

컴포넌트 기반 개발을 위한 CASE 도구의 기능적 요구사항 및 개발관리 도구

Functional Requirements about CASE Tools for Component Based Development and a Development Management Tool

김영희(Kim Young Hee)*, 정기원(Chong Ki Won)**

초 록

컴포넌트 기반 개발에 사용되는 CASE 도구는 기본적으로 모델링 기능, 프로젝트 관리 기능, 그리고 구현 및 테스트 지원 기능 등이 필요하다. 본 논문에서는 컴포넌트 기반 개발 시에 사용되는 CASE 도구의 기능적 요구사항을 제시한다. 기존에 존재하는 컴포넌트 기반 개발 도구들로부터 도구의 기능적 요구사항을 도출하여 분석/설계, 프로젝트 관리, 구현 및 테스트 기타 기능 지원 등으로 분류하여 제시하고, 개발관리 기능을 추가하여 컴포넌트 기반 개발을 위한 CASE 도구를 제안한다.

ABSTRACT

CASE tools supporting component based development should include functions for modeling, project management, and supporting other phases activities of the development process. Functional requirements of CASE tools supporting component based development are proposed in this paper. Required functions are identified by analyzing the existing CBD CASE tool, and are categorized into modeling, project management, and supporting other phases activities.

키워드 : CASE 도구, 컴포넌트 기반 개발, 요구사항, 개발관리

CASE tools, component based development, requirement, development management

본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음.

* 숭실대학교 전산원 소프트웨어 정보학과 교수

** 숭실대학교 컴퓨터학부 교수

1. 서 론

비즈니스 패러다임의 변화에 따른 빠른 시장 대응(Time-to Market)에의 요구 및 비용 감소와 높은 생산성에 대한 요구가 증가하고 있다. 또한, 재사용성 증대를 통한 유연한 어플리케이션을 개발할 수 있는 절차와 매커니즘에 대한 필요성이 대두되며 많은 수의 어플리케이션들이 컴포넌트 기반으로 개발되고 있다. 그동안 어플리케이션의 발전에 비해 소프트웨어 개발에 사용되는 CASE 도구는 상대적으로 빈약한 발전을 보여 왔으나[1], 최근에는 객체지향 시스템 또는 컴포넌트 기반 시스템 개발에 적용할 수 있는 도구들이 속속 등장하고 있으며, 컴포넌트 기반 개발 프로세스에 특화되어 개발된 도구들이 많이 등장하고 있다. IBM사의 ROSE와 CA(Computer Associates)사의 Advantage Joe, 그리고 볼랜드(Borland)사의 Together 등은 UML을 사용하여 컴포넌트 기반 개발을 지원하는 대표적인 CASE 도구로서 분석, 설계 및 구현 초기단계까지의 작업들을 지원하고 있다.

본 논문에서는 컴포넌트 기반 개발에 사용되는 CASE 도구의 기능적 요구사항을 제시한다. 기존의 컴포넌트 기반 개발에 사용되는 개발 프로세스를 분석하여 필수적인 활동 및 작업을 조사하고, 기존의 도구에서 지원하고 있는 개발 프로세스의 활동 및 작업을 분석한다. 이를 기반으로 컴포넌트 기반의 개발 프로세스의 수행에 필요한 도구의 기능적 요구사항을 제시하고, 이를 기반으로 하여 개발관리 측면의 요구사항을 추가한 컴포넌트 기반 개발 CASE 도구 모델을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 컴포넌트 기반 개발 프로세스에 대하여 알아보고, 현재 개발 과제에서 많이 사용되고 있는 대표적인 CASE 도구들에 대해 조사한다. 3장에서는 이들 도구들로부터 컴포넌트 기반 개발을 위하여 도구가 갖추어야 하는 기능적 요구사항을 도출하여 제시한다. 4장에서는 제시한 요구사항을 반영하는 개발 도구를 제안한다. 5장에서는 제안한 도구 모델을 평가하고, 6장에서는 본 논문의 결론 및 향후 연구를 정리한다.

2. 관련 연구

2.1 CBD 방법론

RUP(Rational Unified Process)는 래쇼날 소프트웨어 사에서 제안한 객체지향방법론으로서, 개발 프로세스와 관리 프로세스를 통합한 형태를 띠고 있다. RUP의 단계 구성은 초기(Inception), 정제(Elaboration), 구축(Construction), 그리고 전이(Transition)의 4개 단계로 되어 있으며, 각 단계에서 수행해야 할 비즈니스 모델링, 요구사항분석, 분석 및 설계, 구현, 시험, 그리고 전개 등의 핵심 워크플로우로 구성되어 있다. RUP에서는 각 단계별로 워크플로우의 반복수행을 통해 시스템을 점진적, 진화적으로 개발한다. 하지만 RUP는 컴포넌트 기반 개발을 위한 전용 프로세스라기보다 객체지향 개발을 위한 프로세스의 범주에 속하기 때문에 컴포넌트 개발에 적용하기 위해서는 프로세스의 조정이 요구된다[2, 8].

카탈리시스(Catalysis) 방법론은 OMT 방법론의 정형화를 목적으로 시작된 방법론이다. 카탈리시스는 도메인에 따른 다양한 문맥의 조합을 통해 적용 가능한 패턴 기반의 프로세스를 제공한다. 더불어 모델 간에 잘 정의된 일관성 규칙을 제공하고 있으며, 복잡한 시스템을 기술하기 위해 필요한 다양한 관점을 제공한다. 카탈리시스는 4개의 단계를 통해 각 단계 별로 핵심적인 산출물을 개발한다. 요구사항분석 단계를 통해 도메인 모델과 시스템 문맥을 개발하고, 시스템명세 단계를 통해 목표 시스템의 외부 행위를 기술한다. 아키텍처 설계 단계를 통해 물리적 아키텍처와 논리적 어플리케이션 아키텍처를 개발하고 컴포넌트 내부설계 단계를 통해 각 컴포넌트에 대한 상세 설계를 진행한다[4]. 카탈리시스 방법론을 적용할 경우, 이에 사용될 CASE 도구는 정형화된 모델, 설계, 의미 검사가 가능해야 하며, 카탈리시스에서 제공하는 다양한 범주의 패턴을 제공할 수 있어야 한다.

CBD96은 1996년 스텔링 소프트웨어(Sterling Software)에서 개발한 컴포넌트 기반 개발 방법론 중의 하나이다. CBD96은 구체적인 개발 프로세스는 제시하지 않고 있으며, 단지 컴포넌트를 설계하고 개발하기 위해 필요한 설계 모델들과 이를 산출하기 위한 작업, 이러한 작업들 사이의 연관관계만을 제시하고 있다. 또한 CBD96은 비즈니스 모델에서 컴포넌트 모델로의 도출 관계에 대한 정확한 정의가 없으며, 컴포넌트 내부설계 보다는 인터페이스 개발에 중점을 두고 있는 방법론이다[4].

UML 컴포넌트 방법론은 순수한 개발 프로세스로서 아키텍처 중심의 개발 방법론이다. UML을 사용하여 컴포넌트 모델링을 수행하며, 컴포넌트 사이의 의존관계에 대한 명세를 중심으로 프로세스를 수행한다. 전체 개발 프로세스 상의 워크플로우는 먼저 요구분석단계에서 비즈니스 요구사항을 분석하여 비즈니스 개념모델을 작성하고, 명세단계에서 요구분석단계의 산출물과 사용사례모델 등을 사용하여 컴포넌트 명세 및 아키텍처를 작성하며, 공급단계를 통해 요구사항에 맞는 기존의 컴포넌트를 획득하고, 어셈블리 단계를 통해 컴포넌트를 개발하고 획득한 컴포넌트와 조립한다. 최종적으로 테스트 단계와 전개단계를 거친다[4].

마크미-III 방법론은 한국전자통신연구원에서 개발한 컴포넌트 기반 개발 방법론으로, 어플리케이션 패밀리간의 공통성과 가변성을 통한 컴포넌트의 재사용성 증대에 초점을 맞추고 있다. 사용사례를 기반으로 사용자 요구사항 분석, 컴포넌트 식별, 점진적 개발단위 결정을 수행하고, 구현 플랫폼은 EJB와 .NET 환경을 기반으로 하고 있다. 마크미-III 방법론은 개발 프로세스와 프로젝트관리 프로세스로 나뉘어 있으며, 개발 프로세스는 요구획득 단계, 아키텍처 단계, 점진적개발 단계, 인도 단계의 네 단계로 구성되고, 프로젝트관리 프로세스는 프로젝트시작 단계, 프로젝트진행 단계, 프로젝트종료 단계의 세 단계로 구성된다. 마크미-III 방법론은 전체 시스템을 몇 개의 미니프로젝트로 나누고 점진적, 반복적으로 컴포넌트를 분할하여 시스템을 개발하는 특징을 가지고 있으며, 개발 초기에 전

고한 아키텍처를 정의하고 비즈니스 컴포넌트와 응용 컴포넌트 아키텍처를 결정할 수 있는 아키텍처 중심의 방법론이다[3].

2.2 컴포넌트 개발을 위한 기존의 CASE 도구

IBM 래쇼날 제품군은 래쇼날 소프트웨어에서 개발한 CASE 도구 제품군으로서, 요구사항 분석, 모델링, 그리고 형상관리 등을 지원하는 다양한 도구를 제공하고 있다. RUP를 효과적으로 지원할 수 있도록 RUP 프로세스의 다양한 작업에 적합한 형태의 CASE 도구들을 지원하고 있으며, 각 도구들은 유기적으로 연결되어 사용될 수 있다. 래쇼날 제품군은 RUP를 지원하는 측면에서는 효과적이나, 컴포넌트 개발에 있어서는 개발자가 각 도구의 사용에 대한 계획을 별도로 수립하는 작업이 필요하다[8].

CA사의 컴포넌트 개발을 위한 CASE 도구에는 Advantage Joe가 있다. Advantage Joe는 프로세스 및 데이터를 모델링 할 수 있는 도구가 있으며, 컴포넌트를 분석하고 설계할 수 있는 컴포넌트 모델러가 있다. 요구사항 분석에서 설계 단계까지의 작업에 필요한 산출물을 개발하기 위한 도구와 전체 개발 프로세스 중 구현 초기까지의 작업을 수행할 수 있는 도구를 지원하고 있다. Advantage Joe는 J2EE 기반의 컴포넌트 어플리케이션 개발 시 CBD96에 기반하여 분석, 설계, 구현, 테스트 및 배포에 이르기까지 단계를 자동화하는 통합개발환경을 지원하고 있으며, 웹 어플리케이션 서버 환경의 프로그래밍을 지원하고 있

다. Advantage Joe는 CBD96 방법론에 따라 작업을 수행하도록 지원하고 있으므로, 다른 방법론을 사용할 경우 제품에서 지원하는 여러 다양한 기능들의 사용이 제한될 수 있다 [6].

Together 제품군은 블랜드 사의 CASE 도구 제품군으로서, 개발의 전 과정에서 모델과 코드를 실시간으로 동기화하는 것이 가능하다. CBD방법론을 반복적으로 수행할 수 있도록 지원하고, 어플리케이션을 분석, 설계, 구현, 배포할 수 있도록 어플리케이션 개발 환경을 제공하는 도구이다. 파일 기반 IDE 도구와 통합 작업, 모든 클래스와 라이브러리 지원 및 ROSE에서 저장한 파일 사용 등의 다른 제품과 통합이 가능하고 사용자 보고서 설계를 위한 기능이 지원되고 있다[9].

COBALT Constructor & Assembler는 한국전자통신연구원에서 개발한 도구이며, COBALT Constructor는 도메인 분석을 통한 컴포넌트 식별, 컴포넌트 단위의 설계 및 구현, 전개, 테스트의 컴포넌트 생성 전 단계를 지원하며, COBALT Assembler는 시스템 아키텍처의 모델링, 컴포넌트의 개조와 합성, 전개, 테스트의 컴포넌트 조립 프로세스 단계를 지원한다[18].

2.3. 개발방법론과의 관계에 따른 CASE 도구 활용의 분류

컴포넌트 분석 및 설계에 사용하는 도구들은 개발 과정에 활용하는 방법에 따라 두 가지 범주로 분류할 수 있다. 첫 번째 범주는 CASE 도구가 특정 개발 방법론을 지원하는

형태로서 개발 방법론의 프로세스에 따라 도구가 사용된다. 두 번째 범주는 방법론의 적용 여부와는 관계없이 CASE 도구가 분석 및 설계에 사용되는 다양한 형태의 산출물 작성 기능을 지원하는 형태이다. 분석 및 설계 단계의 산출물들은 UML 표기를 사용한 다이어그램들이 중심이 된다.

첫 번째 범주의 도구 활용 방법은 특정 개발 방법론 프로세스의 단계, 활동 및 작업 진행에 따라 주요 산출물을 개발하며, 산출물들은 프로세스의 다음 단계 입력물로 사용될 수 있다. 이러한 범주의 도구 활용은 산출물들 사이의 일관성이 보장되며, 도구 상에서 프로세스 흐름의 추적이 가능하다. 도구 사용에 익숙할 경우 해당 방법론의 프로세스를 자연스럽게 수행하게 된다. 그러나 해당 개발 방법론에 숙달되어 있지 않거나, 다른 개발 방법론을 적용할 경우 도구의 활용도가 떨어지는 단점이 있다. 두 번째 범주의 도구 활용 방

법은 도구가 갖는 분석 및 설계 모델 생성 기능을 중심으로 이용하는 것으로서, 개발 방법론의 단계, 활동 및 작업 사이의 연관 관계와는 관계없이 필요한 분석 모델 또는 설계 모델을 개발하는 데 주안점을 둔 것이다. 이러한 방법의 경우 각 방법론이 요구하는 다양한 산출물의 개발을 위해 필요한 도구들을 자유로이 사용할 수 있고 분석 및 설계자가 익숙한 도구를 활용할 수 있는 장점이 있다. 그러나 도구에서 프로세스 진행을 반드시 준수해야 하는 것이 아니므로 분석 및 설계자가 프로세스 진행에 차질이 없도록 일일이 조정하며 작업을 수행해야 하고 도구들 사이의 데이터 교환이 불가능할 경우, 산출물 생성을 위한 작업의 입력력물에 대한 부가적인 가공작업이 필요하다. 이 경우에는 산출물 사이의 일관성을 확인하는 작업이 필요하게 된다. 다음 <표 1>은 CASE 도구 활용의 두 분류에 대한 특성을 보여준다.

<표 1> CASE 도구 활용의 분류

특 성	특정 개발 방법론을 지원하는 도구 활용	산출물 생성 중심의 도구 활용
개발 프로세스 지원	개발 프로세스에 따른 단계, 활동 및 작업 지원	분석 및 설계자가 적용하는 프로세스의 활동 및 작업을 식별하고 적용 여부를 결정
산출물의 관리	프로세스 흐름에 따라 산출물의 개발 및 관리 수행	도구에서 지원하는 분류체계에 따라 산출물 개발 및 관리 수행
다이어그램 생성	개발 방법론에 따른 특정 다이어그램 생성 지원 eg : 비즈니스 모델링, 컴포넌트 모델링	지원하는 모델 외의 것을 개발하는 경우 스테레오 타입 등을 활용하여 표현하거나 다른 도구를 겸용
타 방법론 지원	특정 개발 방법론에 종속적이므로 조정 작업이 필요	개발 방법론에 따라 필요한 기능을 구분하여 사용

3. CBD를 위한 CASE 도구의 기능적 요구사항

컴포넌트 기반 어플리케이션 개발 CASE 도구는 2장에 나타난 바와 같이 여러 가지가 있다. 기존 도구에서 지원하고 있는 개발 프로세스의 활동 및 작업을 분석하여 이 도구들로부터 컴포넌트 기반 개발 도구가 지원해야 하는 요구사항을 도출하고 이를 기반으로 컴포넌트 기반 개발을 위한 CASE 도구가 가져야 할 기능적 요구사항을 분석 및 설계, 프로젝트관리, 구현 및 시험, 타 단계 활동의 지원 등으로 분류하여 제시하였다.

3.1 분석 및 설계 단계

컴포넌트 기반 개발 시 분석 및 설계 단계에 필요한 다양한 산출물의 개발 기능과 더불어 산출물의 활용 및 검증 기능이 필요하며, 반복적이며 점진적인 분석 설계를 위한 기능들이 요구된다.

3.1.1 분석 및 설계 모델 생성

가. 컴포넌트 설계 지원 (순공학)

컴포넌트 기반 개발 방법론은 일반적으로 업무 설계를 수행한 후 필요한 컴포넌트를 식별한다. 식별된 컴포넌트는 인터페이스가 정의되며, 기존의 컴포넌트를 획득하거나 새로운 컴포넌트를 개발하여 필요한 컴포넌트를 획득하게 된다. CASE 도구에서 이러한 컴포넌트 기반 시스템 구축 방법을 지원하기 위해서는 업무 설계 및 컴포넌트 식별, 인터페이스 정의 및 설계, 그리고 COTS 컴포넌트 및

직접 개발한 컴포넌트의 통합 지원 기능이 요구된다. 이와 더불어 컴포넌트의 상세 설계를 위한 설계 기능이 지원되어야 한다.

나. 역공학

기존 시스템의 재개발 또는 유사한 시스템의 개발을 위해서는 기존 시스템의 소스 코드로부터 UML 다이어그램을 추출하는 기능이 요구된다. 정적모델의 핵심인 클래스도를 추출할 수 있어야 하며, 필요에 따라 동적모델인 순차도를 제공할 수 있어야 한다. 기존 시스템이 재제지향으로 개발된 시스템이 아닐 경우, 시스템을 클래스로 랩핑(Wrapping)하거나 클래스화를 돕기 위한 정보(변수 및 오퍼레이션의 정보 등)를 제공하여 컴포넌트화할 수 있어야 한다. 데이터베이스 테이블이 존재할 경우 테이블로부터 데이터베이스 모델을 생성할 수 있어야 한다.

다. 비즈니스 프로세스 공학

컴포넌트 기반 개발은 비즈니스 프로세스 공학 수행을 위해 UML의 사용사례도 또는 활동도를 사용하여 업무를 표현할 수 있어야 한다. 개발된 프로세스 모델에 포함된 비즈니스 정보는 설계되는 컴포넌트 또는 컴포넌트 내부의 객체 정보로 변환될 수 있어야 하며, 비즈니스 프로세스 모델 상의 정보들이 설계 모델에 어떻게 반영되었는지 추적할 수 있어야 한다.

라. 분석 및 설계 모델 사이의 입출력 지원

각 단계에서 산출되는 다양한 모델들은 개발 프로세스에서 정의한 입출력 관계에 따라 다음 단계의 입력물로 사용될 수 있어야 한다. 이를 위해서는 개발 방법론에서 정의한 프로세스의 산출물의 연관 관계를 설정할 수

있어야 하며, 설정된 산출물의 정보는 분석 및 설계 작업 진행 시 분석/설계자가 입력된 산출물 정보를 바탕으로 새로운 산출물을 개발할 수 있도록 지원해야 한다.

마. 설계 편의 기능 지원

시스템 설계 시 반복적으로 나타나는 요구사항을 해결하기 위한 방안으로 디자인 패턴이 적용될 수 있다. 이를 위해 분석 및 설계 도구에서는 에릭 감마 등이 제시한 23개의 디자인 패턴[16] 이외에 도메인 별 디자인 패턴(예 - 실시간 시스템, 분산 시스템 등), 소프트웨어 아키텍처에 따른 디자인 패턴, 그리고 배포 환경에 따른 디자인 패턴 등을 설계에 적용할 수 있는 기능을 제공해야 한다. 다양한 디자인 패턴을 적용 조건에 따라 검색할 수 있어야 하며, 선정된 패턴은 템플릿 형태로 제공되어 설계자가 개발하는 시스템에 맞게 조정하여 사용할 수 있어야 한다. 또한 디자인 패턴의 수정 및 확장을 위해 패턴을 편집할 수 있는 기능이 요구된다.

바. 산출물 사이의 일관성 검증

생성된 분석 및 설계 모델이 UML 표기법에 따라 적절한 형태로 작성되었는지 검증하고, 생성된 모델 사이에서 일관성이 유지되었는지 확인할 수 있어야 한다. 모델 내에서는 해당 모델이 모델링 문법에 적합하게 생성될 수 있도록 표기를 제한하여야 하며, 모델을 생성할 때 적합하지 않은 모델을 생성하고자 하면 분석/설계자에게 경고를 줄 수 있어야 한다. 모델이 적합하게 생성되었을 경우 입출력 관계에 있는 타 모델과의 일관성을 검증해야 한다. 분석단계 및 설계단계에서 생성된 각각의 산출물들이 단계 내에서 일관성 있게

생성되었는지 검증할 수 있어야 하며, 분석단계의 산출물이 설계단계의 입력물로 사용될 경우 단계 사이의 일관성이 유지되고 있는지 검증할 수 있어야 한다.

3.1.2 프로토타이핑 및 시뮬레이션

사용자의 요구사항을 식별하고 이를 검토하기 위한 프로토타입을 개발 프로세스 초기에 제시할 수 있는 기능을 지원해야 한다. 이를 위해 사용자 인터페이스를 개략적으로 스케치할 수 있는 도구를 지원할 수 있어야 하며, 스케치된 인터페이스 사이의 항해 시나리오 또는 링크를 연결시킬 수 있어야 한다. 또한 생성된 프로토타입의 시뮬레이션을 위해 스텝 또는 드라이버의 생성기능을 제공해야 한다.

3.1.3 데이터베이스 모델링

데이터베이스의 설계를 위해서는 데이터베이스의 논리모델과 물리모델을 설계할 수 있어야 한다. 논리 모델로 ER 모델을 생성할 수 있어야 하며, 이를 토대로 물리모델을 생성할 수 있어야 한다. 컴포넌트 명세에 따라 DDL을 작성할 수 있어야 하며 물리적 데이터 모형과 명세 클래스 모형을 바탕으로 비즈니스 로직 구현에 필요한 DML을 작성할 수 있어야 하고 작성된 DML을 실행하고 그 결과를 해당 테이블에 반영하는 기능이 요구된다. 또한 관계형 데이터베이스 모델 외에 객체지향 데이터베이스 모델, 객체 관계형 데이터베이스 모델 등도 지원할 수 있어야 한다.

3.1.4 사용자 인터페이스 설계

사용사례 모델, 컴포넌트 아키텍처 모델, 그리고 객체 모델 등을 토대로 사용자 인터페이스를 식별하고, 식별된 인터페이스를 설계하는 기능이 요구된다. 사용자 인터페이스의 명세를 작성할 수 있는 기능이 있어야 하며, 명세를 토대로 인터페이스를 스케치할 수 있는 기능이 요구된다. 스케치된 인터페이스 사이의 연관 관계를 설정할 수 있어야 하며, 사용자 인터페이스와 비즈니스 로직을 담당하는 컴포넌트 또는 클래스와의 연관 관계를 설정할 수 있어야 한다. 또한 이러한 사용자 인터페이스 설계는 프로토타입의 개발에 활용될 수 있어야 한다.

3.1.5 컴포넌트 획득

웹 서비스를 이용하여 컴포넌트 공급자를 선정하고 식별하여서 후보 컴포넌트 목록을 작성하고 컴포넌트의 기능과 요구사항 명세서의 기능 매핑 정도, 사용자 인터페이스와 조직의 비즈니스 부합정도, 컴포넌트에 의해 야기되는 이벤트, 관리 데이터, 환경 및 기술적인 특성 등의 평가를 고려하여 재사용할 컴포넌트를 획득한다.

3.1.6 문서화

분석 및 설계 도구를 통한 모든 생성 모델은 문서화 도구를 사용하여 문서화될 수 있어야 한다. 이를 위해 도구는 문서화 프로그램을 호출하여 사용할 수 있어야 하며, 문서상에서 OLE(Object Linking Embedding) 형태로 생성된 모델을 편집할 수 있는 기능을 지원할 수 있어야 한다. 또한 모델을 수정하였

을 경우 해당 모델이 포함된 문서 산출물이 수정될 수 있도록 산출물에 링크정보를 제공하거나 동적으로 수정된 모델이 모델을 포함한 산출물에 반영될 수 있어야 한다.

3.1.7 모델의 XML화

분석 및 설계 단계에서 생성된 모델들은 도구 고유의 포맷으로 저장될 뿐만 아니라 XML 형태로 입출력이 가능해야 한다. UML의 XMI을 지원하여 생성된 모델을 XML 문서로 생성할 수 있어야 하며, 다른 도구에서 생성된 XML 문서를 입력으로 하여 UML 모델을 생성할 수 있어야 한다. 이와 더불어 필요할 경우 설계된 컴포넌트 모델을 XML 문서로 생성할 수 있어야 한다.

3.1.8 다양한 형태의 출력

도구로부터 생성된 모델들은 다양한 형태로 출력이 가능해야 한다. 분석 및 설계 모델을 구성하고 있는 다양한 요소들을 여러 관점으로 살펴볼 수 있도록 다양한 추상화 단계의 출력 기능을 제공해야 하며, 요소들의 정보를 적절한 형태로 구성하여 보고서를 출력할 수 있어야 한다.

3.2 구현 및 시험 단계

3.2.1 구현물(코드) 생성

설계된 컴포넌트의 명세로부터 구현에 필요한 코드의 뼈대를 생성할 수 있어야 한다. 이를 위해서는 컴포넌트의 배포 플랫폼에 따라 요구되는 인터페이스와 여러 구현물을 생성할 수 있어야 한다. 예를 들어 EJB의 경우

웹 컨테이너에 포함될 JSP 또는 애플릿을 생성할 수 있어야 하며, EJB 컨테이너에 포함될 자바 빈즈의 컴포넌트 인터페이스, 홈 인터페이스 그리고 빈 클래스를 생성할 수 있어야 한다. 이와 더불어 실행 가능한 구현코드 생성을 위해 컴포넌트 명세에 구현 코드를 추가할 수 있는 기능을 포함해야 하고 이를 토대로 생성된 구현코드는 컴파일 후 실행을 할 수 있어야 한다.

3.2.2 모델 사이의 동기화

설계모델에 따라 구현된 후 구현단계에서 구현모델(코드)의 구조가 변경되거나 설계모델이 변경되어 구현모델의 변경이 필요할 경우, 두 단계의 모델 사이의 동기화 기능을 제공할 수 있어야 한다. 설계모델에서 구현모델을 즉시 생성할 수 있어야 하며, 설계모델의 변경이 있을 경우 해당하는 구현모델에 변경사항이 바로 반영되어야 한다. 이러한 동기화는 모델 개발 도중에 사용될 수 있을 뿐만 아니라 개발이 완료된 모델에도 적용되어 사용될 수 있어야 한다.

3.2.3 획득 컴포넌트 재사용

획득한 컴포넌트를 이용하여 재사용이 요구되는 작업에서 웹 서비스를 이용하여 메소드를 호출하여 재사용 컴포넌트를 플러그인, 플레이 또는 결과를 반환 받아서 작업을 계속할 수 있어야 한다.

3.2.4 시험

분석 및 설계 모델을 통해 컴포넌트를 시험할 수 있어야 한다. 분석 모델에 나타난 사용

자의 요구사항이 클래스도 및 순차도와 같은 설계 모델에 제대로 반영되었는지 시험할 수 있어야 한다. 이를 위해 설계 모델은 유사코드 등을 사용하여 실제 기능을 시뮬레이션 할 수 있도록 만들 수 있어야 하며, 컴포넌트 또는 객체 단위의 시험을 위해 드라이버 및 스텝을 생성할 수 있는 도구를 제공해야 한다. 개별적인 컴포넌트의 시험뿐만 아니라 컴포넌트를 통합한 전체 시스템의 시험을 위해 컴포넌트 인터페이스간의 호출과 in-house 및 COTS 컴포넌트들의 인터페이스를 활용하는 시험 기능 등이 요구된다. 이는 설계 단계에서 설계 모델 상의 시험을 위한 기능이며, 시험 단계를 지원하기 위해 테스트케이스 생성 기능을 지원해야 한다.

3.3 프로젝트 관리 지원

3.3.1 형상관리 및 버전관리

많은 CBD 방법론들이 시스템 개발 시 반복적이며 증분적인 개발 특성을 가지고 있다. 따라서 분석/설계 도구를 통해 산출되는 결과물들에 대해 반복 및 증분에 따른 산출물들의 변경 및 확장을 추적하는 기능이 필요하다. 도구를 통해 개발된 산출물은 동일 단계, 활동 또는 작업에 의해 개발된 산출물을 위해 버전관리를 지원해야 하며, 다양한 산출물들의 반복적인 개발과 분할 작업을 지원하기 위해 체크인/아웃과 같은 형상관리 기능이 지원되어야 한다.

3.3.2 품질 보증

분석 및 설계된 시스템의 품질을 보증하기

위해서는 분석 및 설계 산출물들을 토대로 품질을 측정할 수 있는 매트릭이 지원되어야 한다. 컴포넌트 자체의 품질과 인터페이스의 적절한 조립성, 다른 컴포넌트와의 원활한 상호작용, 이를 이용한 어플리케이션 개발의 용이성, 그리고 적절한 동작의 보장 등을 평가하기 위한 평가 항목과 매트릭이 요구되며, 이를 사용하여 기 개발된 컴포넌트나 개발하고자 하는 컴포넌트를 평가할 수 있어야 한다. 이를 위해 설계 명세로부터 인터페이스 및 인터페이스 내의 메소드 개수, 입력 및 출력 개수, 메소드의 구현 비율 등을 측정할 수 있고 컴포넌트의 품질에 대한 평가정보를 제공해야 한다. 컴포넌트 명세에 대한 요구사항 수용여부를 측정하여 컴포넌트의 활용도를 평가할 수 있어야 한다.

3.3.3 프로세스 지원 및 조정

개발 프로세스 상에서 분석 및 설계 단계에 해당하는 활동 및 작업들을 도구 상에서 설정하고, 설정된 프로세스의 흐름에 따라 도구에서 개발될 수 있는 산출물들을 지정할 수 있어야 한다. 이를 위해 도구 내부에 프로세스의 단계, 활동 및 작업을 설정할 수 있어야 하며, 프로세스 흐름에 따라 생성되는 산출물들의 관계를 설정할 수 있어야 한다.

3.4 기타 기능

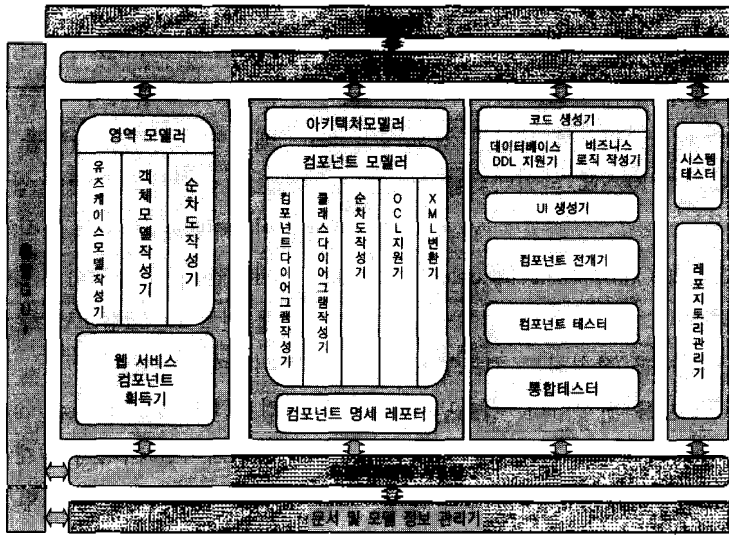
도구에서 지원하지 않는 기능이나 분석 및 설계 단계가 아닌 타 단계의 작업을 지원하기 위해 도구는 타 도구와의 연동 기능을 지원해야 한다. 도구는 분석 및 설계 단계를 제외한

개발 프로세스 나머지 단계의 도구들과 더불어, 프로젝트 관리에서 사용되는 도구들과도 연동이 될 수 있어야 한다. 설계모델로부터 생성된 구현코드는 자연스럽게 구현도구의 입력으로 사용될 수 있도록 연결되어야 하며, 설계단계에서 도구에 의해 생성된 테스트케이스는 시험단계의 입력으로 연결되어 테스트의 자동화를 가능하게 해야 한다. 프로젝트 관리의 입장에서 품질 보증, 일정 관리, 위험 관리 등 여러 가지 측면에서 사용되는 다양한 도구들은 분석 및 설계 도구와 상호간에 필요한 정보를 입출력할 수 있어야 한다.

4. 개발 관리 도구 모델 설계

이 장에서는 3장에서 제시한 기능적 요구사항을 수용하는 개발 관리 도구 모델을 <그림 1>과 같이 제안한다. 제안한 도구는 J2EE와 .NET 플랫폼 중에서 선택한 플랫폼을 기반으로 하여 지원하도록 하였다.

제안한 개발 관리 도구 모델은 컴포넌트 기반 개발 프로세스를 지원하고 이와 더불어 프로젝트관리 프로세스를 지원하도록 설계하였다. 영역 모델러, 아키텍처 모델러, 컴포넌트 모델러 등의 기능을 제어할 수 있도록 툴 관리자가 필요하며, 툴 관리자, 프로젝트관리 서포터와 문서 및 모델 정보 관리기들이 하나의 통합 GUI를 통해서 연결되어서 사용자가 도구를 쉽게 사용할 수 있도록 해야 한다. 프로젝트 관리 서포터와 문서 및 모델 정보 관리기는 서로 연결되어서 각종 문서를 저장할 수 있어야 한다. 영역 모델러, 아키텍처 모델러,



〈그림 1〉 개발 관리 도구 모델

컴포넌트 모델러, 컴포넌트 명세 레포터, 코드 생성기, UI생성기 등의 작업을 수행하면서 문서화 작업이 병행되어야 하는 경우가 있으므로 프로젝트관리 서포터를 사용하여 문서화를 행할 수 있도록 하고 문서 및 모델 정보 관리기를 통해서 저장하도록 하였다.

제안한 개발 관리 도구 모델의 구성요소와 그에 대한 설명은 다음과 같다.

1. 프로젝트 관리 지원

- 프로젝트 관리 서포터 (Project Management Supportor): 프로젝트 관리에 필요한 각종 문서, 즉 프로젝트 계획서, 프로젝트 수행 계획서, 프로젝트 관리 계획서, 품질 보증 계획서 등을 작성하고 수정 및 저장을 위한 문서를 편집한다.
- 저장소 관리자(Repository Manager): 컴포넌트, 객체, 데이터모델, 논리적 데이터베이스 스키마 등의 소프트웨어에 대

한 메타 데이터를 공유하고 재사용할 수 있도록 하며, 객체의 실시간 버전을 저장하거나 참조할 수 있도록 한다. 이들을 위한 백업, 보안, 접근 제어, 오류 복구 등을 지원한다.

2. 분석 및 설계 지원

- 영역 모델러(Domain Modeler): 어플리케이션 영역에 필요한 기능에 대한 패밀리 사용 사례 모델 생성 및 편집과 추출된 클래스, 속성 및 관계 등을 기반으로 UML 기반의 패밀리 객체 모델을 개념 수준에서 작성하여 패밀리 사용 사례별로 객체간의 기능 흐름을 표현한다.
- 웹 서비스 컴포넌트 획득기(WebService Component Acquisitor): 재사용할 컴포넌트를 획득하기 위해서 웹 서비스 UDDI 레지스트리로부터 요구사항을 만족하는 컴포넌트 공급업체의 목록 및 WSDL을 얻고 WSDL을 검토 및 참조

하여 원격 프로시저를 호출하기 위한 스텝을 만든다[19].

- 아키텍처 모델러(Architecture Modeler): 획득한 재사용 컴포넌트를 포함한 컴포넌트 아키텍처와 시스템 아키텍처를 작성한다.
- 컴포넌트 모델러(Component Modeler): 컴포넌트 다이어그램 편집 및 컴포넌트 인터페이스 명세를 작성하고 컴포넌트 단위로 클래스도를 편집한다. 컴포넌트에 할당된 사용 사례별로 순차도를 작성한다. 클래스의 사전/사후 제약 조건을 명세에 나타낸다. 다른 도구와의 호환을 위해 XML로 변환한다.
- 컴포넌트 명세 작성기(Component Specification Reporter): 획득한 재사용 컴포넌트를 포함한 컴포넌트와 컴포넌트 인터페이스 명세를 작성한다.

3. 구현 및 시험 활동 지원

- 코드 생성기(Code Generator): 구현 환경을 고려하여 획득한 컴포넌트는 웹 서비스를 통해 컴포넌트를 호출하도록 코드를 작성하고 작성된 코드를 포함하여 설계 모델의 나머지 클래스에 대한 소스 코드를 생성한다. 컴포넌트를 개발하는 경우, J2EE 플랫폼 기반인 경우는 EJB 빈 컴포넌트를 생성하고 빈의 비즈니스 로직을 작성하고 .NET 플랫폼 기반인 경우는 BLO(Business Logic Object) 설계 요소를 클래스로 구현하고 비즈니스 로직을 작성한다.
- 컴포넌트 전개기(Component Deployer): 구현 환경을 고려하여 J2EE 플랫폼 기

반인 경우에는 EJB 컴포넌트의 DD(Deployment Descriptor)를 생성하고 JAR로 묶어서 어플리케이션 서버에 배치한다. .NET 플랫폼 기반의 경우 필요한 어셈블리에 대한 참조와 클래스 라이브러리의 출력 형식, 강력한 이름을 지정한 후에 빌드하여 COM+ 컴포넌트를 만들고 .dll컴포넌트를 어플리케이션 서버에 배치한다.

- 컴포넌트 테스터(Component Tester): 어플리케이션 서버에 전개된 컴포넌트 단위로 인터페이스를 시험한다.
- 통합 테스터(Integration Tester): 획득한 재사용 컴포넌트를 포함한 컴포넌트 미들웨어와 연결할 컴포넌트 및 인터페이스와 웹 페이지 요소들을 통합하여 페이지를 테스트한다.
- 사용자 인터페이스 생성기(User Interface Generator): 웹에 들어갈 콘텐츠 아이템과 그래픽 디자인에 대한 요구를 정의하고 웹 구조 및 네비게이션을 정의한다.
- 시스템 테스터(System Tester): 통합된 전체 시스템이 기대된 비즈니스 프로세스를 지원하는지 확인한다. 시스템이 요구된 성능을 만족하는지 검증한다. 테스트 케이스는 실제 운영 환경에서 사용되는 데이터를 사용해야한다. 스트레스 볼륨 테스트를 지원한다.

4. 기타 구성 요소

- 통합 GUI: 인터페이스와 개별 CASE도구들 사이의 통신을 위한 일관된 메커니즘을 제공한다.
- 툴 관리자: 툴 간 상호 운용성을 보다

원활하게 수행할 수 있도록 도구들의 행위를 제어한다.

- 문서 및 모델 정보 관리기: 객체 관리 층으로 하여금 데이터베이스와 상호 작용하게 해준다.

5. 제안한 개발 관리 도구의 평가

제안한 개발 관리 도구 모델에 대한 평가 항목은 컴포넌트 기반 개발 방법론들 중에서 컴포넌트 개발보다는 컴포넌트 기반 어플리케이션 개발에 비중[17]을 두며 국내의 실정을 잘 반영하고 있는 마르미-III의 프로세스에 따른 작업에 의해 생성되는 산출물들을 선정하였으며, 비교 대상 도구들은 컴포넌트 기반 어플리케이션 개발에 많이 사용되는 ROSE, Advantage Joe 그리고 Together 도구를 선정하였다. 산출물별 선정한 도구 및 제안한 개발 관리 도구의 지원 정도를 <표 2>와 같이 비교하여 평가하였다[3, 6, 8, 9].

마르미-III의 프로세스는 요구획득 단계에서 업무 모델링을 수행한다. 업무 모델링 시 대부분의 도구는 업무 모델을 생성할 수 있는 기능을 제공하지만, 단순한 UML 표기를 사용한 모델 생성 기능을 제공한다. 프로젝트 관리에 해당하는 계획 수립 및 수행계획서 작성을 위한 기능은 직접적으로 지원하지 않는다. 제안한 개발 관리 도구 모델은 프로젝트 관리 서포터를 사용하여 프로젝트 관리에서 요구되는 각종 계획서의 작성이 용이하다. Together는 메트릭 기능을 제공함으로써 소

프트웨어의 품질 및 사내 표준과의 일치성 검증이 가능하고 시스템의 정량적 품질 측정 결과를 Bar Graph, Distribution Graph 등의 형태로 제공한다. ROSE와 Advantage Joe는 품질 측정의 기능을 거의 지원하지 않는다. 제안한 개발 관리 도구 모델도 품질 보증 계획서는 작성할 수 있지만 시스템의 정량적 품질 측정 기능은 제공하지 못하는 단점이 있다. Together는 형상/버전관리를 위해서 지역 레포지토리 방식이 내장되어 있고 다른 형상 관리 도구와 연동하여 해당 기능을 지원하고 있다. 프로토타입 생성, 요구사항 추적 등의 기능은 제공하지 않는다. Advantage Joe는 형상 관리 기능이 없으며 제안한 도구는 버전관리 기능이 있고 ROSE는 버전관리와 체크인, 체크아웃 기능을 지원하며 형상관리 도구를 연결하여 사용할 수 있다. 다른 일반적인 CASE 도구들이 갖는 분석 및 설계에 활용되는 컴포넌트 아키텍처, 컴포넌트 인터페이스 및 내부설계 기능, 생성된 다이어그램으로부터의 코드 생성은 도구들 모두 지원하고 있다. 그리고 작성된 모델의 문서화 기능을 ROSE와 Advantage Joe는 일부 지원하며 제안한 개발 관리 도구 모델과 Together는 문서 포맷 및 양식을 재정의 할 수 있는 스크립트 방식의 템플릿 디자이너를 제공한다. Together는 사용자에게 맞는 양식을 작성하여서 개발 전 단계에서 작성되는 다이어그램에 대한 정보 소스를 통한 클래스 및 패키지, 함수와 속성 등의 모든 정보 추출이 가능하며 품질 결과 및 시험 결과의 문서화도 가능하다. 제안한 개발 관리 도구 모델, Together와 Advantage Joe는 웹 문서를 생성하고, 실제한 컴포넌트

〈표 2〉 기존 도구와 제안한 개발 관리 도구의 마르미-III 산출물 생성 지원 정도에 따른 평가

마르미-III 산출물	Together	ROSE	Advantage Joe	제안한 도구
시스템비전기술서, 업무정의서	△	△	△	○
현행시스템분석서	△	△	△	○
프로젝트계획서, 프로젝트수행계획서	X	X	X	○
프로젝트관리계획서	X	X	△	○
유스케이스기술서	○	○	○	○
품질보증계획서	○	X	X	△
논리객체관계도, DB설계서, 물리적DB	○	○	○	○
클래스모형기술서, 비즈니스컴포넌트모형기술서, 시스템아키텍처정의서, 컴포넌트상관도, 컴포넌트순차도/협력도, 컴포넌트명세서	○	○	○	○
UI프로토타입, 아키텍처프로토타입	○	X	X	△
UI흐름도, 사용자화면정의서, 컴포넌트인터페이스정의서	○	△	△	○
UI코드, 컴포넌트코드	○	○	○	○
JAR파일생성	○	X	△	○
컴포넌트테스트, 시스템테스트	△	△	△	○
테스트계획서	X	X	X	○
컴포넌트통합테스팅	○	X	○	○
재사용컴포넌트목록및획득	△	△	X	○

지원함:○, 일부지원함:△, 지원안함:X

의 전개 및 시험을 도구 내에서 수행할 수 있으며 설계한 모델의 통합 시험도 수행할 수 있다. 그러나 시험 정보의 관리 기능은 제공하지 않으므로 시험 정보를 관리하기 위해서는 별도의 작업이 요구된다. ROSE와 Together는 기존 시스템의 소스코드로부터 클래스 다이어그램을 생성할 수 있는 역 공학 기능을 제공하지만 Advantage Joe와 제안한 개발 관리 도구 모델은 역 공학 기능을 제공하지 않

는다. 데이터 모델링의 경우 Together를 제외한 나머지 도구들은 가장 보편적으로 사용되는 관계형 데이터베이스 모델의 생성 기능이 마약하므로 별도의 모델링 도구를 활용하기도 한다. 제안한 개발 관리 도구는 웹 서비스를 이용하여 컴포넌트를 획득하고 컴포넌트 프로시저를 호출하여서 결과를 반환 받을 수 있어서 컴포넌트의 재사용성을 높일 수 있다.

6. 결 론

본 논문에서는 컴포넌트 기반 개발 시에 사용되는 도구들로부터 도구의 요구사항을 추출하여 컴포넌트 기반 개발을 위한 도구의 기능을 분석 및 설계, 프로젝트 관리 지원, 구현 및 타 단계 지원, 그리고 기타 기능으로 구분하여 제시하였다. 또한 제한한 도구의 평가는 컴포넌트 기반 어플리케이션 개발에 비중을 두고 있는 국내에서 개발된 컴포넌트 기반 개발 방법론인 마르미-III의 산출물 생성 지원 정도로 평가하였다. 컴포넌트 기반 개발 시 사용되는 도구가 필수적으로 가져야 할 기능들을 나타내고 있으며, 이러한 기능적 요구사항 이외에 추가적으로 개발 관리 기능을 갖춘 도구는 컴포넌트 기반 개발의 각 단계의 산출물 생성을 적절히 지원할 수 있으며 웹 서비스를 이용하여 컴포넌트를 획득하고 컴포넌트를 재사용함으로써 재사용을 높일 수 있을 것으로 기대된다.

또한, 이를 효과적으로 활용하기 위해서는 도구가 갖추지 못한 기능적 요구사항들을 포함하는 타 도구와 연동이 되며 사용하기 용이하게 개발하여야 한다. 본 논문에서는 이러한 기능적 요구를 수용하는 도구를 제시하였다. 이와 더불어 개발 프로세스 상에 사용되는 타 단계의 도구가 지원하는 기능과의 중복을 피하기 위해 도구가 지원하는 기능들의 명확한 경계를 설정하는 작업이 필요할 것이다. 제시한 도구를 토대로 컴포넌트 기반 개발을 위한 통합 프로젝트 지원 환경 수준의 통합 CASE 도구 개발의 연구가 요구된다.

참 고 문 헌

- [1] 정기원, 조병호, "소프트웨어 프로세스 모델 설계 방법과 프로세스 모델 기반 CASE 도구에 관한 연구." 1995.
- [2] 조진희, "컴포넌트 기반 개발 방법론의 비교(컴포넌트 식별 기법 중심)." IT Forum Korea 2001, 2001.
- [3] 한국전자통신연구원, "마르미-III v2.0." 2002.
- [4] Cheesman J., Daniels J., "UML Components." Addison-Wesley, 2001.
- [5] Grtner Research, "Successfully selecting Object-Oriented A&D Tools," 2002.
- [6] http://www.cai.co.kr/solutions/advantage/application_development_tool/joe.shtml
- [7] <http://www.microtool.de/objectiF/de/index.htm>
- [8] <http://www306.ibm.com/software/awdtools/rup/index.html>
- [9] <http://www.borland.com/togethersoft/index.html>
- [10] Jacobson, Griss, Jonsson, "Software Reuse." Addison-Wesley, 1997.
- [11] Jim Conallen, "Modeling Web Application architectures with UML." Communication of the ACM, Vol 42, No 10, 1999.
- [12] Natalia Juristo, Ana M. Moreno, and Andres Silva, "Is the European Industry Moving toward Solving Requirements Engineering Problems?." IEEE Software Nov/Dec 2002, Vol 19, No 6, pp.70 ~ 77, 2002.

[13] Rajiv D. Banker, Robert J. Kauffman, Charles Wright, and Dani Zweig, "Automating Output Size and Reuse Metrics in a Repository-Based Computer-Aided Software Engineering(CASE) Environment." IEEE, 1994.

저 자 소 개



김영희 (E-mail : kyhse@ssuci.ac.kr)
 1974. 송실대학교 전자계산학과(학사)
 1986. 송실대학교 산업대학원 전자계산학과(석사)
 1973~1978. 한국은행 사무개선부 전자계산과
 1995~현재 송실대학교 대학원 전자계산학과 박사과정(수료)
 1979~현재 송실대학교 전산원 소프트웨어정보학과 교수
 관심 분야 소프트웨어 개발 프로세스, 방법론, 모델링



정기원 (E-mail : chong@comp.ssu.ac.kr)
 1967. 서울대학교 전기공학과(학사)
 1981. 미국 알라바마 주립대학(헨츠빌) 전산학과(석사)
 1983. 미국 텍사스 주립대학(알링턴) 전산학과 박사
 1966~1968. 미8군(IBM 기계정비 담당)
 1971~1975. 한국과학기술연구소 전자계산실 연구원
 1975~1990. 국방과학연구소 책임연구원
 1990~현재 송실대학교 컴퓨터학부 교수
 관심분야 소프트웨어 개발 프로세스, 방법론, 모델링, 실시간 응용, 전자거래, 정보시스템 개발 및 평가