스 구현 설계	사용자 인터페이스 구현 설계서	SDD
데이터베이스 설계	데이터베이스 설계서	DBDD
테스트 계획	테스트 계획서	STP
단위 테스트 설계	단위 테스트 설계서	STD
데이터 베이스 구축	물리적 데이터베이스	N/A
컴포넌트 구현 및 테스트	테스트 코드	N/A
	컴포넌트 코드	
	단위테스트 결과서	STR
사용자 인터페이스 구현	사용자 인터페이스 코드	N/A
단위 테스트 수행	단위 테스트 결과서(보완)	STR
통합 테스트 설계	통합 테스트 설계서	STD
통합 테스트 수행	통합 테스트 결과서	STR
시스템 테스트 설계	시스템 테스트 설계서	STD
시스템 테스트 수행	시스템통합테스트결과서	STR
사용자 지침서 작성	사용자 지침서	SUM
운용자 지침서 작성	운용자 지침서	SCOM
시스템 설치 계획	시스템 설치 계획서	SIP
시스템 설치	시스템 설치 보고서	N/A
인수테스트 지원	-	N/A
사용자 교육	교육 보고서	N/A

3. 상황보고 저작기

방법론 개발은 그 업무 특성상 이론적인 면이 강조되어 실용성이 저하되는 경우가 발생할 수 있다. 이에 대한 보완책으로 ‘국방 CBD 방법론’ 개발팀은 방법론의 개발과 함께 개발된 방법론을 적용한 시범 체계의 개발을 병행하였다. 이를 통해 방법론의 문제점 도출 및 타당성 검증이 가능하였다. 방법론 개발과 시범체계 개발간의 반복적인 피드백을 통해 방법론의 실용성과 완성도를 최대

한 높이고자 하였다.

시범 체계는 군에서 가장 수요가 많은 소프트웨어 중의 하나인 상황도 도시 소프트웨어를 대상으로 하였다. C4I 체계의 핵심 소프트웨어인 상황도 도시 소프트웨어의 전체 기능 중 주요 기능을 선별하여 개발 범위를 정하였다. 시범 개발 소프트웨어는 선별된 상황도 도시 기능을 활용하여 상황보고를 효과적으로 할 수 있는 소프트웨어로서 ‘상황보고 저작기’로 명명되었다. 본 장에서는 상황보고 저작기의 기능 및 소프트웨어 아키텍처에 대하여 설명하고, 웹 서비스를 적용하여 분산 시스템을 구축한 내용에 대하여 설명한다.

3.1 기능

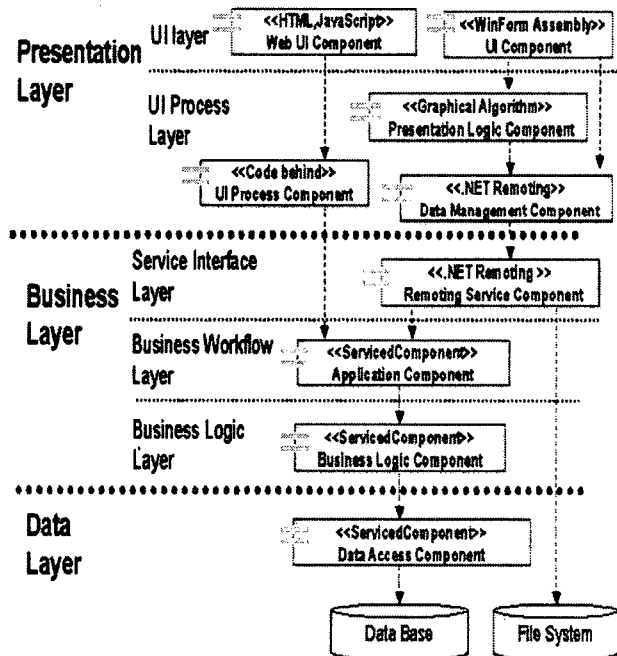
상황보고 저작기는 컴퓨터를 이용하여 상황도 도시 및 상황 보고를 지원하는 기능을 제공한다. 상황도는 지휘관 및 참모가 수행해야 하는 전장기능업무를 수행하는데 있어서 판단 및 결심지원을 위해 필요한 각종 정보를 그래픽 및 상황표를 활용하여 유지관리하며, 응용에서 구축된 데이터베이스와 연계하여 상황도를 작성하고 관리할 수 있다. 또한 작전계획 수립을 도식화하고, 투명도형 명령을 작성/전파하기 위해 투명도를 작성하고 관리하는 기능을 제공한다. 그리고 이와 같은 업무를 지원하기 위한 기능으로 벡터 지도 및 래스터 지도 도시 기능, 군사부호 작성 기능을 제공한다. 이와 같은 상황도 도시 기능과 함께 브리핑 문서 목록 관리 등 상황 보고를 지원하는 기능을 제공한다.

3.2 소프트웨어 아키텍처

국내외에서 아키텍처 중심의 소프트웨어 개발에 대한 연구가 활발히 진행되고 있다.[5] IEEE 1471 [6]에서는 아키텍처를 다음과 같이 정의하고 있다. “아키텍처는 시스템의 기본 구조로서 시스템의 구성 요소와 그들 간의 관계로 시스템을 구체화하는 것이며, 시스템의 설계와 진화를 지배하는 원칙이다”. ‘국방 CBD 방법론’은 아키텍처 중심의

방법론이다. ‘국방 CBD 방법론’ 개발 프로세스의 아키텍처 정의 활동에서 정의되는 아키텍처는 설계 및 구현의 청사진의 역할을 한다. 아키텍처는 시스템에 대한 다양한 관점을 일관성 있고 이해하기 쉽게 표현하는 수단으로, 프로젝트 이해 당사자 간의 원활한 의사 소통을 통해 시스템 개발의 효율성을 증대시킨다.

상황보고 저작기의 소프트웨어 아키텍처를 정의하기 위해서 다음과 같은 사항이 고려되었다. 재사용성을 위하여 컴포넌트를 기반으로 개발되어야 한다는 것이 전제 조건이었다. 또한 일관된 인터페이스를 제공하고, 프로그램의 변경 관리의 용이성을 위하여 웹 환경에서 운용되도록 요구되었으며, 트랜잭션 처리도 중요한 설계 고려 사항이었다. 그리고 실행환경은 .NET 플랫폼이고, DBMS는 MS SQL Server로 정하였다.



위의 그림은 상황보고 저작기의 소프트웨어 아키텍처이다. 상황보고 저작기의 소프트웨어 아키텍처는 레이어 아키텍처 패턴[7]을 사용하여 설계하였다. 상황보고 저작기는 크게 3개의 레이어로 아키텍처를 구성하였다. 사용자와의 상호 작용

기능을 하는 프리젠테이션 레이어를 두었고, 비즈니스 기능을 제공하는 비즈니스 레이어를 두었다. 마지막으로 데이터 레이어는 데이터 저장소에 접근하여 데이터를 검색 및 저장하는 역할을 하도록 하였다. 그리고 각 레이어에 속하는 컴포넌트의 유형을 설계하였다.

비즈니스 레이어는 비즈니스의 흐름을 제어하는 계층인 비즈니스 워크플로우 레이어와 비즈니스 로직을 처리하는 비즈니스 로직 레이어로 다시 분리하였다. 또한 프리젠테이션 레이어도 사용자와의 상호 작용을 다루는 UI 레이어와 프리젠테이션 차원에서의 프로세스 흐름 및 로직 처리를 담당하는 컴포넌트로 구성된 UI 프로세스 레이어로 분류하였다.

본 연구에서 개발한 상황보고 저작기는 프리젠테이션 레이어 컴포넌트가 많은 기능을 포함하고 있는 특성이 있다. 즉 상황보고저작기는 지도 처리 및 투명도 도식 등 사용자에게 정보를 제공하고, 사용자의 입력을 통해 정보를 얻기 위한 방법으로, 복잡한 그래픽 처리가 필수적으로 요구된다. 이에 따라 프리젠테이션 레이어의 구성은 일반적인 웹 기반 시스템과는 차이가 있는 아키텍처로 설계하였다. HTML기반의 사용자 인터페이스 뿐만 아니라, 복잡한 그래픽처리를 위해 윈도우 폼 형태의 사용자 인터페이스가 요구되었다. 이에 따라 UserControl 클래스를 상속 받아 사용자 인터페이스 컴포넌트를 만들고, 이를 HTML의 Object 태그에 포함시켜 웹 브라우저에 윈도우 폼 형태로 삽입되도록 구성하였다. 또한 상황보고저작기의 특성상 프리젠테이션 레이어의 컴포넌트가 재사용이 요구되는 다양한 기능을 포함하고 있으므로, 그래픽처리 알고리즘 등 프리젠테이션 레이어에서 발생하는 복잡한 로직을 재사용 컴포넌트로 구성하도록 설계하였으며, 이러한 컴포넌트 유형을 프리젠테이션 로직 컴포넌트라 칭하였다.

상황보고 저작기는 웹 브라우저 상에 포함되어 클라이언트에서 실행되는 컴포넌트와 서버단의

비즈니스 컴포넌트 간의 데이터 교환을 필요로 한다. 웹 클라이언트와 서버간에 교환되어야 할 데이터의 종류는 문자 및 숫자 등의 기본 자료형 뿐만 아니라, 이미지 객체 및 투명도 객체 (GraphicsPath의 ArrayList)와 같은 복잡한 객체 타입도 포함한다. 이러한 데이터 교환을 효율적으로 수행하기 위해 .NET 리모팅 서비스를 활용하였다. 프리젠테이션 레이어의 데이터 관리 컴포넌트와 서비스 인터페이스 레이어의 원격 서비스 컴포넌트는 실행 시에 서로 다른 컴퓨터에서 동작하면서 데이터를 교환한다. .NET 리모팅 서비스는 원격 객체의 메소드를 호출하는 분산 컴포넌트 간의 통신 기술이다.

트랙잭션 처리를 위해 비즈니스 워크 플로우 레이어의 어플리케이션 컴포넌트에서 시작하여 비즈니스 로직 레이어와 데이터 레이어의 컴포넌트에서는 COM+에서 제공하는 서비스인 자동 트랜잭션 처리를 사용하기 위해 ServicedComponent 클래스에서 파생된 클래스로 구현하였다.

컴포넌트의 식별은 클래스 다이어그램에서 핵심 객체를 추출하는 방식을 기본으로 하고, 소프트웨어 아키텍처를 기준으로 필요한 컴포넌트를 식별하는 방식을 병행하여 사용하였다. 핵심 객체를 통해 컴포넌트를 식별하고, 이를 아키텍처에서 설정한 컴포넌트 유형에 따라 배치하고, 컴포넌트 유형 중 식별된 컴포넌트가 없을 경우, 이를 추가적으로 식별하였다. 결과적으로 각 컴포넌트 유형별로는 한 개 이상의 컴포넌트를 식별하였다. 상황보고 저작기의 컴포넌트를 식별한 결과 프리젠테이션 로직 컴포넌트, 비즈니스 로직 컴포넌트, 데이터 액세스 컴포넌트는 모두 핵심 객체 추출을 통해서 컴포넌트가 식별되었다. 이 3가지 유형의 컴포넌트가 실질적인 재사용 대상 컴포넌트로 평가되었다. 그리고 나머지 유형의 컴포넌트는 대부분 소프트웨어 아키텍처를 통해서 추가적으로 식별되었다.

3.3 분산 시스템 구축

분산 시스템이란 네트워크로 연결된 다수의 독립적인 컴퓨터들이 분산 처리를 수행하는, 즉 서로 일을 분담하여 작업을 처리하는 시스템을 말한다. 상황보고 저작기에 적용한 분산 컴퓨팅 기술에 대해 설명하기에 앞서 객체지향 패러다임 등장 이후의 소프트웨어 기술의 진화 과정을 간략하게 살펴보도록 한다.[8]

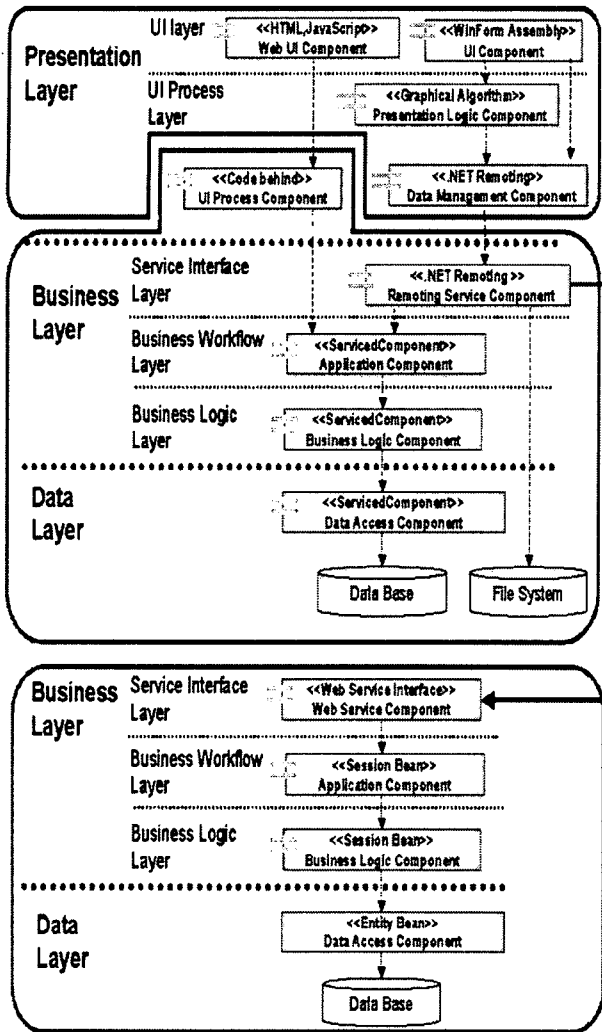
소프트웨어 위기라고 불릴 만큼 소프트웨어 개발의 생산성 향상은 중요한 쟁점이었다. 객체지향 프로그램의 재사용 개념은 프로그램 개발의 생산성 향상에 일조를 하였다. 그러나 객체지향 프로그램에서의 재사용은 소스 코드에 대한 재사용이라는 한계를 가지고 있었다. 이에 따라 소스 코드 수준이 아닌 바이너리 수준에서의 재사용이 요구되었고, 재사용 가능한 바이너리 객체인 컴포넌트라는 개념을 생각하게 되었다. 자바 빈, COM은 이러한 개념을 기반으로 발전된 기술이다. 그리고 네트워크 기술의 발전에 따라 원격지에 존재하는 컴퓨터 간의 연동을 위해 분산 컴퓨팅 기술이 대두되게 되었으며 대표적인 기술로 RPC를 들 수 있다. 분산 컴퓨팅 기술과 컴포넌트 개념의 접목에 따라 발전된 기술이 분산 컴포넌트 기술이다. 이 기술의 예로 EJB, RMI, DCOM, CORBA 등을 들 수 있으며, 이들은 RPC 방식을 기반으로 한다. 이러한 다양한 분산 컴포넌트 모델의 존재로 인하여 현재의 분산 시스템은 특정 언어, 플랫폼, 컴포넌트 모델에 의존적으로 개발되고 있는 실정이다. 즉 서로 다른 환경에서 개발된 컴포넌트 간 연동의 어려움으로 인해 동일 기능의 중복 개발이 이루어지고 있다.

이와 같은 배경에서 특정 언어 및 플랫폼에 독립적인 분산 컴포넌트 기술로서 W3C에서 표준으로 정한 기술이 웹 서비스 기술이다. 이 기술의 표준화 작업에 MS, IBM, SUN, SAP와 같은 주요 IT업체가 적극적으로 참여하고 있으므로, 웹 서비스는 실용적인 표준으로서 자리매김할 것으로 기

대를 모으고 있다. 웹 서비스는 HTTP와 XML을 기반으로 서비스 요청 및 결과 수신 방법을 정의하여 이기종 플랫폼의 분산 컴포넌트 간 연동을 가능하게 한다. 웹 서비스 기술은 다음과 같은 요소로 이루어진다. 서비스 등록과 검색을 담당하는 UDDI, 서비스를 기술(Description)하는 WSDL, 웹 서비스 사용자와 서비스간 서비스 호출과 결과에 대한 전송 방법을 정의 하는 SOAP, 전송 네트워크로서 HTTP의 활용이 웹 서비스에서 표준으로 정의하고 있는 주요 기술이다.

웹 서비스 기술은 시스템간의 상호 운용성 및 재사용성 극대화라는 장점으로 인해 향후 분산 시스템 개발에 폭 넓게 활용될 것으로 생각한다. 컴포넌트 기반 개발의 진화 과정에 있어서 최근 기술인 웹 서비스를 활용한 개발 사례를 제시하는 것은 중요한 의미를 갖는 것으로 생각한다. 따라서 웹 서비스 기술을 활용하여 상황보고 저작기를 개발하였다. 3.2절에서 보여준 아키텍처는 모두 .NET 플랫폼에서 설계 및 구현된 것이다. 편의상 이를 ‘.NET 버전의 상황보고 저작기’라고 칭한다. 그리고 웹 서비스를 활용한 상황보고 저작기를 ‘웹 서비스 버전의 상황보고 저작기’라 하겠다. 웹 서비스 버전의 상황보고 저작기는 웹 서비스를 통해 .NET과 J2EE 플랫폼의 컴포넌트가 상호 연동이 되도록 구축하였다. .NET 버전의 상황보고 저작기의 비즈니스 레이어 컴포넌트 및 데이터 레이어 컴포넌트 중 상황도 도시 기능과 연관된 컴포넌트를 제거하였다. 그리고 동일한 기능의 컴포넌트를 J2EE 플랫폼에서 EJB로 구현하고 이들 컴포넌트의 인터페이스를 웹 서비스 컴포넌트를 통해 노출하였다. J2EE 플랫폼에서의 WAS (Web Application Server)는 Web Logic을 이용하였다. 이렇게 새롭게 개발된 웹 서비스 버전의 상황보고 저작기는 프리젠테이션 레이어는 모두 .NET기반에서 동작하며, 비즈니스 레이어 컴포넌트 및 데이터 레이어의 컴포넌트 중 상황도 도시 기능 관련 컴포넌트는 EJB 기반으로 나머지는 .NET 기반

으로 개발되어 상호 연동하여 실행하도록 구현하였다.



위의 그림은 웹 서비스를 활용한 상황보고 저작기의 소프트웨어 아키텍처를 표현한 그림이다. 굵은 실선으로 둘러싸인 블록은 실행 시에 물리적으로 분리된 플랫폼을 나타낸다. 상위 블록은 .NET기반의 웹 클라이언트 시스템이며, 중간 블록은 .NET기반의 서버 시스템이며, 하위 블록은 J2EE기반의 서버 시스템이다. 중간 블록과 하위 블록간의 연동, 즉 웹 서비스는 그림에서 보는 바와 같이 중간 블록의 원격 서비스 컴포넌트에서 하위 블록의 웹 서비스 컴포넌트를 호출(굵은 실선 화살표)하도록 구성하였다.

4. 결론

본 논문에서 소개한 ‘국방 CBD 방법론’이 국방 정보체계 개발을 위한 표준 개발 방법론으로 지정되어 향후 국방 분야 소프트웨어 개발 프로젝트에 폭 넓게 적용될 것으로 기대된다. ‘국방 CBD 방법론’과 ISO/IEC 12207 및 MIL-STD-498의 연관성은 기존 국방 정보체계 관련 규정과의 상충을 최소화하여 ‘국방 CBD 방법론’의 실용적인 적용에 도움을 줄 것으로 판단한다.

본 연구에서는 ‘국방 CBD 방법론’을 상황보고 저작기 개발에 시범 적용하였다. 이를 통해 방법론의 실용성을 검증하였고, 웹 서비스를 활용한 분산 시스템 개발 사례를 제시하였다. 본 논문에서 소개한 상황보고 저작기의 소프트웨어 아키텍처는 웹 기반의 분산 시스템을 구축하는데 활용될 수 있을 것이다. 또한 개발 과정에서 작성된 모든 산출물은 향후 ‘국방 CBD 방법론’을 적용할 때 참고할 수 있는 모범 사례로 활용되기를 기대한다.

[참고문헌]

- [1] 공동운영환경 구성요소 연구 및 프로토타입 개발, 국방과학연구소, 2002.12
- [2] 국방공동운영환경 컴포넌트 개발 절차, 국방과학연구소, 2003.2.
- [3] ISO/IEC 12207:1995-Information Technology Software Lifecycle Processes.
- [4] MIL-STD-498, Software Development and Documentation, 1994.12
- [5] 소프트웨어 아키텍처 실무(교육자료), 한국소프트웨어공학협회 아키텍처분과위, 2003.8.
- [6] IEEE Std 1471-2000, Recommended Practice for architectural description of Software-Intensive Systems.
- [7] F.Buschmam, Pattern-oriented Software Architecture - A System of Patterns, John-Wiley&Sons, 1996
- [8] 웹 서비스 실전 프로그래밍, 한빛 미디어, 2002.11.