
소프트웨어 개발 명세서의 불일치 요소 검출방안

A Method for Detecting Inconsistent Element of Software Development Specification

최신형

삼척대학교 컴퓨터제어계측공학과

Shin-Hyeong Choi(shchoi@samcheok.ac.kr)

요약

소프트웨어를 개발할 때 작성되는 명세서들은 개별특성으로 인해 연속적인 자동화가 이루어지지 않으며, 소프트웨어 감리에서도 많은 불일치 항목이 발견된다. 본 논문에서는 분석 및 설계단계에서 작성되는 명세서간 일관성 유지를 위하여 감리관련자료를 분석하여 불일치 항목 분류기준을 설정한다. 이를 기초로 명세서 불일치 항목 추출 방법을 제안한다. 이를 이용하면, 분석 및 설계단계에서 작성되는 명세서의 오류를 줄임으로써 최종산출물인 소프트웨어에 대한 품질을 향상시킬 수 있다.

■ 중심어 : | 명세서 | 불일치 | 감리 |

Abstract

The specifications which are drawn up when developing a software cannot achieve continuous automation because of individual qualities of them, and many inconsistent elements can be discovered even from software audit. In this paper, we establish the standards for the classification of inconsistent items by analysing audit related data in order to maintain the consistency among specifications drawn up in the phases of analysis and design. We propose the specification inconsistent item detection method. This can reduce the error of the specification which is drawn up from analysis and design phase with it will be able to improve the quality of the software which is a last product. It can improve the quality of the software which is a last product, by reducing the error of the specification which is drawn up from analysis and design phase.

■ Keyword : | Specification | Inconsistency | Audit |

I. 서론

사용자가 요구하는 소프트웨어가 점차 복잡해지고, 그 크기가 방대해짐에 따라, 개발된 소프트웨어의 정확

성 역시 문제가 되었다. 이런 문제를 해결하기 위해 등장한 개발방법론은 각 소프트웨어 개발단계별로 표준화된 명세서를 작성하도록 규정하였고, 이를 통해 단계별로 산출되는 명세서간의 일관성을 유지할 수 있다. 하지

* 본 연구는 2004년도 삼척대학교 학술연구비 지원에 의해 수행되었음.

접수번호 : #041110-002

접수일자 : 2004년 11월 10일

심사완료일 : 2004년 12월 10일

교신저자 : 최신형, e-mail : shchoi@samcheok.ac.kr

만, 작성되는 명세서들은 개별적인 특성으로 인해 모두 다 연속적으로 자동화될 수는 없다. 현재 소프트웨어 개발 현장에서는 단계별로 자동화가 아닌 수작업에 의해 각종 명세서들을 작성하고 있다. 이로 인해 작성되는 명세서간에는 많은 불일치 항목이 발생한다.

그러므로 본 논문에서는 감리결과보고서 분석을 통해 소프트웨어 개발단계별로 작성되는 명세서들 중 분석 및 설계단계에서 작성되는 명세서의 불일치 항목을 검출하기 위한 분류기준을 제시하고, 테이블 분석방법을 통해 해당 명세서간에 불일치 항목을 검출할 수 있는 방안을 제시한다.

2장에서는 관련연구로서 개발방법론과 단계별 명세서에 대해 살펴보고, 3장에서는 명세서간 오류검출방안을, 4장에서는 실제 개발현장에서 산출한 명세서에 적용한 예와 결과분석을 보여주며, 5장에서는 결론을 맺는다.

II. 관련연구

1. 개발방법론

소프트웨어 개발방법론은 시스템의 효과적 개발을 위한 단계적 절차를 말하며, 단계별 구성원의 역할, 단계별 명세서, 단계별 사용기법을 명시하는 것이다. 또한, 사용자가 원하는 바를 정확하게 파악하고, 새로운 시스템을 사용자에게 쉽게 이해시킬 수 있기 위해 필요하다. 개발방법론은 초창기의 구조적 방법론부터 객체지향개발방법론을 거쳐 차세대 소프트웨어 개발방법론으로 각광받고 있는 컴포넌트개발방법론까지 발전해왔다[4].

2. 단계별 명세서

개발방법론에 따라 약간의 차이는 있지만, 일반적으로 분석, 설계, 개발, 구현 단계별로 각종 명세서가 작성된다. 요구분석을 비롯하여 시스템 설계 및 구현 등 일련의 과정에서는 현재 대부분 UML(Unified Modeling Language)을 사용한다. UML은 업계 표준이라고 할 수 있으며, 객체지향 모델링 언어인 UML에 따라 작성되는 표준 명세서를 정리하면 다음 [표 1]과 같다.

소프트웨어를 개발할 때 작성되는 개발 명세서는 [표

1]과 같이 네 가지 단계로 구분하여 작성된다. 하지만, 표준과 실제 작성되는 명세서간에는 개발할 시스템에 따라서 명세서가 생략 및 추가되기도 한다[3, 4].

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
개발	소스 코드, 통합 테스트 시나리오(결과), 시스템 테스트 시나리오(결과)
구현	구현 계획서, 데이터베이스, 애플리케이션 시스템, 운영 시스템, 사용자 매뉴얼, 운영 매뉴얼

3. 오류검출과 일관성 점검방법

오류 검출 및 일관성 점검은 시스템 개발에 있어서 분석 및 설계 단계에서부터 개발될 시스템의 완전성과 일관성을 확보하기 위하여 반드시 필요한 작업이며, 오류 검출 및 일관성 점검 방법에 관한 많은 연구들이 수행되어 왔다[8, 9, 10, 12, 13].

3.1 역공학을 이용

원시코드로부터 설계모델을 추출하여 설계단계에서 작성한 설계모델과 비교하여 일관성을 검증한다.

3.2 제약언어를 이용

객체모델을 생성할 수 있는 객체 모델 작성기와 작성된 객체 모델을 제약언어를 기반으로 검증할 수 있는 객체 모델 검증기를 포함한 객체 모델 검증 시스템을 이용한다.

3.3 지식베이스를 이용

객체모델, 기능모델, 동적모델을 정형명세화하여 정보를 추출하고, 지식베이스에 담겨있는 추론 규칙을 이용하여 분석하는 오류검사방법이다.

3.4 시나리오 검증기법을 이용

시나리오 정보트리추적과 시나리오 정보테이블참조

과정을 중심으로 완전성과 일관성을 검증한다.

3.5 정형방법론의 일관성 검사방법을 이용

정형방법론의 일관성 검사방법을 유즈케이스 모델링에 적용함으로써 논리적으로 일관되고 구조화된 요구사항모델을 생성할 수 있다.

III. 명세서간 오류검출방안

현업에서 시스템을 개발하는 과정은 업무분석과 사용자 인터뷰와 요구사항 정의서를 기초로 단계별로 진행된다. 이런 과정은 대부분의 개발업체에서 동일하게 실시되지만, 이후의 작업은 개발도구나 개발자의 개발방식에 따라 약간의 차이가 나타난다. 또한, 분석 및 설계 단계별로 작성되는 각종 명세서는 개별 특성으로 인해 자동화 도구에 의해 작성되기보다는 개별적으로 작성된다. 이 결과 분석 및 설계단계를 거쳐 개발단계로 진행할 때 관련되는 명세서에 포함된 항목 간에 오류가 발생할 수 있다. 이런 오류는 명세서 항목간의 불일치에 기인하며, 명세서 항목 간에 지켜져야 할 일관성 규칙을 충족하지 않을 때 발생한다. 이것은 대규모 시스템개발에 대한 감리를 수행할 때는 항상 감리기관의 지적사항이다[6, 11].

1. 불일치의 개념

광의적인 의미에서 불일치는 2개의 기술정보 사이에 유효한 것으로 가정된 특정한 관계를 이 기술정보가 충족하지 못하는 상황을 의미한다[1]. 본 논문의 목적은 소프트웨어 개발단계별로 작성되는 각종 명세서 항목에 대한 불일치 요소를 검출하는데 있다. 이를 위해 본 논문에서는 불일치를 다음과 같이 제한적 범위에서 정의한다. 불일치란 소프트웨어 개발단계별로 작성되는 명세서 중 관련 항목간의 오류를 의미한다. 본 논문에서 정의한 불일치 정의에 의거하여 불일치 요소를 검출하는데 사용되는 명세서를 결정한다. 이를 위해 소프트웨어 개발과정 중에 작성되는 명세서는 분석과 설계단계에서 작성되는 것으로 한정하며, 분석 및 설계단계의 명

세서를 활용하여 테스트 단계에서 추가로 작성되는 명세서도 포함한다.

2. 테이블 분석

앞에서도 기술한 바와 같이 개발단계별로 정확한 개발 명세서를 작성하는 것은 쉽지 않은 일임에도 불구하고 소프트웨어 개발에 있어 매우 중요한 일이다. 정확한 개발 명세서의 작성을 위해서는 작성되는 명세서간에 존재하는 불일치 항목을 점검하는 작업이 필요하다.

2.1 기본 명세서 분석

본 논문에서는 이를 위해 우선적으로 시스템 개발현장의 명세서에 대하여 실시한 감리자료와 한국전산원의 “감리결과 분석을 통한 주요 문제점 및 개선사례 연구” 보고서를 분석하였다. 이 보고서는 실제 시스템 개발현장에서 실시한 감리결과를 바탕으로 1997년의 20건과 1998년 2건, 1999년 1건, 2000년 3건 등 총 26건의 개발 감리보고서에서 지적한 주요 문제점과 개선 권고사항을 대상으로 분석한 결과를 바탕으로 작성되었다[6].

감리보고서를 분석한 결과를 요약하면 [표 2]와 같다.

응용시스템 개발과정에서 단계별로 문제점이 가장 많이 나타난 것은 구현단계로 89건의 오류가 발생했으며, 다음으로 설계단계, 분석단계 순으로 나타났다. 그 중에서 분석 및 설계단계에서는 기능구현과 사용자 인터페이스, 명세서간의 연계성이 주요 문제점으로 지적되었다. 또한, 개발현장의 감리활동을 통해서 작성된 감리결과보고서에 의하면 분석 및 설계단계에서 가장 빈도수가 높게 나타난 오류 유형은 첫째로 표준 명명규칙에 어긋나는 오류이고, 둘째로 명세서간 일관성이 잘 지켜지지 않는 오류이다.

표 2. 감리보고서 분석결과

분석대상	내용
발생 오류수	구현(89)설계(5)분석(1)
주요 문제점	항목간 불일치 발생
오류 유형 (빈도수)	1. 표준명명규칙 어김 2. 일관성 어김

이상에서 고찰한 기본적인 자료를 기초로 해서 본 연

구에서는 [그림 1]과 같이 개발 명세서 사이의 불일치 항목을 분류했다. 즉, 한국전산원의 “감리결과 분석을 통한 주요 문제점 및 개선사례 연구”를 기초로 해서, 분석 및 설계단계에서 작성되는 명세서에 관련되는 불일치 항목 분류 기준을 설정한다. 이것은 표준 명명규칙, 유즈케이스, 엔티티 정의, 화면, 클래스, 보고서, 프로그램, 레이아웃 및 테이블 등이며, 오류발생빈도와 명세서 간 연관성을 고려하여 분류하였다.

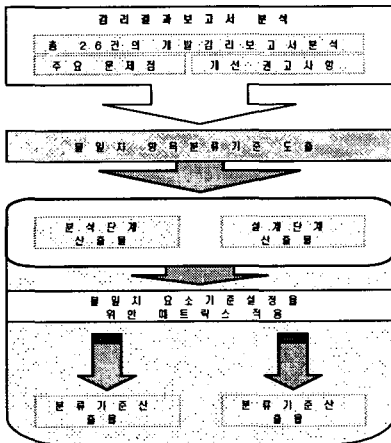


그림 1. 불일치 항목 분류기준 도출과정

다음으로 이 기준을 바탕으로 [그림 2]의 불일치 항목 기준설정을 위한 매트릭스를 사용하여 불일치 항목이 포함된 명세서를 추출한다.

분석단계에서 작성되는 명세서는 일반적으로 9개인데, 이 중에서 “불일치 항목 분류기준”에 해당되는 명세서는 4개로서 유즈케이스 다이어그램, 클래스목록, 클래스 다이어그램, 유즈케이스 시나리오 등이다.

설계단계에서 작성되는 명세서는 일반적으로 21개인데, 이 중에서 “불일치 항목 분류기준”에 해당되는 명세서는 14개로서 클래스 다이어그램, 시퀀스 다이어그램, 프로그램 목록, 프로그램 사양서, ERD, 엔티티 정의서, 화면 목록, 화면 레이아웃, 보고서 목록, 보고서 레이아웃, 레이아웃 목록, 레이아웃 정의서, 테이블 목록, 테이블 정의서 등이다. 분석단계 명세서 4개와 설계단계 명세서 14개를 제외한 나머지 명세서는 관련되는 항목이 없으므로 불일치 항목이 발생될 수 없다.

2.2 추가 명세서 분석

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. 하지만 테스트 시나리오를 작성하기 위해 참고하는 자료들은 개별적인 특징과 필요성을 가지고 있기 때문에 서로 긴밀한 연결 관계를 규정짓기 힘들뿐만 아니라 자동화가 아닌 검사팀이라는 인력에 의존하다보니 상당한 노력과 시간이 소요되는 것이 현실이다. 또한, 수작업에 의존하여 작성함으로써 관계없거나 잘못된 자료 입력과 같은 오류가 발생할 가능성이 높고, 이로 인해 인수 테스트와 같은 후속단계에서 오류를 수정하기 위해 더 많은 시간이 소요된다.

[표 3]은 불일치 항목 분류기준에 따라 추출한 분석단계와 설계단계 명세서를 입력으로 받아들이며 명세서간에 발생할 수 있는 불일치 요소를 검출하는 과정이다.

표 3. 불일치 항목 추출 방법1

<pre> Procedure Detect-Inconsistent-Element-Analysis-Design-Phase Input : analysis phase specification A[1],...,A[4] design phase specification B[1],...,B[14] Output : inconsistent element I[1],...,[n] Begin FOR specifications list which it follows in specifications classification 1. FOR table related standard naming rule 1.1 compare ID of table k with ID of table m 1.2 count number of inconsistent item 2. FOR table related entity 1.1 compare item of entity definition table with item of ERD 1.2 count number of inconsistent item 3. FOR table related screen 1.1 compare ID of screen list table with item of screen layout 1.2 count number of inconsistent item 4. FOR table related class 1.1 compare ID between class list tables 1.2 count number of inconsistent item 5. FOR table related report 1.1 compare ID of report list table with name of report layout 1.2 count number of inconsistent item 6. FOR table related program 1.1 compare ID of program list table with name of program layout 1.2 count number of inconsistent item 7. FOR table related layer 1.1 compare ID of layer list table with name of layer definition table 1.2 count number of inconsistent item 8. FOR table related table 1.1 compare ID of table list table with name of table definition 1.2 count number of inconsistent item </pre>
--

본 논문에서는 분석 및 설계단계에서 작성되는 테이블 정보를 기초로 추가적으로 테스트 단계에서 통합 테스트를 수행하기 위해 사용되는 테스트 시나리오를 작성할 때 적용할 수 있는 테이블 분석방법을 제시한다.

통합 테스트를 할 때 기준은 해당 업무별로 테스트 시나리오를 작성하여 수행하며, 이 때 필요한 항목으로는 해당업무에 대응되는 프로그램 이름을 비롯하여 프로그램 ID, 테이블 정보, 이벤트 정보 등이다.

표 4. 불일치 항목 추출 방법2

```

Procedure Detect-Inconsistent-Element-Test-Phase
Input : process definition table, program and table relation table, table definition table
Output : inconsistent element [(1),...[n]
Begin
collect table data
1. FOR process definition table
1.1 search table item related process ID
1.2 compare table definition table with table item of 1.1
1.2.1 FOR table definition table
collect type and data length of columns related table
1.3 compare table item of program and table relation table with process definition table
1.3.1 count number of inconsistent item
1.3.2 save inconsistent items in inconsistent detection table
    
```

분석 및 설계단계에서 작성되는 명세서를 분석한 결과, 테스트 시나리오를 작성하는데 필요한 항목들을 포함하는 명세서로는 요구사항정의테이블, 프로그램과 테이블상관도, 프로세스정의테이블이다. 하지만, 이들 세 가지 명세서는 개별적인 특성으로 인해 항목 간 불일치 항목을 검출하는데 어려움이 있다.

그러므로 본 논문에서는 테스트 단계에서 신뢰성 있는 통합 테스트를 수행하기 위해 이들 세 가지 명세서에 대해 항목 간 불일치 요소를 검출하는 불일치 항목 추출 방법2를 [표 4]와 같이 제안한다. [표 4]의 불일치 항목 추출 방법2를 적용한 테이블 분석방법은 테스트 단계에서 사용되는 테스트 시나리오를 기술할 때 참조하는 테이블 정보에 대한 정확성을 증가시킴으로써 신뢰성 있는 통합 테스트를 수행할 수 있다.

IV. 테이블 분석을 위한 사례적용

이 절에서는 [표 3]과 [표 4]에 제시한 불일치 항목 추

출 방법1, 2를 실제 시스템 개발현장에서 단계별로 작성된 명세서에 적용한 예를 나타낸다.

1. 적용사례

사례에 이용한 명세서는 도시정보시스템 개발과정 중 분석단계와 설계단계별로 작성된 것으로 [그림 3]과 같이 구성된다. 도시정보 시스템 개발 프로젝트는 지하시설물 전산화사업의 일환으로 조사, 탐사 및 데이터베이스 구축을 위한 데이터 구축, 구축된 데이터의 관리, 데이터의 조회 및 처리를 위한 응용 소프트웨어, 시스템 구성을 위한 기술 구조, 다양한 형태로 제공되는 애플리케이션으로 구성된다. 데이터 구축단계를 통해 작성한 시설물 정보를 도로, 상수도, 하수도, 공통 업무별로 데이터베이스화하여 관리한다. 자료수집 및 데이터베이스화 과정을 거친 정보는 ArcInfo 등의 각종 개발 및 관리 소프트웨어를 통하여 가공되어 분류함으로써 이용된다. 최종적인 시스템은 도로, 상수도, 하수도와 공통 업무별로 기초자료 입력 및 검색 등의 관리기능을 수행하는 애플리케이션으로 구성된다.

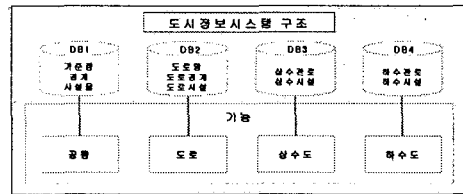


그림 3. 도시정보시스템 기능 및 DB 구조

[표 5]에서는 불일치 항목 추출 방법을 적용한 결과를, [표 6]에서는 상세한 오류정보를 나타낸다.

표 5. 불일치 항목 추출 방법 적용결과

적용 방법	불일치 항목수
불일치 항목 추출 방법1	46개
불일치 항목 추출 방법2	39개

[표 7]은 사례적용결과 분석표로서 불일치 항목이 기존의 수작업으로 실시한 방법보다 85개가 더 많이 검출되었으며, 짧은 검사시간 내에 편리하게 검출할 수 있다. 또한 분석, 설계단계에 추가적으로 테스트 단계에서도 오류를 검출할 수 있으므로 오류유형의 다양성을 확

인할 수 있다. 이러한 장점으로 기존의 검사 팀에 의해서 수행되던 방법보다 정확성을 높이며, 검사시간을 단축시킬 수 있어, 소요되는 시간 등이 기존의 방법보다 우수하다.

구사항 추적표를 중심으로 수작업으로 이루어진다. 이런 수작업으로 인해서 많은 명세서간에 불일치 항목이 발견되며, 전체 개발공정이 지연된다. 또한, 명세서들의 특성 때문에 연속적인 자동화가 이루어지지 않으며, 소프트웨어 감리에서도 불일치 항목이 많이 지적된다.

본 논문에서는 분석 및 설계단계에서 작성되는 개발 명세서간 일관성 유지를 위해서 먼저 개발현장의 명세서에 대하여 실시한 감리자료와 감리사레분석보고서를 조사하였다. 이를 바탕으로 분석 및 설계단계에서 작성되는 명세서간의 특성을 비교하여 관련 있는 항목별로 불일치 항목 기준을 분류하고, 분석, 설계, 테스트 단계별로 적용 가능한 불일치 항목 추출 방법을 제안하였다. 본 논문에서 제안한 불일치 항목 추출 방법을 이용하면, 분석 및 설계단계에서 작성되는 명세서에 오류를 줄임으로써 최종산출물인 소프트웨어에 대한 품질을 향상시킬 수 있다. 향후 연구과제로는 본 논문에서 제안된 불일치 항목 분류기준 보다 더 많은 불일치 항목 기준을 설정하여 불일치 항목 검출효과를 높이는 것이다.

표 6. 단계별 불일치 항목수(상세)

분류기준	분석단계 명세서에서 검출된 불일치 항목수															
	5	6	7	9												
표준명명규칙	1															
유즈케이스	1			1												
엔티티																
화면																
클래스	2															
보고서																
프로그램																
레이어																
테이블																
엔티티																
화면																
클래스	2															
보고서																
프로그램																
레이어																
테이블																
분류기준	설계단계 명세서에서 검출된 불일치 항목수															
	1	2	3	4	5	6	7	8	9	10	13	14	15	16		
표준명명규칙			1						1							
유즈케이스	1															
엔티티				3	3											
화면						2	2									
클래스																
보고서								7	7							
프로그램			3	3												
레이어											3	3				
테이블														1	1	
분류기준	테스트 단계에서 필요한 명세서 간의 불일치 항목수															
	12	19														
누락	22	17														

표 7. 적용결과 분석표

방법	비교항목	불일치 항목 수	지원단계	자동화 여부	검사 시간
제안 방안	$(\alpha + 85)$ 개	분석, 설계, 테스트	○	짧다	

V. 결론

시스템 개발현장의 각종 명세서 관리는 분석단계에서 발생한 요구사항에 대해서 현업 요구사항 정의서 및 요

참고 문헌

- [1] B.Neseibeh, J.Kramer, and A.Finkelstein, "A Framework for Expressing the Relationships between Multiple Views in Requirements Specification," IEEE Trans. on Software Eng., Vol.20, No.10, pp.760~773. 1994.
- [2] M. Fowler, and K. Scott, "UML Distilled Second Edition : A Brief Guide to the Standard Object Modeling Language," Addison-Wesley, 2000.
- [3] L.Kuzniarz, G.Reggio, J. L.Sourrouille, and Z.Huzar, Workshop on Consistency Problems in UML based software development, Workshop Materials, Research Report 2002:06, Blekinge Institute of Technology, Ronneby 2002.
- [4] 최은만, 소프트웨어 공학론(개정판), 사이텍미디어, 2001.
- [5] E. Wallmuller, Software Quality Assurance A

- practical approach, Prentice-Hall, 1994.
- [6] 한국전산원, 감리 결과 분석을 통한 주요문제점 및 개선사례 연구, 2001.
- [7] 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.
- [8] 정기원, 조용선, 권성구, "객체지향 설계방법에서 오류 검출과 일관성 점검기법 연구", 정보처리학회논문지D, 제6권, 제8호, 1999.
- [9] 김도형, 정기원, "객체지향 분석과정에서 오류와 일관성 점검 방법", 정보과학회논문지B, 제26권, 제3호, pp.380~392, 1999.
- [10] 최진재, 황선영, "유즈케이스 모델링을 위한 요구 사항 중복 및 불일치 분석", 정보처리학회논문지 D, 제31권, 제7호, pp.869~882, 2004.
- [11] 한국전산원, 정보시스템 감리기준, 1998.
- [12] 최윤석, "설계단계 순차도와 객체지향 프로그램간의 일관성 검증", 숭실대학교 석사학위논문, 1999.
- [13] 조진형, 배두환, "UML 객체지향 분석모델의 완성 및 일관성 진단을 위한 시나리오 검증기법", 정보과학회논문지B, 제28권, 제3호, pp.211~223, 2001.

저 자 소 개

최 신 형(Shin-Hyeong Choi)

정회원



- 1993년 2월 : 울산대학교 전자계산학과(공학사)
- 1995년 2월 : 경남대학교 전자계산학과(공학석사)
- 2002년 8월 : 경남대학교 컴퓨터공학과(공학박사)

• 2003년 9월~현재 : 삼척대학교 컴퓨터 제어계측공학과 전임강사

<관심분야> : 테스트 및 품질평가, 신뢰도 분석, 컴포넌트 기반 개발방법론, 임베디드 시스템