

## 요구사항의 품질 향상을 위한 자동화 검증 기법

김철진<sup>1\*</sup>

<sup>1</sup>인하공업전문대학 컴퓨터시스템과

### An Automated Verification Technique for Enhancing Quality of Requirement

Chul-Jin Kim<sup>1\*</sup>

<sup>1</sup>Dept. of Computer Systems and Engineering, Inha Technical College

**요약** 소프트웨어의 품질은 요구사항 품질과 강하게 관계되어 있다. 이에 따라서 기업들은 요구사항의 품질을 향상시켜 주기 위한 노력을 하고 있다. 그러나 명세서 형태의 요구사항은 검증하기 어려우며, 분석가의 비정형화된 업무 지식에 의존해야 한다. 또한 명세서 형태의 요구사항 품질을 향상시켜 주기 위한 정형화된 방법론이나 자동화된 기법이 미흡한 상황이다. 본 논문에서는 요구사항의 품질을 향상시켜 주기 위한 요구사항 검증 프로세스 및 자동화 검증 도구를 제안한다. 검증을 위해 요구사항에 대한 외적부 설계와 내적부 설계를 비교한다. 사례연구를 통해 제안된 요구사항 자동화 검증 기법의 타당성을 검증한다.

**Abstract** Software quality is strongly associated with requirement quality. Accordingly, companies are trying effect for enhancing requirement quality. But, it is difficult to verify the requirement of specification format, which have to rely on abnormal business knowledge of analyst. Also, it is insufficient to normal methodology or automated technique for enhancing requirement quality of specification format. In this paper, we propose the process of requirement verification and automated verification tool for enhancing requirement quality. we compare the external view design and internal view design for verifying requirement. In the case study, we evaluate the feasibility of the proposed automated verification technique of requirement.

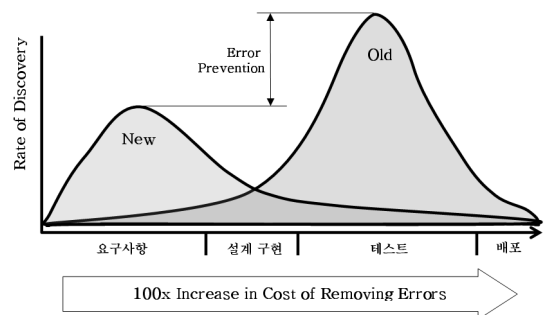
**Key Words** : Requirement Verification, Internal View Design, External View Design, Design Simulation

### 1. 서론

소프트웨어의 품질은 요구사항 품질과 강하게 관계되어 있다. 그림 1에서와 같이 요구사항의 품질에 따라 소프트웨어 개발 비용을 크게 줄 일 수 있다[1].

이에 따라서 기업들은 요구사항의 품질 향상에 많은 관심을 가지고 있으며, CMMi(Capability Maturity Model Integration)[2]나 SPICE(Software Process Improvement Capability dTermination)[3] 등을 통해 요구사항의 품질을 향상시켜 주기 위한 노력을 하고 있다. 그러나 명세서 형태의 요구사항은 검증하기 어려우며, 분석가의 비정형화된 업무 지식에 의존해야 한다. 또한 명세서 형태의 요구사항 품질을 향상시켜 주기 위한 정형화된 방법론이나 자동화된 기법이 미흡한 상황이다. 따라서 본 논문에서는

요구사항의 품질을 향상시켜 주기 위한 요구사항 검증 프로세스 및 자동화 검증 도구를 제안한다.



[그림 1] 요구사항 단계에 의한 비용절감 효과[1]

[Fig. 1] Effect of Cost Reduction by Requirement Phase[1]

\*Corresponding Author : Chul-Jin Kim

Tel: +82-10-6398-7471 email: cjkim@inhac.ac.kr

접수일 12년 06월 18일

수정일 (1차 12년 06월 22일, 2차 12년 06월 25일)

계재확정일 12년 09월 06일

본 논문의 구성은 다음과 같다. 2장에서는 요구되는 기반 기술 및 관련 연구에 대해 알아본다. 3장에서는 본 논문의 핵심연구로서 요구사항 자동화 검증 기법을 제안한다. 4장에서는 커피자판기 사례를 요구사항 자동화 기법에 적용하여 본 연구의 타당성을 검증한다.

## 2. 관련 연구

관련연구에서는 본 논문에서 제안하는 요구사항 자동화 검증 기법에 관련된 기본 기술을 조사하며, 관련된 연구에 대해서 분석한다.

### 2.1 XMI

XMI(XML Metadata Interchange)[4]는 OMG(Object Management Group) 표준으로 XML 기반의 교환 메타데이터 표준이다. 일반적으로 XMI는 UML(Unified Modeling Language) 모델간의 상호교환 표준으로 사용된다. 또한 설계 모델에서 구현 언어로의 변환을 위한 목적으로 사용되기도 한다.

본 논문에서는 요구사항 자동검증을 위해 설계 산출물 간에 비교를 수행하는데, 이때 설계 산출물의 XMI를 비교한다.

### 2.2 산업계의 요구사항 검증

연구 [5]에서는 산업계의 요구사항 검증 사례를 통해 요구사항 검증 기술에 대해 연구한다. 요구사항의 정확한 검증을 위해 어휘(lexical) 수준의 NLP(Natural Language Processing)을 이용하며, 요구사항을 작성하고 체크하기 위한 가이드를 제시한다.

요구사항 검증의 품질을 보장하기 위한 도구들은 RMS(Requirement Management System) 도구 내에 요구사항 검증 기능을 포함하거나, 추적성이나 임팩트 분석을 수행하는 도구 내에 포함되기도 한다. 또한 요구사항의 어휘나 분석을 수행하는 도구에 요구사항 검증 기능을 포함하고 있다.

연구 [5]는 요구사항 검증 기법을 조사하였으나 이러한 기법들은 요구사항 자체의 어휘적인 검증을 제공할 뿐, 고객 요구사항을 만족할 수 있는지의 품질을 검증하는 측면에서는 미흡하다.

### 2.3 소프트웨어 요구사항 관리 사례 연구

연구 [6]에서 소프트웨어 요구사항 프로세스(SRP)는 요구사항 개발 단계와 요구사항 관리 단계로 구분한다.

요구사항 개발 단계는 요구사항 추출, 분석, 검증하는 절차로 구성된다. 요구사항 관리 단계는 요구사항들이 구현되어 제품화될 때까지 변경 사항을 유지 관리 하는 절차로 구성된다. 본 연구에서 요구사항 검증은 사용사례 다이어그램에 대해 Inspection[7], Walkthrough[7] 등의 검토기법을 통해서 사용사례 명세서를 검토한다.

연구 [6]에서는 요구사항 검증에 대해 분석가의 업무 지식에 의존하며, 정형화되고 자동화된 기법을 제공하지 못하고 있다.

### 2.4 기업 아키텍처를 지원하는 요구사항 분석 프로세스에 관한 연구

연구 [8]은 기업의 아키텍처를 효율적으로 수립하기 위한 요구사항 관리 프레임워크, 분석 프로세스를 제시한다. 연구 [8]에서는 요구사항의 불일치와 불완전성을 보완하기 위해 요구사항 관리 프레임워크를 제시하였으며, 프레임워크는 3가지 차원(View, Perspective, Concept)으로 구성된다. View는 분석하고자하는 영역과 설계를 구분하며, Perspective는 분석하고자 하는 참여자의 관점을 정의한다. Concept는 정보 시스템과 업무 아키텍처를 정의한다.

연구 [8]에서는 요구사항에 대한 검증을 위한 프레임워크를 제안하고 있으나, 비정형적인 부분이 많다. 또한 자동화에 대한 연구도 미흡한 상황이다.

### 2.5 객체지향 페트리넷을 이용한 요구사항검증

연구 [9]는 객체지향의 개념을 이용하여 계층구조의 페트리넷인 HOONet을 제안하였으며, 이를 이용하여 복잡한 요구사항을 하향식 분할 방식으로 정의할 수 있다. 특히, 요구사항이 부분적으로 제시되었거나, 전체 요구사항이 분석되지 않더라도 시스템의 모델링에 대한 분석이 이루어 질 수 있도록 한다. HOONet에서 검증할 수 있는 특성들은 교착상태 검사, 생존성 검사, 공정성 검사, 비결정성 검사 등을 수행할 수 있다.

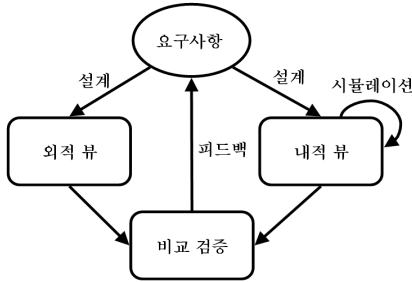
연구 [9]는 복잡한 요구사항을 체계적으로 명세하기 위한 방법으로, 고객 요구사항을 만족시킬 수 있는 품질 검증 방법과는 차이를 갖는다.

## 3. 요구사항 자동화 검증 기법

본 논문에서는 요구사항 자동화 검증을 위한 검증 프로세스를 제안하며, 검증 비교를 자동화할 수 있는 검증 도구를 제안한다.

### 3.1 요구사항 자동화 검증 구조

요구사항의 자동화 검증 구조는 그림 2와 같다. 요구사항에 대해 외적부(External View)와 내적부(Internal View)로 서로 다른 관점에서 설계하여 비교검증 한다. 이렇게 비교 검증된 결과를 요구사항에 피드백을 통해 요구사항의 품질을 향상시킬 수 있다.

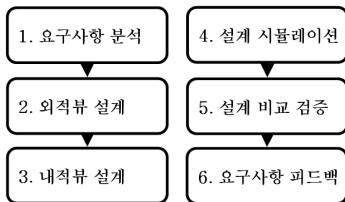


[그림 2] 요구사항 자동화 검증 구조  
[Fig. 2] Automatic Verification Structure of Req.

외적부 설계는 요구사항에 대해 순차 다이어그램 (Sequence Diagram)을 설계하는 것이며, 내적부는 상태 머신 다이어그램(State Machine Diagram)을 설계하여 추적 순차 다이어그램(Trace Sequence Diagram)을 시뮬레이션하는 것이다. 비교 검증은 순차 다이어그램과 추적 순차 다이어그램을 비교하여 차이를 분석한다.

### 3.2 요구사항 자동화 검증 프로세스

요구사항 자동화 검증 프로세스는 그림 3에서와 같이 요구사항 분석, 외적부 설계, 내적부 설계, 설계 시뮬레이션, 설계 비교 검증, 요구사항 피드백의 단계를 통해 요구사항의 품질을 향상시킬 수 있다.



[그림 3] 요구사항 자동화 검증 프로세스  
[Fig. 3] Automatic Verification Process of Requirement

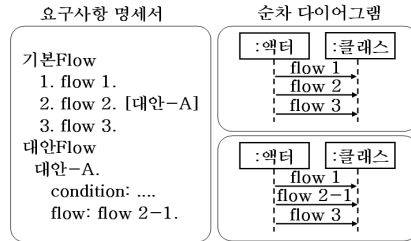
#### 1) 요구사항 분석

요구사항 분석은 사용자 요구사항을 기반으로 사용자 레 다이어그램(Use Case Diagram)을 설계하며, 각각의 사용 사례(Use Case)에 대해 사용 사례 명세서(Use Case Specification)를 정의한다. 사용 사례 명세서의 기본흐름

(Baic Flow)과 대안흐름(Alternative Flow)는 다음 단계의 외적부 설계와 내적부 설계하는 입력 데이터가 된다.

#### 2) 외적부 설계

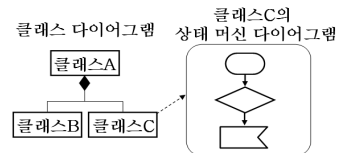
외적부 설계는 요구사항 명세서의 기본흐름과 대안흐름에 대해 순차 다이어그램을 설계한다. 그림 4와 같이 기본흐름과 대안흐름에 의한 순차도를 각각 설계한다.



[그림 4] 요구사항에 대한 순차 다이어그램 설계  
[Fig. 4] Seq. Diag. Design for Requirement

#### 3) 내적부 설계

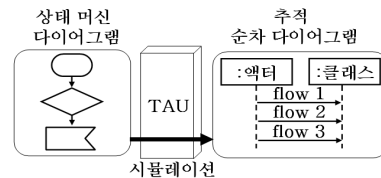
내적부 설계는 외적부 설계와 다르게 시스템 관점에서의 설계로서 요구사항에 대해 클래스 단위로 상태 머신 다이어그램을 설계한다. 본 단계에서는 그림 5에서와 같이 도출된 클래스 각각에 대해 상태 머신 다이어그램을 설계한다.



[그림 5] 클래스에 대한 상태 머신 다이어그램 설계  
[Fig. 5] State Machine Diag. Design for Class

#### 4) 설계 시뮬레이션

설계 시뮬레이션은 내적부 설계에서 도출된 상태 머신 다이어그램을 통해 순차 다이어그램을 생성한다. 본 단계에서는 Rational TAU[10] 도구를 이용하여 시뮬레이션을 하며, 그림 6에서와 같이 상태 머신 다이어그램을 통해 추적 순차 다이어그램을 생성한다.



[그림 6] 추적 순차 다이어그램 생성  
[Fig. 6] Creation of Trace Sequence Diagram

상태머신 다이어그램과 Rational TAU에 의해 생성되는 추적순차 다이어그램의 표기들 간의 관계는 표 1과 같다. 상태머신 다이어그램의 입력 표시는 생성되는 추적순차 다이어그램의 메시지전달에 해당된다.

[표 1] 상태머신 다이어그램과 추적순차다이어그램 관계  
[Table 1] Relationship between SMD and TSD

상태머신 다이어그램	(생성)추적순차 다이어그램
입력	메시지 전달
조건	조건별로 추적 순차다이어그램 생성 (입력 창이 있어 조건값을 입력하여 생성)
상태	추적 순차 다이어그램의 상태

5) 설계 비교 검증

설계 검증은 외적부 설계 단계 산출물과 내적부 설계 단계 산출물을 비교한다. 외적부 설계의 순차 다이어그램과 내적부 설계의 시뮬레이션을 통해 생성된 추적 순차 다이어그램을 비교한다. 두 다이어그램의 비교는 다이어그램의 저장 형태인 XML(XML Metadata Interchange)[4]를 비교하여 일치성을 확인한다.

표 2에서와 같이 비교 요소는 메시지(Message), 객체(Object), 매개변수(Parameter) 이며, 비교 방법은 존재 유무(Existence), 전후 관계(Before/After Msg.), 타입(Type Match), 순서(Order)를 통해 두 다이어그램 간의 차이를 비교한다.

[표 2] 설계 검증 요소

[Table 2] Verification Factors of Design

비교 대상	비교 요소		비교 방법	우선 순위
SD vs. TSD	Message	Between Actor and System	Existence	P1
			Before/After Msg.	P1
		Between Classes within System	Existence	P2
			Before/After Msg.	P2
SD vs. TSD	Object	Existence	P2	
SD vs. TSD	Parameter	Existence	P1	
		Type Match	P1	
		Order	P1	

SD : Sequence Diagram TSD : Trace Sequence Diagram

6) 요구사항 피드백

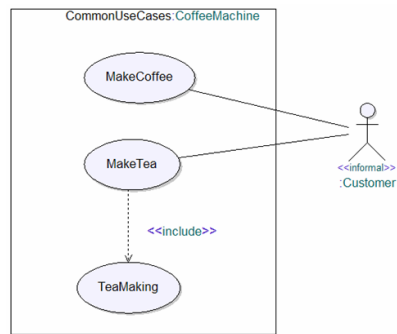
설계 검증 비교를 통해 일치성을 평가하여 불일치하는 요소를 분석하고, 관련된 요구사항을 수정한다. 내적부 설계 산출물인 순차 다이어그램 측면에서의 불일치와 외적부 설계 산출물인 추적 순차 다이어그램 측면에서의 불일치를 도출한다. 일반적으로 자동 생성된 추적 순차 다이어그램의 불일치 내용을 요구사항에 반영한다.

4. 실험 및 평가

본 논문에서 제안한 요구사항 자동화 검증 기법을 커피 자판기 사례에 적용하여 타당성을 검증한다. 커피 자판기는 동전을 삽입한 후 커피나 차를 선택할 수 있으며, 커피를 선택할 경우 자판기 내부에서는 물과 커피를 공급한 후 사용자에게 커피를 제공하는 기능으로 구성된다. 이러한 요구사항을 본 논문의 자동화 검증 프로세스에 따라 적용한다. 본 사례에 대해 설계는 IBM의 Rational TAU를 이용한다.

1) 요구사항 분석

커피 자판기 요구사항에 대한 사용사례 다이어그램과 'MakeCoffee' 사용사례에 대한 사용사례명세서는 그림 7과 그림 8과 같이 정의한다.



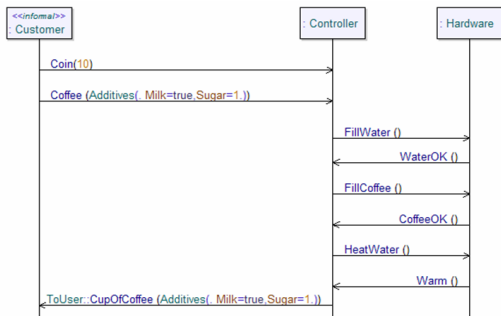
[그림 7] 커피 자판기 시스템의 사용사례 다이어그램  
[Fig. 7] Use Case Diag. of Coffee Machine System

<ul style="list-style-type: none"> <li>기본 Flow                     <ul style="list-style-type: none"> <li>BF-1                             <ol style="list-style-type: none"> <li>1. Customer는 Coffee Machine에 Coin을 넣는다.</li> <li>2. Customer는 Coffee를 선택한다.</li> <li>3. Coffee Machine은 Water를 채운다.</li> <li>4. Coffee Machine은 Coffee를 채운다.</li> <li>5. Coffee Machine은 Water을 태운다.</li> <li>6. Coffee Machine은 Customer에게 Coffee를 제공한다.</li> <li>7. 본 'MakeCoffee' Use Case를 종료한다.</li> </ol> </li> </ul> </li> <li>대안 Flow                     <ul style="list-style-type: none"> <li>None</li> </ul> </li> </ul>
--

[그림 8] MakeCoffee의 사용사례 명세서  
[Fig. 8] Use Case Specification of MakeCoffee

2) 외적부 설계

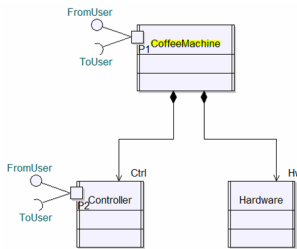
'MakeCoffee' 사용사례에 대한 외적부 설계로서 그림 9와 같이 순차 다이어그램을 설계한다. 'MakeCoffee' 순차 다이어그램은 그림 8의 사용사례 명세서를 기반으로 설계한다. 본 외적부 설계 산출물은 자동 생성된 추적 순차 다이어그램과의 검증 비교 대상이 된다.



[그림 9] MakeCoffee의 순차 다이어그램  
[Fig. 9] Sequence Diag. of MakeCoffee

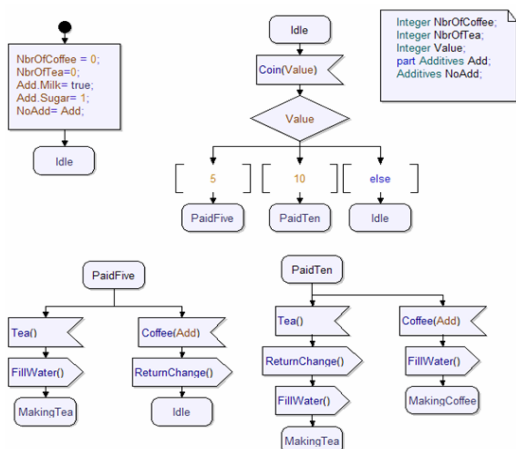
### 3) 내적부 설계

상태 머신 다이어그램은 그림 8의 'MakeCoffee' 사용자 명세서를 기반으로 설계한다. 상태 머신 다이어그램은 그림 10의 클래스 다이어그램(Class Diagram) 중에 'Controller' 클래스에 대해 설계한다.



[그림 10] 커피 자판기 시스템의 클래스 다이어그램  
[Fig. 10] Class Diag. of Coffee Machine System

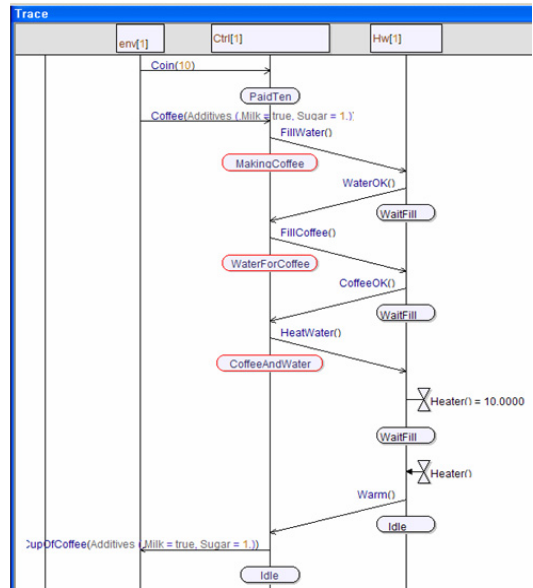
내적부 설계를 위해 그림 11과 같이 상태 머신 다이어그램을 설계한다.



[그림 11] Controller 클래스의 상태 머신 다이어그램  
[Fig. 11] State Machine Diag. of Controller Class

### 4) 설계 시뮬레이션

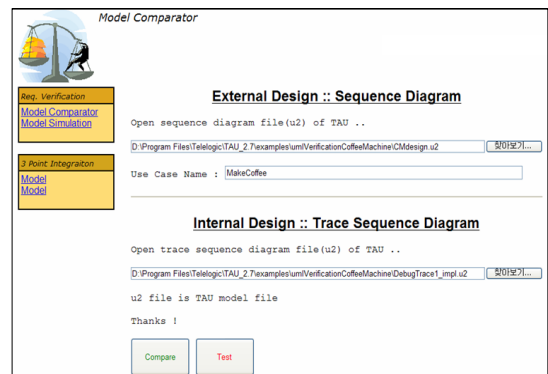
설계 시뮬레이션은 내적부 설계에서 설계한 그림 11의 상태 머신 다이어그램을 통해 그림 12와 같은 추적 순차 다이어그램을 자동 생성한다 (by Rational TAU). 본 추적 순차 다이어그램은 그림 9의 순차 다이어그램과 검증 비교 대상이 된다.



[그림 12] Controller 클래스의 상태 머신 다이어그램  
[Fig. 12] State Machine Diag. of Controller Class

### 5) 설계 비교 검증

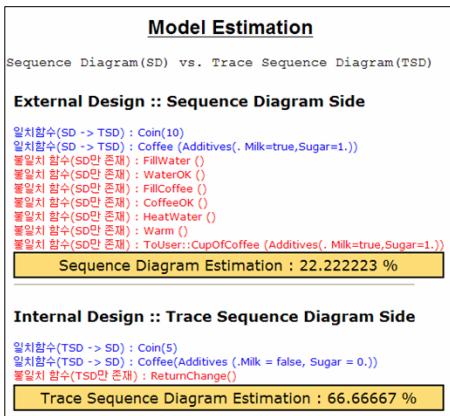
설계 비교 검증은 외적부 설계 산출물인 순차 다이어그램의 XMI와 내적부 설계 산출물인 추적 순차 다이어그램의 XMI를 입력으로 받아 표 2의 요소들을 비교한다.



[그림 13] XMI 모델 비교기  
[Fig. 13] XMI Model Comparator

그림 13은 Rational TAU 설계 산출물인 XMI 형식의 '.u2' 파일을 입력하여 비교하기 위한 요구사항 자동화 검증 도구이다. 본 도구는 Apach Tomcat(5.5) 서버 기반으로 사용자 인터페이스는 JSP로 개발되었으며, TAU 설계 파일을 해석하기 위해 DOM Parser를 이용하여 XMI를 해석한다.

그림 14는 비교된 결과로서 순차 다이어그램 측면에서는 추적 순차 다이어그램과의 일치도가 22%이며, 추적 순차 다이어그램 측면에서 순차 다이어그램과의 일치도는 66%이다. 이 결과를 통해 순차 다이어그램이 'MakeCoffee' 사용사례 명세서에 대해 더 많은 설계 요소들을 포함하고 있다고 할 수 있다.

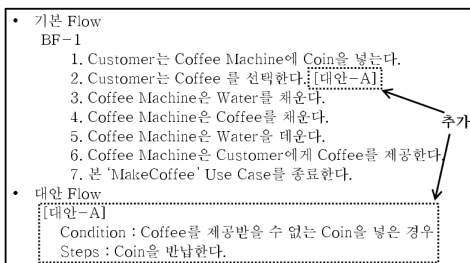


[그림 14] 비교 결과  
[Fig. 14] Comparison Result

6) 요구사항 피드백

검증 비교를 통해 순차 다이어그램에서는 발견하지 못했지만, 추적 순차 다이어그램에서 발견한 'ReturnChange()' 기능에 대해 요구사항에 반영한다.

그림 15에서와 같이 커피자판기에 부족한 'Coin'을 넣은 후 커피를 선택했을 때 대안 흐름이 추가되었다.



[그림 15] 수정된 MakeCoffee 사용사례명세서  
[Fig. 15] Modified Use Case Spec. of MakeCoffee

지금까지의 사례를 통해 내적부 설계 산출물인 순차 다이어그램과 외적부 설계에 의해 자동 생성된 추적 순차 다이어그램을 비교하여 요구사항을 수정할 수 있음을 실험하였다. 이에 따라 본 논문에서 제안한 요구사항 자동화 검증 기법이 실현 가능성을 증명하였다.

5. 결론 및 향후 연구과제

본 논문은 요구사항의 품질을 향상시키기 위해 사용자 관점의 외적부 설계와 클래스 관점의 내적부 설계를 비교 검증하는 기법을 제안하였다. 설계 산출물들을 비교하기 위해 자동화 검증 도구를 제안하였으며, 검증 도구를 통한 검증 결과는 요구사항을 수정할 수 있는 기반을 제공할 수 있음을 파악 하였다. 향후 연구과제로서 자동화 검증 시 다양한 비교 요소들을 고려할 수 있도록 연구한다.

References

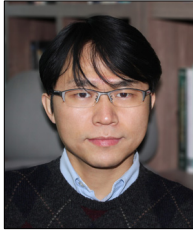
- [1] Boehm, B.W. et al, "Some Experience with Automated Aids to the Design of Large-Scale Reliable Software", IEEE Trans. On Software Engineering, 1975.
- [2] Capability Maturity Model Integration (CMMI), www.sei.cmu.edu/cmmi/, SEI Carnegie Mellon.
- [3] ISO/IEC 15504 TR2, Software Process Assessment and Capability determination, ISO/IEC, 1998.
- [4] MOF 2 XMI Mapping, http://www.omg.org/spec/XMI/, OMG, 2011.
- [5] Gauthier Fanmuy, Anabel Fraga and Juan Llorens, "Requirements Verification in the Industry", Complex Systems Design & Management, Pages 145-160, 2012.
- [6] Choi J. E., Choi S. K., and Lee S. A., "The Case Study of Software Requirement Management", Proceedings of Korean Institute of Information Scientists and Engineers, Vol.29, No.1(B), p445-447, 2002.
- [7] Dean Leffingwell, Don Widrig, Managing Software Requirement, A Unified Approach, Addison-Wesley, 2000.
- [8] Choi B. K. and Leem C. S., "A Study on Requirement Analysis Process that support Enterprise Architecture Design", The Journal of Korean Institute of CALS/EC, Vol.8, No.1, p35-54, 2003.
- [9] Hong J. E., Yoon I. C., and Bae D. H., "Hierarchical Specification and Verification of Requirements using An Object-Oriented Petri Net", The Journal of Korean

Institute of Information Scientists and Engineers,  
Vol.27, No.2, p157-167, 2000.

- [10] Rational TAU,  
<http://www-01.ibm.com/software/awdtools/tau/>, IBM.
- 

김 철 진(Chul-Jin Kim)

[중심회원]



- 2004년 2월 : 숭실대학교 대학원 컴퓨터학과 (공학박사)
- 2004년 4월 : 가톨릭대학교 컴퓨터 정보공학부 강의전담교수
- 2004년 3월 ~ 2009년 2월 :삼성 전자 책임연구원
- 2009년 3월 ~ 현재 : 인하공전 컴퓨터시스템과 조교수

<관심분야>

컴포넌트 기반 개발 방법론, 컴포넌트 커스터마이제이션,  
모바일 서비스, 클라우드 컴퓨팅