

명세서 분석을 통한 오류 검출방안

최신형*, 한군희**

*삼척대학교 컴퓨터제어계측공학과

**천안대학교 정보통신학부

e-mail:shchoi@samcheok.ac.kr

The Error Detection Method through Specification Analysis

Shin-Hyeong Choi*, Kun-Hee Han**

*Dept of Computer Control Engineering, Samcheok National University

**Division of Information and Communication, Cheonan University

요 약

소프트웨어를 개발할 때 작성되는 명세서들은 개별특성으로 인해 연속적인 자동화가 이루어지지 않으며, 소프트웨어 감리에서도 많은 불일치 항목이 발견된다. 본 논문에서는 분석 및 설계단계에서 작성되는 명세서간 불일치를 검출하기 위해 시스템 개발과정에 대해 실시한 감리결과보고서를 분석하여 불일치가 발생하는 연관된 명세서를 내용별로 분류하고, 이를 바탕으로 분석, 설계, 테스트 단계별로 사용되는 명세서간 불일치 추출방안을 제안한다. 이를 이용하면, 분석 및 설계단계에서 작성되는 명세서의 오류를 줄임으로써 최종산출물인 소프트웨어에 대한 품질을 향상시킬 수 있다.

1. 서론

다양한 사용자들의 요구사항으로 인해 소프트웨어는 복잡하고 다양해졌다. 이로 인해 정확성이 문제시 되었고, 이런 문제를 해결하기 위해 개발방법론을 사용하며, 이를 기반으로 단계별로 표준화된 명세서를 작성한다. 하지만, 작성되는 명세서들은 개별적인 특성으로 인해 모든 명세서 작성이 연속적으로 자동화될 수는 없다. 현재 소프트웨어 개발현장에서는 분석, 설계, 개발, 구현단계별로 자동화가 아닌 수작업에 의해 각종 명세서를 작성하고 있다. 이로 인해 작성되는 명세서간에는 많은 불일치 항목이 발생한다. 그러므로 본 논문에서는 소프트웨어 개발 단계별로 작성되는 명세서들 중 분석 및 설계단계에서 작성되는 명세서간 불일치 검출을 위해 명세서를 분류하고, 해당 명세서간에 불일치를 검출할 수 있는 방안을 제시한다.

2. 관련연구

2.1 개발방법론

소프트웨어를 개발하기 위한 방법론은 시스템의 호

과적 개발을 위한 단계적 절차를 말하며, 사용자가 원하는 바를 정확하게 파악하고, 새로운 시스템을 사용자에게 쉽게 이해시킬 수 있기 위해 단계별로 표준화된 명세서가 작성된다.

2.2 단계별 명세서

개발방법론에 따라 약간의 차이는 있지만, 일반적으로 분석, 설계, 개발, 구현 단계별로 각종 명세서가 작성된다. 개발단계별로 작성되는 표준 명세서를 정리하면 표 1과 같다.

표 1. 개발 단계별 표준 명세서

단계	내용
분석	현업요구사항 정의서, 유즈케이스 다이어그램, 클래스 목록, 클래스 다이어그램, 시스템 청사진
설계	클래스 다이어그램, 시퀀스 다이어그램, 프로그램 목록, 화면 레이아웃, 보고서 레이아웃, 프로그램 사양서, 프로그램 테이블 상관도, 테스트 계획서, 통합 테스트 시나리오, 시스템 테스트 시나리오, 테이블 목록, 테이블 정의서

소프트웨어를 개발할 때는 표 1과 같은 표준 명세서에서 개발할 시스템에 따라서 명세서가 생략 및 추가되기도 한다[2, 3].

3. 명세서간 오류검출방안

실제 시스템 개발현장에서의 진행과정은 개발도구나 개발자의 개발방식에 따라 약간의 차이가 나타난다. 또한, 분석 및 설계단계별로 작성되는 각종 명세서는 개별특성으로 인해 수작업에 의존하여 연속적으로 작업되지 못한다. 그 결과 분석 및 설계단계를 통해 작성되는 명세서 항목 간에는 오류가 발생할 수 있다. 이런 오류는 명세서 항목간의 불일치에 기인하며, 명세서 항목 간에 지켜져야 할 일관성 규칙을 충족하지 않을 때 발생한다[4].

본 논문에서는 분석 및 설계단계에서 작성되는 명세서에 포함될 수 있는 불일치를 발견하기 위한 검출방안을 제안한다.

3.1 명세서 분석

먼저, 실제 시스템 개발을 통해 작성된 명세서에 대해 실시한 감리결과보고서를 분석하였다[4]. 이를 통해 시스템 개발과정에서 문제점이 많이 나타난 비율은 구현, 설계, 분석단계 순으로 나타났으며, 분석 및 설계단계에서는 기능구현과 사용자 인터페이스, 명세서간의 연계성이 주요 문제점으로 지적되었다. 추가로 시스템 개발현장의 감리활동을 통해서 작성된 감리보고서에 의하면 분석 및 설계단계에서 가장 빈도수가 높게 나타난 오류 유형은 표준 명명규칙에 어긋나는 오류, 명세서간 일관성이 잘 지켜지지 않는 오류이다.

이상에서 고찰한 기본적인 자료를 기초로 해서 본 논문에서는 그림 1과 같이 분석 및 설계단계에서 작성되는 명세서를 오류발생빈도와 명세서 간 연관성을 고려하여 분류하였다.

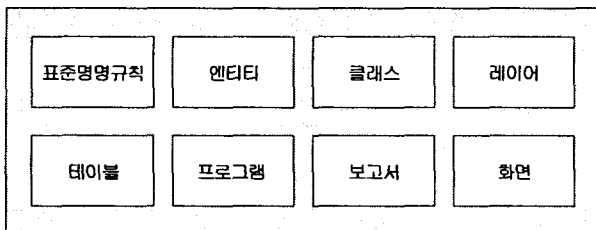


그림 1. 항목별 연관성을 고려한 명세서 분류

분석 및 설계단계에서 작성되는 표준 명세서를 대

상으로 그림 1과 같이 분류된 내용별로 해당 명세서를 추출하면 표 2와 같이 선택된다.

표 2. 명세서 분류에 따른 명세서 목록

내용	해당 명세서
표준명명규칙	클래스목록, 화면목록, 보고서목록, 프로그램목록
엔티티	ERD, 엔티티정의서
클래스	클래스목록, 클래스다이어그램
레이어	레이어목록, 레이어정의서
테이블	테이블목록, 테이블정의서
프로그램	프로그램목록, 프로그램사양서
보고서	보고서목록, 보고서레이아웃
화면	화면목록, 화면레이아웃

그림 2는 감리결과보고서 분석을 통해 표준 명세서로부터 항목 간 불일치가 발생하는 명세서를 선택하는 과정을 나타낸다.

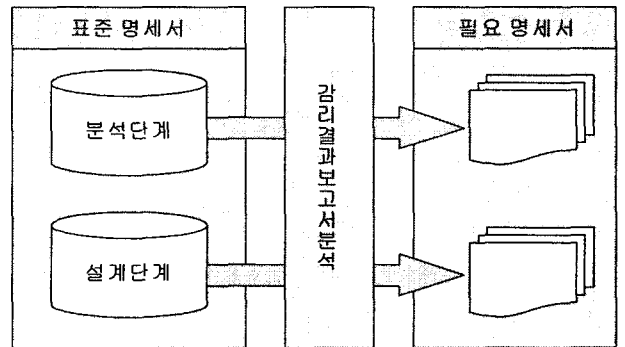


그림 2. 명세서 분석과정

3.2 단계별 불일치 검출방안

본 논문에서는 명세서 항목 간 불일치를 검출하기 위해 두 단계로 나누어서 검출과정을 수행한다. 즉, 분석 및 설계단계와 테스트 단계에서 작성 및 사용되는 명세서에 대해 실시한다.

3.2.1 분석 및 설계단계 명세서

분석 및 설계단계에서 작성되는 명세서에 대한 불일치 검출은 표 2에 나타난 분석단계 명세서 4개와 설계단계 명세서 14개를 대상으로 실시한다. 이를 통해 해당 명세서의 항목 간 불일치를 검출하면 보다 빠른 시간에 불일치 항목을 검출하여 분석 및 설계단계에서 작성되는 명세서의 오류를 줄일 수 있다.

3.2.2 테스트 단계 명세서

일반적으로 테스트 단계에서는 분석 및 설계단계를 통해 작성된 명세서를 기초로 테스트 작업을 수행한다. 단위 테스트는 기본적으로 소스코드에 대해 실시하지만, 통합 테스트는 통상적으로 검사팀이 분석 및 설계단계를 통해 만들어진 각종 명세서를 참고로 테스트 시나리오를 작성하여 실시한다. 이 때 필요한 항목으로는 해당업무에 대응되는 프로그램 이름, 프로그램 ID, 테이블 정보, 이벤트 정보 등이며, 필요한 해당 명세서는 요구사항정의서, 프로그램과 테이블상관도, 프로세스정의서이다. 이들 명세서에 대해 해당 항목 간 비교를 통해 불일치를 검출할 수 있다.

4. 적용사례

본 논문에서 제안한 명세서 항목간 불일치 검출방안을 실제 시스템 개발현장에서 단계별로 작성된 명세서에 적용한 예를 나타낸다. 이들 명세서는 도시정보시스템 개발과정 중 분석단계와 설계단계별로 작성된 것으로 표 1과 같이 구성된다.

도시정보시스템 개발과정 중 분석단계와 설계단계에서 작성된 각종 명세서를 대상으로 단계별 불일치 검출방안을 적용한 결과 표 3과 같이 분석단계 명세서 5개, 설계단계 명세서 41개, 테스트단계에 사용된 명세서 39개의 불일치가 검출되었다.

표 3. 단계별 불일치

단계	검출된 불일치
분석	5개
설계	41개
테스트	39개

검출된 불일치를 수정함으로써 보다 정확한 개발 명세서 작성뿐 아니라 최종 시스템의 신뢰성을 높일 수 있다.

5. 결론

현재 소프트웨어 개발현장에서는 분석, 설계, 개발, 구현단계별로 자동화가 아닌 수작업에 의해 각종 명세서들을 작성하고 있다. 이로 인해 작성되는 명세서간에는 많은 불일치 항목이 발생한다.

본 논문에서는 분석 및 설계단계에서 작성되는 명세서간 불일치를 검출하기 위해 먼저 시스템 개발과

정에 대해 실시한 감리결과보고서를 분석하여 불일치가 발생하는 연관된 명세서를 내용별로 분류하였다. 이를 바탕으로 분석, 설계, 테스트 단계별로 사용되는 명세서간 불일치 추출방안을 제안하였다.

본 논문에서 제안한 불일치 추출방안을 이용하면, 분석 및 설계단계에서 작성되는 명세서에 오류를 줄임으로써 최종산출물인 소프트웨어에 대한 품질을 향상시킬 수 있다.

참고문헌

- [1] Martin Fowler, Kendall Scott, 'UML Distilled Second Edition : A Brief Guide to the Standard Object Modeling Language', Addison-Wesley, 2000.
- [2] Kuzniarz L., Reggio G., Sourrouille J. L., Huzar Z., Workshop on Consistency Problems in UML based software development, Workshop Materials, Research Report 2002:06, Blekinge Institute of Technology, Ronneby 2002.
- [3] 최은만 저, '소프트웨어 공학론(개정판)', 사이텍 미디어간, 2001.
- [4] 한국전산원, "감리 결과 분석을 통한 주요문제점 및 개선사례 연구", 2001.
- [5] Y.C. Kim and C.R. Carlson, "Scenario Based Integration Testing for Object-Oriented Software Development", Proceedings of the 8th Asian Test Symposium IEEE, pp. 283-288, 1999.
- [6] 정기원, 조용선, 권성구, "객체지향 설계방법에서 오류 검출과 일관성 점검기법 연구", 정보처리학회논문지D, Vol.6, No.8, 1998.
- [8] 김도형, 정기원, "객체지향 분석과정에서 오류와 일관성 점검 방법", 정보과학회논문지B, Vol.26, No3, pp.1453-1467, 1996.