

컴포넌트 기반의 무기체계 임베디드 소프트웨어 개발방법론 설계

Design of CBD Methodology for Weapon System Embedded Software

이종이¹, 윤희병²

¹ 서울시 은평구 국방대학교 전산정보학과
E-mail: jongee@yahoo.co.kr

² 서울시 은평구 국방대학교 전산정보학과
E-mail: hbyoon@kndu.ac.kr

요 약

본 논문은 컴포넌트 기반의 무기체계 임베디드 소프트웨어 개발방법론을 제안한다. 이를 위해 먼저 국방 소프트웨어 개발 관련 규정, 무기체계 임베디드 소프트웨어, 컴포넌트 기반의 개발 방법 그리고 임베디드 소프트웨어 개발방법의 특징을 바탕으로 기존 CBD 방법론을 분석한다. 그런 다음 개발방법론 설계 고려요소를 도출하고, 고려요소를 바탕으로 컴포넌트 기반의 소프트웨어 개발(CBSD)과 컴포넌트 개발(CD)에 대해 어떻게 설계를 할 것인가, 산출물간에 어떤 상호작용을 하는지를 설명하는 컴포넌트 기반의 무기체계 임베디드 소프트웨어 표준 개발방법론을 제시한다.

Key Words : 무기체계 임베디드 소프트웨어, CBD, 개발방법론

1. 서 론

국방 소프트웨어는 군사력 건설을 위한 소프트웨어와 군사력 유지를 위한 소프트웨어로 나누고, 군사력 건설을 위한 소프트웨어는 무기체계 임베디드 소프트웨어와 지휘통제체계 소프트웨어로, 군사력 유지를 위한 소프트웨어는 관리정보체계 및 사무자동화체계 소프트웨어로 구분하고 있다. 이는 다시 무기체계 임베디드 소프트웨어를 제외한 지휘통제체계, 관리정보체계, 사무자동화체계 소프트웨어는 자동화정보체계 소프트웨어로 분류하여 무기체계 임베디드 소프트웨어와 구분하고 있다[1].

국방부는 자동화정보체계 소프트웨어 개발에 있어서는 2005년 초 "국방 CBD 방법론"을 발표하고, "국방 아키텍처 프레임워크(MND-AF)와 CBD방법론 적용지시(2005.1.7)"에서 국방 SW 개발방법론을 CBD 방법론으로 채택하는 국방 CBD 방법론의 표준절차를 준수하도록 하였다. 하지만 자동화정보체계 소프트웨어와는 달리 무기체계 임베디드 소프트웨어 개발을 위한 방법론은 별도로 규정하고 있지 않다. 또한 응용 소프트웨어 개발을 위한 방법론인 국방 CBD 방법론을 무기체계 임베디드 소프트웨어 개발에 적용하기 위해서는 테일러링이

필요하다.

CBD 방법론은 공통으로 사용되는 컴포넌트의 확보 및 재사용으로 개발비용을 획기적으로 절감할 수 있으며, 개발 기간도 대폭 단축할 수 있는 장점을 갖고 있어 무기체계 임베디드 소프트웨어 개발에 적용한다면 비용 절감과 개발 기간을 단축할 수 있을 것이다.

따라서 본 논문에서는 무기체계 임베디드 소프트웨어의 상호운용성과 유지보수성(성능개량, 재사용성)을 향상시킬 수 있고 개발비용을 절감하고 개발기간을 단축하기 위해 컴포넌트 기반의 무기체계 임베디드 소프트웨어 개발시 표준으로 적용할 수 있는 방법론을 제시하고자 한다.

2. 관련연구

2.1 국방 소프트웨어 개발에 관한 규정/지침

소프트웨어 개발을 위해서는 국방 획득 업무를 담당하는 방위사업청 규정과 지침을 따라야 한다. 방위사업청의 "방위력 개선 사업관리 규정" 제 175 조 ③항에서 "무기체계 연구개발 사업을 추진함에 있어서 내장형 소프트웨어를 포함하는 경우에는 해당 무기체계의 연구 개발 사업절차를 적용하되, 내장형 소프트웨어의 개

발판리에 대하여는 소프트웨어 개발 프로세스에 관한 지침을 적용한다.”고 명시하고 있다[2]. 하지만 소프트웨어 개발 프로세스는 무기체계 임베디드 소프트웨어 개발 전 분야에 대한 지침을 제공하지 못하고 있어 무기체계 임베디드 소프트웨어 개발을 위한 개발방법론을 표준화 할 필요가 있다[3].

2.2 무기체계 임베디드 소프트웨어

국방 무기체계 분야에 사용되는 임베디드 소프트웨어를 무기체계 임베디드 소프트웨어라 할 수 있다. 무기체계는 하드웨어 구성항목(HWIC), 인터페이스 요구상세 그리고 소프트웨어 구성항목(SWIC)으로 구분되고, 소프트웨어 구성항목(SWIC)은 계층적으로 분할된 여러 개의 구성단위(CSC, Computer Software Components)로 구성된다[1]. 즉 무기체계 임베디드 소프트웨어는 컴포넌트 단위로 구성되어 있다. 이러한 무기체계 임베디드 소프트웨어의 개발 과정에서 고려해야 할 특징은 기술의 발전에 따라 유지 보수가 가능해야 하고, 유지 보수비용을 절감 할 수 있어야 하며, 거대하고 복잡한 소프트웨어를 효과적으로 개발 할 수 있으며 타 무기체계와 연동이 가능한 상호운용성을 보장해야 한다.

2.3 컴포넌트 기반 소프트웨어 개발방법론

컴포넌트 기반의 소프트웨어 개발 방법은 소프트웨어 개발 패러다임 진화의 최첨단에 위치하며 높은 품질의 소프트웨어를 신속하고 효과적으로 개발할 수 있는 방법론이다.

일반적인 컴포넌트 기반 개발은 대체로 요구 분석 및 컴포넌트 획득단계, 컴포넌트 기반 설계 단계, 컴포넌트 조립 단계, 시험 단계, 유지 보수 단계를 거친다[4]. 컴포넌트 기반 개발 방법은 크게 두 가지 형태로 나눌 수 있다. 하나는 컴포넌트 획득 방법 중의 한 방법인 컴포넌트 자체를 개발하는 컴포넌트 개발(CD)이고, 다른 하나는 이미 구축되어 있는 컴포넌트를 사용하여 소프트웨어를 개발하는 컴포넌트 기반 소프트웨어 개발(CBSD)이다[5].

2.4 임베디드 소프트웨어 개발방법론

임베디드 시스템 개발 프로젝트의 생명주기는 시스템 요구분석, 시스템 설계, 항목별 개발(하드웨어, 인터페이스, 소프트웨어), 시스템 통합 및 테스트, 평가 및 인수 검증, 유지보수 및 기능 첨가의 6단계로 나눌 수 있으며 각 단계에 대응하는 검증 절차들을 가진다[6][7]. 임베디드 시스템은 설계 단계에서 하드웨어 설계, 소프트웨어 설계가 동시 개념으로 병행 수행되어야 하며 개발자는 어떤 기능을 하드웨어로

해결해야 하고 어떤 기능을 소프트웨어로 해결해야하는지를 최우선적으로 결정해야 한다. 또한 동시설계 및 동시 진행으로 인해 이전과정으로의 피드백이 불가능하거나 많은 위험요소를 내포할 수 있어 위험 최소화를 위해 폐기형 프로토타이핑 기법을 적용하여 소프트웨어 기능을 점진적으로 최종버전까지 반복 개발하여야 한다.

3. 설계 고려요소 도출

3.1 국내·외 CBD 방법론 분석

가. 국방 CBD 방법론

사실상의 산업표준인 UML 기반의 CBD 방법론에 기초하여 분석, 설계 및 구현 산출물들이 표준화 된 응용분야 국방 표준 개발 방법론이다. 국방 CBD 방법론은 개발 절차를 IDEFO 표기법을 이용하여 사용자 지침서를 작성함으로써 개발방법론을 이해하기 쉽게 상세히 설명하고 있다.

또한 요구사항 정의 활동에서 용어집을 작성하여 공통된 용어를 사용하게 함으로써 원활한 문서 작성 및 의사소통을 가능하게 하고 있다. 하지만 국방 CBD 방법론은 하드웨어와 관련된 개발에 관해서는 언급하고 있지 않다.

국방 CBD 방법론에서 시스템과 관련된 작업은 도메인 모델링 작업, 현행 시스템 분석 작업 그리고 시스템 아키텍처 정의 작업으로 구성되어 있다. 도메인 모델링 작업은 시스템에 대한 이해를 쉽게 할 수 있도록 관심 시스템과 외부 객체와의 관계에 대해 기술하는 시스템 배경 모델 작업이며, 현행 시스템 분석 작업은 개발 대상 시스템에 대한 이해를 돕기 위해서 현행 시스템의 환경과 구성 등을 분석하는 작업이고, 마지막으로 시스템 아키텍처 정의 작업은 시스템을 구성하는 요소의 배치 상태를 정의 할 뿐, 시스템 단계에서 중요한 하드웨어 및 소프트웨어로 해결해야 할 부분을 우선적으로 결정하고 있지 않으며, 또한 하드웨어와 소프트웨어를 동시 설계하는 개념도 포함되어 있지 않다.

국방 CBD 방법론의 컴포넌트 개발 관련 작업을 살펴보면 컴포넌트 개발 과 컴포넌트 획득 방법은 정의하고 기존 컴포넌트의 재사용을 위한 가변성 구현 방안 설계에 대해 설명하고 있을 뿐 컴포넌트를 개발하는 단계에서부터 여러 시스템에서의 재사용을 고려하여 개발하고 있지는 않다.

나. 마르미-IV 개발방법론

마르미-IV 개발방법론은 1994년부터 개발되어 온 한국형 개발방법론인 마르미(Magic and Robust Methodology Integrated) 시리즈

중 임베디드 소프트웨어를 개발하는 프로젝트에 적용하기 위해 국내에서 처음으로 체계화된 제품계열 기반의 개발방법론이다. 마르미-IV 개발방법론에서는 전체 개발 프로세스 경로를 액티비티 다이어그램으로 표시하여 전체 프로세스를 이해하기 쉽게 하였으며, 전체 개발 프로세스 경로를 다시 개발 프로세스의 생명주기상 특징을 고려하여 6개의 개발 경로로 분류하여 각 개발 경로별로 적용하는 작업을 구분하여 설명하고 있다. 또한 영역 활동에서는 작업과 역할, 작업과 산출물 그리고 작업관련 기법도표를 작성하여 이해하기 쉽게 설명하고 있다.

전체 개발 프로세스 경로는 아키텍처 영역 활동에서 시스템 아키텍처 설계, 시스템 아키텍처 평가, 시스템 아키텍처 확정 작업에 대해 설명하고 있다. 컴포넌트 개발 관련 작업을 살펴보면 컴포넌트를 추출하고 컴포넌트간의 상관관계를 결정하며 컴포넌트 코드를 구현하고 테스트 하지만 필요한 컴포넌트를 구현하는 방법은 자세하게 설명하고 있지 않다. 또한 제품군 자산에서 해당 컴포넌트를 선정하는 작업은 있지만 컴포넌트를 개발 할 때 재사용성을 고려한 점은 없다.

나. DESS 개발방법론

임베디드 실시간 시스템을 위한 객체지향/컴포넌트 기반의 소프트웨어 개발방법론으로 고전적인 소프트웨어 개발방법론 중에서 잘 알려진 V모델 개발 사이클과 UP(Unified Process) 반복 점진적인 개발 방법을 종합한 개발방법론이다.

DESS 개발방법론의 실현 워크플로우의 활동들은 V&V 워크플로우의 Review 또는 테스트 활동을 통해 매 활동마다 다음활동으로 진행하기 전에 철저한 검증과 확인을 하고 있어 각 활동에서의 오류를 최소화하는 장점을 가지고 있다. 특히 시스템 디자인 활동에서는 시스템의 기능을 하드웨어로 할 것인지, 소프트웨어로 할 것인지를 구분하고 있고 소프트웨어 요구공학 이후 활동에서 소프트웨어 통합 활동까지는 하드웨어와의 병행 개발을 명시하고 있어 임베디드 소프트웨어 개발 특성을 충족하고 있다. 컴포넌트 개발 관련 활동을 살펴보면 컴포넌트 라이브러리에 존재하는 컴포넌트를 재사용하고자 하나 컴포넌트 개발시 여러 시스템에서의 재사용을 고려하여 개발하고 있지는 않다.

3.2 설계 고려요소 도출

3.1절의 분석결과를 토대로 컴포넌트 기반의 소프트웨어 개발(CBSD) 설계 고려요소로 IDEF0 표기법, 용어집, 컴포넌트 재사용, 각

활동마다 V&V, 임베디드 시스템 개발을 위한 모든 단계 이행을 도출할 수 있고 컴포넌트 개발(CD) 설계 고려요소로 용어집, 컴포넌트 개발 방법 제시와 보완요소로 여러 시스템에서의 재사용, 컴포넌트 구현 방법 설명을 도출할 수 있다.

4. 방법론 설계 및 평가

4.1 컴포넌트 기반의 소프트웨어 개발(CBSD)

제안하고자하는 개발방법론의 컴포넌트 기반의 소프트웨어 개발(CBSD) 프로세스는 방위사업청 지침인 소프트웨어 개발 프로세스에서 무기체계 임베디드 소프트웨어 개발에 적용할 6개 활동과 임베디드 소프트웨어 생명주기의 하드웨어/소프트웨어 역할 분담, 동시설계 및 재통합 그리고 임베디드 시스템 위험요소 최소화의 프로토타이핑, 피드백, 점진적 평가 기능과 3.2절에서 도출된 고려요소를 적용하여 설계하였다. 전체 프로세스는 그림 1과 같이 크게 시스템 요구분석/설계, 소프트웨어 요구분석/설계, 컴포넌트 요구분석/개발(획득) 그리고 통합의 4단계와 하드웨어(폐기형 가상, 실물) 프로토타이핑 단계로 구성하였다.

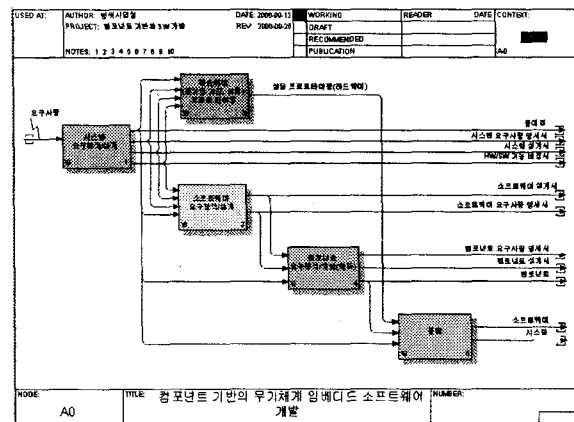


그림 1. CBSD 프로세스

방위사업청 지침인 소프트웨어 개발 프로세스에서 무기체계 임베디드 소프트웨어에 적용할 6개 활동 중 소프트웨어 요구사항 분석, 소프트웨어 구조설계 활동은 소프트웨어 요구분석/설계 단계에, 소프트웨어 상세 설계, 소프트웨어 코딩 및 단위시험은 컴포넌트 요구분석/개발(획득) 단계에, 마지막으로 소프트웨어 통합, 소프트웨어 개발 시험 평가 활동은 통합 단계에서 수행하도록 설계하여 상위 지침을 충족하도록 하였다.

임베디드 소프트웨어 생명주기 특징인 하드웨어/소프트웨어 역할 분담, 동시 설계 기능은

시스템 요구분석/설계 단계에서 하드웨어/소프트웨어 기능 배정서를 작성토록 하여 하드웨어적 기능은 하드웨어 프로토타이핑 단계에서, 소프트웨어적인 기능은 소프트웨어 요구분석/설계 단계에서 동시 설계하도록 하였다.

임베디드 시스템 위험요소 최소화의 프로토타이핑 기법, 점진적 평가를 위해 하드웨어 프로토타이핑 단계를 두었고, 소프트웨어 요구분석/설계 단계에서는 하드웨어 프로토타이핑과의 점진적 평가를 가능하도록 하였으며 피드백 기능은 각 단계의 활동마다 Review 또는 Testing 활동을 추가함으로써 해결하였고 하드웨어/소프트웨어 재통합 기능은 통합단계에서 수행하도록 설계하였다.

4.2 컴포넌트 개발(CD)

제안 개발방법론의 컴포넌트 개발의 핵심은 컴포넌트의 적용성과 재사용성을 높이기 위해 CBD 방법론의 컴포넌트 개발(CD) 절차와 컴포넌트를 한 시스템에서 보다는 여러 시스템(제품군)에서의 재사용을 목표로 제품 계열 중심의 핵심자산 개발 방법, 마르미-IV 개발방법론의 자산 개발 프로세스를 도입한 것이다. 즉 국방 도메인의 표준이나, 여러 시스템(제품군)에서 컴포넌트의 공통적인 기능 즉, 공통성을 기준으로 공통적으로 사용될 컴포넌트 핵심자산을 설계하고 이 컴포넌트 핵심자산에 특정 시스템의 컴포넌트가 가지고 있는 가변성을 추가해서 시스템의 특성에 적합하게 특화된 컴포넌트를 개발하여 재사용성을 향상시키고자 하였다.

컴포넌트 개발(CD)의 프로세스는 그림 2와 같이 5단계로 구성하였다.

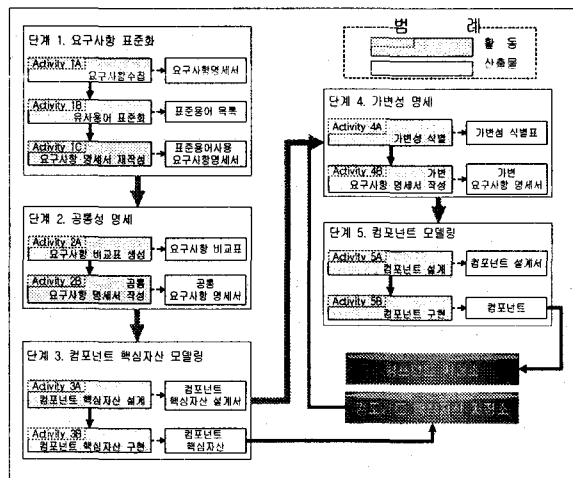


그림 2. CD 프로세스

첫 번째 “요구사항 표준화”는 여러 시스템(제품군)의 요구사항을 수집해서 유사용어를

식별하고 식별된 유사용어를 표준화 하고 이 표준화된 용어를 사용하여 요구사항을 재 작성하는 단계이고 두 번째 “공통성 명세”는 표준화된 요구사항들을 비교하여 그 안의 공통적인 기능들을 식별하여 명세화 하는 단계이고 세 번째 “컴포넌트 핵심자산 모델링”은 공통 요구사항 명세서를 이용하여 컴포넌트 핵심자산을 설계 및 구현하고 구현된 컴포넌트 핵심자산을 컴포넌트 핵심자산 저장소에 저장하여 재사용 준비를 하는 단계이고 네 번째 “가변성 명세”는 컴포넌트의 가변성을 식별하여 명세하는 단계이며 마지막 다섯 번째 “컴포넌트 모델링”은 컴포넌트 핵심자산에 식별된 가변성을 추가하여 제품에 사용할 컴포넌트를 생성하는 단계이다. 제품에 사용할 컴포넌트는 컴포넌트 저장소에 저장하도록 설계하였다.

5. 결론

컴포넌트기반의 소프트웨어 개발과 컴포넌트 개발을 포함하는 컴포넌트 기반의 무기체계 임베디드 소프트웨어 개발방법론을 제시한 논문은 공통으로 사용되는 컴포넌트의 확보 및 재사용으로 개발비용을 획기적으로 절감, 개발 기간 대폭 단축 및 표준화된 개발방법론을 사용함으로써 소프트웨어의 상호운용성과 유지보수성 향상에 기여할 것이다.

본 논문은 개발방법론 설계를 위한 접근방법을 제시한 것으로 향후 세부적인 활동과 기법, 산출물 양식 등에 대한 추가적인 연구 활동이 필요하다.

참 고 문 헌

- [1] 최상영, 『무기체계 개발방법론 소프트웨어 개발 발전 연구』, 국방대, 2003.
- [2] 방위사업청, 『방위력개선 사업관리 규정』, 2006. 5. 1.
- [3] 방위사업청, 『소프트웨어 개발 프로세스』, 2006. 1. 25.
- [4] 김현숙 외 2명, 『국방 공통운용환경 컴포넌트 개발 절차』, 국방과학연구소, 2004.
- [5] 임춘봉, 『CBD 방법론 개요』, 화이트정보통신(주), 2004.
- [6] 나중화, 『임베디드 시스템 프로그래밍』, 사이텍미디어, 2005.
- [7] Software Technology Support Center, *Guidelines for Successful Acquisition and Management of Software-Intensive Systems (GSAM)*, Department of the Air Force, Ver. 3.0, 2000. 5.