

임베디드 소프트웨어 재사용을 위한 문서 자산의 분류

차정은⁰ 양영중
한국전자통신연구원 임베디드 S/W연구단 S/W공학팀
{jecha⁰, yiyang }@etri.re.kr

Classification of Documentation assets for Reusing Embedded Software

Jung-Eun Cha⁰ Yung-Jong Yang
Electronics and Telecommunications Research Institute

요 약

임베디드 시스템에서 임베디드 소프트웨어의 비중이 크게 확대됨에 따라 소프트웨어 재사용 자산에 대한 가치는 현저하게 증가하고 있다. 그러나 환경적 변화에 따른 유사한 특성을 지닌 임베디드 소프트웨어에 대한 반복적인 수요가 증가함에도 불구하고, 현재에는 임베디드 소프트웨어의 자산화를 위한 구체적인 지침 제공이 전무하여 임베디드 소프트웨어를 단지 개발 과정에서 우연히 발생하는 임시방편적인 산출물로 인식하고 있어 조직 내 임베디드 소프트웨어의 생산은 항상 비용 소모적인 오류를 만들어 내고 있다. 따라서, 본 논문에서는 하드웨어에 의존적이며 외부 환경과 다양하게 상호협력 해야만 하는 임베디드 소프트웨어 자산들의 생성과 활용을 위해 문서화 관점에서 재사용 자산을 정의하고 분류하여 이들간의 상호 관계를 명확히 기술함으로써 임베디드 소프트웨어 재사용을 위한 표준 지침을 제공하고자 한다.

1. 서 론

소프트웨어 개발의 낮은 생산성과 품질의 문제를 혁신적으로 개선시켜줄(*silver bullet*) 방법으로 소프트웨어 재사용이 제시되었다. 그러나 소프트웨어 재사용은 매우 어려운 작업으로 오늘날의 주된 재사용 대상인 컴포넌트와 같이 범용적이며 보다 큰 규모의 재사용은 이전의 라이브러리와 같은 작은 재사용 모듈 개발에 비해 3배의 비용을 요구하며 재사용 자산으로 신뢰성을 인정 받기가 힘들다 [1, 2]. 재사용의 이상적인 효과에도 불구하고 생산성 향상에 효과적일 결과를 가져오지 못하는 것은 재사용 대상이 되는 재사용 자산에 대한 정확한 정의나 이를 개발하기 위한 구체적인 절차가 미흡하고, 재사용 자산을 공급하고 이를 사용하는 이해당사자들 간에 재사용 자산의 개발 단계별 형태에 대한 명확한 동의가 없기 때문이다. 특히 임베디드 소프트웨어는 제한된 시스템 자원에서 최적화된 성능을 실현해야 하므로 높은 신뢰성을 요구할 뿐더러, 동일 서비스라도 다양한 환경적 요소에 따라 다른 속성을 가져야 하므로 소프트웨어의 가변성(*variation*)만 잘 명시된다면 높은 재사용성을 가질 수 있다[3, 4]. 본 논문에서는 임베디드 소프트웨어 자산의 주요 기술적 속성에 따라 3가지로 구분하고 자산의 개발 단계별로 개발 산출물들을 체계적으로 정의하였다.

또한, 임베디드 소프트웨어 자산의 재사용을 극대화를 목적으로 문서화 자산의 양식을 '간소화', '최적화' 라는 2가지 관점에서 정의하였다. 이를 위해 필수적으로 고려되는 핵심 문서 자산 선별을 위해 문서 자산들의 특징과 개별 문서 자산에 포함해야 될 속성들을 식별하였으며, 식별된 속성들의 산출 시기와 속성 정보들의 중복성, 그리고 개발자들의 요구를 기반으로 필수 자산을 선별하고 문서 자산들간의 통합을 수행하였다. 이에 대한 결과로 전체 5가지의 문서 자산을 정의하였다.

2. 임베디드 소프트웨어 재사용 자산

2.1 임베디드 소프트웨어 자산의 정의

소프트웨어의 재사용을 위해서는 소프트웨어 개발을 시작할 때부터 소프트웨어 재사용자를 고려해야만 재사용자가 개발된 소프트웨어를 잘 이해하고 적용할 수 있다. 만약 그렇지 않고 소프트웨어 개발 과정에서 문서 작성을 간과하거나, 개발자 자신만의 패턴으로 코딩 한다면 재사용자 입장에서는 재사용에 앞서 이해하기 위한 노력 소모가 많아 재사용의 효과를 획득하기 힘들다. 본 문서에서는 소프트웨어 재사용 자산을 다음과 같이 정의한다.

재사용 가능한 일정한 형태로 가공된 재사용 모듈과 이와 연관된 문서 및 관련 자료들이다. 재사용 모듈은 코딩된 원시 코드와 실행 가능한 바이너리 코드의 집합을 의미하며, 문서에는 재사용 자산 개발 과정에서 작성된 요구사항 정의서, 설계서, 아키텍처 등이 포함된다. 또한 관련 자료는 재사용 자산의 개발 및 운영에 사용되었던 DB, 이미지 파일, 그래픽 라이브러리 등이 있다.

2.2 임베디드 소프트웨어 문서 자산의 분류
2.2.1 분류 기준 및 종류

임베디드 소프트웨어 자산의 종류와 그에 따른 형태는 다양하며 자산이 생성되는 시기나 자산의 성격 혹은 자산의 규모에 따라 다르게 나타날 수 있기 때문에 본 논문에서는 이러한 개별 기준들을 다 포함할 수 있는 임베디드 소프트웨어 계층을 1차 기준으로 하여 분류하고, 그것들을 다시 자산의 개발단계별로 구분하였다. 즉, 임베디드 시스템의 실시간성과 자원의 경량성, 저전력 등의 속성을 충분히 반영해야 하는 기술적 특성에 따라 임베디드 운영체제, 디바이스 드라이버, 유무선 통신 프로토콜 같은 시스템 소프트웨어와, 임베디드 자바 플랫폼 및 분산 미들웨어(RT-CORBA, COM, 등), 제어, 통신, 멀티미디어 미들웨어 등의 미들웨어 소프트웨어, 그리고 멀티미디어 재생기, 임베디드 브라우저, 게임, GPS 등의 응용 소프트웨어로 구분한 후, 자산 개발 과정에서 필요한 산출물들을 정의하였다. 다음의 <표 1>은 임베디드 소프트웨어의 개발 단계에 따라 분류된 문서 자산들로 자사용자의 필요성에 따라 필수, 선택 자산으로 구분된다[5].

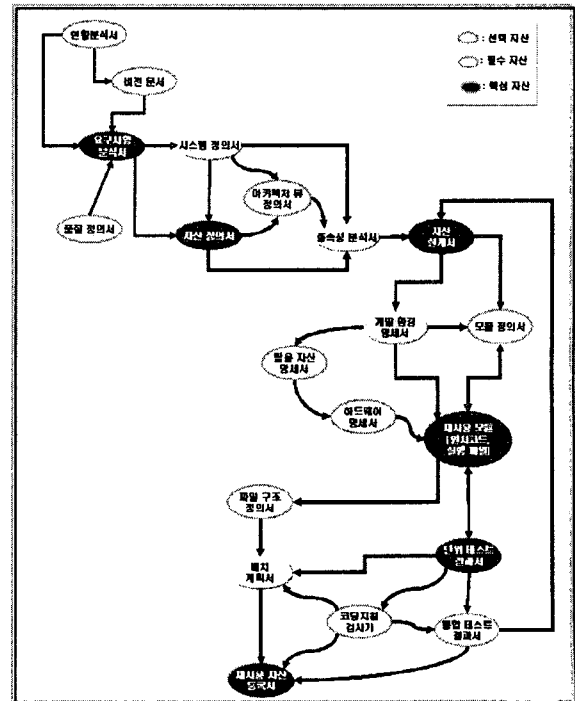
<표 1> 개발 단계에 따른 문서 자산의 분류

분류	자산명	내용
요구사항	필수	요구사항 분석서 타겟 시스템에 대한 요구사항들을 수집하여 정형화된 형식으로 문서화한 것임. 자산의 정의, 목적, 요구사항(기능/품질), 활용 범위 등을 나타냄
	선택	시스템 정의서 목표 시스템의 개발 및 운영, 유지보수 정보 및 제약 조건 등을 다양한 측면(즉, 비즈니스 측면, 기술적 측면, 환경적 측면)에서 상세히 정의한 문서임
	선택	현황 분석서 현재 대체되는 제품들(시스템)의 특징 및 문제점, 관련 이해당사자들이 제기하는 발전 방향, 그리고 목표 시장에서의 경쟁과 효과 등을 정의한 문서임.
정의	비전 문서	구축할 시스템에 대해 누가 사용하고, 어떤 영역에 적용되며, 어떻게 활용할 수 있는지를 정의한 문서로, 이행 당사자들이 달성할 목표와 범위를 비즈니스와 기술적 관점에서 나타냄
	품질 정의서	요구사항 정의서에서 품질(Quality) 요구사항에 관한 사항만을 추출하여 품질 분류표 작성을 통해 시스템이 달성하고 평가 받을 품질 요소를 정의함
아키텍처 설계	필수	아키텍처 정의서 목표 제품 개발을 위한 구현 방법, 프로세서, 언어, 필요 요소들의 획득 방법 등을 정의하고 목표 시스템 전체의 연관 구조(블록 다이어그램)를 정의
	선택	기능모듈 정의서 아키텍처 정의서의 전체 시스템의 구조에 포함된 소프트웨어 블록들의 기능적 정의를 나타낸 문서임
	선택	종속성 분석서 환경 종속적 부분에 대한 정보를 제공하는 산출물로, 하드웨어 제약사항이나, 응용 계층 및 시스템 계층 소프트웨어에 대한 제약사항 등을 제시함
선택	모듈별 정의서 소프트웨어 기능 모듈 정의서에서 설명된 각 소프트웨어 기능 모듈의 상세한 명세를 작성한 산출물	

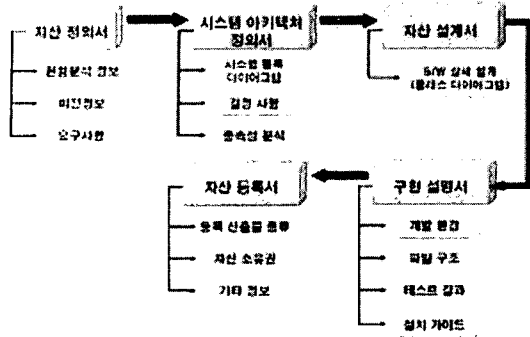
필수	자산 설계서	재사용 모듈의 설계문서
자산 설계	개발환경 명세서	타겟 OS, 교차 개발 도구, 참조 타겟보드 종류, 하드웨어 제약사항 등의 정보를 제시
	활용자산 명세서	커널, 부트로더, 디바이스 드라이버, 그래픽 라이브러리 및 상용 컴포넌트 등에 대한 정보를 제시
자산 구축	필수	재사용 모듈 원시코드+실행 파일
	선택	테스트 결과서 단일 테스트 및 통합 테스트 결과를 제시
	선택	파일구조 정의서 재사용 모듈에 사용된 물리적인 파일의 구조 및 개별 파일들의 주요 정보들을 제시
배출	필수	재사용 자산 등록서 재사용 지원 시스템에 등록을 위해 재사용 지원 시스템 형태에 필요한 정보를 포함하는 산출물
	선택	배치 계획서 자산 설치 및 설정에 관련된 매뉴얼 및 자산의 라이선스 등에 관련한 정보를 정의

2.2.2 자산들 간의 정보 흐름

(그림 1)은 자산의 개발 과정에서 정제와 상세화 작업을 통해 일관성 있는 정보 흐름을 가지는 임베디드 소프트웨어 재사용 자산들 간의 관계성을 도식화 한 것이다. (그림 1)에서 격자 모양 타원은 반드시 만들어야 할 산출물을 의미하고, 사선 모양의 타원은 필수는 아니지만 재사용 자산 생성에서 원활한 정보 흐름을 지원하며 자산의 풍부한 정보 생성을 위한 산출물이다. 그리고 솔리드 타원은 재사용 자산 개발에 가장 핵심적인 산출물들이다.



(그림 1) 임베디드 소프트웨어 문서 자산간의 정보 흐름

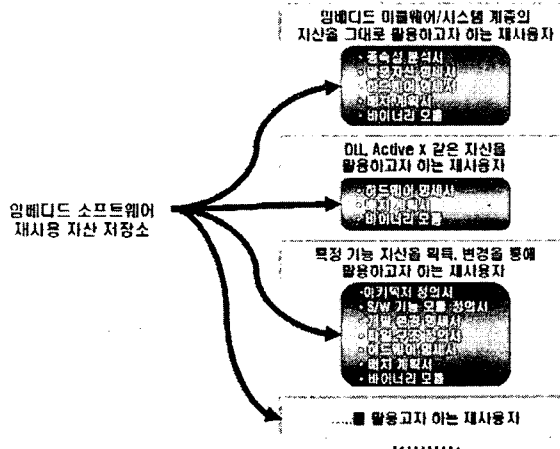


(그림 2) 필수적인 임베디드 소프트웨어 문서 자산

3. 필수 문서 자산

<표. 1>의 자산들을 산출 시기와 개별 자산에 포함된 속성 정보들의 중복성과 자산 개발자들의 요구를 기반으로 필수 자산을 선별하였다. 이들 자산 개발자의 문서 자산 생성 부담을 축소시키고 나아가 자산 재사용자에게 꼭 필요한 정제된 정보만을 제공함으로써 재사용자가 자산 이해를 위해 수반해야 할 의사 결정의 발생 빈도를 줄이기 위한 것이다. 이들 자산들은 분산 정의된 문서 정보들을 통합하여 간략화 한 것으로 자산 공급자의 작업량과 자산 수요자의 필요성에 맞추어 정리하였다(그림 2).

또한 재사용자의 수준에 따라 필요한 자산 범위나 활용 방법은 달라진다. 임베디드 재사용 자산에 대해 완벽히 이해하고 있어 바이너리의 재사용 모듈만을 필요로 하는 경우는 간단히, 설치 가이드나 하드웨어 명세서와 같은 산출물만 필요하지만, 재사용자가 자산에 대한 사전 지식이 없는 경우라면, 자산에 대한 보다 많은 이해 정보 획득을 위해 가능한 많은 산출물들을 획득할 필요가 있다. 따라서, 본 문서에서는 사용자에 따라 요구되는 문서 자산의 종류를 개략적으로 분류하여 도식화하였다(그림 3).



(그림 3) 사용자 수준에 따른 문서 자산의 분류

4. 현재 이슈 및 결론

본 논문은 임베디드 소프트웨어의 재사용 체계를 정립하기 위한 1차적인 선행 연구의 결과이다. 재사용 대상이 되는 임베디드 소프트웨어 자산을 문서 관점에서 분류하고 재사용자에게 제공할 문서 자산을 계층화 시킴으로써 자산 개발자와 자산 수요자와 사이의 일치성 있는 자산에 대한 이해를 수립할 수 있도록 하였다.

재사용은 재사용을 하고자 하는 조직 내의 표준화된 자산 개발 방식과 그 결과물을 생산을 위한 상호 협조와 필요하며 이를 위해 본 논문에서 임베디드 소프트웨어 자산의 문서 자산에 대한 분류를 서술하였다.

현재는 다양한 특성의 임베디드 소프트웨어 자산을 크기와 물리적인 형태별로 정의하고 있으며 향후에는 자산의 특성에 는 맞춤형 개발 기법으로 생산, 배포할 수 있는 있도록 지침을 마련할 것이다.

5. 참고 문헌

[1] McGary, F., G. Page, et al. 1984, ' An Approach to Software Cost Estimation' , NASA Software Engineering Laboratory, SEI-83-001(Feb.)
 [2] BiggerStaff, Ted, and Alan J. Perlis, 1989, Software Reliability, New York: ACM Press
 [3] 나중화, 강순주. 임베디드 시스템 프로그래밍 : 이론 및 실습, 사이텍미디어, 2004년 10월
 [4] David E. Simon, 임문자를 위한 임베디드 시스템, 사이텍 미디어, 2003년 11월
 [5] 임채덕 외, 임베디드 소프트웨어 기술동향 및 산업발전 전망, Journal of IITA, Vol 4., No. 3, 2002년 9월