

임베디드 소프트웨어 개발을 위한 국방 CBD 프로세스 테일러링

정경철^o, 이길섭, 이승종

국방대학교 전산정보학과

ruuum@hanmail.net^o, {gislee, ljc}@kndu.or.kr

Tailoring the Defense Component Based Development Process for Embedded Software Development

Kyoung-Chul Jung^o Kil-Sup Lee Sung-Jong Lee

Dept. of Computer & Information, Korea National Defense University

요 약

유비쿼터스 시대에 진입하면서 임베디드 소프트웨어의 개발에 대한 관심이 높아지고 있다. 국방 분야에서는 국방 무기 및 비무기 내장형 소프트웨어 관리지침과 공군에서 운영되는 항공무기 소프트웨어 개발 프로세스 등이 운용되고 있으며, 국방 CBD 방법론은 주로 응용체계 개발을 위한 컴포넌트 기반의 구체적인 방법론이다. 이러한 국방 CBD 방법론을 임베디드 소프트웨어 개발에 적용가능 여부는 발표되지 않았다. 따라서 본 논문에서는 임베디드 소프트웨어 개발을 위한 국방 CBD 프로세스 테일러링 방안을 고찰하고자 한다. 이를 위하여 국방 무기 및 비무기 내장형 소프트웨어 관리지침에 근거한 조정, 임베디드 소프트웨어 개발 방식에 따른 조정, 컴포넌트 및 객체 지향기술, 그리고 UML 기술 적용으로 구분하여 테일러링 방안을 제시한다. 그 결과 국방 CBD 방법론의 적용 범위의 확대 및 개발 프로세스의 표준화에 기여할 것으로 기대된다.

1. 서 론

유비쿼터스 시대에 진입하면서 모든 사물에 내장된(embedded) 소프트웨어 개발이 필요하게 되었다. 이와 관련하여 개발 프로세스에 대한 연구가 활발히 진행되고 있다. 대표적인 추세는 컴포넌트와 같은 재사용성 강조, 클래스, 개체, 상속 등 객체지향 특성의 적용, UML[1]과 같은 통합 모델링 언어의 사용을 들 수 있다.

국방 분야에서는 국방 무기 및 비무기 내장형 소프트웨어 관리지침[2]과 공군에서 운영되는 항공무기 소프트웨어 개발 프로세스[3] 등이 운용되고 있다. 이 두 가지 절차는 MIL-STD-498[4]의 체계 요구 분석 및 설계 후 소프트웨어와 하드웨어 트락으로 구분하여 개발하는 절차를 가진다. 그러면서 특정 개발방법이나 프로그래밍 언어 등 구체적인 방법을 한정하지는 않는다.

한편, 2005년 초에 국방부에서는 국방 CBD 방법론[5]을 발표하여 국방 소프트웨어 개발에 적용을 추진하고 있다. 그러나 국방 CBD 방법론은 주로 응용 소프트웨어 개발을 목적으로 고려하고 있어 임베디드 소프트웨어 개발에 대한 적용가능 여부는 발표되지 않았다.

따라서, 본 논문에서는 임베디드 소프트웨어를 위한 국방 CBD 프로세스 대한 테일러링 방안을 고찰하고자 한다. 이를 위하여 상위 지침인 국방 무기 및 비무기 내장형 소프트웨어 관리지침에 근거하여 국방 CBD 프로세스의 조정, 임베디드 소프트웨어 개발방식에 따른 조정, 컴포넌트 및 객체 지향기술, 그리고 UML 기술 적용으로 구분하여 테일러링 방안을 제시한다.

논문의 구성은 2장에서는 임베디드 소프트웨어 개발절차에 대해 살펴보고 3장은 국방 CBD 프로세스 테일러링 방안을 기술한다. 4장은 테일러링을 통한 개선사항에 대한 토의, 결론 및 향후 연구에 대하여 기술한다.

2. 관련 연구

2.1 국방 분야 임베디드 소프트웨어 개발 절차

국방 분야에서는 무기체계 정밀화·첨단화에 따른 소프트웨어 의존도 증대되어 소프트웨어 성능개선 및 유지보수 문제를 해결하기 위해 국방부는 2002년부터 「무기 및 비무기 체계 내장형 소프트웨어 개발관리지침」으로 규정하였다. 이를 통하여 무기 및 비무기 체계에 내장되어 해당 장비의 임무에 전용으로 제공되는 소프트웨어 개발을 별도의 개발절차를 적용하지 않고, 표준화 개발을 목적에 부합되도록 세부 품목을 선정하여 요구사항에 부합되게 수행되도록 하고 있다. 이 지침은 MIL-STD-498에 근거하여 계획, 체계 분석 및 설계, 소프트웨어 요구분석 및 설계, 구현 및 테스트, 인도 등 5개 단계를 13개 활동, 25개의 작업으로 구성된다.

국방 CBD 방법론은 국방정보체계 개발에 있어 여러 정보체계 간에 공통적으로 지원되는 소프트웨어의 요소를 식별하여 재사용과 통합이 가능하도록 표준화된 관리를 제공하는 컴포넌트 기반 방법론이다. 이 방법론은 UML 기반 CBD 방법론에 기초하여 국제표준인 ISO/IEC 12207과 MIL-STD-498 등을 분석 및 참조하여 개발된 프로세스로서 분석, 설계, 구현 및 테스트, 인도를 포함하는 4개 단계, 12개 활동, 37개의 작업으로 구성된다.

한편, 내장형 소프트웨어 개발관리지침은 각종 무기체계의 시뮬레이터, 항공용 소프트웨어, 전자전 소프트웨어에서 적용이 되고 있다. 국방 CBD 방법론은 주로 응용 소프트웨어를 주요 적용대상으로 하고 있으며, 현재는 교육 및 전파 단계로 응용 소프트웨어에 대한 적용사례도 미약하고, 임베디드 소프트웨어는 적용가능 여부도 발표되지 않고 있는 상황이다.

2.2 임베디드 소프트웨어 개발 방법

임베디드 체계의 소프트웨어 개발방법은 초기에 체계의 소프트웨어의 비중이 하드웨어의 비중보다 적어 구조적 방법을 적용하였다. 고정된 하드웨어에 맞게 소프트웨어를 개발하여 하드웨어를 직접 제어할 수 있는 어셈블리 언어로 구현함으로써 처리속도와 효율성에 많은 장점이 있었다.

그러나 현재 소프트웨어와 하드웨어가 동시에 개발되고 고기능을 요구함에 따라 고정된 하드웨어로 가정하기 힘들고 어셈블리어로 작성한 소프트웨어 부분의 프로그램 확장과 디버깅에 어려움이 많아 차지하는 비중이 많이 줄어들고 있다.

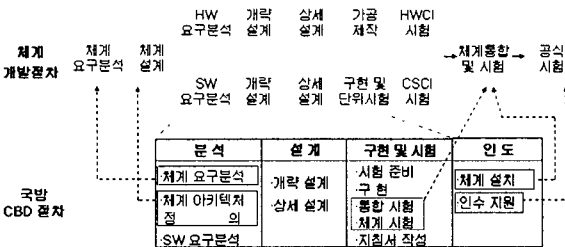
위에서 언급한 구조적 방법의 문제점을 극복하기 위해 프로세스와 데이터를 분리하지 않고 새로운 기술 및 운영환경 변화의 단위를 프로세스와 데이터를 동시에 가진 객체를 두어 생산성 향상과 유지보수의 복잡성을 감소시키는 객체지향 방법으로 발전하였다. 이러한 방법론으로 초기 객체지향 기술인 OMT[6]와 실시간 지원을 위한 ROPES[7], UML 프로토타입을 이용한 임베디드 소프트웨어 특성의 정성적 및 정량적 모델링을 위한 ACCORD/UML이 있다.

최근에는 소프트웨어를 독립적으로 사용 가능한 컴포넌트 단위로 쪼개어 표준화된 네트워크 기반의 이기종 분산 환경에서 신속하게 변화하는 요구사항에 즉각 대처할 수 있도록 형상관리 및 유지보수가 용이하고 재사용성과 기존 체계와의 통합이 유리한 CBD(Component Based Development) 방법으로 나아가고 있다. 이러한 CBD 방법론으로 초기의 Catalysis[8], 반복 및 아키텍처 기반의 통합 프로세스, 오늘날에는 컴포넌트, 제품계열[9] 및 모델 중심 아키텍처 (MDA)[10] 등이 있다.

3. 국방 CBD 프로세스 테일러링

3.1 내장형 소프트웨어 개발관리지침 적용

내장형 소프트웨어 개발관리지침은 앞에서 언급된 바와 같이 체계 요구분석 및 설계를 바탕으로 하드웨어와 소프트웨어의 트랙을 구분하되 소프트웨어 개발만을 가정한 후에 하드웨어와 병행을 하는 절차로 구성이 되어 있다.



[그림 1] 임베디드 소프트웨어 개발 프로세스

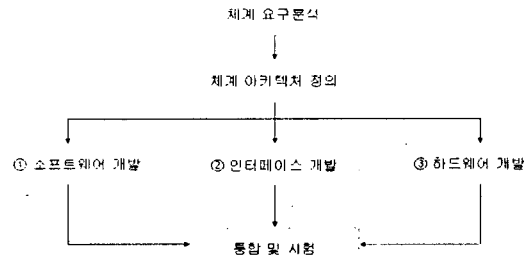
반면에 국방 CBD 방법론은 초기에 체계 아키텍처를 분석을 하지만 하드웨어는 개발환경으로 고정된 것으로 간주를 하고 있다. 따라서 이 둘 간의 개념차이를 상위 개

념인 지침에 맞추어 주는 것이 필요하다.

[그림 1]은 내장형 소프트웨어 개발관리지침과 국방 CBD 프로세스를 병합하는 개념을 보여주고 있다. 국방 CBD 프로세스의 주요 활동을 지침에 맞게 조정하는 것이 필요하다. 주요 내용은 체계 요구정의와 체계 아키텍처 정의를 체계 프로세스로 조정하고 체계 설치 및 인수 지원을 공식시험 전후로 조정이 필요하다.

3.2 임베디드 소프트웨어 개발 방식

임베디드 소프트웨어를 개발하는 방식은 개발 범위에 따라 소프트웨어만 개발 (경우 I: ①), 소프트웨어와 인터페이스의 개발 (경우 II: ①+②), 소프트웨어와 하드웨어 병행 개발 (경우 III: ①+②+③) 등 세 가지[1]로 나뉠 수



[그림 2] 임베디드 소프트웨어 개발 방식

있다. [그림 2]는 체계 요구분석과 체계 아키텍처 설계를 거쳐 3개의 개발과정으로 구분되었다가 통합되는 과정을 보이고 있다.

경우 I은 하드웨어는 사전에 준비되어 있다고 가정하고 순수하게 응용차원에서 소프트웨어를 개발한다. 경우 II는 기존의 하드웨어 상에 펌웨어(firmware) 또는 응용 소프트웨어 형태로 개발된 체계와 새로이 개발되는 소프트웨어와 인터페이스가 개발되는 경우이다. 경우 III는 소프트웨어와 하드웨어가 병행하여 개발되는 경우이다.

여기서 인터페이스 개발은 내장형 체계에서 반드시 고려해야 할 요소로서 기계적(Mechanical), 전자적(Electronic), 기능적(Functional), 그리고 절차적(Procedural) 요소를 포함한다. 이 중 기계적이고 전자적인 것은 하드웨어 요소이고, 기능적이고 절차적인 요소는 소프트웨어 요소이다. 하드웨어 개발은 회로도, 회로보드배치도, 구성품 목록, 회로배치구성도 등 전자적 요소와 조립순서, 구성품 목록, 자재 명세, 설계도 등 기계적 요소를 포함한다.

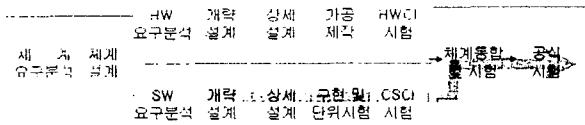
병행개발은 하드웨어와 소프트웨어가 동시에 개발해야 하기 때문에 전통적인 하드웨어 개발 방법과 결합되어서 접근해야 한다. 하드웨어와 소프트웨어 양쪽 영역에 있어 다양한 선택의 분석으로 최적의 구현방법을 찾을 수 있도록 초기 체계 아키텍처를 개발하는 경우에 기능적 매핑이 고정하지 않으며 하드웨어와 소프트웨어 아키텍처 설계단계에서 하드웨어와 소프트웨어의 기능적인 분산을 최적화 할 수 있도록 고찰 후 체계 아키텍처를 완

1) 그 외의 경우는 소프트웨어 관점에서 본 연구에서 제외한다.

성한다.

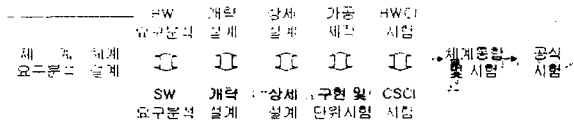
3.3 개발방식 적용 테일러링

응용차원 소프트웨어 개발방식에 따른 개발 프로세스는 체계요구분석과 체계설계를 통하여 체계상에서 소프트웨어의 기능적인 요소를 식별하여 소프트웨어 요구분석을 한다. [그림 3]과 같이 임베디드 소프트웨어를 단순히 하드웨어의 응용 소프트웨어로 간주하여 개발하는 방식이다. 이는 하드웨어는 개발환경으로서만 존재하고 응용 소프트웨어에 대한 영향은 최소화 된 경우이다.



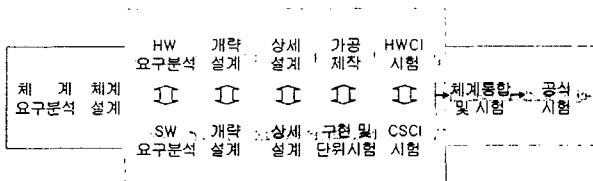
[그림 3] 응용차원 소프트웨어 개발방식

인터페이스 개발방식에 따른 개발 프로세스는 하드웨어와 소프트웨어 각각의 인터페이스 중 기능들을 수행하기 위해 상호 연결 및 통신을 하는 인터페이스를 체계 설계 단계에서 공통의 하드웨어 및 소프트웨어 인터페이스를 설계하고, 소프트웨어의 개발이 진행되면서 하드웨어 관련 인터페이스의 적절성을 검토하여 개발해야 한다. 이 때, 이질적인 멀티프로세서들을 컴포넌트로 통합할 수 있도록 하여 하드웨어 설계를 시험하거나, 소프트웨어 기능을 검증할 수 있도록 한다.



[그림 4] 인터페이스 개발방식

병행 개발방식에 따른 개발 프로세스는 [그림5]와 같이 하드웨어와 소프트웨어가 동시에 개발되는 프로세스로서 각 단계별 검증 작업도 하드웨어와 소프트웨어에서 동시에 수행되어야 하며, 이런 개발의 중간 단계에서 하드웨어와 소프트웨어의 동시 검증능력이 올바른 설계를 위한 필수적인 요건이 된다.



[그림 5] 병행 개발방식

4. 결론

본 연구에서는 응용 소프트웨어의 개발을 고려한 국방 CBD 방법론을 체계 프로세스와 개발방식을 고려한 테일러링 방안을 고찰하였다. 그리고 오늘날 재사용을 위한 컴포넌트 및 객체지향 기술의 적용을 고려하였다.

먼저, 체계 프로세스 적용은 국방 CBD 프로세스의 활동과 과업을 상위 지침인 체계 프로세스에 맞추어 조정을 하여주면 간단하게 달성을 할 수 있다. 그러나 소프트웨어와 하드웨어 개발을 고려한 개발방식을 조정하는 사항은 개발방식에 대한 개념을 소프트웨어만을 개발, 소프트웨어와 인터페이스 개발, 소프트웨어와 하드웨어 병행 개발의 세 가지 상황을 가정하여 프로세스 조정을 이루었다.

소프트웨어만을 개발하는 경우는 기존의 체계개발 프로세스와 동일하게 적용이 가능하다. 하지만 인터페이스 개발이 요구되는 경우는 기계적, 전자적, 기능적, 절차적 요소를 하드웨어와 소프트웨어 요소로 구분하고 개발이 요구된다. 본 연구에서는 하드웨어와 소프트웨어 개발 주요 이정표에서 인터페이스 적용 여부를 검토하는 것을 제안하였다. 마지막으로 하드웨어 개발까지 포함되는 경우는 하드웨어의 기계적이고 전자적인 요소를 포함하여 인터페이스 검토를 통하여 개발을 제안하였다.

끝으로 오늘날 컴포넌트 및 객체지향 기술의 적용은 국방 CBD 프로세스가 내재적으로 보유하고 있는 요소들이라서 세부적인 논의를 생략하였다. 그렇지만, 2장에서 거론된 컴포넌트 및 객체지향 기술의 근본적인 특징들을 고려하여 실시간 지원을 위한 UML 2.0의 지원, 반복 및 아키텍처의 적용, 제품계열 적용 등에 대하여는 패턴 또는 사례에 대한 연구가 추가적으로 필요하다.

참고문헌

- [1] Ivan Jacobson, UML 사용자 지침서, 인터비전, 1999.
- [2] 국방부, "무기 및 비무기체계 내장형 소프트웨어 개발관리 지침", 국방부, 2001.
- [3] 공군본부, 무기체계 소프트웨어 개념 및 발전방향, 공군본부, 2003.
- [4] DoD, Software Development and Document, MIL-STD-498, 1994.
- [5] 국방부, 국방 CBD 방법론, 국방과학연구소, 2005.
- [6] James Rumbaugh, "Object-Oriented Modeling Technology(OMT)", OOPSLA, 1994.
- [7] Bruce Powel Douglass, ROPES : Rapid Object-Oriented Process for Embedded system, I-Logix, 1999.
- [8] Desmond Francis D'Souza, Object, Component and Frameworks with UML, Addison Wesley, 1998.
- [9] Northrop L, "Framework for software Product line Practice", SEI Report, 2002.
- [10] OMG, MDA Guide Version 1.0.1, OMG, 2003.