

컴포넌트기반 원격프로세스 심사지원 도구의 설계와 구현

박정민^o 이은석
성균관대학교 컴퓨터공학과
{jpark^o, eslee}@selab.skku.ac.kr

Design and Implementation of Component based remote process supporting assessment tool

Jeong-Min Park^o Eun-Suk Lee
Dept. of Computer Engineering, Sungkyunkwan University

요 약

소프트웨어를 구성하는 요소들의 재사용은 소프트웨어 공학에서 가장 중요하게 생각하는 부분으로써 기존의 소프트웨어의 유지보수나 새로운 소프트웨어의 개발에서 생산성을 높일 수가 있다. 이러한 소프트웨어의 경제성, 재사용성을 위한 개발방법론이 CBD(Component Based Development)이다.

CBD는 소프트웨어 개발 방법론들 중의 하나로써 컴포넌트들을 특정 목적에 맞게 조립하고 배치하여 품질 보증된 높은 생산성의 컴포넌트를 재사용할 수 있게 한다. 이러한 장점 때문에 CBD가 유력한 차세대 개발방법론으로 주목받고 있는 것이다. 따라서 본 논문은 기존에 구축된 on-line 심사 프로세스 지원시스템을 컴포넌트기반 시스템으로 재구축하여 컴포넌트화의 효과를 알아보고자 한다.

1. 서 론

기존의 소프트웨어는 절차적 코딩 중심의 개발방식으로 수정사항이나 사용자의 추가적인 요구사항 발생시 설계와 분석과정 등을 처음부터 다시 수행해야 하기 때문에 개발시간과 유지비용이 많이 투입되는 비효율성이 존재하고 있었다. 이런 단점을 보완하고자 개발된 것이 바로 컴포넌트기반 개발(CBD: Component Based Development) 기술이다.

컴포넌트기반으로 소프트웨어를 개발한다면 소프트웨어를 처음부터 개발함으로써 생기는 비용이나 위험요소들을 감소시킬 수 있고, 소프트웨어공학에서 주요 이슈라고 있는 문제들에 효과적으로 대응이 가능하기 때문에 기존 시스템보다 확장성, 이식성, 유지보수성이 높다.

본 논문에서는 화상 컴포넌트, 채팅 컴포넌트, 화이트보드 컴포넌트, 파일 컴포넌트 등의 재사용 가능한 CSCW(Computer Supported Cooperative Work) 컴포넌트들을 이용하여 기존의 on-line 심사 프로세스 시스템을 컴포넌트기반 시스템으로 재구축하여 컴포넌트화의 효과를 알아보고자 한다.

2. 관련연구

2.1 소프트웨어 재사용

소프트웨어 재사용(reuse)은 소프트웨어 생산과 유지보수에 있어서 기존에 개발된 소프트웨어의 표준화된 공통부품의 사용을 극대화함으로써 품질과 생산성을 향상시키며 유지보수에 능동적으로 대처할 수 있도록 하는 계획되고 체계화된 행위들의 집합이다 [2]. 재사용을 통해 이용되는 부품들은 이미 기능과 역할을 충분히 검증받았기 때문에 소프트웨어의 생산성과 품질 향상에 기여할 수 있다[3][4].

컴포넌트기반 방법론은 과거 구조적 방법이나 객체지향 기술이 제대로 해결하지 못한 개발 생산성, 소프트웨어 재사용성, 시스템 유지 보수성을 향상시킬수 있는 대안으로 주목받고 있으며, 이외에도 요구사항 획득 및 다른 소프트웨어의 생산, 납기 지연, 비용 초과 등 소프트웨어 위기를 초래한 고질적인 문제들을 해결할 수 있는 방안이 되고 있다[5][6][7].

2.2 CBD(Component Based Development) 프로세스

CBD는 독립적인 비즈니스 프로세싱 모듈로서 컴포넌트들의 통합에 의한 높은 융통성과 유지보수성을 가진다. 표 1 에서 나타낸 것처럼 CBD 프로세스의 진행은 관점에 따라 두 가지 부분으로 구분된다. 그러므로 실질적인 프로젝트 수행에서 요구되는 필수적인 작업들을 중심으로 재구성한 큰 그림은 그림 1과 같다. 컴포넌트 저장소 중심의 재사용 요소의 "획득-이해-적용"이라는 기본적인 재사용 원칙에 준하며 컴포넌트 확보 과정에서는 도메인 공학을 기반으로 하는 재사용 컴포넌트의 식별 및 추출을 위한 세부 프로세스를 포함하고 있다. 기존의 컴포넌트들을 다양한 형상으로 조립하고 개조(Adaptation)하여 구축될 수 있는 넓은 관점을 표현한다[8][9].

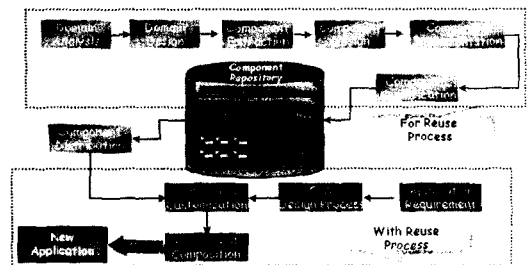


그림 1 CBD 프로세스

표 1 두 가지 관점에서의 CBD 프로세스

구분	설명
Development for Component Reuse	컴포넌트의 식별, 추출, 개발에 관련된 일련의 행위들로 재사용 가능성이 있는 컴포넌트를 생산하는 과정
Development with Component Reuse	이미 존재하는 컴포넌트와 관련 결과물들의 활용을 극대화함으로써 시스템 구축을 지원하는 과정

2.3 기존 on-line 심사 프로세스 시스템

기존에 구축된 SPICE 심사지원 시스템은 전통적기반 개발방법론을 적용한 시스템으로 각 기업의 내부 심사자와 외부 심사자간의 원격 인터페이스와 각각의 산출물들을 온라인으로 평가할 수 있는 장점이 있었다. 하지만 온라인 심사 프로세스 시스템은 요구사항 변경시 시스템 확장과 유지보수가 용이 하지 않고, 최신 기술의 지속적 수용이 어렵다. 그림 2는 컴포넌트 기반 개발방법과 전통적 기반 개발 방법론을 비교한 것이다.

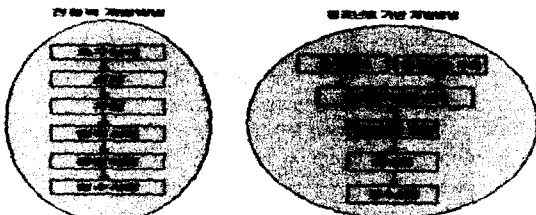


그림 2 전통적 기반 개발 방법론과 컴포넌트기반 개발방법론

3. 제안시스템

3.1 요구사항 분석

- 심사원들 간의 자연스러운 대화와 현장 심사의 인터뷰를 대체할 수 있는 화상 대화 기능 및 보안 요소
 - 채팅기능
 - 화상회의 기능
 - 쪽지 기능
- 심사원들 간의 토의에 사용되는 프리젠테이션 공유
 - 화이트보드 기능
- 문서 심사를 원격으로 수행할 수 있는 기능
 - 파일뷰어 기능
- 심사에 필요한 도움말을 제공하는 기능
- 심사 프로세스 지원 기능

3.2 컴포넌트 추출

표 2는 요구사항으로부터 예상되는 컴포넌트를 추출한 것이다.

표 2 예상 컴포넌트 추출

예상 컴포넌트	내용
화상회의 컴포넌트	현장 심사의 인터뷰를 대체할 수 있는 화상 대화 기능
채팅 컴포넌트	보안 요소
화이트 보드 컴포넌트	심사원들 간의 토의에 사용되는 프리젠테이션 공유 기능
파일뷰어 컴포넌트	문서 심사를 원격으로 수행할 수 있는 기능 및 도움말 제공 기능
심사 기능 컴포넌트	심사 프로세스를 지원하는 기능

3.3 컴포넌트 기반 설계

UML(Unified Modeling Language)은 컴포넌트 설계 기술을 잘 표현할 수 있는 방법론으로써 컴포넌트의 설계와 동적, 물리적 설계를 제공한다. 컴포넌트 인터페이스와 각 공용 기능에 대한 순차도를 작성하여 인터페이스와 컴포넌트안의 객체 사이의 관계를 보여준다. 그림 3, 그림 4는 컴포넌트 워크플로우와 구조도를 정의한다.

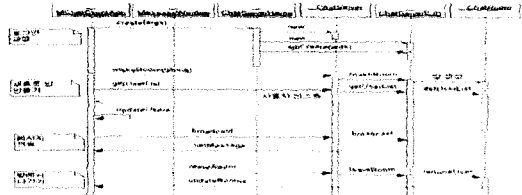


그림 3 컴포넌트 워크플로우

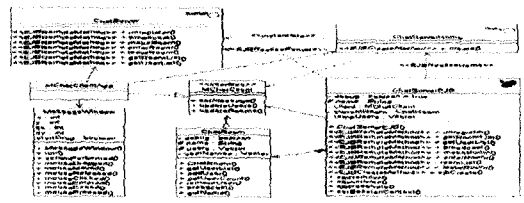


그림 4 컴포넌트 구조도

3.4 시스템 구성

본 논문에서 제안하는 컴포넌트기반 원격프로세스심사 지원도구는 그림 5의 시스템 전체 구성도 에서도 볼 수 있듯이 SPA(Software Process Assessment)를 위한 컴포넌트와 CSCW 컴포넌트로 구성한다. CSCW 컴포넌트는 화상 컴포넌트, 화이트 컴포넌트, 채팅 컴포넌트 등으로 컴퓨터기반의 협동작업을 지원하고, SPA 컴포넌트는 소프트웨어 프로세스의 문서 심사를 지원한다. 각각의 모듈의 정보공유는 네트워크기반으로 이루어지며, 주요 기능 단위가 컴포넌트화 되어 있다. 표 3은 각 컴포넌트의 기능을 정리해 놓았다.



그림 5 컴포넌트기반 시스템 구성도

표 3 컴포넌트 기능

컴포넌트 종류	기능
화상 컴포넌트	심사원들 간의 원활한 의사소통 및 의견 조정을 지원
채팅 컴포넌트	화상 컴포넌트의 보완적인 통신 지원
화이트보드 컴포넌트	화상 컴포넌트, 채팅 컴포넌트의 보완적인 것으로 의견 조율시 이해도를 높임
파일 뷰어	심사과정의 도움말 기능을 지원, 프리젠테이션을 지원
파일 컴포넌트	SPA를 위해서 문서 심사를 지원함
심사 프로세스 컴포넌트	반복주기마다의 심사결과를 데이터베이스에 저장 조율가능

4. 구현 및 평가

4.1 시스템 구현

EJB 기반의 컴포넌트 플랫폼에서 실제 컴포넌트 빈을 구현한다. 데이터베이스 스키마를 구축하며, 최종적으로 EJB서버에 컴포넌트 빈을 배치하기 위해 관련 클래스들을 JAR파일로 묶어준다. 구현된 빈에 대해서는 완성도를 높이고, 에러를 찾기 위한 테스트 단계를 수행한다. 그림 6은 컴포넌트 기반의 원격 프로세스 심사도구의 실행 화면이다.

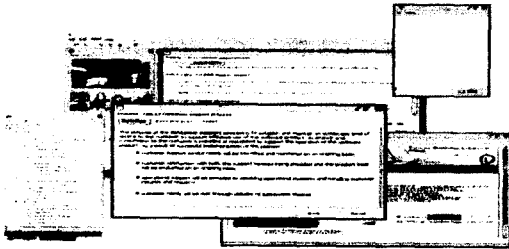


그림 6 컴포넌트기반의 원격 프로세스 심사 시스템

4.2 컴포넌트기반 시스템의 효과

컴포넌트기반 시스템은 재사용성, 유지보수성, 적응성, 확장성의 측면에서 우수성을 가진다.

- **재사용성:** 컴포넌트 소프트웨어 개발시 이미 검증받은 컴포넌트를 재사용하여 시스템을 확장하므로 재사용성이 높다.
- **유지보수성:** 반복을 기반으로 하는 프로세스로서 가시성 확보를 위한 전략은 객체지향방법론과 같지만, 체계적인 관리를 위해 컴포넌트 중심 아키텍처를 초기에 작성하여 컴포넌트를 중심으로 개발 공정상의 관리 난이도를 최소화하기 때문에 유지보수성이 높다.
- **적응성:** 컴포넌트를 중심으로 기간 시스템과 웹기반 다계층 아키텍처 등 다양한 환경에서 완전한 추상개념을 유지할 수 있어 이를 통한 적응성 확보가 유리하다.
- **확장성:** 보다 큰 단위의 컴포넌트를 중심으로 반복을 계획하여 프로세스의 이해도를 높일 수 있고, 추상개념을 기존 프로세스 및 환경을 고려하여 확장 구축을 할 수 있다.

5. 시스템 평가

평가는 컴포넌트를 ISO/IEC 9126의 소프트웨어 품질 특성을 기준으로 평가하였다. 표 4는 제안시스템의 평가 기준이다. 표 5은 평가기준에 따라 CSCW 컴포넌트의 완성도를 정량적으로 평가하였다. 표 6은 기존 시스템과 컴포넌트 기반 시스템을 정성적 비교평가 하였다.

표 4 평가기준

속성유형	평가제도	속성값	근거
소프트웨어 품질 특성	Y: 만족 N: 만족안함, NA: 적용불가능	0 ≤ X ≤ 100 일 때, X ≥ 80 이면 A 70 ≤ X < 80 이면 B 60 ≤ X < 70 이면 C 50 ≤ X < 60 이면 D	컴포넌트에 관한 근거를 기술

표 5 컴포넌트 완성도

속성유형	평가제도	속성값	근거
기능성	Y	A	각각의 컴포넌트별 관련 기능들이 충실히 수행됨
신뢰성	N	B	다수의 시스템 결속시 시스템 유지를 불만족
효율성	Y	A	컴포넌트기반으로 재사용면에서 효율성이 있음
유지보수성	Y	A	스스코드 변경시 재배치가 용이
이식성	Y	B	대체로 만족하나 이식성에 대한 속성 어려움

표 6 컴포넌트기반 시스템과 전통적 기반 심사 시스템 비교

구분	평량제기반 심사 시스템	컴포넌트기반 시스템
처리방식	• 제한적인 분산처리 방식	• 완전한 분산처리 방식
복잡	• 개발난이도가 높음 • 운영 난이도가 높음	• 개발난이도가 상대적으로 낮음 • 운영난이도가 상대적으로 낮음
합류 시스템 확장성	• 확장 가능하나 쉽지 않음	• 확장성이 매우 뛰어나
장애 복구	• 어려움	• 강에민인 식별 용이, 빠른 복구지원
유지보수성	• 쉽지 않음	• 모듈성이 높고, 유지보수성 뛰어나
프랜져션	• 처리가능	• 프랜져션 지원 우수
정보기술	• 기존의 기술에 의존	• 최신 기술의 지속적 수용 가능

6. 결론

본 논문에서는 컴포넌트기반의 원격프로세스 심사 지원 도구를 재구축하여, 컴포넌트기반의 시스템이 전통적 기반의 기존 시스템보다 우수함을 나타내었다.

그러나 CBD에 경험이 없는 조직이라고 가정하더라도 시스템 기능을 위한 프레임워크와 컴포넌트들이 있을 수 있겠지만, 프로젝트마다 다양한 요구 사항을 모두 충족시키지 못하기 때문에 일정 부분의 비즈니스 컴포넌트는 새로 만들 수밖에 없다. 따라서 보다 정제된 시스템을 위해서 추가적인 변경과 원격에서 프로세스가 체계적으로 진행하기 위한 추가적인 요구사항과 그에 대한 컴포넌트의 확장을 향후의 과제로 남긴다.

참고문헌

- [1] SEI in Carnegie Mellon University, http://www.sei.cmu.edu/str/descriptions/cbsd_body.html
- [2] Carma McClure, The Three Rs of Software Automation : Re-Engineering, Repository, Reusability, Prentice Hall, 1992.
- [3] Bruce A. Burton et al., "The Reusable Software Library," IEEE Software, Vol. 4, No. 4, pp. 25-33, 1987.
- [4] Jingwen Cheng, "A Reusability-Based Software Development Environment," ACM SIGSOFT, Vol. 19, No. 2, pp. 57-62, 1994.
- [5] Eliseo Mambella, Roberto Ferrari et al, "An Integrated Approach to Software Reuse Practice," SSR '95, Seattle WA, USA, pp. 63-71, 1995.
- [6] Keith Short, Component Based Development and Object Modeling, Sterling Software, 1997
- [7] 한국소프트웨어컴포넌트컨소시엄, SW컴포넌트 개발산출물 표준, 2002
- [8] 차정은, 김행근, "CBD 프로세스 지원을 위한 컴포넌트 저장소의 구축", 정보과학회지 제29권 제7호, 2002. 8
- [9] 한국전자통신연구원, 마르미-III v3.0, 2003
- [10] Bulter Group, "What is a component", Interact, 1998
- [11] Peter Herzum, Oliver Sims, Business Component Factory : A Comprehensive Overview of CBD for the Enterprise, OMG press, December, 1999