

## 하드웨어와 소프트웨어가 포함된 제품개발을 위한 프레임워크

### A Framework for Product Development including HW and SW Components

도남철\*, 채경석\*

\* 경상대학교, 산업시스템 공학부 (dnc@gnu.ac.kr, pro-cgs@hotmail.com)

#### Abstract

This paper proposes a framework for product development including hardware and software components. The framework provides separation of the hardware dependent software, an integrated product development process, and integration of software components with product configurations and product structures. In order to separate the hardware dependent software, the framework considers product configuration modules and engineering changes of associated hardware and software components. The proposed product development process integrates development of the hardware dependent software into the existing product development process. In order to integrate the hardware dependent software with product configurations and product structures, the framework represents software components by existing product data models in Product Data Management (PDM). The framework is applied to development of a robot system including hardware and software components in order to show its effectiveness.

#### 1. 서론

현재의 제품개발은 다양한 고객 요구를 만족시키는 유연한 기능을 제공하기 위하여 많은 소프트웨어를 포함한다. 예로, 자동차의 경우 제조원가의 20%가 소프트웨어를 포함한 전자 부품이 차지하며 최신 Hybrid 자동차는 그 수치가 47%까지 올라간다. 자동차의 전자 부품 중 소프트웨어가 차지하는 비중은 2005년에 10%에서 2015년에는 20%로 증가될 것으로 예상되며, 이러한 통계치는 소프트웨어가 전자 부품의 부속품이 아닌 중요한 독립 부품으로 인식되어야 함을 보여준다. (Choi, 2005)

하드웨어와 소프트웨어가 통합된 제품개발은 서로 이질적인 두 분야의 특성, 개발 방법, 그리고 관련 지식의 차이로 여러 가지 문제점을 내포하고 있다. 현재는 하드웨어와 소프트웨어가 독립적으로 개발되고 있으며 단순히 각 분야의 결과를 다른 쪽에 전달하여 제품개발을 하고 있다. 그러나 하드웨어와 소프트웨어의

통합이 일반화되고 복잡해짐에 따라 단시간 내에 경쟁력 있는 제품개발을 위해서 두 분야의 제품개발을 지원할 수 있는 통합 제품정보관리 시스템이 필요하다.

현재 소프트웨어 공학자들은 소프트웨어 개발을 위한 Software Configuration Management (SCM)와 하드웨어 개발을 위한 Product Data Management (PDM)에 대한 통합 프레임워크를 제안하고 있으며, 일부는 PDM의 선진화된 제품구성(Product Configuration) 기술을 소프트웨어 개발에 도입하려는 시도도 있었다.

이 논문은 PDM 관점에서 소프트웨어를 포함하는 제품개발을 지원하기 위한 통합된 프레임워크를 제안한다. 프레임워크는 일반적인 소프트웨어 개발을 하드웨어 제품개발에 통합하기 위한 의존적인 소프트웨어(The Dependent SW)를 분류할 수 있는 기본원칙과 하드웨어 제품을 위한 기존의 제품구성 및 조립구조(Assembly Structure)에 소프트웨어를 통합하기 위한 방법을 포함한다. 프레임워크는 기본원칙, 통합된 제품 자료 모델, 그리고 의존적인 소프트웨어를 지원하는 제품개발 프로세스에 대한 설명으로 구성되며, 프레임워크의 유효성을 검증하기 위하여 네트워크 기반 로봇 시스템 개발에 적용되었다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 관련연구에 대해서 서술한다. 3장에서는 하드웨어 및 소프트웨어 통합 개발을 위한 제품자료 모델과 통합된 개발 프로세스로 구성된 프레임워크를 제안한다. 4장에서는 네트워크 기반 로봇 시스템 개발에 적용된 프레임워크의 적용에 대하여 설명하고, 5장에서 논점을 재정리하며 결론을 도출한다.

#### 2. 관련 연구

하드웨어와 소프트웨어 통합에 관한 연구로 SCM 전문가들의 연구가 있다. Estublier (1998)의 SCM과 PDM 비교에 관한 연구, Dahlqvist (2004)의 SCM과 PDM 통합에 관한 연구, 그리고 Krebs (2005)의 SCM에 PDM의 제품구성 개념 적용에 관한 연구 등이다. 하지만 SCM 관점에서의 연구는 단순히 정적인 제품자료 구조의 통합을 시도하고 있으며, 통합의 기본 원칙과 제품개발 프로세스에서 PDM과 SCM 사이의 복잡하고 다양한 상호작용에 대한 고려사항을 제공할 수 없었다.

1998년 Estublier는 PDM과 SCM을 통합 관점에서 Fundamentals, Product Model, Versioned Product Model, Work Space, 그리고 Process Model 등의 5가지 측면으로 비교하였다. 이 연구는 일반적인 소프트웨어 전체를 통합을 위한 PDM의 비교 대상으로 하였으며, 제품개발 프로세스와 제품구성 방법에 대한 차이점은 언급하지 않고 있다.

2004년 Dahlqvist는 PDM과 SCM의 개발 프로세스에서 자료 공유와 교환의 필요성 그리고 두 영역의 공통된 기능 등을 언급하였다. 이 연구에서는 개발 프로세스 사이의 기능을 통합하는 공통의 개발 도구에 초점을 맞추고 통합 가능성을 분석하였다. 이 연구에서 API를 이용한 PDM과 SCM 도구의 통합 그리고 하드웨어와 소프트웨어를 포함하는 자료구조를 제안하였지만 제품개발 프로세스상의 설계변경을 위한 자료의 정합성 문제에 대한 해결 방법은 언급하지 않고 있다.

2005년 Krebs는 Software Mass Customization을 위한 Software Product Families (Mannisto, 2000)에 PDM의 제품구성 개념을 적용한 Software Product Lines과 Model-based Configuration을 제안하였다. 하지만 하드웨어와 소프트웨어가 통합된 제품개발을 위한 방법론은 언급하지 않고 있다.

위의 관련 연구를 종합하면 PDM과 SCM을 비교한 연구는 개발 프로세스와 제품구성 관점의 연구가 부족하고, PDM과 SCM 도구의 통합에 관한 연구는 제품자료의 일관성 유지에 관한 연구가 미흡하며, PDM의 개념을 SCM에 적용한 연구는 하드웨어와 소프트웨어가 통합된 제품개발에 관한 연구가 미흡하다.

그러므로 본 연구에서는 하드웨어와 소프트웨어의 개발 프로세스를 통합하고, 제품구성을 위한 통합 제품자료 모델과 제품자료의 일관성 유지를 위한 프레임워크를 제안한다. 이 프레임워크는 일반적인 소프트웨어에서 하드웨어와 연관된 소프트웨어를 분리하고, 통합된 제품개발 프로세스와 제품자료 모델로써 일관성 있는 제품자료를 위한 체계적인 방법을 제공한다.

### 3. 하드웨어와 소프트웨어가 포함된 제품개발 프로세스를 위한 프레임워크

제안된 프레임워크는 방향을 제시하는 기본원칙, 통합 제품개발 프로세스, 그리고 통합 제품자료 모델을 포함한다.

#### 3.1 기본원칙

제안된 프레임워크는 다음의 원칙을 기반으로 한다.

- 1) PDM을 중심으로 SCM의 특징을 통합한다.
- 2) 통합 개발을 위하여 일반적인 소프트웨어 개발에서 의존적인 소프트웨어 개발을 분리한다.
- 3) 통합 제품개발 프로세스와 통합 제품자료 모델로 구성된다.

통합 프레임워크의 첫 번째 원칙은 소프트웨어 개발을 위한 SCM을 하드웨어 개발을 위한 PDM에 적용하는 것이다. PDM은 적용 대상이 구체적이고 오랫동안 적용된 경험이 있으며, ISO STEP과 같은 표준이 있기 때문에 SCM에 비하여 정교한 관리가 가능하다. 특히 최근에 SCM에서 PDM의 구성 관리를 도입하여 적용하려는 Software Product Family 또는 Product Line 등의

기존 연구 등을 검토하여 첫 번째 원칙을 결정하였다. Software Product Family는 고객들의 다양한 요구를 만족시키기 위하여 소프트웨어 모듈의 조합에 초점을 맞춘 것으로 PDM의 제품 구성 관리로부터 개념을 가져온 것이다. Dahlqvist (2004)도 기존 PDM 제품 구조에 소프트웨어 부품을 통합하기 위해 PDM 제품 구조 확장을 제안하였다.

두 번째 원칙은 일반적인 소프트웨어에서 하드웨어와 통합이 필요한 의존적인 소프트웨어를 분리하는 것이다. 복잡도를 줄이기 위하여 하드웨어와 관계없는 소프트웨어는 기존의 소프트웨어 개발 방법론에 따라 독립적으로 개발된다. 현재의 제품들에 포함되는 소프트웨어는 Embedded 소프트웨어, Firmware, OS, 그리고 클라이언트 서버 응용 프로그램을 포함하는 다양한 종류로 구성된다. 기존의 연구는 소프트웨어 개발 전체를 구분 없이 하드웨어 개발과 통합하려는 모델을 제시하였다. 하지만 의존적인 소프트웨어의 분리가 없다면, 제품개발 프로세스에 많은 종류의 소프트웨어가 포함되기 때문에 통합은 불가능하다. 제안된 프레임워크는 의존적인 소프트웨어를 분리하고 이를 기존의 PDM에 통합하는 방안을 제시한다.

제안된 프레임워크는 하드웨어와 소프트웨어 통합을 위한 제품 자료 모델의 변화뿐만 아니라 하드웨어와 소프트웨어의 개발 프로세스의 변화도 제시한다. 하드웨어와 소프트웨어의 특성 차이로 인하여 제품개발 전 단계에 걸쳐 많은 차이점이 존재하고 이를 해결하기 위하여 변경요소를 제안한다.

#### 3.2 통합된 제품개발 프로세스

Fig.1은 제안된 프레임워크의 제품개발 프로세스를 보여주고 있다. 제품개발 프로세스의 특징은 소프트웨어 개발에서 분리된 의존적인 소프트웨어를 하드웨어 개발에 통합한 것이다.

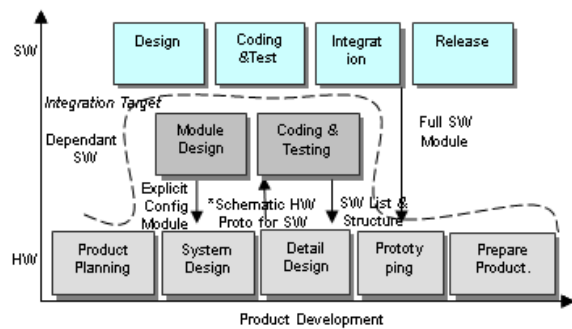


Fig.1 프레임워크의 제품개발 프로세스

의존적인 소프트웨어는 하드웨어와 밀접하게 연관된 소프트웨어이다. 다음의 특징에 따라 일반적인 소프트웨어에서 의존적인 소프트웨어를 분리 한다:

- 1) 물리적으로 하드웨어를 구동하거나 하드웨어로부터 신호를 받는 소프트웨어
- 2) 하드웨어의 제품구성을 위하여 모듈화 될 필요가 있는 소프트웨어
- 3) 하드웨어의 설계변경에 영향 받는 소프트웨어

제품개발 프로세스 관점에서 하드웨어 개발은 Fig.1의 하단과 같은 5단계로 구성된다. 두 번째 단계인 System Design 단계는 미리 정의된 요소를 조합하여 다양한 제품구성을 만드는 제품구성 모델을 만들게 된다. 이때 소프트웨어가 포함된 제품구성을 위하여 하

드웨어 제품 모듈에 연관된 소프트웨어 모듈이 제공되어야 한다.

세 번째 단계인 Detail Design 단계에서 설계자들은 제품구성, 재료, 그리고 제조 방법을 포함하는 제품 사양을 산출한다. 이때 소프트웨어는 Coding과 Testing을 하게 되는데, 의존적인 소프트웨어는 프로그램을 구동시키거나 입력을 할 하드웨어가 필요하게 된다. 그러나 하드웨어 개발은 이 단계에서 설계가 완료되지 않으므로 소프트웨어 Test를 위한 최소한의 입력력 기능을 하는 Schematic HW Prototype을 제공하게 된다. Detail Design의 최종 단계에서 의존적인 소프트웨어 개발자는 하드웨어 제품구조에 대응하는 소프트웨어 목록과 구조를 제공해야 한다. 이는 소프트웨어 부품으로 PDM에 통합되며 추후 설계변경 시 그 영향을 고려하게 된다.

### 3.3 통합된 제품자료 모델

Fig.2는 제품자료 모델에 대한 프레임워크를 보여주고 있다. 이 제품자료 모델은 일반적인 PDM에서 채택하는 제품자료 모델이다. 제품구성은 미리 정의된 Option이라는 모듈의 조합으로 구성되며, 각 Option은 이를 구성하는 물리적 부품인 Part로 구성된다. Part는 다른 Part를 하위 부품으로 가지면서 제품구조를 표현할 수 있으며, 각 부품은 다수의 공학 문서(Document)를 가질 수 있다. PDM에서 공학 문서는 CAD Model File을 저장하는데 사용되며 Version 관계를 통하여 설계이력을 관리한다.

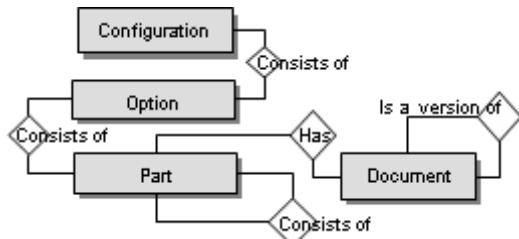


Fig.2 프레임워크의 제품자료 모델

### 통합된 제품구성 표현

현재 제품 개발에서는 다양한 고객 요구를 만족시키기 위하여 미리 정의된 Option의 조합으로 다양한 제품 구성을 생성하는 Implicit 제품구성 모델을 주로 사용하고 있다. 소프트웨어도 각 제품구성을 위한 소프트웨어 Option을 제공하여야 한다. 이때 하나의 제품구성에 여러 개의 소프트웨어 Option이 선택될 수도 있으나 제안된 프레임워크는 하나로 통합된 소프트웨어 Option을 제공한다(Explicit 제품구성). 이는 소프트웨어가 주로 하나의 파일로 Compile 되어 사용되는 특성을 반영한 것이다.

Fig.3은 통합된 제품구성 모델 기반의 제품구조를 보여준다. 예에서 제품구성 Config A와 Config B는 각각 하드웨어 Option A, B 그리고 B, C로 구성되며, 각 제품구성 별로 하나의 소프트웨어 Option A와 B를 제안한다. 하드웨어와 다르게 각 소프트웨어는 하나의 Option을 제공하는 것을 볼 수 있다.

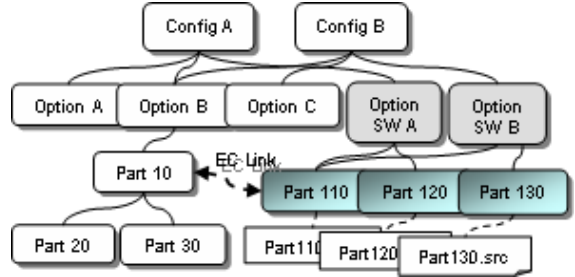


Fig.3 통합된 제품구성을 위한 소프트웨어 모듈의 예

### 소프트웨어 부품의 구현

소프트웨어 부품은 소프트웨어 Option (Option SW A, B)하에 하드웨어 부품과 동일하게 그 제품구조가 구성된다. 소프트웨어 부품은 메타 정보를 표현하는 Part 객체(Part 110, 120, 130)를 이용하여 추상화되기 때문에 제품구조를 이용하여 다른 소프트웨어 부품과 Source Code는 공유할 수 있다. Source Code는 Document 객체(Part 110, 120, 130.src)로 관리된다. 서로 연관된 하드웨어와 소프트웨어 부품은 하드웨어 부품의 변경이 관련된 소프트웨어 부품에 전달될 수 있도록 설계변경 Link (EC Link)로 연결된다.

## 4. 프레임워크 적용

### 4.1 네트워크 기반의 로봇 시스템 개발

제안된 프레임워크는 네트워크 기반 로봇 시스템의 개발에 적용되었다. 연구 목적을 위한 시제품 로봇 시스템은 무선 네트워크 기반의 탐색 로봇이다. 로봇은 사용자가 원거리에서 로봇을 직접 제어하거나 무선 네트워크를 통하여 제어 프로그램을 Upload 하여 자율 행동을 할 수 있다. Fig.4는 로봇 시스템의 아키텍처와 로봇의 사진을 보여준다.

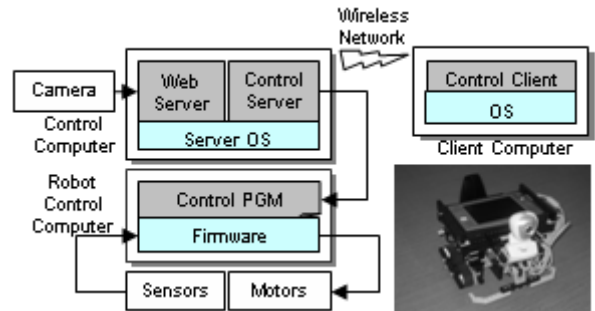


Fig.4 로봇 시스템의 아키텍처와 로봇의 사진

이 개발에서 Dassault Systems SmarTeam V5R14와 Microsoft Visual Source safe 6.0이 주요 PDM과 SCM으로 사용되었다. 이 외에도 다양한 소프트웨어 모듈과 프로그램 개발을 위하여 다양한 다른 소프트웨어 개발 환경과 프로그램 언어를 사용하였다.

### 4.2 프레임워크 적용

#### 의존적인 소프트웨어 분리

로봇 시스템은 제어 서버와 클라이언트를 위한 OS, 로봇 제어 컴퓨터를 위한 Firmware, 원격 제어를 위한 네트워크 프로그램, 동영상 전송을 위한 서버 프로그램, 그리고 로봇 행동을 위한 로봇 제어 프로그램을 포함한 다양한 종류의 프로그램을 요구한다. 이들 프로그램

램들 중, 프레임워크에서 제안된 의존적인 소프트웨어 로써 로봇 제어 프로그램을 분리한다. 로봇 제어 프로그램은 입력 신호와 모터 제어에 의한 로봇의 자율 행동을 구현하기 위하여 로봇 제어 컴퓨터에서 실행된다. 따라서 프로그램은 센서와 모터 같은 하드웨어와 매우 연관이 깊고, 제품구성과 설계변경에 영향을 받게 된다. 로봇 제어 프로그램은 소프트웨어 부품으로 추상화되고 PDM에 의하여 관리된다. Remote 제어를 위한 서버와 클라이언트 프로그램을 포함하는 다른 소프트웨어는 SCM 도구에 의하여 관리된다.

**제품구성을 위한 소프트웨어 모듈화**

Fig.5는 로봇의 제품구성을 보여준다. 로봇 시스템은 두 가지 제품구성인 Argos Type 1S와 Argos Type 1D를 제공한다. 접촉센서의 종류에 따라 나누어지는 두 가지 제품구성은 5개 중 4개의 하드웨어 Option을 공유한다.

프레임워크는 각 구성에 대하여 각각의 소프트웨어 Option을 제공한다. 소프트웨어 Option을 위한 Explicit 제품구성 모델은 각 구성에 대한 소프트웨어 Option을 쉽게 개발, Compile, 그리고 Test 하도록 프로그래머를 지원한다.

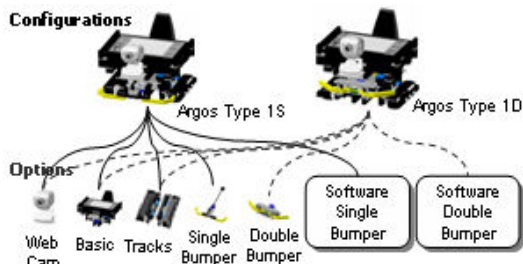


Fig.5 로봇의 제품구성 모델

**Schematic HW Prototype**

프레임워크에서 설명했듯이, Detail Design 단계의 초기에 하드웨어 기술자는 로봇 제어 프로그램 개발을 위해 Schematic HW Prototype을 프로그래머에게 제공된다. Fig.6은 Schematic HW Prototype의 사진(a)과 각 개발 프로세스에서 Prototyping 단계를 위한 다른 Prototype을 보여준다(b, c).

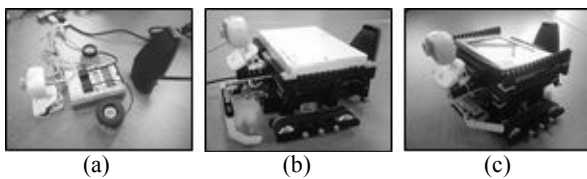


Fig.6 Schematic HW Prototype (a) 과 Prototyping Phase (b, c)

**소프트웨어 부품과 제품구조**

Detail Design의 최종 단계에서, 프로그래머는 소프트웨어 부품목록과 제품구조(Part List and Product Structure)를 제공한다. Fig.7은 예제에서 사용된 소프트웨어 부품구조를 보여준다. 하드웨어 제품구조를 사용하여, 소프트웨어 부품은 다른 소프트웨어 부품과 Source Code를 공유한다. PDM의 문서 관리 특징은 프로그래머들이 Source Code를 공동으로 관리하고 개발하도록 한다. 설계변경 Link는 또한 하드웨어 부품의 설계변경 시 연관된 소프트웨어 부품을 프로그래머가

재검토 하도록 지원한다.

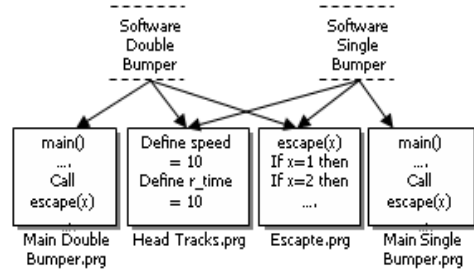
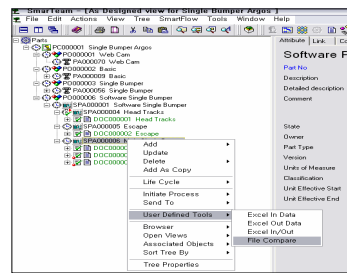


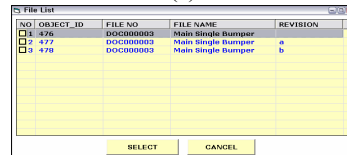
Fig.7 소프트웨어 부품구조의 예

**소프트웨어 부품 관리를 위한 PDM의 기능 확장**

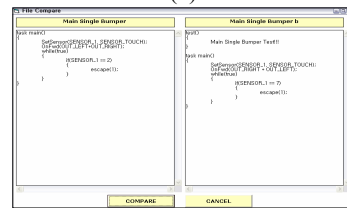
제안된 프레임워크에서는 일반적인 소프트웨어에서 하드웨어 의존적인 소프트웨어를 분리하여 PDM에서 관리하게 된다. 기존 PDM은 Version 관리 기능을 통하여 의존적인 소프트웨어의 Version 관리는 가능하지만, SCM의 주요 기능인 Source Comparison 기능은 포함하고 있지 않다. 그러므로 Source Comparison 기능을 PDM에 확장하여 구현하였다 (Fig.8).



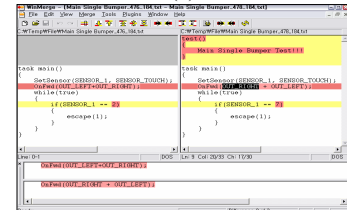
(a)



(b)



(c)



(d)

Fig.8 PDM에 Comparison 기능을 확장(a)하고 실행한 사진(b, c, d)

Fig.8에서 소프트웨어 부품을 선택하여 File Compare를 실행하면(a), 선택된 소프트웨어 부품과 연관된 Source Code의 목록들이 보이고(b), 비교하고자 하는 Source Code를 선택하면 Source Code의 내용이 보인다.(c) Source Code의 내용 확인 후 Comparison 기능을 실행하면 두 Source Code의 비교 결과가 나온다(d).

## 5. 결론

본 논문은 하드웨어와 소프트웨어가 포함된 제품개발을 위한 프레임워크를 제안하였다. 제안된 프레임워크는 소프트웨어를 포함하는 통합된 제품개발을 위하여 의존적인 소프트웨어 개념을 제안하고, 제품구성과 조립구조 안에 소프트웨어를 통합하기 위하여 소프트웨어 부품으로써 소프트웨어를 추상화하였다.

의존적인 소프트웨어를 적용한 통합된 제품개발 프로세스와 제품자료 모델로써 제품구성 및 설계변경에 따른 제품자료의 일관성을 유지할 수 있었다.

제안된 프레임워크를 검증하기 위하여 이를 네트워크 기반 로봇 시스템 개발에 적용하였으며, 확장된 PDM을 이용하여 소프트웨어 개발을 위한 SCM 기능을 지원하였다.

## 참고문헌

- B. S. Choi and D. K. Bok (2005), Trends of Convergence in Automobile and IT, (in Korean) *SERI Economic Focus*, No 65.
- J. Estublier, J-M Favre and P. Morat (1998), Toward SCM/PDM Integration, *System Configuration Management, SCM-8, Lecture Notes in Computer Science 1439*, Springer, pp.75-94
- A. P. Dahlqvist, I. Crnkovic and M. Larsson (2004), Managing Complex Systems Challenges for PDM and SCM, *Software Configuration Management, SCM-10, 23th ICSE*.
- I. Crnkovic, U. Asklund, and A. P. Dahlqvist (2003), *Implementing and Integrating Product Data Management and Software Configuration Management*. Artech House.
- T. Mannisto, T. Soininen and R. Sulonen (2000), Configurable Software Product Families, *ECAI 2000 Configuration Workshop*.
- T. Krebs, K. Wolter and L. Hotz (2005), Model-based Configuration Support for Product Derivation in Software Product Families, *PuK 2005*.