

CMMI 지원을 위한 SIR-CM 분석 및 설계에 관한 연구

박은주, 이성욱, 신승우, 김행곤
대구가톨릭대학교 컴퓨터정보통신공학부
e-mail:{ejpark, sojiro, selab, hangkon}@cu.ac.kr

A Study on Analysis and Design of SIR-CM for supporting CMMI

Eun-Ju Park, Sung-Wook Lee, Seung-Woo Shin, Haeng-Kon Kim
Dept. of Computer Information & Communication Engineering,
Catholic University of Daegu, Korea

요 약

현재 많은 SI업체나 IT기업들은 소프트웨어 프로세스의 품질 향상을 위해서, 사실적 표준이 되어가고 있는 CMMI 인증을 취득하려 하고 있다. 조직은 이를 위해 여러 CMMI 지원 도구들을 이용하게 되는데 각 도구별 산출자산들의 데이터 형식이나 종류가 다를 수 있다. 따라서 그들을 표준화하여 통합 저장하고 관리할 수 있는 통합 저장소가 필요하다.

본 논문에서는 이질적인 산출물들의 효과적인 관리 지원을 위한 SIR-CM(SPIC Integration Repository - Configuration Management) 시스템을 제안한다. 이것은 소프트웨어 프로세스 개선 센터(SPIC)의 각 지원 도구로부터 산출되는 이질적인 산출물들을 통합하여 저장, 관리 할 수 있다. 그리고 처음 SPI를 위해 CMMI의 인증을 받으려는 기업에게 충분히 필요한 시스템이라 판단되고 더욱 질 좋은 소프트웨어 프로세스 개선을 가져다 줄 것이다

1. 서론

최근 SI업체나 IT기업들이 소프트웨어 프로세스의 품질 인증을 획득하기 위해 지속적으로 노력하고 있다. 품질 인증 획득을 통해 조직의 소프트웨어 개발 및 IT시스템 개발 능력을 객관적으로 검증 받을 수 있어서, 결과적으로 조직의 대외 경쟁력을 높여 수익성 향상을 가져올 수 있다. 국제적으로 공인된 SPI 모델에는 미국식인 CMMI 모델과 유럽식인 SPICE(ISO/IEC 15504) 모델이 주로 쓰이고 있다. 그 중에서도 CMMI가 소프트웨어 프로세스 개선에 있어서 사실적 표준이 되어가고 있으며, 우리나라 IT 기업들도 CMMI 인증을 획득하려 하고 있다. 그러나 CMMI에서는 여러 프로세스가 정의되어 있고, 단순히 실행해야 할 Practice만을 이야기하고 있기 때문에, 일반 기업들은 어떻게 CMMI를 적용해야 하며 어떤 데이터들을 저장해야 하는지 잘 이해하지 못하는 경우가 많다.

일반적으로 CMMI 인증을 획득하기 위해서 기업은 여러 CMMI 지원 도구들을 이용한다. 이 과정에서

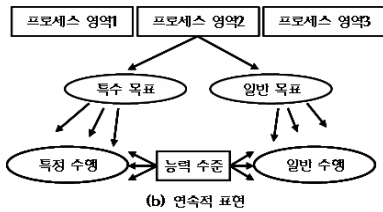
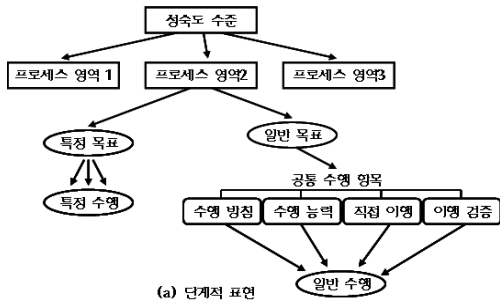
산출되는 모든 자산들은 저장소를 기반으로 조직 내에서 공유되고 관리되어야 한다. 그러나 이 과정에서 산출되는 자산들은 데이터 형식이나 종류가 다양하다. 따라서 이러한 이질적인 자산들을 통합하여 저장하고 관리할 수 있는 통합 저장소가 필요하다. 본 논문에서는, 이러한 이질적인 자산들을 효과적으로 관리하고 지원하기 위한 SIR-CM(SPIC¹⁾ Integration Repository - Configuration Management) 도구를 제안한다.

2. 관련연구

2.1 CMMI(Capability Maturity Model Integration)

CMMI는 1991년 미 국방부의 요청으로 미국 카네기 멜론 대학 소프트웨어 공학연구소에서 개발한 소프트웨어 프로세스 성숙도 평가 모델로 조직의 프로세스 개선 활동을 효율적으로 지원하는 모델이다. 2006년 8월 SEI에서 발표한 1.2 버전에는 총 22개

1) SPIC : Software Process Improvement Center



(그림 1) CMMI 모델 구성요소

의 프로세스 영역을 가지며 그림 1에서 나타내는 것처럼 CMMI 모델은 (a)단계적(Staged) 표현과 (b)연속적(Continuous) 표현 두 가지 방법으로 표현된다. 지침서에서는 각 프로세스 영역의 특정 목표와 일반 목표를 기술하고, 이에 따르는 특정 수행과 일반 수행을 기술한다[1][2][3].

2.2 형상관리

형상관리란 하드웨어 혹은 소프트웨어 개발에 있어서 발생하는 형상을 관리하는 것이다. 개발 과정에서 생기는 형상들이 어떻게 변경되는지를 기록하고 관리하는 프로세스로 그 정보를 저장소에 저장 관리한다. 기본적으로 형상관리에서는 형상들이 바뀔 때마다 버전을 매겨 이 정보를 저장소에 기록하고 관

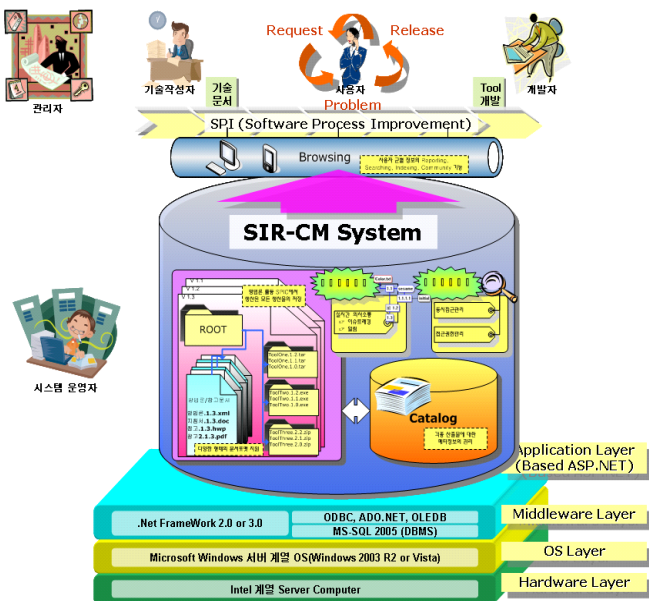
리한다. 이는 변경관리 혹은 버전관리라고 하는 기능으로 이 정보를 바탕으로 개발자들은 전체적인 프로젝트 흐름에 반영하여 개발한다. 게다가 하나의 큰 프로젝트에 대해서 여러 명이 수행할 경우 다수의 병렬 작업을 지원하는 작업공간관리 기능이 있다. 이러한 형상관리를 도입한 소프트웨어 형상관리 도구들이 다양하게 존재한다. 오픈소스 프로젝트에 많이 쓰이는 CVS를 비롯하여 이를 좀 더 개선시킨 서브버전이 있고 상용화된 툴로는 ClearCase, PVCS, VSS 등 많은 제품들이 있다[4][5].

3. SIR-CM

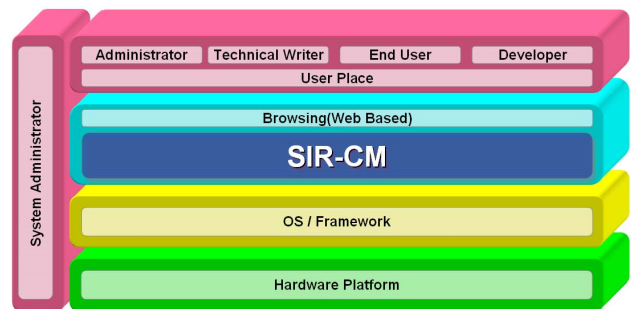
3.1 개요

본 논문에서 제안하는 SIR-CM(SPIC Integration Repository - Configuration Management) 시스템은 소프트웨어 프로세스 개선을 위한 지원 도구들의 실질적인 산출물들을 통합하여 저장, 관리할 수 있는 저장소이다. SIR-CM을 중심으로 각 도구로부터의 산출물들이 저장되고, 관리된다. 또한 SIR-CM시스템에 작업 공간 관리와 프로세스 관리를 포함하는 형상관리 기법을 적용하여, 과거 프로세스 정보들을 형상 관리의 측면에서 체계적으로 관리할 수 있도록 한다.

SIR-CM의 전체적인 시스템 구조는 그림 2와 같다. SPI를 지원하는 도구들의 모든 산출물을 입력 데이터로 저장하고 이를 각 도구에 서비스한다. SPI 지원 도구의 산출물로 CMMI의 지침서와 각 프로세스 별 측정데이터와 기타 방법론 및 도구를 포함한다. SIR-CM에 접근할 수 있는 유저로는 CMMI 심사위원, 프로젝트 관리자, 개발자, 시스템 운영자 등이 된다. SIR-CM에 저장되는 데이터는 형상 관리를 통해 사용자는 언제든지 브라우저로 다양한 형태의 서비스를 제공받는다. 이러한 SIR-CM은 기본적인 x86계열의 서버 플랫폼과 Microsoft사의 Windows Server를 바탕으로 .Net Framework 기반의 언어로 개발한다. 이를 바탕으로 한 SIR-CM의 계층도는 그림 3과 같다.



(그림 2) SIR-CM 시스템 구조



(그림 3) SIR-CM 계층도

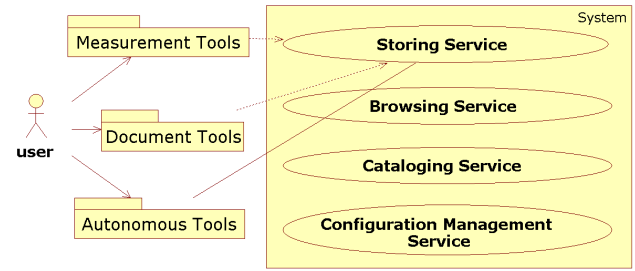
3.2 SIR-CM 분석

SIR-CM은 조직이나 프로젝트 단위의 프로세스를 CMMI를 통해 프로세스 개선활동을 돕는 도구들과 연계를 통해 좀 더 원활한 프로세스의 수행이 이루어지도록 한다. SPI활동에 있어서 개발자 및 해당 프로세스 영역의 사람에게 필요한 지침서나 자산을 단순히 저장하는 것에 그치지 않고, 사용자가 요구하는 데이터를 마이닝 하여 보고하거나 관련된 프로세스를 검색한다. 프로세스 영역별로 나오는 산출물을 표준화하여 변환하거나 조직에서 사용하는 지침서로 제공하고, CMMI 지원도구들 사이의 데이터 연계성을 토대로 해당 도구에서 사용하는 데이터 타입으로 사용할 수 있도록 자동화 하는 기능을 가진다. 기존에 개발된 도구들의 데이터를 수집, 재가공(XML형태 구조화 된 문서)하여 산출되는 메트릭 값을 저장하고 현재 프로세스에서 필요한 목표치를 확인하여 개선한다. 산출되는 데이터는 버전관리 및 작업공간 관리를 통해 저장소의 데이터 무결성을 보장한다.

본 논문에서는 다양한 CMMI 지원도구를 세 가지로 분류하여 표기한다. 측정값을 저장하는 측정 도구, 지침서와 같은 문서형태로 나오는 산출물을 위한 도구, 프로세스 수행 시 자동화된 처리를 수행하는 도구이다. SIR-CM은 다음과 같은 기능을 포함한다.

- 표준화된 문서규격을 지원 (XML, html)
- 기존 개발된 도구와 표준화 된 방법으로 통신 (SOAP, WSDL)
- 기존에 만들어진 도구는 많은 수정 작업 없이 재 사용하여 SIR-CM과 연동
- 프로세스의 영역별 산출되는 측정값은 측정을 위한 도구, 자동화를 지원하는 도구, 혹은 지침서 등을 저장
- 각종 산출물에 대한 메타정보 관리
- 프로세스 영역별 혹은 수준별로 분류를 통해 지침서, 측정값을 확인
- 프로젝트 관리자 혹은 개발자는 접근영역에 따라 다양한 Reporting 기능과 Searching기능을 제공
- 각 도구들에서 메시지 형태의 데이터가 송수신 되었을 때 그 데이터의 신뢰성을 보장
- 수행하는 조직단위나 프로젝트 단위로 구성
- 측정된 메트릭 값의 이전 결과도 저장 되어 필요할 경우 확인 가능
- 측정된 메트릭 값을 그래프, 뷰 등을 통해 Reporting 기능 제공

그림 4는 이러한 요구사항 명세를 유스케이스 다이어그램으로 나타낸 것이고, 표 1은 이 유스케이스 다이어그램에 대한 구체적인 설명을 나타낸다.



(그림 4) SIR-CM의 유스케이스 다이어그램

3.3 SIR-CM 메타데이터

CMMI 모델에 따라 SIR-CM에 저장하는 산출물은 크게 지침서, 22개의 프로세스 영역에 따른 측정값, 지원 도구와 도구내부 데이터가 있다. 표 2는 이 데이터에 대한 메타데이터를 표현한다.

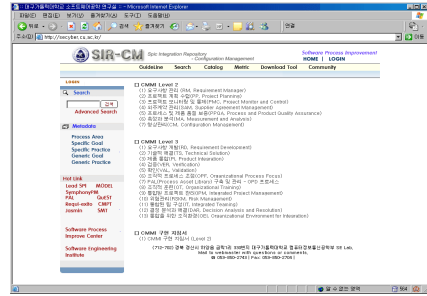
SIR-CM의 Meta-data는 기본적으로 CMMI 지침서를 프로세스 영역별로 가진다. CMMI에서는 연속적 혹은 단계적으로 표현할 수 있지만 SIR-CM에서는 지원도구들의 산출물이 속하는 프로세스 영역에 대해서만 명시하므로 구분하지 않는다. CMMI를 적용시킬 수 있는 Organization Unit(이하 OU라고 표기)은 조직, 프로젝트, 프로세스 3가지가 된다. 한 OU가 개선활동을 할 때 각 도구들 사이의 산출물들은 SIR-CM에 의해 이동한다. 또한 OU와 산출물, 그리고 추가되는 도구들을 모아 발전된 지침서를 제시한다. CMMI는 한 번 인증 받았다고 해서 영원히 지속되는 것이 아니기 때문에 SIR-CM은 OU의 자산으로써 활용가치가 높다.

<표 1> SIR-CM의 유스케이스

Actor & Usecase	Description
User	관리자, 기술 작성자 및 개발자 등의 프로젝트에 관련된 모든 man-power
Storing Service	사용자로부터 들어오는 모든 데이터를 저장하는 서비스 제공
Browsing Service	사용자가 원하는 데이터에 대한 searching과 reporting 등의 서비스 제공
Cataloging Service	사용자가 CMMI의 해당하는 PA를 확인하거나 메트릭 측정 요소들의 분류에 대한 레벨 별 분류
Configuration Management Service	저장된 데이터들의 형상관리 서비스 담당
Measurement Tools	CMMI를 통해 프로세스 개선활동 중 측정기법(ex: GQM, PSM)을 통한 메트릭 측정 도구들로 본 논문에서는 개념적인 도구로 도입
Document Tools	CMMI를 통해 프로세스개선을 수행하면서 나오는 문서 산출물들을 관리해주는 도구들로 본 논문에서는 개념적인 도구로 도입
Autonomous Tools	CMMI 프로세스 개선업무에 불필요한 사람의 공수를 제거하기 위한 도구들로 본 논문에서는 개념적인 도구로 도입

<표 2> SIR-CM 메타데이터

메타데이터 카테고리	메타데이터 이름
ORGANIZATION	ORGANIZATION_ID
	ORGANIZATION_NAME
	CMML_LEVEL
PROJECT	PROJECT_ID
	PROJECT_NAME
	CMML_LEVEL
	RELATE_PROCESS_AREA
PROCESS	PROCESS_ID
	PROCESS_NAME
	RELATE_PROCESS_AREA
GUIDELINE	GUIDELINE_ID
	LEVEL
	PROCESS_AREA
	SPECIFIC_GOAL
	SPECIFIC_PRACTICE
	GENERAL_GOAL
TOOL	TOOL_ID
	TOOL_NAME
	TOOL_URL
ARTIFACT	ARTIFACT_ID
	ARTIFACT_NAME
	PROCESS_AREA
	METRIC



(그림 7) SIR-CM 브라우저

Configuration Management를 제외하고는 인터페이스를 통한 연결로 상호 직접적인 메서드의 조작 등은 지원하지 않도록 설계하였다. 시스템은 앞서 언급되었던 기존 개발도구와 메시지를 통하여 데이터를 주고받는다. 그림 7은 위의 분석 및 설계의 결과를 기반으로 작성한 프로토타입 예이다.

3.4 SIR-CM 설계

SIR-CM은 크게 네 개의 컴포넌트의 단위로 구성된다. 각 컴포넌트의 단위는 Configuration Management와 Storing Service를 제외한 나머지 컴포넌트는 인터페이스를 통해 직접적인 연결을 최대한 배제하여 설계한다. 이것을 UML로 설계한 패키지 다이어그램은 그림 5와 같다.

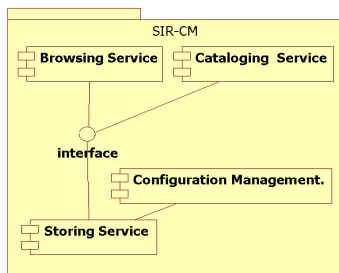
전체 프로그램이 하나의 패키지를 구성하고 있으며 각각의 기능을 컴포넌트단위로 설계하였다. Browsing Service와 Cataloging Service, Storing Service는 모두 직접적으로 연결되지 않고 인터페이스로만 연결되며 실제 데이터 이동도 메시지 단위로 이루어진다. 그림 6은 패키지 다이어그램에 대해 상세한 클래스 다이어그램을 나타낸다. Searching Service와 Reporting Service 두 가지의 클래스로 Browsing Service가 나뉜 것을 제외하고는 패키지 다이어그램의 컴포넌트와 동일하다고 할 수 있다. 외부에서 읽어지는 데이터는 모두 뷰(view)를 통해 보여지고 내부에서 데이터가 처리되는 것도 Storage와

4. 결론 및 향후연구

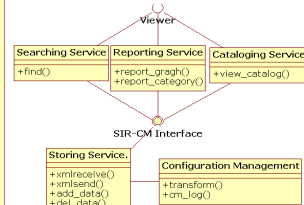
본 논문은 CMMI를 위한 지원 도구들의 이질적인 산출물들을 통합하여 저장, 관리할 수 있는 저장소인 SIR-CM 시스템을 제안했다. CMMI 지원 도구들 사이의 연계성과 효율적인 관리를 위해 저장소에 CM을 도입하였다. 이에 SIR-CM은 각 도구들의 산출물을 표준화된 데이터로 통합하여 관리한다. 이는 처음 SPI를 위해 CMMI의 인증을 받으려는 기업에게 충분히 필요한 시스템이라 판단되고 SPI에 있어서 더 나은 결과를 가져다 줄 것이다. 향후 본 논문에서의 분석 설계를 바탕으로 실질적인 시스템을 위한 프로토타입을 개발하고, 보다 효율적인 소프트웨어 프로세스 개선을 위해 자동화 기술에 대한 연구가 필요하다.

참고문헌

[1] Software Engineering Institute, "Capability Maturity Model Integration(CMMISM), Version 1.1", Carnegie Mellon University, August 2002.
 [2] Carnegie Mellon SEI site, <http://www.sei.cmu.edu/cmmi/cmmi.html>.
 [3] Mary beth Chrissis, Mike Konard, Sandy Shrum, "CMMI", Addison Wesley, 2003.
 [4] Anne Mette Jonassen Hass, "Configuration Management Principles and Practice", pp.3-27, Addison Wesley, 2002.
 [5] Assembling configuration Management Environments (for Software Development) site, http://www.cmcrossroads.com/bradapp/acme/scm-defs.html#SEL_Software_CMM.



(그림 5) 패키지 다이어그램



(그림 6) 클래스 다이어그램