

# 국방 분야에서 내장형 소프트웨어 산출물의 재사용 프로세스에 관한 연구

김영균, 유석진, 안효철, 김영수  
 현대·기아 자동차그룹사 현대로템(주) 기술연구소

e-mail: ygkim@hyundai-rottem.co.kr

## A Study on a Reuse Process of the Embedded Software Artifacts in Defense

Young-Gyun Kim, Yu Suk Jin, Hyo-Chul Ahn, Young-Soo Kim  
 R&D Center of Hyundai-Rotem Company, Hyundai-Kia Motors Group

### 요 약

소프트웨어 재사용(Software Reuse)은 소프트웨어 생산성을 높이는 방법임에도 불구하고 내장형 소프트웨어 개발 분야에서의 적용은 많은 어려움이 있다. 내장형 소프트웨어는 개발된 소프트웨어의 실행 환경이 서로 다르고, 하드웨어 의존성이 높기 때문이다. 국방 분야에서의 내장형 소프트웨어는 보안성이라는 특성상 타 프로젝트의 모범사례를 적용하는 데는 정보 공유의 어려움이 있어 왔다. 본 논문에서는 MIL-STD-498에 기반하여 국방 분야의 내장형 소프트웨어 재사용 프로세스에 관해 연구하고 적용 결과를 제시하였다.

### 1. 서론

소프트웨어의 재사용에 관한 많은 연구들이 진행되었음에도 불구하고 내장형 소프트웨어 분야에서는 소프트웨어 재사용의 적용에 있어서 많은 어려움을 겪고 있다. 내장형 소프트웨어는 그 특성상 하드웨어 의존성이 강하고, 실행 환경이 이질적인 환경으로 인하여 재사용이 어렵다는 점 등의 문제점을 갖고 있다. 표1은 국방 분야의 무기체계 내장형 소프트웨어의 특징을 추출한 것으로 숫자가 높을수록 상관관계가 높다.

<표1> 무기체계 내장형 소프트웨어의 특징[1]

구분	실시간성	테스트 어려움	고 신뢰성	가격 민감성	지 전력성	목적 한정성	개발 어려움	하드웨어 통합성
다양성	0	3	1	0	0	1	0	0
복잡성	1	1	1	0	1	1	1	3
고가성	3	3	3	1	1	3	3	1
진부성	0	3	1	0	0	1	3	0
실패위험	3	3	3	1	0	3	3	3
...								
상관관계값	10	17	12	6	3	11	16	10

최근 자동차 내장형 소프트웨어 분야에서는 재사용성이 강화된 컴포넌트 기반의 AUTOSAR(AUTomotive Open System ARchitecture)라는 표준을 제시하고 있다. 재사용에 관한 프로세스 관점의 접근 방식으로 국방 분야의 사실상의 표준이라고 할 수 있는 CMMI v1.2[2] 에서는 TS(Technical Solution) 프로세스의 SP 2.4에서 재사용에 대해 언급하고 있다.

국방 분야에서 소프트웨어 산출물의 정의는 방위사업청의 ‘무기체계 내장형 소프트웨어 획득 및 관리 지침(제2009-17호)’[3]에 “계약을 만족시키기 위하여 생성된 소프트웨어 및 관련 정보를 말한다. 여기에는 개발 중 작성된 소프트웨어 기술문서, 개발된 컴퓨터 파일(소스코드, 실행 파일, 목적파일, 데이터베이스 등) 등이 포함 된다.”고 정의 되어 있다. 또한 ‘재사용 가능한 소프트웨어’에 관한 정의는 “하나의 사용목적을 위하여 개발되었지만 다른 사용 목적을 가질 수 있거나, 특별히 다수의 프로젝트를 위하여 혹은 한 프로젝트에서 다수의 역할을 수행하기 위하여 개발되는 소프트웨어 산출물”로 정의 되어 있다.

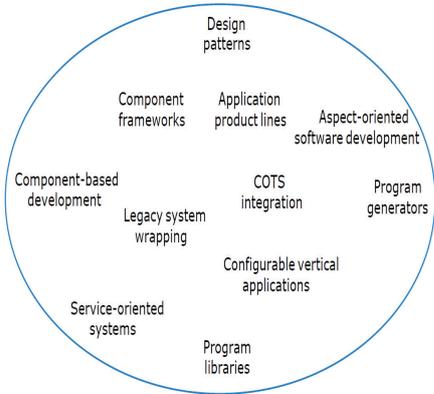
소프트웨어 산출물의 재사용은 결함(Defects)의 감소 경향, 리스크의 감소, 표준의 준수, 개발의 가속화로 개발기간 단축 및 비용 절감 등의 많은 이점을 갖고 있다. 본 논문에서는 국방 분야에서 널리 사용되고 있는 내장형 소프트웨어 개발 프로세스인 MIL-STD-498에 기반하여 소프트웨어 산출물의 재사용에 관하여 연구하였다.

### 2. 재사용에 관한 연구들

#### 2.1 기존 연구들

소프트웨어 재사용에 관한 기존 연구로서는 Design patterns, Component-based development, Application frameworks, Legacy system wrapping, Service-oriented systems, Application product lines, COTS integration, Program libraries 등의 많은 접근 방법들이 제시되었다 [4]. 그림 1은 재사용의 다양한 형태를 보여주고 있다.

재사용에 있어서 흔히 사용하는 방법은 Program libraries 라는 접근방식으로 재사용할 수 있는 클래스나 함수 라이브러리를 사용하는 방법이다. 또는 COTS integration이라고 해서 기존의 응용 프로그램들을 사용하여 통합함으로써 소프트웨어를 개발하는 접근 방식이다.

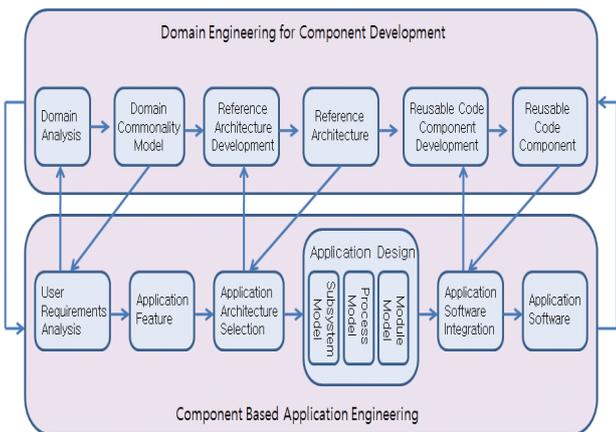


(그림 1) 재사용의 유형[4]

상용 툴의 기능이 향상됨에 따라 Program generators라는 접근방식을 사용하기도 한다. 이 방식은 특정 분야의 응용 프로그램에서 사용되는 코드를 생성할 수 있는 툴을 사용한 접근 방식이다.

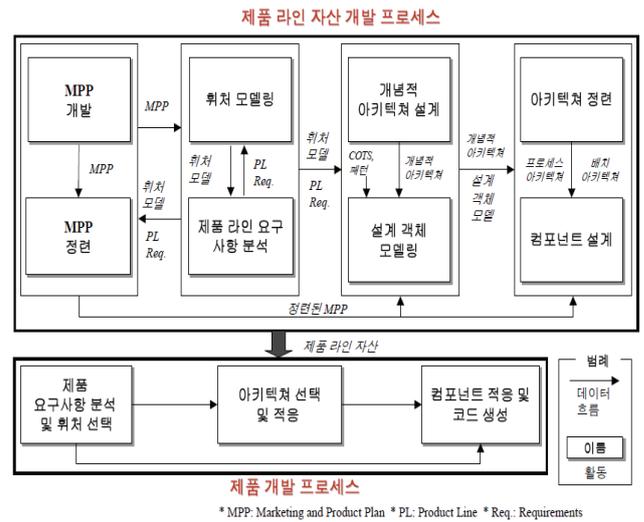
널리 사용하고 있는 접근 방식 중 하나는 그림 2와 같이 CBD(Component-based development) 방식으로 Component model 표준을 준수하는 Components(객체들의 집합)를 통합함으로써 응용 소프트웨어를 개발하는 접근 방식이다.

기존의 응용 소프트웨어를 재사용하기 위해 인터페이스의 집합을 정의하여 제공함으로써 응용 소프트웨어를 재사용하는 Legacy system wrapping 방법도 사용되고 있다.



(그림 2) 기존 CBD 방식의 재사용 접근 방식[5]

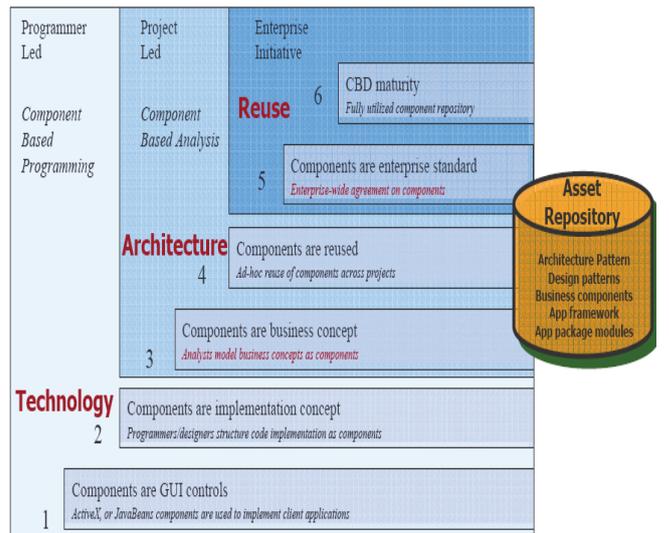
최근 국방 분야에서는 그림 3과 같이 재사용을 높이기 위해 공통적인 아키텍처에 기반하여 응용 소프트웨어를 개발하는 Product Lines 접근방식을 적용하고자하는 움직임이 일고 있다.



(그림 3) Product Lines 접근방식[6]

기존에 연구된 많은 접근방식들이 코드 중심의 재사용에 초점을 맞추고 있는 것이 현실이다. 그러나, 코드 중심의 재사용은 이질적인 실행 환경을 갖고, 하드웨어 의존성이 강한 내장형 소프트웨어 분야에서 재사용성을 높임으로써 생산성 향상을 높이는 데는 한계를 갖고 있는 것도 사실이다. 이질적인 실행환경을 갖는 내장형 소프트웨어의 재사용에 대한 문제점을 해결하기 위하여 모델에 기반한 PIM(Platform Independent Model) 접근방식이나 코드 자동 생성 접근 방식 등이 연구되고 있다.

조직내의 재사용의 성숙도에 관한 모델로서는 그림 4와 같이 6단계의 재사용 성숙도를 제시하고 있는 Butler Group의 CBD Maturity Levels[7]가 있다.



(그림 4) CBD Maturity Levels

본 논문에서는 소프트웨어 개발 수명주기(Life-Cycle)의 단계별 산출물의 재사용성에 관해 연구하였다. 국방 분야에서는 문서화가 엄격히 요구되고 있어 수명주기에 기반

한 재사용성이 코드 중심의 재사용이나 특정 단계의 산출물의 재사용에 기반한 접근방식보다 생산성 향상에 기여할 것이라고 생각한다. 표2는 국방분야에서 널리 적용하고 있는 MIL-STD-498 표준 프로세스의 산출물들의 목록을 보여 주고 있다.

<표 2> MIL-STD-498 표준의 소프트웨어 산출물들

MIL-STD-498
Software Development Plan(SDP)
Software Test Plan(STP)
Software Installation Plan(SIP)
Software Transition Plan(STrP)
Operational Concept Description(OCD)
System/Subsystem Specification(SSS)
Software Requirements Specification(SRS)
Interface Requirements Specification(IRS)
System/Subsystem Design Description(SSDD)
Software Design Description(SDD)
Interface Design Description(IDD)
Database Design Description(DBDD)
Software Test Description(STD)
Software Test Report(STR)
Software Product Specification(SPS)
Software Version Description(SVD)
Software User Manual(SUM)
Software Input/Output Manual(SIOM)
Software Center Operator Manual(SCOM)
Computer Operation Manual(COM)
Computer Programming Manual(CPM)
Firmware Support Manual(FSM)

**2.2 소프트웨어 재사용의 문제점**

소프트웨어 재사용에 있어서 흔히 발생하는 문제점 중 하나는 기존의 산출물들이 새로 개발되는 응용 소프트웨어에 100% 재사용이 어려운 경우에 발생한다. 특히, 기존 코드의 일부를 수정 및 개선하여 재사용하는 경우, 품질에 심각한 문제를 야기할 수 있다. 사람의 안전성에 대한 요구가 강한 국방 및 의료 분야의 내장형 소프트웨어에 있어서는 더욱 중요한 이슈가 되고 있다. 따라서, 기존의 코드를 재사용하는 경우, 기존 코드의 재사용에 앞서 철저한 검증 및 확인(Verification and Validation) 절차가 포함되어야 한다. 검증 및 확인 절차를 거치는 경우, 추가의 시간과 비용이 소요되는 것을 사전에 예측하고 계획 수립을 해야 한다.

무기체계 내장형 소프트웨어 품질 속성에 관한 대표적인 연구로는 ‘무기체계 임베디드 S/W 아키텍처 설계를 위한 품질속성 추출’[8]이 대표적인 연구이다. 이 연구에서는 무기체계 S/W의 대표적인 품질 속성으로 가용성, 사용성, 성능, 시험용이성, 보안성의 품질 속성을 추출하였다. 내장형 소프트웨어의 테스트 프로세스에 관한 연구로는 임베디드 소프트웨어 테스트 성숙도 모델로 Emb-TMM을 제시한 연구[9]가 있으며, 국방 분야의 소프트웨어 테스트 프로세스에 관한 연구로는 국방 시험 성숙도 모델인 MND-TMM(Test Maturity Model)을 제시한 연구[10]가 있다.

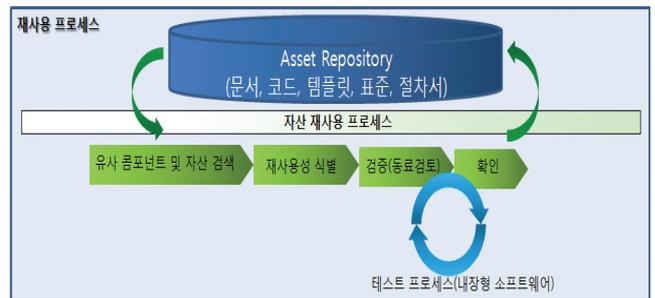
**2.3 프로세스 관점에서의 접근**

프로세스 관점에서의 재사용은 국방 분야의 프로세스에 있어서 사실상의 표준 모델이라고 할 수 있는 CMMI v1.2의 TS(Technical Solution) 프로세스의 SG 2 Develop Design, SP 2.4 Perform Make, Buy, or Reuse Analyses에서 1개의 프랙티스로 제시하고 있다. 그러나, 재사용이 조직적인 관점과 프로젝트 관점에서 정착되어 생산성을 높이기 위해서는 재사용에 관한 별도의 체계적인 프로세스를 갖는 것이 바람직하다. 특히, 다양한 접근방식의 재사용이 혼재되어 있는 경우는 접근방식에 따른 일정 지연, 비용 상승과 결함을 내포하지 않도록 문제점을 면밀히 검토해야 할 것이다.

재사용 프로세스 관점에서 재사용 분석의 수행(Perform a reuse analysis)을 언급하고 있는 대표적인 연구로는 ‘A Reuse-Based Software Development Methodology’[11]가 있다. 이 연구에서는 MIL-STD-498 표준 이전의 미국방성의 표준인 DoD-STD-2167A 프로세스 표준을 기반으로 하고 있으며, Reuse Activity Model을 제시하고, 재사용에 대한 계획 또는 전략을 개발하고, 해결책을 확인하는 6단계의 재사용 활동을 제시하고 있다.

**3. 내장형 소프트웨어 산출물의 재사용 프로세스**

본 논문에서는 기존에 적용하고 있는 국방 분야의 MIL-STD-498 프로세스에 그림 5와 같이 품질을 고려한 재사용 프로세스를 추가하였다. 공통 저장소로서 자산 레파지토리(Asset Repository)를 두고, 자산 레파지토리는 각 프로젝트별로 생성되는 각종 문서(계획서, 요구분석서, 설계서, 코드, 테스터 케이스, 템플릿, 표준문서, 절차서 등)를 저장하고, 검색하여 재사용 가능하도록 제공한다.

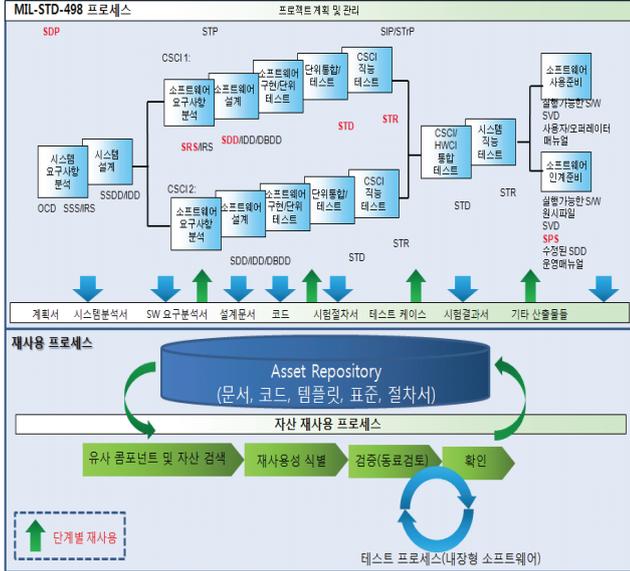


(그림 5) 재사용 컴포넌트의 품질을 고려한 프로세스

새로운 프로젝트 추진시 자산 레파지토리에서 유사 자산 및 컴포넌트를 검색하여 재사용 가능한지 재사용성 식별 과정을 거쳐, 재사용 가능한 경우, 공통부분과 가변부분을 식별하고, 가변부분에 대해서는 수정/삭제 및 추가를 통해 수정을 하고, 수정된 컴포넌트와 자산에 대해 검증(Verification) 과정을 거친다. 소스 코드인 경우 수정된 부

분들에 대해 추가의 테스트 프로세스를 거치고, 확인 (Validation) 절차를 통해 재사용 할 수 있는 컴포넌트와 자산으로 재등록한다.

기준에 국방 분야에서 적용하고 있는 MIL-STD-498 프로세스에 추가하여 재사용 컴포넌트의 품질을 고려한 개발 프로세스는 그림 6과 같다.



(그림 6) 재사용 프로세스를 적용한 개발 프로세스

소프트웨어 컴포넌트 및 자산을 등록하고 검색하여 재사용할 수 있도록 설계 및 구현한 결과는 그림 7 및 그림 8과 같다. 사용자의 접근성을 높이기 위해 웹기반으로 구현되었다.



(그림 7) 소프트웨어 자산의 등록 화면



(그림 8) 자산의 검증(동료검토) 화면

등록하는 자산의 유형에 따라 소스코드, 보고서, 도면, 템플릿, 절차서 등을 식별하여 등록한다. 등록된 문서, 소스코드 등에 대해 검증(동료검토) 절차를 웹기반의 온라인으로 접속하여 가능 하도록 구현 되었다.

#### 4. 결 론

본 논문에서는 국방 분야에서 적용하고 있는 기존의 MIL-STD-498 프로세스에 기반하여 컴포넌트 및 자산의 품질을 고려한 재사용 프로세스에 관하여 연구하였다. 기존의 재사용 접근방식에서 흔히 발생할 수 있는 결함을 제거하기 위해 검증 및 확인 과정과 테스트 프로세스를 통해 재사용 컴포넌트와 자산의 품질을 높이도록 설계하였다. 웹기반의 소프트웨어 구현을 통해 생산성과 접근성을 높이도록 하였다. 향후 요구사항의 자동 추적 기능과 연계하여 재사용 가능한 컴포넌트 및 자산의 추천, 컴포넌트 그리드 등의 연구와 연계할 필요가 있다.

#### 참고문헌

- [1] 윤희병, "국방분야 내장형 소프트웨어 개발 프로세스", 국방항공신뢰성컨퍼런스 2009.3.20
- [2] Mary Beth Chrissis, CMMI 2nd Edition, Addison Wesley, 2007, pp. 552 ~ 553
- [3] 방위사업청, 무기체계 내장형 소프트웨어 획득 및 관리 지침(제2009-17호), 2009. 2. 5, pp. 1
- [4] Ian Sommerville 2004, Software Engineering, 7th edition
- [5] 강교철, 'Issues in Component-Based Software Engineering', International Workshop on Component-Based Software Engineering, May, 1999, pp. 209
- [6] 강교철, 이재준, 김병길, 김사중, "임베디드 시스템 개발 생산성 향상을 위한 소프트웨어 제품 라인 공학", 정보과학회지 제22권 제6호 통권 제181호, 2004년 6월호, pp.25-35
- [7] Butler Group, CBD Maturity Model, 1999
- [8] 임종삼, 윤희병, "무기체계 임베디드 S/W 아키텍처 설계를 위한 품질 속성 추출", Proceedings of KFIS Autumn Conference 2006 Vol. 16, No. 2. pp.268~291
- [9] 백상훈, 윤희병, "임베디드 소프트웨어 테스트 성숙도 모델(Emb-TMM) 설계 절차 및 사례연구, 한국국방경영분석학회지, 제33권 제2호, 2007.12, pp.49-60.
- [10] 박준영, 류호연, 최호진, 백중문, 김진수, "국방 소프트웨어의 시험 프로세스 개선을 위한 국방 시험 성숙도 모델", 정보과학회 논문지: 소프트웨어 및 응용 제 35권 제5호(2008.5), pp. 288-296
- [11] 강교철, A Reuse-Based Software Development Methodology, CMU/SEI-92-SR-4, January 1992, pp.3