

사례 구축을 통한 임베디드 SW 재사용 지침 검증

차정은*, 최유희, 하수정, 박창순

*한국전자통신연구원 임베디드 SW 연구원

e-mail:{mary2743, yhchoi, hsj, cpark}@etri.re.kr

Verification of Embedded SW Reuse Guideline through constructing the Case Studies

Gil-Dong Hong*, Cheol-Soo Kim**, Young-Hee Lee*

*Dept of Computer Science, Han-Kook University

**Dept of Computer Engineering, Dae-sung University

요 약

SW 자산 가치에 현저한 증가와 환경적 변화에 좌우되는 임베디드 SW의 재사용을 통한 생산성 강화의 필요성이 급격히 증가하는데 반해, 임베디드 SW의 자산화를 위한 구체적인 지침이나 사례연구의 제공이 전무하여 임베디드 SW의 생산은 항상 비용 소모적인 오류를 만들어 내고 있는 실정이다.

본 논문에서는 임베디드 SW 재사용 사례 구축을 통해 임베디드 SW의 자산화를 위한 임베디드 SW 지침을 검증하고, 실용적 지침으로 보급한다.

1. 서론

최대부분의 임베디드 S/W는 개발 및 탑재되는 외부 환경의 미소한 차이에 따라 동일한 기능을 수행하더라도 상호 호환되지 못하고 다른 S/W 산물로 인식되어 불필요한 비용을 소모하고 있는 실정이다. 특히 임베디드 S/W가 많이 활용되고 있는 휴대단말기나 자동차 제어 시스템 등의 영역은 비즈니스 프로세스가 거의 동일함에도 불구하고 임베디드 S/W가 가지는 하드웨어 의존성과 실시간성 등의 특성 때문에 영역에서 요구하는 적시성(time-to-market)을 위한 기존 임베디드 S/W의 재사용을 시도하지 못하고 있다.

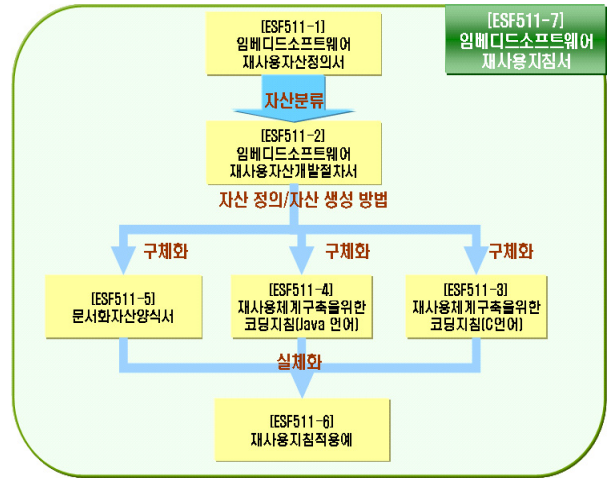
본 논문에서는 임베디드 S/W의 재사용성 향상을 위한 실질적 방법 제시를 목적으로 임베디드 S/W의 모듈화 방법과 모듈 개발 절차에 따른 문서 작성 및 코딩을 위한 지침 등 임베디드 SW 재사용의 타당성을 검증한다.

2. 임베디드 SW 재사용 지침

임베디드 S/W 재사용 지침은 임베디드 S/W를 의미있는 재사용 단위로 자산화시키는 방법과 절차를 정의하고 그 과정에서 생성되는 문서자산들의 양식과 작성 기법을 제시하며, 코딩시 가독성을 높일 수 있는 권장 가이드라인과 성능, 메모리관리 등 특정한 임베디드 S/W의 품질 속성을 향상시킬 수 있는 코딩 패턴 제공한다.

(그림 2)는 임베디드 SW 재사용 지침의 체계적 구성도이다.본 문서에는 임베디드 시스템 영역에서 재사용하고자 하는 임베디드 S/W 재사용 자산의 정의와 종류를 명확히 정의한 재사용 자산 정의서와, 임베디드 S/W 재사용 자

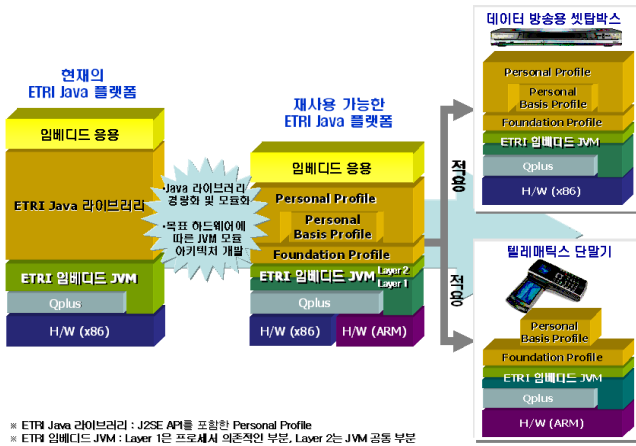
산의 개발을 위한 모듈화 방법과 모듈 개발 절차에 따른 문서 작성 지침 및 코딩을 위한 지침, 지침에 따라 재사용 예제가 내용으로 포함되어 있다.



(그림 1) 재사용 지침서의 구성

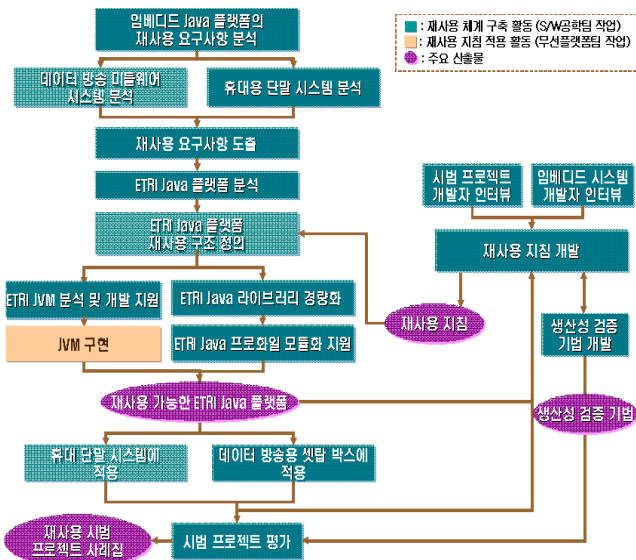
3. 임베디드 SW 재사용 사례

재사용 사례로 본 기관의 개발 산물인 임베디드 Java 플랫폼의 Java VM과 Java 라이브러리 기술을 선정했다. 이는 하드웨어 변경 사항에 대응하지 못하고 있으며 임베디드 시스템에 필요한 형태로 모듈화 되어 있지 않으나, ETRI Java 플랫폼을 재사용 가능한 형태로 만들 경우 활용 분야가 매우 광범위하기 때문이다.



(그림 2) Java VM과 Java 라이브러리 모듈화 개념도

재사용 사례 구축 작업에서 “임베디드 Java 플랫폼”기술을 타겟 H/W의 변경에 대해 대응할 수 있는 구조, 즉 재사용이 가능한 “임베디드 Java 플랫폼”화 하고, 이를 특정 임베디드 시스템에 적용되는 사례를 구축하여 “생산성 향상”을 입증할 수 있었다.



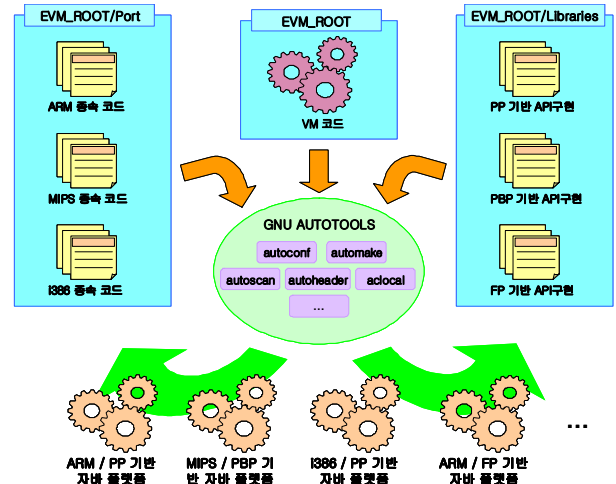
(그림 3) Java VM과 라이브러리 재사용 사례 구축 프로세스

(그림 3)과 같은 재사용 사례 개발 프로세스를 통한 결과는 J2ME의 세 가지 프로파일로 재사용될 수 있도록 모듈화되었다.

- 경량화 이전의 라이브러리 사이즈: 8.3 M Bytes
- PP 버전 : 3.64 MBytes
- PBP 버전 : 3.42 MBytes
- FP 버전: 2.08MBytes

또한 기존 라이브러리의 기능이 각각의 용도에 따라 불필요한 부분을 제거하고 적은 메모리 용량의 환경에서 사용할 수 있도록 되었다. 각각의 라이브러리는 autoconf 체계를 이용하여 재구조화 되어 사용시 패키지 설정에서 --with-j2me-classlib 라는 옵션을 통해서 사용될 수 있도록 configure 체계를 수정하였다.

ETRI Java 플랫폼 전체에 대한 재사용 메커니즘을 도식화한 것이다. 그림에서는 타겟 시스템의 사양 및 용도에 따라 각각 다르게 API Library와 VM의 형태를 정의하여 구분하고, 컴파일 시 사용자의 옵션 입력값에 따라 적절한 API Library와 VM을 조합하는 방법으로 사용자의 요구에 맞는 Java 플랫폼을 만들어 내는 방법을 표현하고 있다.



(그림 4-5) ETRI Java 플랫폼의 재사용 메커니즘

재사용 지침의 적용을 통해 시행한 본 사례 연구를 통해 Java 라이브러리가 3가지 용도로 모듈화되고 경량화되었다.

- ▶ PP 버전 : 56.2% 감소, ▶ PBP 버전 : 58.8% 감소
- ▶ FP 버전: 74.9% 감소

또한, 적용 VM의 다양화가 이루어져 ETRI VM외에 Kaffe VM 지원이 추가로 가능하게 되었다. 이를 통해 다양한 프로세서를 지원하는 VM을 지원하게 되어 Java 라이브러리의 활용성을 높일 수 있었다. 예를 들어 ARM이나 MIPS 프로세서를 지원하는 VM을 사용할 경우 ETRI Java 라이브러리를 해당 VM과 용이하게 결합하여 x86외에 타 프로세서에서 쉽게 J2ME CDC 응용환경을 구현할 수 있다. 그리고, 지원 라이브러리의 다양화 이루어져 커널 2.4 기반의 GTK 1.2 및 GTK 2.x 상에서 활용할 수 있게 되었다.

4. 결론

임베디드 S/W 재사용 지침은 S/W 재사용 문화를 확산 시킴으로써 조직의 생산성 향상을 위한 작고 실제적인 재사용 템플릿을 보급하고 적용하는 것을 목표로 개발되었다. 따라서 재사용 지침을 통해 중복 개발 비용 최소화 및 개발 경험 공유로 조직 내 S/W 자산의 가치를 지속적으로 증대시킬 수 있을 뿐 아니라 개발자가 동의하는 재사용 필수 항목들만을 양식화하여 지침 적용의 부담을 최소화시킬 수 있다.

재사용 지침을 적용한 Java VM과 라이브러리의 모듈

화를 통해 경량화와 다양한 VM 지원이 가능해졌음을 확인할 수 있었다. 즉, 메모리 사이즈 별로 다양한 임베디드 환경에서 필요에 맞게 쉽게 적용 가능하도록 되었고, VM을 교체함으로써 다양한 프로세서를 지원할 수 있게 되었고 그래픽 라이브러리의 지원범위를 확대함으로써 재사용 활용이 증진될 수 있도록 하였다.

[1] Roger S. Pressman "Software Engineering A Practitiners' Approach" 3rd Ed. McGraw Hill
[2] McGary, F., G. Page, et al. 1984, 'An Approach to Software Cost Estimation', NASA Software Engineering Laboratory, SEI-83-001(Feb.)
[3]BiggerStaff, Ted, and Alan J. Perlis, 1989, Software Reliability, New York: ACM Press
[4] 나종화, 강순주. 임베디드 시스템 프로그래밍 : 이론및 실습, 사이텍미디어, 2004년 10월
[5] David E. Simon, 입문자를 위한 임베디드 시스템, 사이텍미디어, 2003년 11월
[6] KOSTA, "S/W 프로젝트 수행 환경 및 생산성에 관한 실태 조사", 20061
[8] Jorge L. Díaz-Herrera[†] and Vijay K. Madiseti, The Yamacraw Embedded Software(YES) Methodology: A Technical Analysis", Yamacraw (YES) Technical Report CSIP-TR-00-01, 1/31/2000.Georgia Tech, 1999.