

테스트 시나리오 구현을 위한 UML 기반 CASE 도구의 기능 요구사항

정근애, 최진영
고려대학교 컴퓨터정보통신대학원 디지털정보미디어공학과
e-mail : jungka@korea.ac.kr

Functional Requirements of UML-Based CASE Tool for Test Scenarios

Geun-Ae Jung, Jin-Young Choi
Dept. of Digital Information & Media Engineering, Korea University

요 약

기업환경은 지속적으로 변화하고 있으며, 그 변화의 주기도 짧아지고 있다. 그리고 시스템은 어떠한 비즈니스 요구사항에도 유연하고 민첩하게 지원할 수 있어야 한다. 따라서 변화에 대한 신속한 대응을 위해 UML 방법론을 참조한 CASE 도구의 기능적 요구사항을 제시함으로써 화면 설계 표준과 테스트 기능을 지원함으로써 짧아지는 시스템 개발 주기와 고객의 요구사항을 신속하게 대응할 수 있는 표준화된 User Interface Diagram 으로 활용한다.

1. 서론

기업의 환경과 패러다임의 변화에 맞춰 소프트웨어 개발에서도 비용 감소와 높은 생산성에 대한 요구가 증가하고 있다. 재사용성 증대를 통한 유연한 어플리케이션 개발 절차와 매커니즘에 대한 필요성이 대두되면서 컴포넌트 기반 개발이 확산되었고, 프로젝트 개발 기간이 짧아지고 단기 개발을 선호하는 양상을 보인다. 그러나 컴포넌트 기반 개발에 익숙해 있거나 소프트웨어 공학에 대한 이해가 부족한 개발자와 고객사는 프로젝트를 수행하면서 산출되는 요구사항분석서, 화면설계서, 테스트계획서 등과 같은 문서에 대한 이해와 관리가 어려운 것도 사실이다. 또한 그동안 어플리케이션의 도구의 발전에 비해 소프트웨어 개발에 사용되는 CASE 도구의 상대적으로 빈약한 발전을 보여 왔다[1]. IBM 사의 ROSE 와 CA(Computer Associates)사의 Advantage Joe, 볼랜드(Borland)사의 Together 등은 UML 방법론을 적용하고 있으며, 컴포넌트 기반 개발을 지원하는 대표적인 CASE 도구로서 분석, 설계 및 구현 초기 단계까지의 작업들을 지원하고 있다.

본 논문에서는 환경 변화에 기민하게 대처할 수 있는 UML 기반의 CASE 도구의 기능적 요구사항을 제시하여 화면설계(User Interface) 및 테스트 기능을 지원함으로써 짧아지는 시스템 개발 주기와 고객 요구사항을 신속하게 대응하여 비용과 시간을 축소시켜 소프트웨어 개발 생산성을 높이는 데 기여할 것으로 기대한다.

본 논문은 다음과 같이 구성된다. 2 장에서는 대표적인 UML 기반 CASE 도구들의 특징과 기능에 대해

조사하고 해당 도구들이 갖춰야 할 기능적 요구사항을 도출한다. 3 장에서는 새롭게 제안한 User Interface Diagram 의 구성과 기능을 제시하고 4 장에서 예시를 통해 제시한 요구사항의 유용성을 검증한다. 5 장에서는 본 논문의 결론 및 향후 연구를 정리한다.

2. 관련 연구

컴포넌트 기반 어플리케이션 개발 CASE 도구는 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 기민한 개발 방법론 구현을 위한 UML 기반의 CASE 도구의 새로운 기능을 제시한다.

IBM 레쇼날 제품군은 레쇼날 소프트웨어에서 개발한 CASE 도구 제품군으로서 요구사항분석, 모델링, 그리고 형상관리 등을 지원하는 다양한 도구를 제공하고 있다. RUP(Rational Unified Process) 프로세스의 다양한 직업에 적합한 형태의 CASE 도구들을 지원하고 있으며, 각 도구들은 유기적으로 연결되어 사용될 수 있다. 레쇼날 제품군은 RUP 를 지원하는 측면에서는 효과적이나, 컴포넌트 개발에 있어서는 개발자가 각 도구의 사용에 대한 계획을 별도로 수립하는 작업이 필요하다[2].

CA 사의 컴포넌트 개발을 위한 CASE 도구에는 Advantage Joe 가 있다. Advantage Joe 는 프로세스 및 데이터를 모델링 할 수 있는 도구가 있으며, 컴포넌트 모델러가 있다. 요구사항분석에서 설계단계까지의 작업에 필요한 산출물을 개발하기 위한 도구와 전체

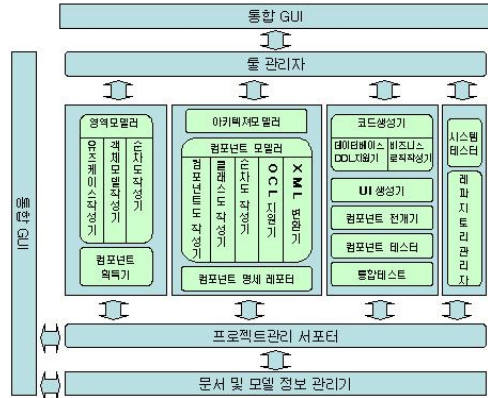
개발 프로세스 중 구현 초기까지의 작업을 수행할 수 있는 도구를 지원하고 있다. Advantage Joe 는 J2EE 기반의 컴포넌트 어플리케이션 개발 시 CBD96 에 기반하여 분석, 설계, 구현, 테스트 및 배포에 이르기까지 단계를 자동화하는 통합 개발 환경을 지원하고 있으며, 웹 어플리케이션 서버 환경의 프로그래밍을 지원하고 있다. Advantage Joe 는 CBD96 방법론⁽¹⁾에 따라 작업을 수행하도록 지원하고 있으므로 다른 방법론을 사용할 경우 제품에서 지원하는 여러 다양한 기능의 사용이 제한될 수 있다.[3]

Together 제품군은 불렛사의 CASE 도구 제품군으로서, 개발의 전 과정에서 모델과 코드를 실시간으로 동기화하는 것이 가능하다. CBD 방법론을 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 ROSE 에서 저장한 파일 사용 등의 다른 제품과 통합이 가능하고 사용자 보고서 설계를 위한 기능이 지원되고 있다.[4]

COBALT Constructor & Assembler 는 한국전자통신연구원에서 개발한 도구이며, COBALT Constructor 는 도메인 분석을 통한 컴포넌트 식별, 컴포넌트 단위의 설계 및 구현, 전개, 테스트의 컴포넌트 생성 전 단계를 지원하며, COBALT Assembler 는 시스템 아키텍처의 모델링, 컴포넌트의 개조와 합성, 전개, 테스트의 컴포넌트 조립 프로세스 단계를 지원한다[5].

이처럼 컴포넌트 기반 어플리케이션 개발 CASE 도구는 다양하며, 각자 고유한 특징을 가지고 있다. 기존 CASE 도구에서 지원하고 있는 개발 프로세스의 활동 및 작업을 분석하여 컴포넌트 기반 개발 도구가 지원해야 하는 요구사항을 (그림 1)과 같이 정리하였다[6]. 개발 관리 도구 모델은 컴포넌트 기반 개발 프로세스를 지원하고 이와 더불어 프로젝트 관리 프로세스를 지원하도록 설계되었다. 영역 모델러, 아키텍처 모델러, 컴포넌트 모델러 등의 기능을 제어할 수 있는 톨관리자 및 통합 GUI 를 통해 사용자가 쉽게 도구를 사용할 수 있어야 한다. 또한 CASE 도구의 문서작성 기능이 병행되어 문서관리 기능을 꺾하였다. 이렇게 작성한 개발 관리 도구 모델을 근간으로 기존 CASE 도구에 대해 비교하여 본다.

제한한 개발 관리 도구 모델을 기존 CASE 도구를 평가함에 있어 산출물 지원 정도에 따라 구분해 보았다. <표 1> 대부분의 CASE 도구는 업무 모델을 생성할 수 있는 기능을 제공되지만 단순한 UML 표기를 사용한 모델 생성 기능을 제공한다. 또한 프로젝트 관리에 해당하는 계획 수립 및 수행계획서 작성을 위한 기능은 직접적으로 지원하지 않는다.



(그림 1) 개발 관리 도구 모델

컴포넌트 전개 및 시험을 도구 내에서 수행 할 수 있는 시스템테스트 및 컴포넌트 테스트의 지원이 부족하며, 특히 시험 정보 관리 기능은 제공하지 않아 시험 정보를 관리하기 위해선 별도의 작업이 요구된다.[6]

<표 1> 산출물 지원 정도에 따른 기존 CASE 도구 평가

산출물	Together	ROSE	Advantage Joe
업무정의서	△	△	△
시스템분석서	△	△	△
프로젝트계획서, 프로젝트 수행계획서	X	X	X
프로젝트관리계획서	X	X	△
유즈케이스 기술서	○	○	○
품질보증계획서	○	○	○
ERD, DB설계서, 물리적 DB	○	○	○
클래스모델기술서, 비즈니스 컴포넌트 모형 기술서, 시스템 아키텍처정의서, 컴포넌트 상관도, 컴포넌트 순차도/협력도, 컴포넌트 명세서	○	○	○
UI프로토타입, 아키텍처프로토타입	○	X	X
UI흐름도, 사용자화면정의서, 컴포넌트 인터페이스 정의서	○	△	△
컴포넌트테스트, 시스템테스트	△	△	△
테스트계획서	X	X	X
통합테스트	○	X	○

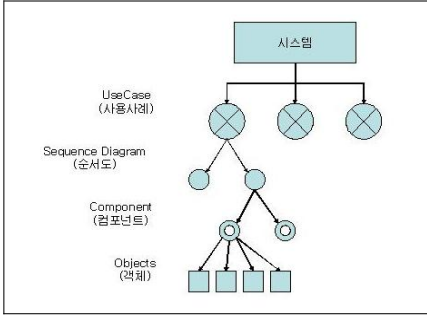
지원함: ○, 일부지원함: △, 지원안함: X

3. 새로운 CASE 도구의 기능적 요구사항

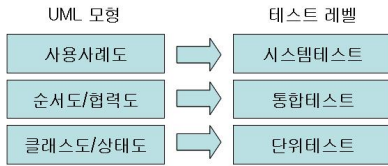
새로운 CASE 도구의 기능 요구사항은 화면 설계도(User Interface Diagram, 이하 UID*)를 통해 단위, 통합, 시스템 테스트 계획서를 작성하는 것이다. 화면 설계도는 Object 레벨로 모형 대상을 추출하고 (그림 2)와 같이 각 추출 단위별로 테스트 대상 레벨을 설정한다. 테스트 레벨에 따라 테스트의 레벨도 분리하여, 시스템 테스트는 사용 사례도(Use case diagram)과 연계하고, 통합 테스트는 UML의 순서도(sequence diagram), 그리고 단위 테스트는 클래스(Class diagram)와 연계하여 UML 모형들과 테스트 레벨과의 관계를

⁽¹⁾ CBD(Component based Development)96 은 1996 년 스탠랜드 소프트웨어에서 개발한 컴포넌트 기반 개발 방법론 중의 하나이다. CBD96 은 구체적인 개발 프로세스는 제시하지 않고 있으며, 단지 컴포넌트를 설계하고 개발하기 위해 필요한 설계 모듈들과 이를 산출하기 위한 작업 및 인터페이스 개발에 중점을 두고 있는 방법론이다.

(그림 3)과 같이 구성하였다[7].

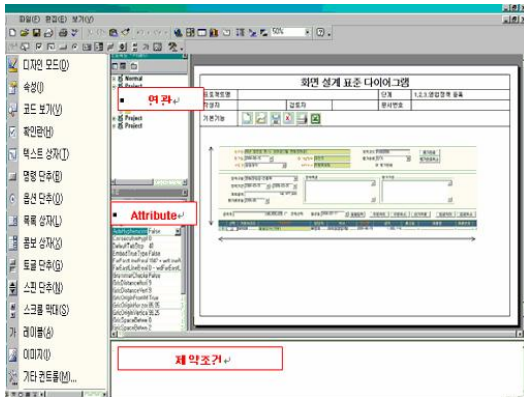


(그림 2) UML 모형 테스트 대상 추출



(그림 3) UML 모형과 테스트 레벨의 관계도

UML 기반의 UID* 도구는 (그림 4)와 같이 각 화면단위의 객체들과의 연관관계 계층도와 Object 단위의 속성 정보 창, 그리고 객체별 다양한 제약사항을 기록할 수 있는 제약사항 창을 구성된다. 연관, 속성, 제약사항은 테스트 시나리오 대상 정보로 활용된다.



(그림 4) 화면 설계 다이어그램

UID*의 객체의 정의는 <표 2>와 같이 어플리케이션 개발 시 사용되는 객체 단위로 구성하였으며, 각 객체의 Event를 정의하여 Event별 제약사항을 제약조건 항목에서 정의할 수 있도록 설계하였다. 객체는 Component 형식임으로 새로운 컴포넌트를 추가 및 삭제가 가능하다. (그림 4)에서는 C/S 환경의 UID*를 예시로 하였지만 어플리케이션의 환경과 프로그램의 종류는 제약사항이 아니다. 따라서 웹 환경 및 Embedded 환경의 UID* 설계도 가능하다.

<표 2> 화면설계 다이어그램의 객체 정의

Icon	Object Element	Object Events
	check box	Checked unChecked
	Text box	Entered Changed
	button	Clicked
	radio button	Checked unChecked
	label	N/A
	list box	Selected
	combo box	Selected
	picture	Clicked DbClicked
	group	N/A
	sheet	Clicked DbClicked
	form	Open Close

UID*의 객체 속성 정보는 그 객체 일반정보와 UML의 상속을 연관할 수 있는 상속정보로 구분된다. 일반 정보는 객체 ID, 객체의 데이터 타입, 길이 등이며, UML 연관 정보는 사용사례 다이어그램 NAME, 클래스 다이어그램 NAME, 순차 다이어그램 NAME으로 해당 UML 다이어그램을 상속받는다. 각 상속은 필수 조건은 아니다.

<표 3> 객체 속성 정보

Attribute Type	Attribute Desc.
Object ID	객체 ID
Object Value Type	객체 값 타입
Length	객체 값 길이
UseCaseName	사용사례 다이어그램의 액터 ID
ClassName	클래스다이어그램의 클래스 ID
SequenceName	순차다이어그램의 객체 ID

UID*의 제약조건 정보는 제약조건 우선 순위를 나타내는 제약순서, 제약대상이 값/이벤트/조건식 등을 구분할 기준항목, 해당 제약대상의 상세한 제약 조건식을 명시할 제약조건 항목, 테스트의 성격을 구분할 테스트기준과 성격에 따른 상세 항목 등으로 구성된다. <표 4>

<표 4> 제약조건 항목

제약조건 항목명	내용
제약순서	제약조건 우선순위
기준	Object Value, Event, 조건식 구분
제약조건	부등호, 조건문(IF), UI내의 다른 Object ID와의 관계 등의 제약조건
테스트기준	기능시험, 성능시험 구분
테스트항목	정량적 시험항목의 세부내용(*)

(*)<표 5> 참조

제약조건은 부등호, 조건문, 연산, 타 객체의 값 (ObjectID.value 형식) 등으로 표현 가능하다.

(그림 5) 화면 설계 다이어그램을 이용한 테스트계획서[이용한도 기준]

테스트계획서										
Component ID	카드시스템	Process ID	신규입회	시험대상 (화면명)	신용카드신청 등록 P/G		시험일			
테스트 순서	시험 사례 식별자	Use CaseName	ClassName	SequenceName	Object ID	테스트구분	제약조건	테스트 기준	테스트 항목	통과 여부
1	TW1	신용카드신청관리	신용카드신청:이용한도	SequenceName	이용한도	값	< {신용등급.value}	기능테스트	인수기준	
2	TW2	신용카드신청관리	신용카드신청:이용한도	SequenceName	이용한도	이벤트	이용한도.Changed:is_not_null	기능테스트	예외사항	
3	TW3	신용카드신청관리	신용카드신청:이용한도	SequenceName	이용한도	조건식	= {이용한도.value} * IF(mid({주민번호.value}, 7, 1)="2", 0.05, 0)	기능테스트	예외사항	

<표 5> 제약조건 테스트 항목 인수 조건 매트릭

내용	세부내용	설명
기능시험	인수기준	인수를 위한 정량적 기준 정의
	예외사항 정의	인수 후 세부적인 사용 사례에 대한 예상 예외 사항 정의
	시스템 기능	인수 후 전체 기능에 대한 예상결과 정의
성능시험	사용자 권한	인수 후 사용자별 권한 정의
	반응시간	인수 후 예상반응 시간 정의
	처리소요시간	인수 후 예상 처리 소요시간 정의
	처리률	인수 후 예상 처리률 정의

UID*에서는 (그림 5)와 같이 화면 단위별 테스트 계획서 양식을 제공하며, 테스트 결과는 데이터베이스로 관리된다.

4. 새로운 CASE 도구의 활용 예시

신용카드 신규 입회관련 요구사항을 UML 방법론으로 분석한 결과 다음과 같다.

- 사용사례 다이어그램: 신용카드신청관리 액터도출
- 클래스 다이어그램: 고객정보, 신용카드신청, 신용카드
- 순차 다이어그램: 신청요청->신용평가->신청서등록->카드발급->카드수령

이 중에 ‘신청서등록’ 항목에서 ‘이용한도’에 대한 제약사항은 1) 신용등급별 한도금액이 설정되며 한도를 초과해서 등록할 수 없다. 2) 여성고객일 경우 0.05% 추가 한도액을 부여된다. 이것을 이용한도_고객속성정보, 이용한도_제약조건은 (그림 6,7)과 같다.

Attribute Type	Attribute Desc.
Object ID	이용한도
Object Value Type	Number
Length	10
UseCaseName	신용카드신청관리
ClassName	신용카드신청:이용한도
SequenceName	신청서등록

(그림 6) 이용한도_객체 속성 정보

제약 순서	기준	제약조건	테스트기준	테스트항목
1	값	< {신용등급.value}	기능테스트	인수기준
2	이벤트	이용한도.Changed:is_not_null	기능테스트	예외사항
3	조건식	= {이용한도.value} * IF(mid({주민번호.value}, 7, 1)="2", 0.05, 0)	기능테스트	예외사항
4				

(그림 7) 이용한도_제약조건 정보

또한 해당 객체에 대한 테스트계획서는 (그림 5)와 같이 생성된다.

5. 결론

빠르게 변화하는 개발 환경 속에서 테스트가 강화된 UML 기반 CASE 도구에 대한 추가 기능 요구사항은 테스트가능성(Testability)[11]을 높이고 테스트 작업이 효과적으로 수행되기 위하여 미리 체크할 수 있는 기능을 제공한다.

본 연구에서 제안한 CASE 도구의 기능의 유용성을 검증하기 위해 다양한 어플리케이션 사례 연구를 통해 검증이 요구된다. 또한 테스트 결과는 시스템으로 처리되며, 해당 결과를 리포트 형식의 산출물로 출력할 수 있는 프로세스가 필요할 것이다. 그리고 소프트웨어 테스팅은 입력력 데이터 사이의 관계로만 정의하므로 함수 단위로 입력력 값의 변화나 함수 내의 각 Statement 단위의 연산결과 값의 변화를 관찰하여 결과값이 올바른 지를 판단하기 위한 연구가 와 다양한 테스트 이력에 대한 관리연구가 향후 필요할 것이다.

참고문헌

- [1] 정기원, 조병호 “소프트웨어 프로세스 모델 설계 방법과 프로세스 모델 기반 CASE 도구에 관한 연구”, 1995
- [2] <http://www-01.ibm.com/software/awdtools/rup>
- [3] <http://supportconnectw.ca.com/public/cool/joe/manuals/JBoss30Joe30.pdf>
- [4] <http://www.borland.com/us/products/together/index.html>
- [5] http://www.etri.re.kr/kor/search/view_01.jsp?div=01&sno=99
- [6] 김영희, 정기원, “컴포넌트 기반 개발을 위한 CASE 도구의 기능적 요구사항 및 개발관리 도구”, 2004
- [7] 김청아, 최병주, “UML 기반의 테스트 데이터 자동 생성 도구: AUTEG(Automatic UML-based Test Data Generating Tool)”, 2002
- [8] 서광익, 최은만, “요구명세의 테스트 가능성 검토와 측정 방법”, 2003
- [9] Sterling Software Component-Based Development Method, <http://www.sterling.com>
- [10] A.G Sutcliffe, N.A Maiden, S. Minocha, and D. Manuel, “Supporting Scenario-Based Requirements Engineering,” Software Eng., vol.24, no.12, pp. 1072-1088, Dec., 1998
- [11] R. Binder, “Design for Testability in Object-Oriented Systems”, *Communications of the ACM*, 1994, pp87-101