프로세스 패턴에 기반한 컴포넌트 품질보증 프로세스의 구축

황 선 명[†] · 김 길 조[††] · 김 진 삼[†††]

요 약

컴포넌트 기반 소프트웨어를 개발하기 위해서는 검증되고 표준화된 소프트웨어 컴포넌트가 요구된다. 본 논문에서는 재사용가능한 프로세스 패턴을 기반으로 하여 컴포넌트 품질 보증 프로세스를 구축하는 접근방법을 제시하고 있다. 프로세스 패턴을 기술하기 위한 메타모형과 컴포넌트 품질 보증을 위한 프로세스 패턴들을 정의하여, 컴포넌트 개발 상황에 맞게 품질보증 프로세스를 구축할 수 있도록 하였다. 또한 이 접근방법을 실제 프로젝트에 적용한 사례를 보여주고 그 장단점을 토의하였다.

A Component Quality Assurance Process based on Process Patterns

Sun-Myung Hwang[†] · Gil-Jo Kim[††] · Jin-Sam Kim[†††]

ABSTRACT

Developing a component-based software requires verified and standardized software components. This paper presents an approach to build component quality assurance process based on process patterns. To describe process patterns, a metamodel is suggested and several process patterns for component quality assurance are developed that can play the building blocks to establish a domain-specific quality assurance process for software components. The process pattern approach was applied to real projects and its advantages and disadvantages are discussed.

키워드 : 컴포넌트 품질보증(Component Quality Assurance), 프로세스 패턴(Process Pattern), 메타모델(Metamodel)

1. 서 론

소프트웨어는 기술의 발전과 다양한 사용자의 요구를 수용하면서 개발되어야 하나 대부분의 소프트웨어는 품질 미비, 납기 및 비용 초과 등의 문제점을 안고 개발된다[1]. 최근 품질이 보장된 소프트웨어 재사용(reusability)을 통해 시스템을 개발함으로써 이러한 문제점을 해결하려고 하고 있으며, 소프트웨어 컴포넌트 기술이 핵심 역할을 수행하고 있다.

소프트웨어 컴포넌트(이하 컴포넌트)는 컴포넌트 모형을 따르면서 추가적인 변경없이 독립적으로 생산되고, 구매되고, 배치될 수 있는 소프트웨어 요소들을 말한다[2]. 컴포넌트의 기능은 인터페이스를 통해 표현되고, 개발자는 인터페이스를 통해 개발을 하고, 사용자는 인터페이스를 통해 사용할 수 있다. 표준 컴포넌트 모형에는 OMG(Object Management Group)의 CORBA(Common Object Request Broker Architecture), MicroSoft사의 COM+(Component Object Model), Sun 사의 EJB(Enterprise Java Beans) 등이 있다. 컴포넌트 기반 개발(Component Based Development, 이하 CBD)은 표준화되고 검증된 컴포넌트를 활용하여 소프트웨어 시스템을 빠른 시간에 구축하려는 시도이다. 그 결과 CBD에서는 개발 위주의 성숙도 모형에 대한 중요성이 줄어들고, 대신 구매 위주의 성숙도의 중요성이 높아진다는 특징이 있다[3]. 이러한 컴포넌트의 개발 및 활용을 지원하기 위한 다양한 도구들과 CBD 방법론 등이 개발되었으며, 컴포넌트 기술은 금융시스템, 생산시스템, 통신 시스템 등 다양한 분야에 적용되고 있다.

컴포넌트 개발에는 높은 개발 노력, 명확한 요구사항 식별, 유지보수에 대한 고려, 변경에 대한 신뢰성 및 민감성 등에 따른 위험 요소들이 있으며, 특히 컴포넌트의 신뢰성(reliability), 인증(certification) 여부, 예측가능성(predictability)의 확보는 컴포넌트의 확산을 위해 필수사항이라고 할 수 있다[4]. 많은 조직들이 코드 공유 또는 디자인 패턴 등을 통해 비공식적인 재사용을 하고 있으나 복수의 애플리케이션 및 프로젝트에서 컴포넌트를 충분히 체계적으로 재

※ 본 연구는 한국과학재단 목적기초연구(R01-2001-000-00343-0(2003)) 지원으로 수행되었음.
† 종신회원 : 대전대학교 컴퓨터공학과 교수
†† 정 회 원 : 에이비앤아이(주) 수석컨설턴트
††† 정 회 원 : 한국전자통신연구원 책임연구원
논문접수 : 2003년 2월 25일, 심사완료 : 2003년 6월 3일

사용하지는 못하고 있다.

컴포넌트의 특성 및 컴포넌트 프레임웍이 시스템 특성을 예측하는데 사용될 수 있지만, 컴포넌트가 복합적이거나 다른 컴포넌트를 채택하여 개발된 경우 시스템 특성을 예측하는 것은 매우 어렵다[5]. 따라서 시스템의 특성을 예측하기 위해서는 컴포넌트 개발 프로세스 특성을 포함시키는 것이 바람직하다[6]. 비록 다양한 CBD 방법론[7-10]이 제안되었고 일부는 컴포넌트의 품질 보증을 강조하고 있으며[12], 품질 목표를 수립하고 이를 달성하기 위한 컴포넌트 품질 보증 프로세스가 제안되었다[13].

본 논문에서는 컴포넌트 품질 보증을 위한 프로세스 패턴(Process Patterns) 접근방법을 제시하고 있다. 프로세스 패턴은 소프트웨어를 성공적으로 개발하기 위한 일반적인 기법, 작업 또는 활동들의 집합들로 성공적인 접근방법으로 알려진 것이다[14]. 본 논문에서는 컴포넌트 품질보증에 공통적으로 적용가능한 프로세스 패턴들을 제시하였고, 이 패턴들을 이용하여 개발 대상 컴포넌트의 특성, 환경, 프로젝트의 상황 등을 고려하여 적절한 컴포넌트 품질 보증 프로세스를 구축할 수는 방법을 제시하고 있다. 프로세스 패턴을 기술하기 위한 메타모형을 정의하였고, 컴포넌트 품질 보증을 위한 핵심 패턴을 도출하고 패턴 기술 체계에 따라 베스트 프랙티스로 구성된 프로세스 패턴들을 정의하고 있다. 이러한 프로세스 패턴들은 재사용 가능하고, 상황에 따라 재구성 가능한 하나의 훌륭한 지식베이스의 역할을 수행한다.

2장에서는 본 논문와 관련된 기존의 관련 연구들을 기술하고 있다. 3장에서는 컴포넌트 품질 보증을 위한 프로세스 패턴 접근 방법을 설명하고 있고, 4장에서는 그 적용 예를 보여주고 있다. 5장에서는 본 접근방법의 타당성을 설명하고, 6장에서는 결론 및 향후 연구방향을 제시하고 있다.

2. 관련 연구

2.1 컴포넌트 개발 접근방법 및 품질 보증

컴포넌트 개발은 전통적인 개발 방법과 다른 방법으로 개발된다. 컴포넌트 기반 개발 방법론에는 Rational Unified Process(RUP)[15], 카탈리시스(Catalysis)[9], Sterling의 CBD 96[7], SELECT Perspective[8], UML Components[11], 한국전자통신연구원의 마르미-III[10] 등이 있다. 이러한 방법론들은 많은 공통점을 가지고 있으며, 단지 표기법이나 사용하는 모형 요소에서 약간의 차이를 보이고 있다. Cai 등[12]은 컴포넌트 개발 프로세스 및 전체 시스템 개발 프로세스를 포괄하는 품질 보증 모형과, 컴포넌트 요구분석, 컴포넌트 개발, 컴포넌트 인증, 컴포넌트 조정, 시스템 아키텍쳐 설계, 시스템 통합, 시스템 시험, 시스템 유지보수의 8개에 대한 프로세스를 제시하고 있다. 컴포넌트 인증은 컴포넌트를 외부에서 조달하는 경우 시스템 요구사항을 충족하는지를 판단하기 위한 프로세스이다.

EU의 지원으로 수행되는 OOSPICE 프로젝트에서는 기존의 SPICE 모형을 확장시켜 컴포넌트 기반 개발을 포함하여 프로세스를 심사하고 개선할 수 있는 모형을 개발하고 있다[6]. 이 프로젝트의 주목적은 통합 CBD 프로세스 메타모형, 심사 방법론 및 지원도구, 컴포넌트 공급자의 능력 프로파일, CBD 개발방법론을 개발하는 것이다.

컴포넌트 품질 평가시에는 ISO/IEC 9126[16]에 정의되어 있는 품질 속성(quality attributes)들의 의미를 달리 해석할 필요가 있다. 예를 들어 Allen[17]은 10개의 CBD 관련 품질 속성으로 정확성(accuracy), 적응성(adapatability), 명확성(clarity), 유연성(flexibility), 상호운영성(interoperability), 유지보수성(maintainability), 수행성(performance), 대체가능성(replaceability), 재사용성(reusability), 확장성(scalability)을 정의하였다. 그 중 명확성(clarity), 유연성(flexibility), 재사용성(reusability), 확장성(scalability)는 ISO/IEC 9126 표준의 특성, 부특성에 포함되어 있지 않은 CBD에 특별한 품질 속성이라고 볼 수 있다. Woodman[18] 등은 여기에다 신뢰성(reliability)을 추가하여 11개의 품질속성을 제시하고 있다.

신뢰성 있는 컴포넌트가 되기 위한 방법 중의 하나는 제3자에 의한 컴포넌트의 인증이다. 인증(certification)은 어떤 것과 연관하여 특성값을 검증하는 과정이며 타당성 확인의 증거로 인증서를 주는 것이다. 제 3자에 의한 인증은 소프트웨어 컴포넌트가 잘 정의된 표준에 따르고 있음을 보장하는 방법이고, 이러한 인증에 근거하여 컴포넌트들이 결합될 경우 신뢰성이 높은 시스템이 구축될 수 있다는 것이다. 이러한 컴포넌트 인증에 관련된 경험들은 노르웨이 에릭슨사[19] 및 미국 NASA[20]에서 찾아볼 수 있다.

2.2 프로세스 패턴 접근방법

패턴(Patterns)은 공통의 문제점 또는 이슈에 대한 일반적인 해결책을 기술한 것으로, 특정 문제에 대한 자세한 해결방안을 알려 준다[21, 22]. 패턴은 작고, 함축적이고, 자체로 충분한 지식의 조각들(small, coherent, self-sufficient piece of knowledge)로서 다양한 상황, 다양한 규모에 적용이 가능하다. 패턴은 규모와 무관하며(scale-invariant), 다양한 추상화 수준에도 적용가능하다[23].

프로세스 패턴(Process Pattern)은 소프트웨어를 성공적으로 개발하기 위한 일반적인 기법, 작업 또는 활동들의 집합들이다. 즉 프로세스 패턴은 성공적인 접근방법으로 여겨지는 것이라고 할 수 있다.

프로세스 패턴 개념의 기본적인 개념은 프로세스 지식을 구조적이고 잘 정의된 모듈화된 방식으로 기술하고 문서화하여 재사용하자는 것이다. 즉 동일한 프로세스 패턴을 다양한 상황에서 적용할 수 있게 되면, 적용이 쉽고 업무의

중복성을 줄일 수 있다.

프로세스 패턴은 개발 프로세스를 안내하고, 개발산출물을 문서화하는데 사용될 수 있다. 프로세스 패턴의 중요한 점은 무엇을 해야 하는지 정확히 알려주는 것이라고 할 수 있다.

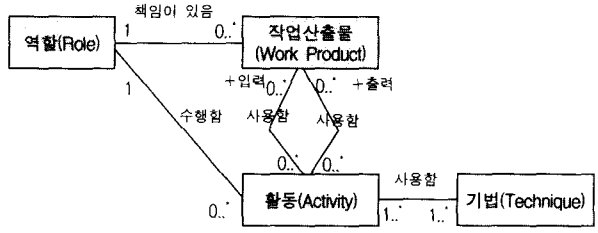
이러한 프로세스 패턴이 도입되는 배경은 다음에 잘 기술되어 있다[24].

- 이해성(understandability) 향상 : 대부분의 전통적인 프로세스들은 한덩어리(monolithic)로 되어 있고, 학습이 어렵다. 실무자들은 이해가 쉽고 쉽게 적용할 수 있는 잘 구조화된(well-structured) 패턴과 같은 프로세스 모형을 원한다.
- 유연성(flexibility) 향상 : 많은 개발자들은 상황에 적절하지 않은 개발 프로세스로 인해 어려움을 느끼는 경우가 많다. 그 결과 extreme programming(XP)과 같은 프로세스에 대한 관심도 높아졌다.
- 정확성(precision) 향상 : 개발 프로세스는 자동 실행 및 정형 분석이 가능하도록 일부분은 정형화될 필요가 있다.
- 구성품 구조(fractal structure) : 컴포넌트 기반 개발방법이 도입되면서, 개발 프로세스도 패턴 형태의 작은 요소를 가지는 구조들을 결합하여 사용할 수 있다.

프로세스 패턴은 기존 모형으로부터 특별한 개발 프로세스를 조립할 수 있도록 한다. 기존 모형으로부터 특별한 개발 프로세스를 조립하기 위해서는 일반적인 프로세스 모형의 빌딩블록과 그들간의 관계를 식별하고 그 관계를 명확히 정의하고 있는 프로세스 프레임웍(process framework)이 필요하다. 프로세스 프레임웍은 다양한 기존 프로세스 모형을 조합할 수 있게 한다 또한 프로세스 프레임웍은 기존의 다른 모형들의 자산 및 장점을 통합하여 다양한 프로젝트 요구사항 및 상황에 유연하게 적응할 수 있는 공통 근거(common basis)가 된다.

3. 컴포넌트 품질보증을 위한 프로세스 패턴

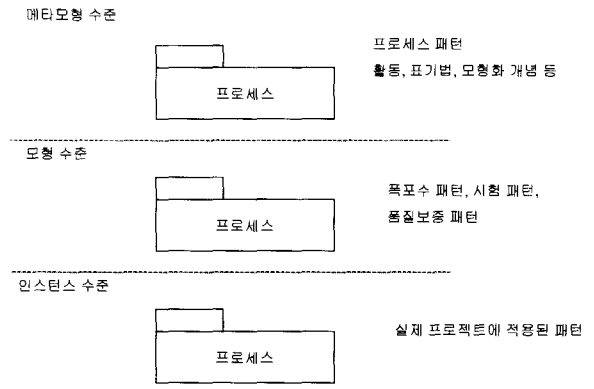
3.1 프로세스 패턴 기술 방법



(그림 1) 프로세스 패턴 메타모형의 구조

프로세스 패턴 메타모형은 프로세스 패턴을 정의하기 위한 것이다. (그림 1)에서 보는 바와 같이 프로세스 패턴 메타모형은 역할, 활동, 작업산출물, 기법으로 이루어진 프로세스를 정의하기 위한 표기법이다.

위에서 정의한 프로세스 패턴 메타모형에 따라 다음과 같은 수준의 프로세스들이 정의될 수 있다. 이러한 수준은 OMG의 소프트웨어 프로세스 메타모형인 SPEM[25]의 M0, M1, M2수준에 따르면 메타모형은 M2에 해당한다.



(그림 2) 프로세스 패턴의 수준

패턴을 적용하는 방법 및 적용 결과에 대한 정보를 제공하기 위해 본 논문에서는 다음 표와 같은 템플릿을 사용한다.

〈표 1〉 프로세스 패턴 기술 항목

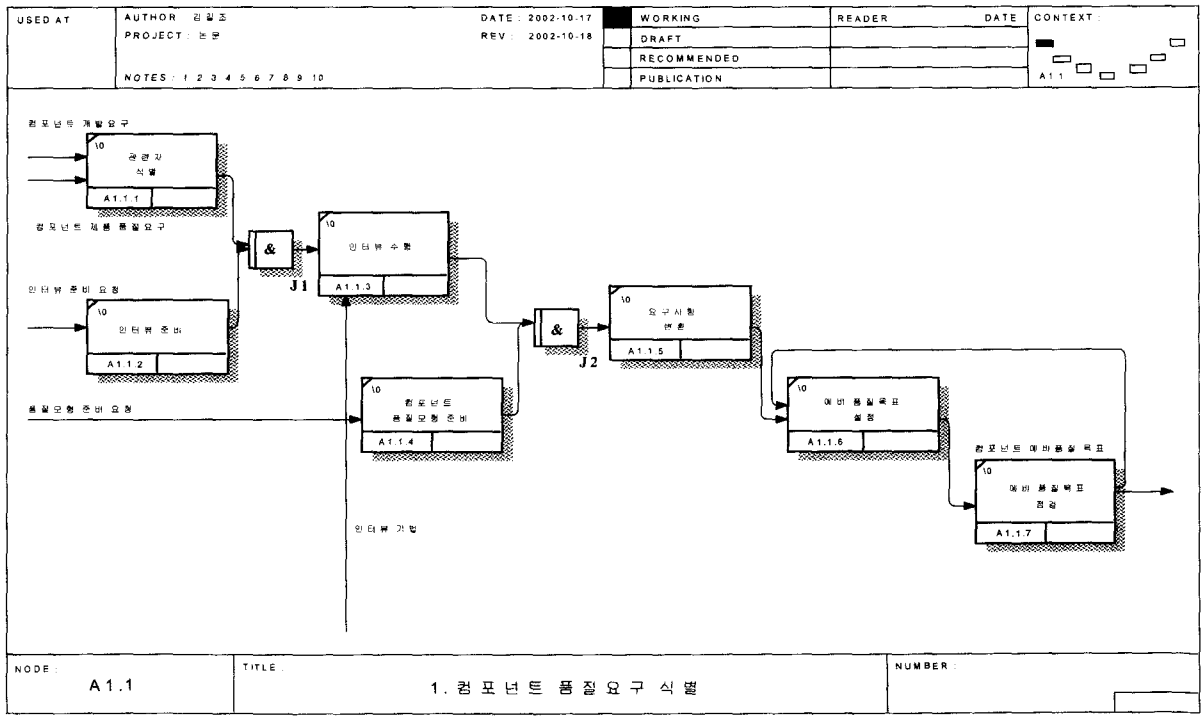
항 목	프로세스 패턴 기술 내용
패턴명	패턴의 이름
키워드	패턴에서의 키워드
목 적	패턴의 당위성 및 의도에 대한 간략한 설명. 패턴에서 다루고 있는 특정 개발에 관한 이슈나 문제를 언급함
문제 영역	문제, 적용 조건, 패턴의 목적을 보여주는 시나리오. 패턴을 적용하기 위한 사전 조건, 적용가능한 상황
해결책	이 패턴이 위에서 언급한 문제 영역을 어떻게 해결해 줄 수 있는지를 언급함
활동수행자	이 프로세스 패턴에서 중심역할을 수행하는 사람들과 그들의 역할
활 동	패턴을 구성하는 작업들의 목록. 작업들간의 흐름도는 별도의 그림으로 표현함
성공적 구현결과	이 패턴이 성공적으로 구현되었음을 알 수 있는 결과
입력물	프로세스 패턴에서 사용되는 입력
출력물	프로세스 패턴 적용 결과로 생성되는 출력
기법, 도구, 템플리트	프로세스 패턴을 활용할 때 사용되는 기법, 도구, 관련 템플리트
관련 패턴	대안이 되거나 유용한 관련 패턴

3.2 컴포넌트 품질보증을 위한 프로세스 패턴의 정의

본 논문에서는 컴포넌트 품질보증 프로세스를 위해 다음의 7개 핵심 프로세스 패턴을 정의하고 있다. 각 핵심 프로세스 패턴은 라이브러리에 저장하여 향후 활용할 수 있어야 한다. 각 패턴에 대한 설명은 다음과 같다.

패 턴 명	컴포넌트 품질요구 식별	키 워 드	컴포넌트 품질요구, 컴포넌트 품질목표
목 적	개발대상 컴포넌트의 사용자 요구사항을 파악하는 단계에서 컴포넌트의 품질에 관한 요구사항을 확인하여 품질목표를 설정하기 위한 것이다.		
문제 영역	소프트웨어와 마찬가지로 컴포넌트의 품질에 관한 사용자의 요구사항을 획득하기 힘들다. 즉 사용자는 컴포넌트 품질에 대한 개념이 부족하고, 사용자의 요구사항 중 품질 요구사항으로 변환하기가 어려우며, 다양한 사용자를 고려할 필요가 있다.		
해 결 책	컴포넌트 품질 요구사항은 컴포넌트 관련자로부터 획득할 필요가 있다. 이 활동은 시장의 요구사항, 경쟁제품의 품질, 비용, 개발 기간 등의 다양한 사항들을 고려하여야 하므로, 품질 전문가에 의한 인터뷰 기법을 활용하여 요구사항을 획득하고, 이 요구사항을 품질모형 기반의 품질요구사항으로 변경하여 개괄적인 품질 목표를 수립하도록 한다. 이 품질목표가 가능한지를 점검하고 불가능한 경우 품질 목표를 다시 수립한다.		
활 동	1. 관련자 식별 2. 인터뷰 준비 3. 인터뷰 수행 4. 컴포넌트 품질모형 준비 5. 요구사항 변환 6. 예비 품질목표 설정 7. 예비 품질목표 점검	활동수행자	프로젝트 관리자 : 컴포넌트에 대한 요구를 검토하고, 예비 품질목표를 설정함 품질전문가 : 인터뷰를 통해 예비 품질 목표를 설정하고, 그 타당성을 검토함
성공적 구현결과	컴포넌트 품질에 대한 사용자의 요구가 파악된다. 컴포넌트에 대한 예비 품질 목표가 설정된다.		
입 력 물	컴포넌트 개발 요구 컴포넌트 품질 요구	출 력 물	컴포넌트 예비 품질 목표
기법, 도구, 템플리트	인터뷰 기법 컴포넌트 품질모형(4장 참조) 컴포넌트 예비 품질목표 프로파일	관련 패턴	컴포넌트 품질목표 설정 패턴이 연결될 수 있다.

(그림 3) 컴포넌트 품질요구 식별 프로세스 패턴



(그림 4) 컴포넌트 품질요구 식별 패턴의 활동 흐름도

- 컴포넌트 품질요구 식별 : 개발대상 컴포넌트 사용자들의 품질 요구사항을 식별하고 예비 품질목표를 수립한다.
- 컴포넌트 품질목표 설정 : 컴포넌트의 예비 품질목표를 기반으로 상세 품질목표를 수립하고, 컴포넌트 품질평가 기준을 수립한다.
- 프로세스 메트릭 계획 수립 : 컴포넌트의 상세 품질목표에 따라 컴포넌트 개발 프로세스에서 측정되어야 할 메트릭을 식별하고, 메트릭 측정 계획을 수립한다.
- 데이터 수집 : 메트릭 측정계획에 의거하여 적절한 시점에서 측정데이터를 수립한다.
- 메트릭 통제 : 측정데이터를 통해 메트릭을 계산하여 필요시 시정조치를 취한다.

- 컴포넌트 품질 평가 : 설정된 컴포넌트 품질목표에 의거하여 컴포넌트 품질을 평가하고 품질보고서를 작성한다.
- 경험베이스 갱신 : 프로세스 패턴의 유효성을 판단하고, 축적된 경험을 패키지화여 경험 베이스를 갱신한다.

다음은 패턴 기술방법을 활용하여 "컴포넌트 품질요구식별" 프로세스 패턴을 기술한 예를 보여준다.

위 패턴에서 제시된 활동들간의 관계는 다음과 같이 IDEF [26]의 형식으로 표현할 수 있다.

컴포넌트 예비 품질 목표는 컴포넌트 품질모형을 기반으로 하여 다음과 같은 프로파일로 표현할 수 있다.

〈표 2〉 컴포넌트 상세품질목표 프로파일의 예

컴포넌트명	품질특성	품질부특성	메트릭	요구사항번호	요구사항수준
B2B경매 컴포넌트	기능성	정확성	정확도	Req.13	높음
	기능성	완전성	완전도	Req.17	보통
...	...	...		...	...

4. 컴포넌트 품질 보증 프로세스 패턴의 적용 사례

프로세스 패턴은 기존 모형으로부터 특별한 개발 프로세스를 조립할 수 있도록 한다. 기존 모형으로부터 특별한 개발 프로세스를 구축하기 위해서는 프로세스 패턴을 빌딩블록으로 하여 패턴간의 관계를 식별하고 프로젝트의 특성에 맞추어 이를 조정하여(tailoring) 정의함으로써 가능하다. 본 절에서는 적용사례를 통해 핵심 프로세스 패턴으로부터 프로젝트에 적합한 품질보증 프로세스가 구축되는 예를 보여준다.

수요가 많은 컴포넌트를 시범적으로 개발하여 보급하기 위해 E연구소 주관의 컴포넌트 개발 시범사업이 수행되었다. 시범사업은 4개 분야에 걸쳐 컴포넌트 개발이 이루어졌으며, 각 분야별로 서로 다른 개발업체에 의해 수행되었다. 다음 표에서 보는 바와 같이 EJB로 개발된 각 컴포넌트는 1개 이상의 세션빈(session bean) 또는 엔터티빈(entity bean)으로 구성되며, 각 빈(bean)은 하나 이상의 클래스(class)로 구현되었다.

〈표 3〉 시범 사업의 컴포넌트 개발 요약

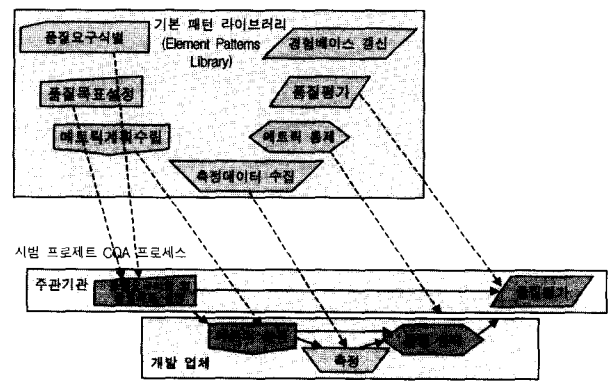
분 야	컴포넌트 수	빈(Bean) 수		클래스 수
		세션빈	엔티티빈	
XML/EDI 서버 공통 컴포넌트	12	3	11	47
B2B가상시장 공통 컴포넌트	6	7	27	8
B2C 쇼핑몰 공통 컴포넌트	5	5	7	51
비즈니스 공통 컴포넌트	5	4	12	60
총 계	28	19	57	166

시범 사업의 특성상 개발되는 컴포넌트의 품질 보증 프로세스는 다음과 같은 요구사항을 고려하여 구축되어야 한다.

- 주관기관(E연구소)과 개발업체의 역할이 명확히 구분되어야 한다.
- 주관기관은 개발업체에게 컴포넌트가 달성해야 할 측정가능한 품질 목표를 제시하여야 하고 이 목표에 따라 평가하여야 한다.
- 시범사업에서 컴포넌트는 단기간에 개발되므로 품질보증 단계는 간결한 형태로 제시될 필요가 있다.
- 개발이 끝난 컴포넌트가 품질 목표의 달성여부를 평가받기 위해 개발업체는 필요한 산출물을 제출하고, 주관기관은 품질 평가를 수행하여야 한다.

컴포넌트 품질 보증 프로세스는 구체적인 품질 보증 활동을 제시하여 개발자에게 제공하여야 하며, 각 활동 및 작업산출물은 적절한 시점에서 검토되어야 한다. 이러한 검토는 개발자 또는 별도의 조직에 의해 수행되는 것이 바람직하다. 그러나 시범 프로젝트가 짧은 기간에 이루어져야 하며, 개발자는 별도의 품질관리 조직을 운영하지 않고 주관기관이 품질관리의 많은 부분을 담당하여야 한다. 또한 개발된 컴포넌트는 반드시 품질 평가를 통해 품질을 검증받도록 하고 있다. 이러한 상황을 고려하여 품질 보증 프로세스는 품질 평가를 위한 핵심 단계만으로 구성된 프로세스의 형태가 필요하였다.

(그림 5)는 프로세스 패턴에서 이 활동들이 유도되는 과정을 보여주고 있다. 각 활동들은 프로세스 패턴들을 결합하고 조정하여 구축되며, 개발업체와 주관기관의 역할을 명확히 구분하고 있다.



(그림 5) 시범프로젝트의 컴포넌트 품질보증 프로세스

컴포넌트 시범사업에는 특정 CBD 프로세스가 적용되지 않았으며, 평가는 1회에 걸쳐 이루어졌기 때문에 경험베이스 구축은 포함되지 않았다. 다음은 시범사업에서 각 활동별로 수행된 중요 내용들은 다음과 같다.

● 컴포넌트 품질 요구 식별 및 품질 목표 설정

시범사업에서 개발된 컴포넌트는 <표 4>와 같은 품질

목표를 설정하였다.

〈표 4〉 시범사업의 컴포넌트 품질 목표

품질 요구사항	품질 달성 목표
컴포넌트 구성요소를 갖출 것	컴포넌트 사용에 필요한 정보를 별도로 제공함
	제공된 정보가 시험평가에 사용할 수 있도록 충분함
안전하고 설치가 용이할 것	컴포넌트가 바이러스에 감염되어 있지 않음
	사용자가 직접 운영 환경에 설치/제거할 수 있음
기능이 제대로 구현될 것	컴포넌트 문서에 기능이 제대로 설명되어 있음
	컴포넌트 문서에 명시된 기능이 모두 구현됨
	컴포넌트 문서에 명시된 기능이 모두 정확하게 구현됨
상호운영성을 만족할 것	컴포넌트가 인터페이스 표준을 준수하여 구현됨
	입출력 데이터 표준이 명시된 경우 이를 준수하여 구현됨

● 메트릭 설정

컴포넌트 품질요구를 만족시키기 위해 컴포넌트 개발과정에서 관리되어야 할 메트릭은 개발업체에서 정의하여 사용하도록 하였다.

● 측 정

시범사업에서는 특정 개발프로세스가 채택되지 않았기 때문에 개발업체 나름대로의 품질계획단계를 수행하고, 필요한 데이터를 측정하도록 하였다.

● 품질 통제

다음 표는 특정 컴포넌트의 기능의 구현여부와 구현 정확성을 기록한 예로서 시험사례를 보완하거나 코드 재작성 등이 필요하다는 조치 내용을 보여준다.

〈표 5〉 컴포넌트의 기능 구현 현황 및 조치의 예

기능 id	기 능	구현 여부	구현 정확성	조치 내용
F1-1	AddAuction	○	×	시험사례 보완
F1-2	ModifyAuction	○	×	시험사례 보완
F1-3	DeleteAuction	○	×	시험사례 보완
F1-4	gETAUCTIONINFO	○	○	-
F1-5	GetAuctionInfo	○	×	코드 재작성
F1-6	getSuccessfulBidInfo	○	×	코드 재작성
F1-7	checkRemainsTime	○	○	-
F1-8	DecideWinner	○	×	시험사례 보완
F1-9	DecideLoser	○	×	시험사례 보완

● 품질 평가

시범사업에서의 품질 평가는 각 컴포넌트 별로 수행되었다. 다음 표는 시범사업의 한 컴포넌트의 품질 평가 결과를 일부 요약한 것이다. 품질평가의 결과는 대부분이 기능성 위주로 수행되었다. 컴포넌트의 특성상 신뢰성 및 안정성 등의 평가는 많은 시간을 필요로 하기 때문에 본 시범사업에서는 제외되었다.

〈표 6〉 시범사업 컴포넌트 평가 결과의 예

메트릭(값 유형)	목표 대비 실제 측정값	
Virus 부재(Yes/No/N.A)	목표	Yes
	실제	Yes
설치 및 제거 가능성(Yes/No/N.A)	목표	Yes
	실제	Yes
기능 완전성(%) (구현 기능수/전체 기능수)	목표	100%
	실제	100%(4/4)
기능 정확성(%) (정확하게 구현된 기능수/전체 기능수)	목표	100%
	실제	75%(3/4)
인터페이스 준수성	목표	N.A
	실제	N.A
입출력데이터 표준준수성(Yes/No/N.A)	목표	Yes
	실제	Yes

(N.A : 적용대상이 아님)

5. 프로세스 패턴 접근방법의 효용성

컴포넌트 품질보증 프로세스 구축을 위한 프로세스 패턴 접근방법은 다음과 같은 측면에서 강점 및 약점을 가질 수 있다.

- 이해가능성 : CBD 사용시 별도로 품질보증프로세스로 분리하지 않고 있는 반면, 프로세스 패턴 접근방법은 가용한 프로세스 패턴으로 별도의 품질보증프로세스를 구축할 수 있다. 이 때 각 프로세스 패턴에 대한 이해가 우선될 필요가 있다.
- 학습성/운영성 : 프로세스 패턴 접근방법은 필요한 패턴 및 담당자(활동책임자)가 명확히 정의되어 있기 때문에 패턴의 활용에 대한 교육이 쉽고, 운영 또한 용이하다.
- 신속성 : 프로세스 패턴 접근방법은 상황에 따라 필요한 품질보증프로세스를 구축하여야 하기 때문에 시간이 소요된다.
- 분석성 : CBD 사용시 개발과정에서 수집되는 데이터를 분류하여 품질보증프로세스와 관련된 데이터를 별도로 정제하여 사용하여야 하나, 프로세스 패턴 접근방법은 패턴별로 필요한 데이터를 수집하고 분석할 수 있다.
- 수정가능성 : CBD는 상황에 따라 품질보증프로세스를 수정하려면 전체 개발과정을 고려하여 일관성을 유지하여야 하지만, 프로세스 패턴 접근방법은 필요한 패턴을 먼저 선정하고, 해당 패턴에서 필요한 사항을 조정할 수 있다.

6. 결 론

본 연구는 재사용 가능한 프로세스 패턴을 활용하여 프

로젝트의 특성 및 상황에 따라 조정하여 품질 보증 프로세스를 구성할 수 있는 접근방법을 제시하고 있다. 소프트웨어 프로세스 패턴 메타모델을 사용하여 재사용가능한 빌딩 블록으로서 컴포넌트 품질보증을 위해 핵심적인 프로세스 패턴들을 정의하였고, 이 패턴들이 실제로 활용가능함을 적용사례를 통해 보여주었다. 이러한 패턴들은 컴포넌트가 특정 CBD 프로세스를 통해 개발될 경우 품질 목표를 달성하기 위해 특정 측면에 초점을 맞추어 노력을 기울일 수 있도록 유도한다.

본 연구를 통해 다음과 같은 향후 연구방향을 설정할 수 있다.

첫째, 프로세스 패턴을 가능하면 빨리 적용할 수 있도록 프로세스 패턴의 정형성(formality)를 높이는 방법을 찾아 볼 수 있다. 이를 통해 상황에 맞는 프로세스 패턴을 쉽게 발견하고, 이들을 결합하여 수정내용을 최소화하여 사용할 수 있으면 프로세스 패턴이 성공적으로 적용될 수 있을 것이다.

둘째, 경험베이스는 성공적인 컴포넌트 품질 개선을 위한 핵심적인 부분을 담당할 수 있기 때문에 향후 실제 사례의 수집 및 축적을 통한 경험베이스의 구축 및 활용이 요구된다. 컴포넌트 품질 보증 및 개선을 위한 본 접근방법은 지속적인 경험, 데이터의 활용을 통해 보다 효율적으로 적용할 수 있다.

셋째, 기존에 사용되고 있는 CBD 프로세스들이 패턴화되어 라이브러리에 있는 경우 이들과의 결합하여 사용할 수 있는 방법을 찾아볼 필요가 있다. 컴포넌트 품질보증 프로세스가 CBD 프로세스와 밀접하게 결합되어 활용될 경우 보다 강화된 CBD 프로세스가 생성될 수 있을 것이다.

참 고 문 헌

[1] Standish Group International, Inc., *Chaos Report*, West Yarmouth, MA, 1995.

[2] B. Council, G. T. Heineman, Definition of a Software Component and Its Elements, in Component-Based Software Engineering, edited by G. T. Heineman and W. T. Council, 2001.

[3] J. Voas, Maintaining Component-based Systems, IEEE Software, July/August, 1998.

[4] I. Crnkovic, Software Focus, 2002 ; Component-based Software Engineering-New Challenges in Software Development, Vol.2, issue.4, winter, 2001.

[5] F. Bachman et al., Technical Concepts of Component Based Software Engineering, SEI, CMU/SEI-2000TR-008, Vol.2, May, 2000, http://www.sei.cmu.edu/pub/documents/00.reports/pdf/00tr008.pdf.

[6] 웹페이지, www.oospice.com.

[7] K. Short, "Component based development and object modeling," Sterling Software, 1997.

[8] Princeton Softech, "Component based development-a roadmap to eBusiness success," Princeton Softech, 2000.

[9] D. D'Souza, A. C. Wills, "Objects, Components, and Frameworks with UML," Addison-Wesley, 1998.

[10] 한국전자통신연구원, "마르미 Ⅲ-컴포넌트 기반 시스템 개발 방법론", 기술문서, 한국전자통신연구원, 2001.

[11] J. Cheeseman, J. Daniel, UML Components, Addison-Wesley, 2001.

[12] X. Cai, M. R. Lyu, K. Wong, "Component-based Software Engineering : Technologies, Development frameworks, and Quality Assurance Schemes," in Proceedings APSEC 2000, Seventh Asia-Pacific Software Engineering Conference, Singapore, December, 2000.

[13] 김길조, 장진호, 황선명, "컴포넌트 품질관리 프로세스 개발 사례", 정보처리학회논문지D, 제8-D권 제6호, 2001.

[14] http://www.ambysoft.com/processPatternsPage.html.

[15] I. Jacobson, G. Booch, J. Rumbaugh, The Unified Software Development Process, Addision-Wesley, 1999.

[16] ISO/IEC 9126-1, Information technology Software engineering-Software quality characteristics and metrics Part 1 : Quality characteristics and sub-characteristics, 1998.

[17] P. Allen, Realizing e-Business with Components, Addison-Wesley, 2001.

[18] M. Woodman et al., Issues of CBD Product Quality and Process Quality, in the proceeding of 4th ICSE Workshop on Component-Based S/W Engineering, 2001.

[19] P. Mohagehghi, R. Conradi, Experiences with certification of reusable components in the GSN project in Ericsson, Norway, in the proceeding of 4th ICSE Workshop on Component-Based S/W Engineering, 2001.

[20] D. Giannakopoulou and J. Penix, Component Verification and Certification in NASA Missions, in the proceeding of 4th ICSE Workshop on Component-Based S/W Engineering, 2001.

[21] S. W. Ambler, Process Patterns-Building Large-Scale Systems Using Object Technology, Cambridge University Press, 1998.

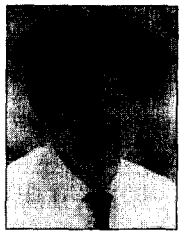
[22] S. W. Ambler, More Process Patterns-Delivering Large-Scale Systems Using Object Technology, Cambridge University Press, 1998.

[23] M. Gnatz, F. Maschall et al., Toward a Living Software Development Process Based on Process Patterns, in V. Ambriola(Ed) : EWSPT 2001, LNCS 2077, 2001, Springer-Verlag, 2001.

[24] H. Storrle, Describing Process Patterns with UML in V.Ambriola(Ed) : EWSPT 2001, LNCS 2077, pp.173-181, Springer-Verlag, 2001.

[25] 웹페이지, www.omg.org.

[26] Platinum Technology, BPWin Method Guide, 1998.



황 선 명

e-mail : sunhwang@dju.ac.kr

1982년 중앙대학교 전자계산학과(이학사)

1984년 중앙대학교 소프트웨어공학전공(이학석사)

1987년 중앙대학교 소프트웨어공학전공(이학박사)

1997년~현재 ISO/IEC JTC7/WG10 한국운영위원

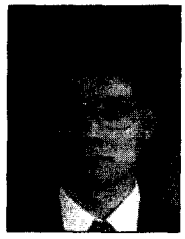
1998년~현재 한국정보통신기술협회TTA 특별위원

1989년~현재 대전대학교 컴퓨터공학과 교수

2000년~현재 한국S/W프로세스심사인협회(KASPA) 이사

2000년~현재 한국정보처리학회 논문지 편집위원

관심분야 : 소프트웨어 프로세스 모델, 품질 메트릭스, 소프트웨어 공학 표준화, 컴포넌트 품질측정, 테스팅 방법론 등



김 길 조

e-mail : kgj@abni.net

1987년 서울대학교 산업공학과(공학사)

1989년 한국과학기술원 경영과학과(공학석사)

1994년~1998년 시스템공학연구소 선임연구원

1998년~2003년 한국전자통신연구원 선임연구원

2003년 대전대학교 컴퓨터공학과(공학박사)

2000년~현재 한국S/W프로세스심사인협회(KASPA) 이사

2003년~현재 에이비앤아이(주) 수석컨설턴트

관심분야 : IT 프로세스 개선, 소프트웨어 품질 측정, 소프트웨어 공학 표준화, 소프트웨어 개발 환경 등

김 진 삼

e-mail : jinsam@etri.re.kr

1984년 중앙대학교 전자계산학과(학사)

1986년 중앙대학교 시스템소프트웨어공학전공(석사)

1987년~1998년 시스템공학연구소 선임연구원

1998년~현재 한국전자통신연구원(ETRI) 책임연구원

2001년~현재 대전대학교 소프트웨어공학전공(박사)

관심분야 : 소프트웨어 공학 표준화, 소프트웨어 개발 환경, 품질 메트릭스, 컴포넌트 품질 측정, 소프트웨어 프로세스 개선 등