

요구사항 추적테이블을 이용한 객체지향 통합 테스트 시나리오 작성지원 도구의 설계 및 구현

최 신 형[†] · 한 판 암^{††}

요 약

본 논문에서는 명세를 기반으로 한 통합 테스트를 수행할 때 사용되는 시나리오 작성지원 절차를 제시하고, 시나리오 작성지원 도구를 구현하였다. 이를 위해 기본적으로 요구사항 정의테이블, 프로세스 정의테이블, 프로그램 대 테이블 상관도를 사용한다. 추가로 앞의 3가지 테이블 기능을 요구분석, 설계, 테스트 단계별로 나누어 한곳에서 신속·정확하게 체크할 수 있는 요구사항 추적테이블을 생성한다. 본 연구의 통합 테스트를 위한 시나리오 작성지원 도구의 출력물은 시나리오 작성을 위한 지침서 역할을 한다. 그 결과 검사팀의 시나리오 작성노력을 상당부분 줄일 수 있을 뿐만 아니라 요구사항의 누락을 방지하고, 해당 프로세스에 대한 테이블 정보를 알 수 있으므로 테스트할 때 잘못된 값을 입력하는 것도 방지할 수 있다.

Design and Implementation of a Scenario Composition Supporting Tool for Object-Oriented Integration Testing Using the Requirement Trace Table

Shin Hyeong Choi[†] · Pan Am Han^{††}

ABSTRACT

This paper suggests the scenario composition supporting method, which can be used to perform integration test based on specification, and implemented a scenario composition supporting tool. To compose scenario for object-oriented integration testing, requirement specification, process definition table and program to table relation diagram are used. In addition, requirement trace table is created. It can check user's requirements, quickly and exactly, dividing former three tables into the stages of requirement analysis, design, and test. The output of scenario composition supporting tool plays a guide part in composing scenario. This tool can reduce scenario composition effort of quality for the checking team and prevent an omission of requirement and input of errors during the test.

키워드 : 테스트 시나리오(Test Scenario), 객체지향 테스트(Object-Oriented Testing)

1. 서 론

소프트웨어 시스템이 복잡해짐에 따라 이에 대처하기 위해서는 더욱더 많은 노력과 비용이 투입되어야 한다. 더욱이 경쟁적인 시장에서는 꼭 짜인 스케줄로 인해 완벽한 시스템을 개발하기가 쉽지 않다. 이런 문제점이 대두된 후 이 분야에 대한 많은 연구가 진행되어왔고, 소프트웨어 분야의 발전에 힘입어 그 성과가 실제 업무에 활용되고 있다. 이런 소프트웨어 분야의 진전은 크게 프로그래밍과 컴퓨터 환경으로 나누어 살펴볼 수 있다.

우선, 프로그래밍 측면에서는 새로운 프로그래밍 방법으로 객체지향 프로그래밍(Object oriented programming : OOP)이 제시된 후, 소프트웨어 위기를 대처하기에 OOP가 적합하다고 생각했었다. 그러나 실제로 OOP 개념을 제공하지 않았

던 비주얼 베이직에 의해 도화선을 당긴 컴포넌트 시장이 비약적으로 활성화되었다.

다음으로 컴퓨터 환경분야에서는 1980년대 중반까지 단일 메인프레임 시스템이 주를 이루었고, 여러 문제점으로 인해 클라이언트-서버 환경이 제안되었다. 그러나 시스템 규모가 커짐에 따라 시스템 구조, 유지보수, 확장성에 있어서 한계에 도달했다. 그 대안으로서 분산객체 환경이 제안되었다[1]. 특히, 네트워크 환경하의 대규모 시스템이 주류를 이루는 현 상황에서는 컴포넌트 기반 개발방법이 대규모 어플리케이션 소프트웨어를 구축하기에 최적의 솔루션으로 각광받고 있다[1-4].

고성능 PC와 충분한 대역폭을 제공하는 네트워크에 의해 분산컴퓨팅이 가능하게 됨에 따라 분산시스템은 다양한 이기종 컴퓨터 환경에서 데이터와 응용 프로그램을 분산 운용하면서 프로그램의 상호운영성과 사용자에게 분산의 투명성을 제공할 수 있게 되었다. 하지만 다양한 프로그래밍 언어, 운영체제, 통신 매커니즘, 인터페이스 등의 사용으로

[†] 준 회원 : 경남대학교 컴퓨터공학과
^{††} 종신회원 : 경남대학교 컴퓨터공학과 교수
논문접수 : 2002년 2월 18일, 심사완료 : 2002년 5월 24일

다양한 변경 혹은 그에 따른 오류가 발생할 가능성이 더욱 더 높아지므로 소프트웨어 신뢰성 향상과 품질 측정을 위한 테스트 작업이 필요하다.

따라서 본 논문에서는 소프트웨어를 개발하는데 고품질을 유지할 수 있도록 사용자 요구사항부터 테스트단계까지의 각종 정보를 한곳에서 관리할 수 있는 요구사항 추적테이블 중심의 통합 테스트를 위한 시나리오 작성지원 방안을 제시한다. 추가적으로 각종 테이블 정보를 이용한 통합 테스트 시나리오 작성지원과 이의 자동화 방안에 대해서도 고찰한다.

본 논문의 구성은 2장에서는 관련연구로서 소프트웨어 테스트와 기존의 통합 테스트 방안의 소개와 문제점을 살펴본다. 3장에서는 통합 테스트를 위한 시나리오 작성 지원 방안을 제시하고, 4장에서는 시나리오 작성지원 도구를 설계하며, 5장에서는 적용사례를 살펴보고, 6장에서는 결론 및 향후 연구과제를 나타낸다.

2. 관련 연구

2.1 소프트웨어 테스트

소프트웨어 테스트는 소프트웨어 생명주기의 한 단계로서, 시스템이 정해진 요구를 만족하는지, 예상과 실제 결과가 어떤 차이를 보이는지 수동 또는 자동방법을 동원하여 검사하고 평가하는 일련의 과정이다.

2.1.1 테스트 전략

소프트웨어 테스트는 크게 단위 테스트, 통합 테스트, 시스템 테스트로 단계를 구분할 수 있다[5, 6].

단위 테스트는 테스트 담당자를 중심으로 각 프로그램 단위별 업무흐름의 순서에 준하여 데이터의 흐름에 이상 또는 오류가 있는지를 검증하는 것과 개발자가 각 함수 및 서브루틴별 정합성 및 오동작 여부를 검증하여 기능상의 오류를 제거하는 것으로 소스 코드상의 구문오류를 발견하기 위한 작업이다.

통합 테스트는 기능 테스트에서 검증한 시스템의 전체기능을 업무 흐름과의 정합성을 체크하는 것을 목적으로 요구사항에 대한 각종 이벤트 흐름이 올바르게 동작하는지를 검사하며, 업무 흐름상 발생 가능한 예외인 경우에 대해 대응이 가능한지를 검증하여 시스템의 보완 및 업무진행 방안의 재정의의 통하여 시스템 운영에 대비하도록 한다[7, 8].

시스템 테스트는 개발 머신에서 볼륨 테스트를 진행하고 운영시스템으로 이관을 진행한다. 이관 후는 개발사용 테스트 데이터베이스와 운영 데이터베이스의 구성을 별개로 유지하며 개발자와 최종사용자가 별개의 데이터를 관리할 수 있도록 한다.

2.1.2 테스트 방법

테스트 방법은 크게 화이트 박스 테스트와 블랙 박스 테스트로 구분한다[9]. 화이트 박스 테스트는 소프트웨어를 구

성하는 모듈 안의 작동을 자세히 관찰하는 방법으로 원시 코드의 내부 구조를 분석하여 테스트하는 프로그램 기반 테스트이다. 테스트 순서는 모듈 내의 제어흐름을 간선으로 표시한 그래프인 논리 흐름도를 작성한 다음 이를 기초로 테스트 케이스를 생성하여 실시한다. 블랙 박스 테스트는 모듈이 요구에 맞게 잘 작동하는가에 초점을 맞춘 것으로 프로그램의 원시코드를 분석하지 않고 기능중심의 명세기반 테스트라고 할 수 있다.

2.1.3 객체지향 소프트웨어 테스트 방법론

객체지향 구조의 특성으로 인해 기존의 테스트 방법을 달리할 필요가 발생되었다. 상속성, 다형성, 캡슐화와 같은 객체지향이 가지는 장점이 테스트 분야에서는 여러 가지 요인을 고려해야만 하는 사항을 발생시킨다[10, 11]. 초기의 테스트 방법론들은 객체 개념으로 인해 블랙박스 테스트를 주로 사용하였으나 객체 내부의 소스코드에 대한 화이트박스 테스트 방법도 적용되고 있다. 객체지향 테스트에 관한 연구는 다음과 같다[12].

① Weyker가 제안한 적합성 테스트 기준

Weyker가 제시한 공리들은 테스트 수행자들이 직관적이고 경험적인 관점에서 공감하는 것들을 형식화한 것이다.

② Perry와 Kaiser의 테스트 방법

Weyker의 적합성 공리를 클래스에 적용하여 객체지향 프로그램의 테스트를 제안하였고 메소드를 상속받을 시 테스트 문제, 메소드의 재정의시 테스트 문제 등이 테스트 대상이다.

③ Harrold와 McGregor의 테스트 방법

상속성 구조를 가지는 특성에 따라 기본 클래스의 테스트 정보를 재사용하여 상속된 클래스의 테스트를 수행한다.

④ Smith의 FOOT

객체지향 프로그램 구조에서 테스트를 실시해야할 여러 가지 전략을 소개하고 FOOT라는 객체지향 프로그램 테스트 구조를 소개하였다.

현재의 소프트웨어들은 날이 갈수록 그 요구사항이 증가하고 규모 역시 커져서 여러 사람이 팀을 이루어 개발하는 것이 일반적이므로 서로 다른 개발자가 구현한 규모가 큰 코드들을 통합하기가 무척 어려운 일이다. 또한, 하나의 시스템이 언어나 운영체제 등의 소프트웨어 개발환경의 한계를 벗어나서 서로 다른 여건에서 구현되는 추세이므로 이들의 통합 역시 중요한 이슈로 떠오르고 있다. 그러나 아직 통합을 위한 테스트는 단위 테스트나 시스템 테스트 레벨에 비하여 매우 연구가 미흡한 상태이다.

2.2 기존의 통합 테스트 방안과 문제점

객체지향 시스템의 통합 테스트는 시스템 단계에서는 보이지 않는 메모리 이벤트나 조건도 테스트하는 것이다. 그러므로 기존의 시스템의 통합 테스트는 원시코드로부터 방

향성 그래프를 추출하여 MM(Method-Message)-Path를 구하여 이들의 조합으로 테스트 케이스를 생성하였다[13]. 그런데 이런 방법은 다음과 같은 문제점이 발생한다[14].

- ① 메시지의 전달을 통한 멤버 함수의 호출 순서만 테스트 하고 멤버 데이터의 경우는 배제하였다. 즉, 멤버 함수의 실행 결과에 대한 테스트는 따로 했거나 그 멤버 함수가 맞게 실행한다는 가정 하에 테스트가 이루어진다.
- ② 통합 테스트는 객체 상호간의 동적인 작용을 테스트하는 것이므로 모든 경우를 고려하여 자동화된 테스트를 할 수는 없다.
- ③ 상태 기반 통합 테스트에서 상태란 객체의 멤버의 값을 의미하지만 실제로 그 값들을 추출하기가 어렵다. 그 이유는 상태 표현에서 멤버 함수의 실행 결과를 구체적인 자료의 값이 아닌 추상적인 상태로 표현하기 때문이다. 그러므로 테스트 케이스를 이용하여 테스트를 한 후 그 결과를 컴퓨터를 이용하여 비교할 수 없다.

이와 같이 원시코드만으로는 통합 테스트에 필요한 정보를 추출하기 어렵다. 이를 보완하기 위해 시나리오를 이용한 통합 테스트 방안이 제시되었다[14, 15]. 이는 시나리오가 연속된 멤버 함수의 집합으로 테스트케이스를 형성할 때 순서를 결정하는데 도움을 주기 때문에 기존의 문제점을 상당수 해결하였다. 하지만 이 방법 역시 인스턴스화된 시나리오를 생성하기 위해서는 사용자가 직접 작성한 매핑 시나리오를 이용하여야 한다.

3. 시나리오 작성 지원도구의 체계

본 논문에서는 통합 테스트를 위해 시나리오를 이용한다는 전제 하에 명세를 기반으로 통합 테스트를 수행할 때 사용될 수 있는 시나리오 작성지원방안을 제시하고, 이를 위한 시나리오 작성지원 도구를 구현한다. 통합 테스트의 시나리오를 구성하기 위하여 참조되는 문서는 요구사항 정의테이블, 프로세스 정의테이블, 프로그램 대 테이블 상관도로 한다.

3.1 시나리오 작성을 위한 테이블

3.1.1 요구사항정의테이블

<표 1>의 요구사항 정의 테이블은 개발하고자 하는 시스템 하드웨어나 소프트웨어의 규격, 요구 조건 등에 관한 완전한 명세를 기술해 놓은 테이블로서 대중소로 구분한 요구사항별로 구성한다.

<표 1> 요구사항 정의테이블 구성

업무명	요구 ID	입력	출력	처리내용	수용여부
수용가시설관리-급수공사처리부관리	RQFWT-031	급수공사민원접수내역, 급수공사처리부	급수공사처리부	급수공사처리부 입력/수정/조회/출력	○

3.1.2 프로세스정의테이블

<표 2>의 프로세스 정의 테이블은 개발될 시스템의 기능별로 프로세스 명칭과 ID를 부여하고 관련테이블 항목에 프로세스 수행에 관련된 테이블 정보를 포함한다.

<표 2> 프로세스 정의테이블 구성

기능명	단위 프로세스ID	단위 프로세스명	설명	관련테이블
수용가시설관리	26.2.1	급수전 관리대장 입력	급수전 정보입력	WTT_급수전관리(C) WTT_급수전업종변경이력(C) WTT_급수전시설변경이력(C)

3.1.3 프로그램 대 테이블 상관도

<표 3>의 프로그램 대 테이블 상관도는 프로그램 목록상의 모든 프로그램이 정확히 상관도 상에 나열되었는지를 검토할 수 있는 자료로서 각 프로그램에서 사용하는 테이블에 대한 상관관계가 정확히 반영되었는지를 체크할 수 있다.

<표 3> 프로그램 대 테이블 상관도 구성

테이블		WTL_기타정보형시설	...	WTT_급수전관리	...	WTT_이용불편사항	...
프로그램ID	프로그램						
WTC010CD	급수전 입력/수정			CRUD		CRUD	

3.1.4 요구사항 추적테이블

본 논문에서는 사용자 요구사항에 대한 효과적이고 빠짐 없는 검토와 적용을 위해 요구사항 정의테이블, 프로세스 정의테이블, 프로그램 대 테이블 상관도 기능을 요구분석, 설계, 테스트 단계별로 나누어 한 곳에서 신속하고 정확하게 체크할 수 있도록 <표 4>와 같은 요구사항 추적테이블을 별도로 작성한다. 이때 프로세스 ID와 프로그램 ID의 대응관계를 정리해 놓은 프로그램 정의테이블을 활용한다.

<표 4> 요구사항 추적테이블 구성

요구분석		설계단계		테스트단계		
업무명	요구 ID	프로세스 ID	프로그램 ID	빌드명	빌드 ID	테스트 여부
급수공사처리관리	RQFWT-031	26.2.1	WTC 010CD	급수공사처리	급수공사처리관리-T	○

요구사항 추적테이블은 복잡한 각종 테이블 항목 중에서 명칭과 ID 중심으로 연결관계를 정리해 놓은 것으로서 나머지 3가지 테이블의 중심에서 연결관계에 따른 각 항목의 전달기능을 담당한다.

즉, 요구사항 추적테이블이 생성되면 먼저 요구사항 정의테이블로부터 업무에 해당하는 요구 ID를 입력받아 동일 레코드의 대응되는 프로세스 ID를 선택한다. 다음으로 프로세스 정의테이블을 검색하여 선택된 프로세스 ID에 관련된 테이블정보를 구한다. 또한 프로세스 ID에 해당하는 프로그램 ID를 선택하여 프로그램 대 테이블 상관도에 전달한다. 입력된 프로그램 ID의 프로그램명과 상관도상에 나타난 관

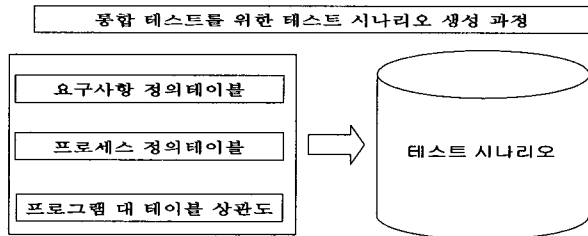
런 테이블 정보를 구한 다음 프로세스 정의테이블에서 구한 테이블 정보와 일치 여부를 확인한다. 이때 프로세스 정의테이블에 있는 관련 테이블 항목과 프로그램 대 테이블 상관도에 관련표시가 되어있는 테이블 항목이 일치하지 않으면 요구분석 및 설계과정에서 잘못 기재한 경우이다. 따라서 요구분석 및 설계팀에 검토를 요청하여 수정된 자료를 다시 입력받는다. 그리고 일치하면 테이블 정의서로부터 해당되는 테이블의 칼럼 형식, 길이 등의 정보를 수집한다. 여기서 두 가지 테이블 정보의 일치 여부를 확인함으로써 요구분석 및 설계단계에서 작성된 명세서 내용이 정확한지를 파악할 수 있으며, 오류가 없는 정보를 테스트 단계에 전달할 수 있으므로 테스트의 신뢰도를 높일 수 있다.

또한, 요구사항 추적테이블은 통합 테스트를 위한 시나리오 작성단계의 테스트 단위별로 구성되어있으므로 요구사항별로 테스트 여부를 확인할 수 있다.

3.2 시나리오 작성지원 방안

통합 테스트를 위한 시나리오 작성지원은 (그림 1)을 기초로 <표 5>와 같은 단계로 진행된다.

(그림 1)처럼 요구분석 및 설계과정을 통해 작성된 명세서에 기초하여 통합 테스트를 실시하는데 사용되는 시나리오를 작성할 수 있다. 이때 <표 5>에 제시한 시나리오 작성지원 절차를 따르면 요구사항에 해당하는 각종 정보를 포함하는 시나리오 작성지원 보고서를 출력할 수 있다.



(그림 1) 통합 테스트를 위한 시나리오 생성과정

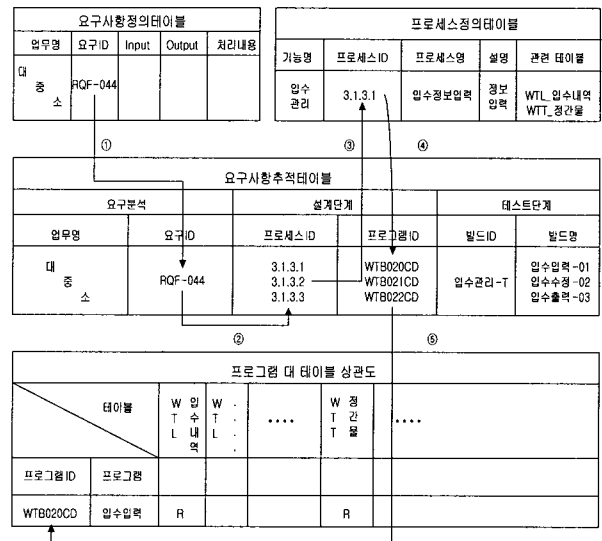
<표 5> 시나리오 작성지원 절차

- Step 1 : 요구사항 정의테이블에 대중소로 구분된 업무명을 참고로 하여 해당 업무의 요구 ID를 선택
- Step 2 : 요구사항 추적테이블에서 Step 1의 요구 ID에 대응하는 설계단계의 해당 프로세스 ID를 찾음
- Step 3 : 프로세스 정의테이블에서 Step 2의 프로세스 ID에 해당하는 관련 테이블 정보를 얻음
- Step 4 : 테이블 정의서에서 Step 3의 테이블에 해당하는 칼럼들의 형식과 길이 정보를 얻음
- Step 5 : 요구사항 추적테이블에서 Step 2의 프로세스 ID에 대응하는 설계단계의 프로그램 ID에 해당하는 프로그램을 선택
- Step 6 : 프로그램 대 테이블 상관도에서 프로그램과 관련된 테이블을 재확인

(그림 2)는 통합 테스트를 실시할 때 필요한 테스트 시나리오 작성지원 보고서를 만드는 과정으로 요구사항 추적테

이블 중심의 연결관계로 설명하고 있으며, 5장의 적용사례에서 다룰 시설물 종합관리 시스템의 업무중 문서관리업무에 대해 <표 5>에 제시된 절차에 따른 작업흐름을 나타낸다.

생성된 요구사항 추적테이블을 중심으로 먼저 요구사항 정의테이블로부터 입수관리 업무에 해당하는 RQF-044라는 요구 ID를 입력받아 동일 레코드에 대응되는 프로세스 ID인 3.1.3.1을 선택한다. 다음으로 프로세스 정의테이블에서 3.1.3.1에 관련된 테이블 정보인 WTL_입수내역, WTT_정간물을 구한다. 다시 요구사항 추적테이블에서 3.1.3.1에 해당하는 프로그램 ID인 WTB020CD를 선택하여 프로그램 대 테이블 상관도에 전달한다. 입력된 WTB020CD라는 프로그램 ID의 프로그램명과 상관도상에 나타난 관련 테이블 정보인 WTL_입수내역, WTT_정간물을 구한 다음 앞서 프로세스 정의테이블에서 구한 테이블 정보와 일치 여부를 확인한다. 두 가지 테이블 정보 모두 WTL_입수내역, WTT_정간물이라는 관련 테이블을 가지므로 테이블 정의서로부터 WTL_입수내역, WTT_정간물에 해당하는 입수제목 (CHARACTER, 30), 입수번호(NUMBER, 10) 등의 테이블 정보를 구할 수 있다.



(그림 2) 통합 테스트를 위한 시나리오 작성지원 보고서 산출 과정

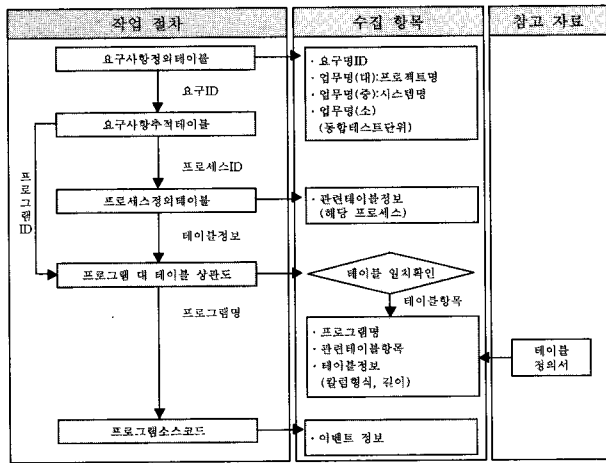
이와 같은 과정을 통해 통합 테스트를 실시할 때 사용되는 시나리오 작성을 지원하는 보고서의 출력정보를 수집한다.

4. 시나리오 작성 지원도구의 구현

본 논문에서 제안한 통합 테스트를 위한 시나리오 작성 지원방안을 작업절차, 수집항목, 참고자료로 나누어 구성하면 (그림 3)과 같다.

테스트 기능별로 시나리오를 작성하기 위해 우선적으로 요구사항 정의테이블에서 업무의 대·중·소에 따라 프로

젝트명, 시스템명, 업무명을 구할 수 있고, 그에 해당하는 요구 ID도 구한다.



(그림 3) 통합 테스트를 위한 시나리오 작성지원 작업절차 및 출력물

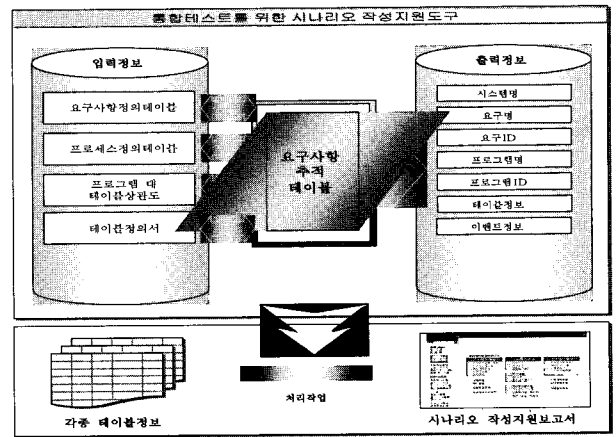
다음으로 요구 ID에 대응하는 프로세스 ID를 프로세스 정의테이블 검색을 통해 관련테이블 정보를 수집한다. 이 테이블 정보를 프로그램 대 테이블 상관도에서 해당 프로그램의 테이블 정보와 일치 여부를 확인한 후 테이블 정의서를 참고하여 테이블 항목, 칼럼 형식과 길이 등의 테이블 정보와 프로그램 명을 구할 수 있다. 마지막으로 해당 프로그램을 소스코드를 참고하여 <표 6>의 각종 이벤트 정보를 구한다. 여기서 이벤트 정보란 해당 프로그램에서 선택할 수 있는 메뉴나 버튼과 같은 컨트롤 도구에 관련된 각종 동작만을 기술해 놓은 것을 말한다.

<표 6> 이벤트 정보

순번	항목	설명
1	Form	단위 프로그램의 실행화면
2	Data	데이터베이스 파일과 관련 테이블
3	Control	각종 컨트롤 도구에 관련된 사항
4	Procedure	컨트롤 도구에 관련된 이벤트를 기술해 놓은 코드

(그림 3)의 작업절차에 따른 테스트 시나리오 작성지원 도구의 구성은 (그림 4)와 같고 출력물로서 시나리오 작성지원 보고서를 구할 수 있다.

통합 테스트를 위한 시나리오 작성 도구는 입력정보로서 요구사항 정의테이블, 프로세스 정의테이블, 프로그램 대 테이블 상관도, 그리고 보조자료로서 테이블정의서를 사용한다. 그런 다음 입력된 정보를 기초로 요구사항 추적테이블을 생성하여 시나리오 작성지원 보고서의 출력항목이 될 시스템명, 요구명, 요구 ID, 프로그램명, 프로그램 ID, 칼럼·길이 등의 테이블 정보와 소스코드를 통해 얻을 수 있는 이벤트 정보를 구한다.



(그림 4) 통합 테스트를 위한 시나리오 작성지원도구 구성

(그림 4)의 출력항목별로 시나리오 작성지원 보고서가 완성되면 검사팀은 기존의 방식처럼 요구분석 및 설계단계에서 산출된 각종 명세서를 일일이 찾아서 대조해가며 시나리오를 작성하는 불편을 하지 않아도 된다. 이 과정에서 상당한 시간과 노력을 줄일 수 있다.

5. 적용 사례

본 연구를 통해 개발한 통합 테스트를 위한 시나리오 작성지원 도구를 S사의 시설물 종합관리 시스템 개발에 적용하여 시나리오를 생성한 결과를 기술하였다. S사의 시설물 종합관리 시스템은 <표 6>의 요구사항 정의테이블에 나타난 바와 같이 크게 3가지 업무로 이루어져 있으며, 각 업무는 데이터베이스 입력 및 수정 등의 관리와 조회업무로 구성된다.

5.1 입력으로 사용되는 테이블

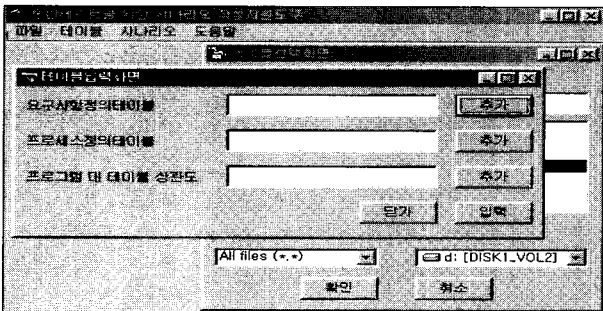
통합 테스트에 사용될 시나리오를 작성하기 위해 입력되는 테이블은 S사의 시설물 종합관리 시스템 개발과정 중 요구분석 및 설계단계에서의 산출물 중에 <표 7>과 같은 산출물을 대상으로 하였다.

<표 7> 시나리오 작성을 위한 입력 테이블

- 요구사항 정의테이블
 - 시설물종합관리 시스템을 3가지 업무(도로, 상수도, 하수도)로 나누고 세부업무에 대해 현업 사용자의 요구를 기술해 놓은 테이블로서 5개의 필드(업무명, 요구 ID, 입력, 출력, 수용여부)로 구성(<표 1> 참조).
- 프로세스 정의테이블
 - 크게 3가지 업무로 나누어진 시스템을 상·하수, 도로시설물 관리와 같은 기능에 따라 프로세스별로 구성된 테이블로 5개의 필드(기능명, 프로세스 ID, 프로세스명, 설명, 관련테이블)로 구성(<표 2> 참조).
- 프로그램 대 테이블 상관도
 - 프로세스를 구현한 프로그램에서 사용되는 테이블에 대한 상관 관계를 기술한 테이블로서 3개의 필드(프로그램 ID, 프로그램명, 테이블)로 구성(<표 3> 참조).

5.2 시나리오 작성지원 절차

(그림 5)는 5.1절에서 기술한 3가지 테이블을 입력하는 화면으로 이것을 통해 본 논문에서 제안한 통합 테스트를 위한 시나리오 작성지원방안의 중심 테이블인 요구사항 추적테이블을 생성할 수 있다. 즉, 임의의 경로에 있는 해당 데이터베이스 파일을 선택하여 요구사항 정의테이블, 프로세스 정의테이블, 프로그램 대 테이블 상관도를 각각 선택한다. 그런 다음 상단 메뉴 중 테이블의 하위 메뉴인 테이블 생성을 클릭하면 중심연결 역할을 하는 요구사항 추적테이블이 생성된다.



(그림 5) 기초 테이블 입력화면

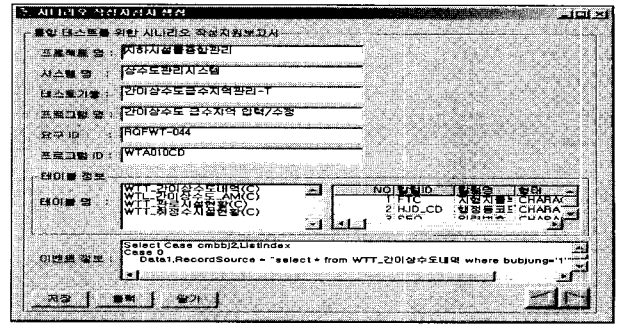
(그림 6)은 3가지 입력 테이블을 바탕으로 요구사항 추적테이블을 생성한 결과이다.

이름	요구ID	프로세스ID	프로그램ID	비드문	비드문	테이블명
간이상수도급수지역관리	ROFWT-044	2.1.3.1	WTA010CD	간이상수도급수지	간이상수도급수지역관리	X
간이상수도급수지역관리	ROFWT-044	2.1.3.2	WTA010CD	간이상수도급수지	간이상수도급수지역관리	X
간이상수도급수지역관리	ROFWT-044	2.1.3.3	WTA010CD	간이상수도급수지	간이상수도급수지역관리	X
수입발포도관리	ROFWT-045	2.1.1.1	WTF010CD	수입내역조회-01	수입관리-1	X
수입발포도관리	ROFWT-045	2.1.1.2	WTF010CD	수입내역조회-02	수입관리-1	X
수입발포도관리	ROFWT-045	2.1.1.3	WTF010CD	수입내역조회-03	수입관리-1	X

(그림 6) 요구사항 추적테이블 생성결과

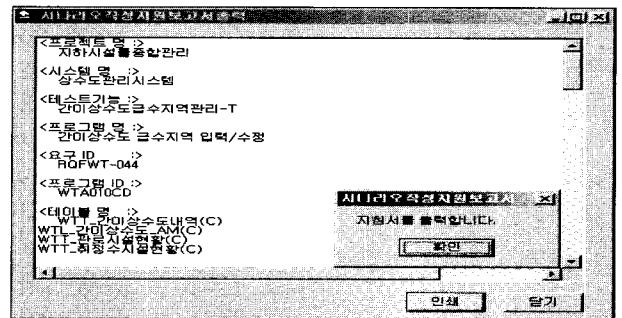
이 테이블은 개발과정을 요구분석, 설계, 테스트 단계별로 나누어 한 곳에서 신속하고 정확하게 체크할 수 있으므로 품질보증분야에도 이용된다. 즉, 생성된 요구사항 추적테이블의 필드값이 비어있다면 분석, 설계과정 중 해당자료를 누락하였거나 잘못 기입했을 경우가 높다. 이런 경우에는 해당업무에 대해 분석, 설계팀으로의 피드백을 통해 해당자료를 확인·정정후 다시 실시한다. 이렇게 보완된 자료를 이용함으로써 통합 테스트는 좀 더 정확한 자료를 바탕으로 실시될 수 있다.

생성된 요구사항 추적테이블을 중심으로 3.2절과 (그림 3)의 통합 테스트를 위한 시나리오 작성지원절차에 따라 시나리오 작성지원 보고서를 생성한 결과는 (그림 7)과 같다. 또한, (그림 8)은 테스트 단위별 시나리오 작성지원 보고서의 출력양식이다.



(그림 7) 항목별로 생성된 시나리오 작성지원 보고서

(그림 8)과 같은 시나리오 작성지원 보고서가 출력되면, 이것을 참고로 하여 검사팀은 테스트 기능별로 통합 테스트를 실시하기 위한 시나리오를 작성할 수 있다. 우선, 시스템명과 테스트 기능 항목값을 참고로 하여 상수도 관리 시스템의 기능중 간이상수도 급수지역관리-T 기능에 대해 시나리오 작성을 시작할 수 있다. 다음으로 프로그램 ID 항목값인 WTA010CD라는 프로그램 소스를 분석한 이벤트 정보와 WTT_간이상수도 내역과 같은 관련테이블 정보인 칼럼형태와 길이를 참고로 시나리오 항목을 작성한다.



(그림 8) 통합 테스트를 위한 시나리오 작성지원 보고서 출력결과

예를 들면, 항목별로 생성된 시나리오 작성지원 보고서인 (그림 7)의 이벤트 정보를 살펴보면, cmbbj2라는 이름을 가지는 콤보박스의 각 항목을 선택할 때 해당 테이블을 불러오는 코드(Data1.Record Source = "select * from WTT_간이상수도내역 where bubjung = 1")가 작성되어있다는 것을 알 수 있다. 즉, 이벤트 정보 중 콤보박스를 통해 행정동, 법정동, 관리기관, 준공년월일, 수원종류를 선택할 수 있다. 그리고 테이블 정보 중 행정동 코드는 문자형 10자리, 관리기관 코드는 문자형 7자리라는 조건으로 검색을 해야함을 알 수 있다.

한편 (그림 9)의 이벤트 정보를 통하여 초기화, 확인, 취소와 같은 커맨드 버튼을 선택함으로써 작업흐름을 변경할 수 있다는 것도 알 수 있다. 이와 같은 정보들의 집합인 시나리오 작성지원 보고서를 참고로 시나리오 항목을 작성해 나갈 수 있다.

```

...
Begin VB.CommandButton cmdcancel
Caption = "취소"
...
Caption = "확인"
...
Caption = "초기화"
...
Begin VB.ComboBox cmbhjl
DataField = "hangjung"
DataSource = "dsrSimpleWTInfo"
...
Private Sub cmbhjl_Click()
Select Case cmbhjl.ListIndex
Case 0
Data1.RecordSource = "select * from WTT_간이상수도내역
where hangjung = '1'"
Data1.Refresh
...
Private Sub cmdinitial_Click()
cmbhjl.Text = ""
...
Private Sub Form_Load()
...

```

(그림 9) 소스코드 분석을 통한 이벤트 정보

5.3 적용결과 분석

현업에서 프로젝트 수행 중 통합 테스트를 실시할 때는 통상적으로 검사팀이 요구분석 및 설계단계를 통해 만들어진 각종 명세서를 참고로 (그림 10)과 같은 시나리오를 작성하여 실시한다. 그리고 시나리오의 특성상 사용자와 시스템의 상호작용을 사용자가 이해하기 쉽게 단계별로 각 항목을 수작업에 의해 자연어로 기술하는 것이 보통이다. 그 결과 다음과 같은 문제점을 가진다.

상수도		2001-09-22
<input type="checkbox"/> 단위 <input type="checkbox"/> 통합 <input type="checkbox"/> 시스템 간이상수도급수지역관리기		
	과일	2001. 10. 5
1.	메뉴를 선택하면 간이상수도급수지역 조건검색창이 띄워진다.	아
2.	조건검색 항목을 입력한다. 검색박스에 해당 코드가 로드된다. 검색조건 - 행정동:종원동, 준공년월일:20010913, 수원종류:지하수	아
3.	초기화 버튼을 누르면 모든 항목의 내용을 Clear 시킨다.	아
4.	검색창의 취소버튼 클릭하면 데이터 셋을 Close 시킨 후 창을 닫는다.	아
5.	확인 버튼을 누르면 조건검색 마른 데이터들이 목록창에 띄워진다.	아

(그림 10) 현업에서의 시나리오 작성예

첫째, 방대한 자료를 참고로 하다보면 요구사항 정의테이블에 명시되어있는 모든 요구사항 중 누락되는 것이 발생할 수 있다.

둘째, 테스트 기능별로 시나리오를 작성할 때 여러 가지 명세서를 참고하므로 자료검색에 검사팀의 상당한 시간이 소요되며, 해당 기능에 대해 잘못된 검색조건으로 시나리오 항목을 작성할 수 있다. 즉, 특정 항목에 대해 문자형을 숫자로, 혹은 10자리를 5자리로 하는 것과 같은 실수를 할 수 있다.

셋째, 시나리오 작성에 검사팀의 상당한 시간이 소요되므로 전체 개발기간에도 영향을 미친다.

본 논문에서 제안한 요구사항 추적테이블에 기초한 시나리오 작성 지원절차에 따른 자동화된 도구를 이용하면 여러 가지 문제점을 아래와 같이 개선할 수 있으며, 출력물인 시나리오 작성지원 보고서가 시나리오 작성을 위한 지침서 역할을 하므로 결과적으로 최종산출물인 소프트웨어에 대한 품질을 향상시킬 수 있다.

The screenshot shows a software analysis tool interface. On the left, there are input fields for '총 요구사항 수:' (148), '테스트 시나리오 작성 수:' (121), '수용 불가 항목 수:' (27), and '테이블 불일치 수:' (21). On the right, there is a table with columns for '항목명', '요구번호', '단위', '분석', '출력', and '관련내역'. Below this, a detailed view of a requirement is shown with a table containing columns for '프로그램명', 'process', 'program', 'net_equal', 'str_table', and 'str'. The detailed table shows several rows of data with values like '1.1.1.1', '6', '7', 'RDL_연선 A', etc.

(그림 11) 작성 지원도구를 사용한 분석결과

(그림 11)은 총 148개의 요구사항을 가지는 시설물 종합 관리 시스템에 대한 적용 결과를 나타내고, <표 8>은 기존의 방식과 본 논문에서 제안하는 테스트 시나리오 작성 지원절차에 따른 결과를 비교·분석하여 항목별로 개선효과를 나타낸 것이다.

<표 8> 제안방식의 항목별 개선효과

항목	개선 효과	비 고	
		기 존	110개
요구사항 누락율	기존방식에 비해 9%의 요구사항 누락율 감소	기 존	110개
테이블 정확성	기존방식에 비해 관련 테이블 불일치 수 21개 감소	제 안	121개
검색조건 정확성	관련없는 테이블 참고를 방지함으로써 잘못된 검색조건 가진 테스트 시나리오 항목 차단	관련 테이블 항목값이 상이한 단위 프로세스 수	소스코드분석을 통한 이벤트 정보와 올바른 관련 테이블 정보 이용

즉, 본 논문에서 제안한 방식은 기존의 방식에 비해 148개의 총 요구사항에 대해 27개의 수용불가 항목을 제외하더라도 11개의 요구사항 누락을 감소시킴으로써 9%의 요구사항 누락율 감소를 보일 뿐 아니라, 관련 테이블 불일치 수도 21개나 감소하였다.

추가 사례로서 소규모 시스템이라 할 수 있는 자료실 도서관리시스템 개발과정에서 작성된 기초자료에 적용한 결과이다. 대규모 시스템에 비해 검색된 테이블간 불일치 수는 작지만 수용불가항목을 제외한 기능에 대해 테스트 시나리오 작성지원이 용이함을 알 수 있었다.

<표 9> 적용사례 2 결과

항 목	값
총 요구사항 수	40
테스트 시나리오 작성 수	35
수용 불가 항목 수	5
테이블 불일치 수	2

그러므로, 본 논문에서 제안한 통합 테스트 시나리오 작성 지원절차에 의해 생성된 작성 지원보고서를 참고하면 요구사항의 누락을 감소와 정확한 테이블 정보를 참고할 수 있으며, 그 결과 올바른 검색조건을 가진 테스트 시나리오 항목을 작성할 수 있다. 또한, 잘못된 값을 입력하여 발생하는 오류로 인해 소요되는 불필요한 시간낭비를 줄일 수 있으므로 시나리오 작성지원 보고서를 통해 시나리오를 작성하는 검사팀의 노력과 시간, 그리고 잘못을 상당히 줄일 수 있다.

6. 결 론

본 논문에서는 명세를 기반으로 한 통합 테스트를 수행할 때 사용될 수 있는 시나리오 작성 지원절차를 제시하고, 이를 위한 시나리오 작성지원 도구를 구현하였다.

통합 테스트를 위한 시나리오 작성지원 보고서를 출력하기 위해 요구사항 정의테이블, 프로세스 정의테이블, 프로그램 대 테이블 상관도와 이를 기초로 생성한 요구사항 추적테이블을 입력정보로 사용한다.

요구사항 추적테이블은 복잡한 각종 테이블 항목 중 명칭과 ID 중심으로 연결관계를 정리해 놓은 테이블로서 나머지 3가지 테이블의 중심에서 연결관계에 따른 각 항목의 전달기능을 담당한다.

본 연구의 통합 테스트를 위한 시나리오 작성지원도구의 출력물은 시나리오 작성을 위한 지침서 역할을 한다. 그 결과 검사팀의 시나리오 작성노력을 상당부분 줄일 수 있을 뿐만 아니라 요구사항에 대한 누락을 방지하고, 해당 프로세스에 대한 테이블정보를 알 수 있으므로 테스트할 때 잘못된 값을 입력하는 것도 방지할 수 있다.

향후 연구과제로는 시나리오 작성을 위한 지침서 역할을 하고 있는 결과물을 좀 더 보완하여 정확한 시나리오 항목을 출력할 수 있는 지원도구를 개발하는 것이다.

참 고 문 헌

[1] Holger D. Hofmann, Jeanne Stynes, "Implementation Reuse and Inheritance in Distributed Component Systems," IEEE, 1998.
 [2] Kozaczynski, W. and Booch, G., "Component-Based Software Engineering," IEEE Trans. on Soft. Eng, 1998.
 [3] Stephen S. Yau and Bing Xia, "Object-Oriented Distributed Component Software Development based on CORBA," IEEE, 1998.
 [4] Clemens Szyperski, "Component Software," Addison Wesley, 1998.
 [5] Paul C. Jorgensen, "Software Testing-A Craftsman's Approach," CRC Press, 1995.
 [6] Myers, G., "The art of Software Testing," Wiley & Sons, 1979.
 [7] A. Jefferson Offutt, Aynur Abdurazik and Roger T. Al-

exander, "An Analysis Tool for Coupling-based Integration Testing," Proceedings of the 6th IEEE International Conference on Engineering of Complex Computer Systems, pp. 172-178, 2000.
 [8] Delamaro ME, Maldonado JC, Mathur AP, "Interface mutation : An approach for integration testing," IEEE Trans. on Soft. Eng, Vol.27 No.3, pp.228-247, 2001.
 [9] 최은만 저, "소프트웨어 공학론(개정판)", 사이텍미디어간, 2001.
 [10] D. Kung, P. Hsia and J. Gao(Eds), "Testing Object-Oriented Software," IEEE Computer Society, 1998.
 [11] Y. Labiche, P. Thevenod-Fosse, H. Waeselynck, "Testing levels for object-oriented software," Soft. Eng. (ICSE 2000), 2000.
 [12] 한규정, 김치수, "객체지향 소프트웨어의 테스트 방법론", 정보통신부 시스템공학연구소 최종보고서, pp.15-23, 1996.
 [13] Paul C. Jorgensen et al, "Object-Oriented Integration Testing," CACM, Vol.37, No.9, pp.30-38, 1994.
 [14] 김은주, 최은만, "시나리오를 이용한 객체지향 시스템의 통합 테스트", 정보처리학회논문지, 제5권 제9호, 1998.
 [15] Youngchul Kim and C. Robert Carlson, "Scenario Based Integration Testing for Object-Oriented Software Development", Proceedings of the 8th Asian Test Symposium IEEE, pp.283-288, 1999.
 [16] 최은만, "컴퓨터를 이용한 시나리오 응용 방안", 정보처리학회지, 제3권 제1호, pp.335-338, 1996.



최 신 형

e-mail : shchoi@kndi.re.kr

1993년 울산대학교 전자계산학과 졸업 (학사)

1995년 경남대학교 대학원 전자계산학과 (공학석사)

2000년 경남대학교 대학원 컴퓨터공학과 (박사과정 수료)

1995년~1998년 해군사관학교 교수부 전산과학과 전산학교관
 2000년~현재 경남발전연구원 지역정보센터 연구원
 관심분야 : 소프트웨어 테스트 및 품질평가, 신뢰도 분석, 컴포넌트 기반 개발방법론 등



한 판 암

e-mail : pahan@kyungnam.ac.kr

1969년 동국대학교 졸업

1975년 동국대학교 경영대학원 졸업 (경영학석사)

1989년 명지대학교 대학원 졸업 (공학석사)

1992년 인천대학교 대학원 졸업(경영학박사)
 1980년~현재 경남대학교 공과대학 컴퓨터공학과 교수
 관심분야 : 소프트웨어 품질관리 및 신뢰성, 소프트웨어 개발환경, 정보공학 등