

명세서 분석을 통한 오류 검출 시스템 개발

최신형^{1*} · 한군희²

Development of Error Detection System through the Specifications Analysis

Shin-Hyeong Choi^{1*} and Kun-Hee Han²

요약 실제 시스템 개발현장에서는 사용자 요구사항과 관련 자료의 분석을 시작으로 크게 네 가지 단계 즉, 분석, 설계, 개발, 구현단계별로 각종 명세서들을 작성하고 있다. 명세서 일부는 자동화된 도구를 사용하여 연속적으로 작성되기도 하지만 모든 명세서가 자동화로 작성되는 것은 아니다. 이로 인해 작성되는 명세서간에는 많은 불일치 항목이 발생한다. 본 논문에서는 시스템을 개발할 때 각 단계별로 작성되는 명세서에 포함된 오류를 검출하는 방안을 제안하고, 이를 기초로 명세서 오류 검출 시스템을 설계 및 구현한다.

Abstract In the actual system development site, the various specifications are drawn up into each phase, namely analysis, design, development and implementation, beginning with the analysis of user requirements and data involved. The part of the specifications are continuously drawn up by using automated tool, but all specification are not automatically drawn up. Therefore, many inconsistent items occur between the specifications. In this paper, we propose the method which detects the error which is included in the specifications, and we are going to design and implement a specification error detection system based on this.

Key Words : audit, specification, automation

1. 서 론

컴퓨터의 빠른 처리능력과 기술력 뒷받침의 지원을 경험한 사용자들은 다양하고도 방대한 양의 자료를 신속하고 정확하게 처리하는 시스템을 요구한다. 이와 같은 사용자 요구는 복잡하고 다양하며, 이로 인해 개발되는 시스템의 정확성이 문제시 되었다.

이런 문제를 해결하기 위해 개발자들은 표준화된 개발방법론을 사용하며, 이를 기반으로 단계별로 표준화된 명세서를 작성하면서 개발을 진행한다. 하지만, 단계별로 작성되는 명세서들은 개별적인 특성뿐만 아니라 작성되는 형식의 상이함으로 인해 모든 명세서 작성이 연속적으로 자동화될 수는 없다.

실제 시스템 개발현장에서는 사용자 요구사항과 관련 자료의 분석을 시작으로 크게 네 가지 단계 즉, 분석, 설계, 개발, 구현단계별로 각종 명세서들을 작성하고 있

다. 명세서 일부는 자동화된 도구를 사용하여 연속적으로 작성되기도 하지만 모든 명세서가 자동화로 작성되는 것은 아니다. 이로 인해 작성되는 명세서간에는 많은 불일치 항목이 발생한다.

그러므로 본 논문에서는 시스템을 개발할 때 각 단계별로 작성되는 명세서에 포함된 오류를 검출하는 방안을 제시한다.

본 논문은 다음과 같이 구성된다. 2장에서는 표준화된 개발방법론에 따라 작성되는 명세서와 감리분야에 대해 관련연구로서 나타내고, 3장에서는 명세서간 오류 검출방안을 제시한다. 4장에서는 오류 검출 시스템을 설계 및 구현하고 5장에서는 사례적용결과를 보이며, 6장에서는 결론을 맺는다.

2. 관련연구

2.1 개발 명세서

개발방법론에 따라 약간의 차이는 있지만, 일반적으로 분석, 설계, 개발, 구현 단계별로 각종 명세서가 작성된다. 개발단계별로 작성되는 표준 명세서를 정리하

¹삼척대학교 컴퓨터제어계측공학과

²천안대학교 정보통신학부

*교신저자: 최신형 (shchoi@samcheok.ac.kr)

면 Table 1과 같다.

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

2.2 감리방법론

2.2.1 정보시스템 감리

정보시스템 감리란 정보시스템의 계획, 개발, 운영, 유지보수에 관한 제반 절차 및 산출물(문서, 시스템)을 독립적인 입장에서 종합적으로 점검, 평가하여 감리의뢰인, 피감리인, 이해 관계자들에게 개선이 필요한 사항을 조정, 권고함에 의해 효과성과 효율성을 증진하기 위한 일련의 활동을 지칭한다[9].

2.2.2 감리방법론 체계

감리방법론은 작업 특성을 지니고 있는 19개의 작업 항목과 작업항목을 그룹화한 4개의 단계로 이루어져 있다. 또한, 단계 및 작업 항목 간에는 업무수행의 선후관계를 명시하고 있다. 그리고 본 방법론의 근간을 이루는 작업항목은 목적과 작업수행절차, 산출물, 그리고 작업 시 고려사항 등 4가지 항목으로 구성되어 있다[9].

1) 목적 : 작업항목이 감리업무를 효율적으로 추진하기 위하여 수행하여야 하는 이유 및 달성해야 하는 사항을 명시한다.

2) 작업수행절차 : 작업항목별로 명시된 목적을 달성하기 위하여 수행되는 활동과 활동 간의 선후관계나 관련성을 명시한다.

3) 산출물 : 작업항목을 수행하는데 필요한 명세서와 작업항목 수행결과로 도출되는 산출물을 모두 명시한다.

4) 작업 시 고려사항 : 활동을 수행하는 과정에서 주의하거나, 감리 사안에 따라 고려되어야 할 특이한 사항들을 명시한다.

Table 1. Standard Specification.

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

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 항목간에 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 Fig 1과 같이 분석 및 설계단계에서 작성되는 명세서를 오류발생빈도와 명세서 간 연관성을 고려하여 분류하였다.

분석 및 설계단계에서 작성되는 표준 명세서를 대상으로 Fig 1과 같이 분류된 내용별로 해당 명세서를 추출하면 Table 2와 같이 선택된다.

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

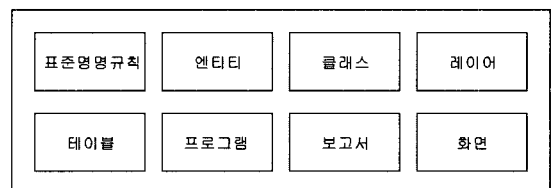


Fig. 1. The specifications classification which considers relation between items.

Table 2. The specifications list which it follows in specifications classification.

내용	해당 명세서
표준명명규칙	클래스목록, 화면목록, 보고서목록, 프로그램목록
엔티티	ERD, 엔티티정의서
클래스	클래스목록, 클래스 다이어그램
레이어	레이어목록, 레이어정의서
테이블	테이블목록, 테이블정의서
프로그램	프로그램목록, 프로그램 사양서
보고서	보고서목록, 보고서 레이아웃
화면	화면목록, 화면 레이아웃

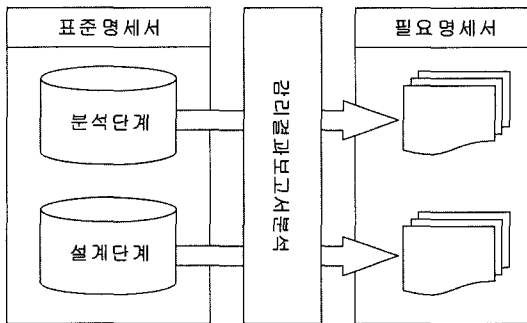


Fig. 2. Specifications analysis process.

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 Table 2에 나타난 분석단계 명세서 4개와 설계단계 명세서 14개를 대상으로 실시한다.

Fig 3의 분석 및 설계단계의 명세서 오류 검출 알고리즘을 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 실시하

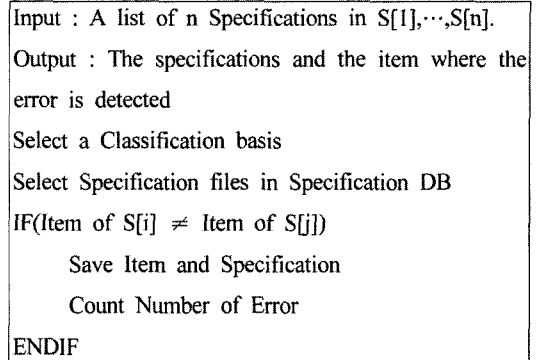


Fig. 3. Specifications error detection algorithm of analysis and design phase.

지만, 통합 테스트는 통상적으로 검사팀이 분석 및 설계단계를 통해 만들어진 각종 명세서를 참고로 테스트 시나리오를 작성하여 실시한다. 이때 필요한 항목으로는 해당업무에 대응되는 프로그램 이름, 프로그램 ID, 테이블 정보, 이벤트 정보 등이며, 필요한 해당 명세서는 요구사항정의서, 프로그램과 테이블상관도, 프로세스 정의서이다. 이들 명세서에 대해 Fig 4의 테스트 단계의 명세서 오류 검출 알고리즘을 이용하면 해당 항목 간 비교를 통해 명세서에 포함된 불일치를 검출할 수 있다.

4. 오류 검출 시스템 구성

4.1 시스템 설계

3장에서 제안한 명세서간 오류 검출 방안을 기초로 한 명세서 오류 검출 시스템은 Fig 5와 같이 크게 3부분으로 나뉜다.

- 1) 명세서 선택 모듈
- 2) 오류 검출 모듈

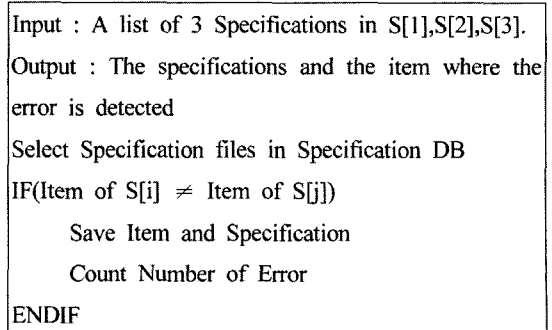


Fig. 4. Specifications error detection algorithm of test phase.

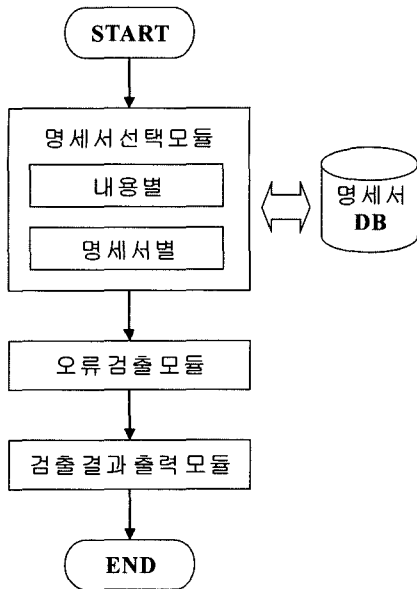


Fig. 5. The whole system block diagram.

3) 검출 결과 출력 모듈

명세서 선택 모듈에서는 감리관련자료 분석을 통해 설정한 명세서 분류기준의 내용별로 해당 명세서를 명세서 DB로부터 선택하며, 오류 검출 모듈에서는 3장에서 제안한 분석 및 설계단계와 테스트 단계별 오류 검출 알고리즘을 이용하여 각 단계별로 명세서간 오류를 검출한다. 검출 결과 출력 모듈은 단계별로 검출된 오류수를 요약하여 나타낸다.

4.2 시스템 구현

명세서 오류 검출 시스템은 Visual Basic언어로 작성되었으며, 각 모듈은 폼을 기반으로 구현된다.

Fig 6은 명세서 분류 기준 중 하나를 선택하는 것으로, 이를 통해 Fig 7과 같이 해당 명세서를 선택할 수 있다.

Fig 8, 9는 단계별로 검출된 오류를 나타내며, 이를 통해 어느 명세서에서 어떤 종류의 오류가 얼마만큼 발생하였는지 알 수 있다.

Fig 8은 3.2.1의 분석 및 설계단계 명세서 오류 검출 알고리즘을 적용하여 분석 및 설계단계 명세서에 포함된 오류를 검출한 결과이다. 오류가 검출된 명세서, 항목값과 오류내용을 나타낸다.

Fig 9는 3.2.2의 테스트단계 명세서 오류 검출 알고리즘을 적용하여 테스트 단계에서 사용되는 명세서인 프로세스 정의 테이블과 프로그램과 테이블 상관도에 포함된 오류를 검출한 결과이다. 오류가 검출된 명세서와 불일치 항목수를 나타낸다.

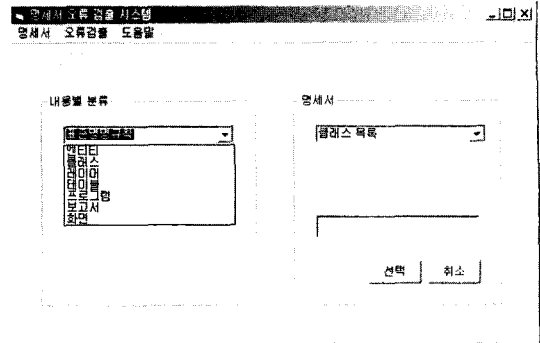


Fig 6. Specifications classification standard selection screen.

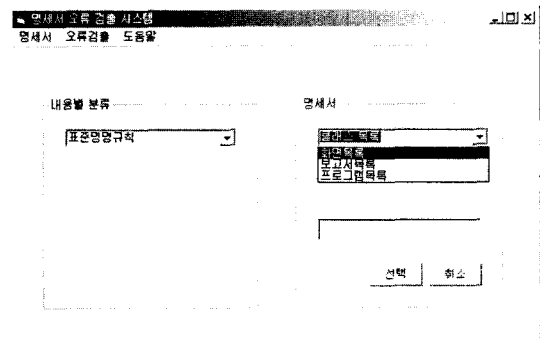


Fig 7. Corresponding specifications selection screen.

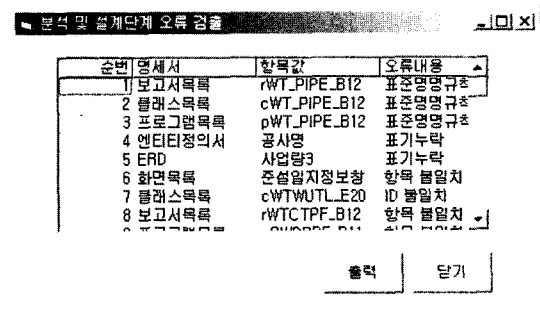


Fig. 8. Error detection result screen 1.

5. 적용사례

5장에서 본 논문에서 제안한 명세서 항목 간 불일치 검출방안을 실제 시스템 개발현장에서 단계별로 작성된 명세서에 적용한 예를 나타낸다.

이들 명세서는 도시정보시스템 개발과정 중 분석단계와 설계단계별로 작성된 것으로 공통 기능, 상수도 기능, 하수도 기능, 도로 기능의 4가지로 구성된다.

도시정보시스템 개발과정 중 분석단계와 설계단계에

순번	명세서	불일치 항목수
1	프로세스 정의 테이블	1
2	프로세스 정의 테이블	2
3	프로세스 정의 테이블	2
4	프로세스 정의 테이블	2
5	프로세스 정의 테이블; 프로그램과 테	2
6	프로세스 정의 테이블; 프로그램과 테	2
7	프로세스 정의 테이블; 프로그램과 테	2
8	프로세스 정의 테이블; 프로그램과 테	2

Fig. 9. Error detection result screen 2.

서 작성된 각종 명세서를 대상으로 단계별 오류 검출방안을 적용한 결과 Table 3과 같이 분석단계 명세서 5개, 설계단계 명세서 41개, 테스트단계에 사용된 명세서에서 39개의 불일치가 검출되었다.

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

5. 결 론

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 개발과정에 대해 실시한 감리결과보고서 등 감리관련 자료를 분석하여 불일치가 발생하며, 항목 간에 연관된 명세서를 내용별로 분류하였다. 이를 바탕으로 분석, 설계, 테스트 단계별로 사용되는 명세서간 불일치 추출방안을 제안하였다. 이를 기초로 명세서 오류 검출 시스템을 구현하였으며, 실제 시스템 개발 현장에서 작성된 각종 명세서에 적용하였다.

본 논문에서 제안한 불일치 추출방안을 이용하면, 분석 및 설계단계에서 작성되는 명세서에 포함된 오류를

Table 3. Number of inconsistent items.

단계	검출된 불일치
분석	5개
설계	41개
테스트	39개

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)", 사이텍미디어, 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.
- [7] 김도형, 정기원, "객체지향 분석과정에서 오류와 일관성 점검 방법", 정보과학회논문지B, Vol. 26, No. 3, pp.1453-1467, 1996.
- [8] 한국전산원, "정보시스템 감리방법론 연구", 2000.
- [9] Grady Larsen, "Designing Component-Based Frameworks using Patterns in the UML", Communications of th ACM, 1999.