

# 유스케이스 모델링을 위한 요구사항 중복 및 불일치 분석

## (Requirements Redundancy and Inconsistency Analysis for Use Case Modeling)

최진재<sup>†</sup>    황선영<sup>\*\*</sup>

(Jin-Jae Choi) (Sun-Young Hwang)

**요약** 본 논문에서는 정형방법론의 일관성 검사방법을 유스케이스 모델링에 적용함으로써 논리적으로 일관되고 구조화된 요구사항 모델을 생성할 수 있는 방안을 제안하였다. 제안된 방안을 통해 서로 중복되거나 충돌할 수 있는 다중 사용자관점의 부분적인 요구사항들을 구조화된 요구사항 모델로 통합할 수 있다. 요구사항 모델의 구조는 문맥목표와 관심영역의 중복구조를 통하여 분석되며, 요구사항의 모델의 일관성은 요구사항 통합의 수단으로 요구사항 명세의 중복관계에 기반한 일관성 검사방법을 사용함으로써 획득된다.

제시된 방법을 사례분석에 적용한 결과 제안된 방안이 요구사항의 중복관계 및 불일치를 도출할 수 있을 뿐만 아니라 다중 사용자관점의 부분적인 요구사항들을 일관성을 갖는 통합된 요구사항 모델로 전환함으로써 소프트웨어의 행위와 기능을 명확하게 규명할 수 있음을 보여 주었다. 제안된 방안을 통하여 개발 초기단계의 유스케이스 모델링에 있어서 요구사항 불일치 도출능력을 강화할 수 있으며, 고객만족을 위해 사용자와 개발자간의 의사소통을 촉진할 수 있다.

**키워드** : 요구공학, 유스케이스, 요구사항 중복, 요구사항 불일치

**Abstract** This paper proposes an effective method to create logically consistent and structured requirement model by applying consistency control approach of the formal method to the use-case modeling. This method integrates the multi-perspective scattered requirement segments that may overlap and conflict each other into a structured requirement model. The model structure can be analyzed based on context goal and concerned area overlap analysis. The model consistency can be achieved by using specification overlap-based consistency checking method as an integration vehicle.

An experimental application to case study shows that the proposed method can successfully identify requirement overlaps and inconsistency. It can also transfer multi-viewpoint requirement segments into a consistently integrated use-case model to clarify software behaviors and functionality. This method helps users to enhance capability to identify specification inconsistency in the use-case modeling at the early stage of software engineering development. The proposed approach can also facilitate communication between users and developers to ensure customer satisfaction.

**Key words** : Requirement Engineering, Use Case, Specification overlap, Requirement Inconsistency

### 1. 서론

정보화시대를 맞이하여 소프트웨어는 기업에서 많은 업무를 자동화 작업으로 대체함으로써 기업의 생산성

향상에 지속적으로 기여해 오고 있다. 이러한 소프트웨어의 긍정적인 효과에도 불구하고 소프트웨어 프로젝트는 사용자의 요구사항에 충족할 수 없는 많은 실패를 야기하고 있다. 스탠디쉬 그룹의 조사에 따르면 소프트웨어 프로젝트의 31.1%는 완성 전에 취소되었고, 계획된 시간과 예산 내에서 완성된 프로젝트는 단지 16.2%에 불과한 것으로 나타났다[1]. 그 주된 원인으로 사용자의 부적절한 입력물(12.8%), 불완전한 요구사항(12.3%), 비현실적 기대사항 및 불분명한 목표(11.2%)

· 본 논문은 "대학 IT 연구센터 육성지원 사업"의 연구결과로 수행되었습니다.

† 정 회 원 : 지엠대우오토엔테크놀로지 Worldwide Purchasing 차장  
jinjae.choi@gmdat.com

\*\* 종신회원 : 서강대학교 전자공학과 교수  
hwang@ccs.sogang.ac.kr

논문접수 : 2003년 11월 18일

심사완료 : 2004년 4월 17일

등 부적절한 요구사항 정의 및 관리를 지적하고 있다 [1]. 또한 소프트웨어의 복잡성이 심화되면서 기업 경영자의 48%가 5년 전보다 소프트웨어 개발에 있어서 더 많은 실패를 겪고 있다고 응답하고 있다[1,2]. 요구사항 분석에 소요되는 비용이 전체 비용에 5%에 불과하지만 이 과정이 전체 소프트웨어 품질 개선의 50%를 좌우한다는 결과를 통해서 보듯 소프트웨어의 요구사항은 소프트웨어 개발에 있어서 중요한 문제로 인식되고 있다[3].

소프트웨어 개발 실패의 가장 큰 원인은 사용자 요구사항의 잘못된 이해와 분석에서 발단된다[4,5]. 사용자 요구사항에 있어서 본질적인 문제는 요구사항에 있어서 정형성과 비정형성 간의 관계에서 비롯된다[6]. 정형화된 소프트웨어의 행위를 이해하기 위해서는 사용자의 비정형적 요구사항을 정형화된 요구사항으로 전환해야 하며, 이 과정에서 사용자와 분석자간의 오해는 요구사항 의미의 모호성, 요구사항 정의의 부정확성, 변경에 따른 다른 요구사항과의 충돌, 일관성의 결여, 요구사항의 불완전성을 야기한다[5,7]. 따라서 일관되고 완전한 요구사항 모델을 얻기 위해서는 반복적인 요구사항 추출 및 분석과정을 통하여 상위단계의 목표와 시스템 내부의 관계로 명확하게 구조화된 요구사항을 정의하고 명시해야 한다[8]. 또한 요구사항의 불일치를 분석하고 지속적으로 관리하여야 하며 핵심과정으로 시스템을 요청하는 사용자 관점에서 동의할 얻어야 한다[6,9].

요구공학을 주제로 객체지향기술을 이용하는 많은 연구 중에서 유스케이스 중심의 분석은 사용자 관점에서 요구사항을 기술할 수 있는 모델링 언어를 제안함으로써 주목 받고 있으며, 객체지향 분석설계 및 컴포넌트 기반의 개발에서 요구사항을 분석하고 명세화하는데 널리 사용되고 있다[10,11,12]. 그러나 유스케이스는 서로 독립적이지 않으며, 상황을 발생시키고 종결시키는 상황을 가지고 있기 때문에 그들 간에 서로 겹쳐질 수 있고 동시에 발생할 수도 있다. 이로 인하여 유스케이스는 요구사항을 구조화하기 어렵고, 요구사항 간의 영향관계를 분석하기 어려우며, 요구사항 간의 불일치를 식별하기 어렵다는 단점을 갖고 있다[9]. 반면에 정형 방법론 [13-16]은 예측 가능한 수학적 계산법에 의하여 엄격하고 정형화된 요구사항 명세를 표현하고 있기 때문에 요구사항의 일관성 및 완전성을 검증할 수 있다[17]. 그러나 이 방법론은 사용자가 이를 이해하기 어렵기 때문에 사용자 관점에서 요구사항의 타당성 검사가 어렵다 [18,19].

현재 사용되고 있는 유스케이스 모델링 접근방식은 요구사항의 불일치를 식별하고 이를 관리할 수 있는 명백한 공정이 제시되지 않고 있으며, 요구사항 불일치를

분석하는 정형 방법론은 사용자의 관점에서 이를 이해하고 가시적으로 구조화하는데 한계를 갖고 있다. 본 연구는 상기에 기술된 단점을 보완하기 위한 대안으로써 요구사항을 구조화하고, 관점들간의 유사성을 분석하여 그 중복관계를 명시하며 요구사항의 불일치를 분석하는 방안을 유스케이스 모델을 이용하여 제시한다. 본 논문의 구성은 다음과 같다. 2절에서 관련 연구를 설명하고, 3절에서는 요구사항 중복관계 규명을 통해 요구사항 불일치 분석방법을 유스케이스에 적용하는 방안을 제시한다. 4절에서는 3절에서 제시된 요구사항 불일치 분석방안과 유스케이스 모델의 확장을 예제에 적용하여 그 타당성을 검증하고, 5절에서 결론을 맺고자 한다.

## 2. 관련연구

### 2.1 요구사항의 중복 및 불일치

요구사항의 “중복(Overlap)”이란 서로 다른 요구사항 명세의 내용이 개발코자 하는 대상 시스템의 공통된 측면을 명시하는 것을 의미한다. “중복”과 “불일치”는 요구사항 명세간의 존재하는 “간섭”의 2 가지 수준을 말하는 것으로 중복은 불일치의 전제조건이 된다[20]. Finkelstein은 요구사항 불일치의 전제조건인 요구사항 중복에 대하여 요구사항 명세사이에 발생할 수 있는 중복관계를 정의하고, 이러한 중복관계의 속성을 규명함으로써 요구사항의 불일치를 검출할 수 있는 방안을 제시하였다[16]. 제안된 방안은 일차 술어논리를 요구사항 명세형식으로 사용한다. 요구사항 명세상에 존재하는 객체상수 및 술어기호간에 존재할 수 있는 모든 중복관계는 비중복(Null Overlap), 완전중복(Total Overlap), 포함중복(Inclusive Overlap), 부분중복(Partial Overlap)의 4가지로 분석된다. 이러한 4가지 중복관계는 객체상수와 술어기호를 단언한 사용자가 이들 기호에 대하여 인식하고 있는 의미를 기반으로 식별된다. 요구사항의 불일치는 요구사항 명세가 요구사항 어휘의 중복관계에서 유추된 일관성 규칙을 충족하지 않을 때 발생한다 [21]. 두개의 요구사항 명세와 이들에 대해 단언된 모든 중복관계를 일차술어합수로 전환한 정형식의 합집합으로부터 요구사항이 특정한 일관성 규칙을 어기고 있는지를 검사할 수 있다. Finkelstein이 제안한 이 방안은 일차 술어논리를 이용하여 정형화된 요구사항 명세를 표현하고 있기 때문에 요구사항 기술에 있어서 모호함을 제거할 수 있다. 그러나 이 방안은 요구사항의 구조화에 대한 구체적인 방안이 제시되지 못하고 있으며, 사용자 관점에서 요구사항 명세의 타당성 검사가 어렵기 때문에 개발자와 사용자 사이의 합의를 쉽게 도출하기 어렵다는 단점이 있다[18,19].

2.2 유스케이스 중심의 분석(Use Case Driven Analysis)

유스케이스 모델은 기본적인 개념으로 액터와 유스케이스를 사용한다. 이러한 개념은 시스템의 외부에 존재하는 대상인 액터와 시스템에서 수행되어야 하는 대상인 유스케이스를 정의함으로써 개발되어야 할 시스템의 범위를 결정하는데 도움을 준다. 사용자는 시스템을 사용하는 실제의 대상이지만, 액터의 개념은 사용자가 수행하는 특정 역할을 나타낸다. 유스케이스는 사용자가 특정 액터의 입장에서 시스템과의 상호작용을 통해 수행되는 트랜잭션의 시퀀스를 의미한다. 각 유스케이스의 개별적 실행은 유스케이스 클래스의 인스턴스로 간주할 수 있다. 유스케이스 인스턴스는 유스케이스가 실행되는 동안에만 존재하는 개념으로 사용자가 요구하는 모든 인스턴스에 대한 서비스를 제공하기 위해서 유스케이스 클래스는 시스템의 완벽한 기능을 규정할 수 있어야 한다. 이러한 유스케이스 중심의 접근방식을 통해 시스템 전체의 아키텍처를 사용자가 원하는 서비스에 의해서 결정할 수 있다. 유스케이스 중심의 접근방식은 사용자와 개발자가 액터의 사용적 측면에서 서로 다른 유스케이스들을 독립적으로 발견하고 분석하도록 함으로써 하나의 유스케이스에 초점을 둘 수 있게 하며, 사용자의 요구사항에 따라 정확한 시스템이 개발되고 있는지를 쉽게 검증할 수 있도록 한다. 유스케이스 중심 분석의 가장 주된 단점은 구조화의 지원이 약하다는 점과 이들 유스케이스들 간의 영향관계를 알 수 없다는 것이다[9]. 유스케이스 모델은 개별적으로 사용자의 관점을 갖는 부분적인 요구사항으로서 단지 유스케이스들의 흩어져 있는 모임이기 때문에 유스케이스의 시나리오들간에는 불일치가 발생할 수 있다[9]. 본 논문에서는 이러한 유스케이스 중심의 분석에서 발생하는 문제점을 보완하기 위하여 요구사항의 중복관계 및 불일치 분석을 이용하여 유스케이스 모델을 확장하는 방안을 제시하고자 한다.

3. 유스케이스 모델링을 위한 요구사항 분석 방안

3.1 불일치의 개념

광의적인 의미에서 불일치는 “2개의 기술정보 사이에 유효한 것으로 가정된 특정한 관계를 이 기술정보가 충족하지 못하는 상황”을 의미한다[22]. 본 논문의 목적은 소프트웨어 개발 초기단계에서 시스템의 행위와 구조의 근간이 되는 사용자 요구사항의 불일치를 분석하는데 있다. 따라서 본 논문에서는 불일치를 다음과 같이 제한적 범위에서 정의한다. 불일치란 “고객이 성취코자 하는 목표를 달성하기 위하여 사용자에게 의해 단언된 요구사항 명세들로부터 지켜져야 할 관계와의 논리적 모순이 유추되는 상황”을 의미한다. 본 논문에서 제안된 불일치

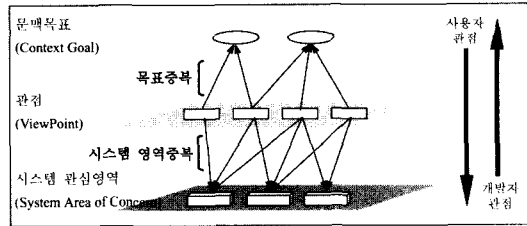


그림 1 중복측면에서 사용자 관점과 시스템간의 관계

의 정의에 의거하여 불일치 분석을 위한 두 가지 조건이 설정된다. 첫째는 불일치의 전제조건으로 사용자에게 의해 단언된 요구사항들이 관심영역 또는 시스템의 공통 측면에서 중복되어야 한다는 것이며[23], 둘째로 일관성 검사의 전제조건으로 사용자의 요구사항 사이에 지켜져야 하는 관계가 일관성 규칙으로 명확하게 표현되어야 한다는 것이다. 그림 1은 이러한 두 가지 조건을 고려하여 상위수준에서 사용자 관점과 소프트웨어 시스템간의 관계를 보여 준다.

“문맥목표”란 시스템이 궁극적으로 제공할 서비스를 식별한 것으로 비즈니스 목표를 만족하게 하는 수단이 된다. “관점”은 사용자가 특정 문제영역에 대하여 갖고 있는 이해와 시각을 대변한다[24]. 동일한 이해관계자가 독립된 다수의 관점을 갖고 있을 수 있기 때문에 관점은 특정 영역에 대한 부분적인 요구사항만을 나타내며, 중복여부에 따라 요구사항 불일치의 주체가 된다. “시스템 관심영역”은 요구사항을 시스템 모델측면에서 구조화하기 위해 요구사항 분석단계에서 정의되고 정제되는 소프트웨어의 기능군 또는 공통특성을 의미한다[25]. 이런 3가지 계층사이에는 “목표중복”과 “시스템 영역중복”의 2가지 중복구조가 성립한다. 이러한 분류는 비즈니스의 목표가 시스템의 기능으로 전환될 수 있도록 관점들간의 불일치를 관리하기 위한 접근이다. 전체 시스템에 대하여 모든 요구사항의 일관성을 검사하는 것은 많은 비용과 시간을 요구하는 작업이다[26]. 따라서 목표중복과 시스템 영역중복을 기준으로 관점을 분류하게 되면 특정 중복영역에 대해서 유효한 일관성 규칙을 적용하여 이 중복영역과 관계된 관점들의 요구사항에 대해서만 제한적으로 일관성을 검사하는 전략을 취할 수 있다.

3.2 관점의 명세화

개발초기 단계의 사용자 요구사항은 단지 부분적인 요구사항을 다루는 흩어져 있는 사용자 관점의 인스턴스 집합일 뿐이다. 따라서 잘 정의된 요구사항 명세는 전체 시스템 관점에서 통합되고 전환될 수 있도록 명확하게 구조화될 수 있는 형식을 갖추어야 한다. “관점 명세”는 시스템 행위를 규정하는 사용자의 기능적 요구사항을 구조화할 수 있도록 확장된 명세의 형식을 제안한

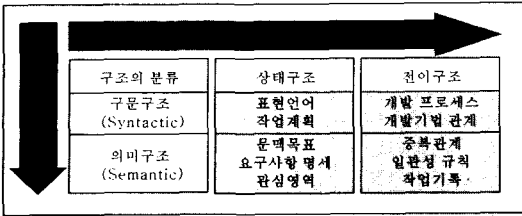


그림 2 요구사항 구조의 분류 및 구조화 지원요소

다. 그림 2는 요구사항의 구조화를 지원하기 위해 명세 형식의 요소를 추출하기 위한 요구사항 구조의 분류와 구조화 지원 요소를 보여준다.

구문구조는 요구사항을 표현하는 형식을 정의함으로써 비정형화된 사용자의 요구사항을 정형화된 표현으로 전환할 수 있도록 표현의 구조화를 지원한다. 의미구조는 요구사항이 규정한 시스템 행위의 논리적인 연관관계를 식별함으로써 일관된 요구사항 모델로 통합할 수 있도록 의미의 구조화를 지원한다. 구문구조와 의미구조는 현재의 요구사항 명세에 대한 의미적 연계성 및 구조관계를 나타내는 상태구조와 다음 단계로 전환되어야 할 개발 프로세스 및 소프트웨어 모델과의 연계성 및 구조관계를 나타내는 전이구조로 식별된다. 상태구조는 요구사항 명세의 개별형식을 결정하지만, 전이구조는 개별적으로 기술한 요구사항 명세로부터 유추되는 요구사항 간의 관계를 설정하는 것으로 요구사항 명세의 개별형식과는 분리되어 다루어진다. 요구사항의 상태구조에 따라 식별된 요구사항 명세 요소들을 포함하여 관점은 다음과 같은 형식으로 명세화하게 된다.

- 에이전트 : 요구사항을 제시하거나 기술하는 사용자 또는 사용자 그룹을 식별한다.
- 스타일 : 관점명세에 적용되는 표현언어 및 구문형식을 정의한다.

- 작업계획 : 다음단계의 개발 프로세스에서 전개해야 할 업무행위를 정의한다.
- 문맥목표 : 시스템이 제공해야 할 궁극적인 서비스를 규정하여 단문으로 표현한다.
- 요구사항 명세 : 사용자 관점에서 수행코자 하는 시스템의 요구사항을 기술한다.
- 관심영역 : 모델링 및 아키텍처 관점에서 중복되는 역할 및 기능군을 도출한다.
- 작업기록 : 요구사항 명세의 개발상태 및 변경이력, 변경의 사유를 포함한다.

요구사항 모델을 통합하기 위하여 관점의 명세형식에서 사용된 접근방식은 요구사항의 의미적 중복구조를 규명하는데 있다. 또한 스타일과 작업계획을 재사용될 수 있도록 사전에 정의하는 것은 서로 다른 관심영역의 상이한 요구사항 명세를 기술하는데 있어서 동일한 표현형식과 개발과정을 제시함으로써 요구사항 모델로 통합될 수 있는 개발환경을 구축하는 역할을 한다.

### 3.3 유스케이스 모델 확장

관점의 명세형식을 유스케이스로 확장하는 최초 단계는 관점의 스타일을 유스케이스 모델로 정의하고 사용자가 기술된 요구사항 명세로부터 액터와 유스케이스를 추출하는 것이다. 관점의 명세형식의 “관심영역” 요소는 유스케이스 모델의 스타일이 적용될 때 “액터”와 “시스템 기능영역”의 요소로 확장된다. 그러나 추출된 액터와 유스케이스들은 단순히 요구사항의 흩어져 있는 집합이며, 이를 일관된 요구사항의 모델로 통합해야만 시스템의 완전한 기능들을 추출할 수 있다. 그림 3은 관점 명세와 유스케이스와의 추상화된 대응관계를 도식화한 것으로 관점의 명세를 유스케이스 모델로 확장하기 위하여 요구사항의 관점이 유스케이스 집합과 어떠한 대응구조를 갖는지를 보여준다.

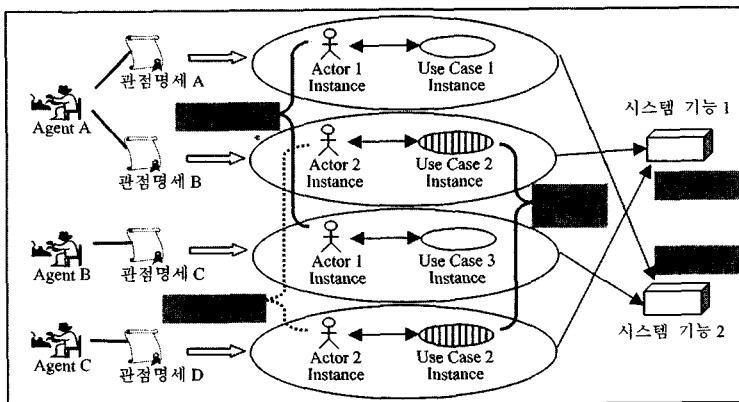


그림 3 관점 명세의 요소와 유스케이스 집합과의 대응관계

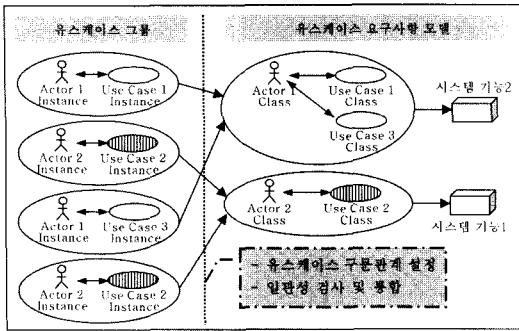


그림 4 유스케이스 집합과 유스케이스 모델간의 전환관계

액터, 유스케이스 및 기능 중복에서 나타난 바와 같이 관점명세에서 추출된 유스케이스 그룹은 사용자의 관점에서 부분적 요구사항을 다루고 있기 때문에 서로 중복될 뿐만 아니라, 시스템에서 제공하는 완전한 서비스 기능 중에서 특정상황에서 발생하는 유스케이스의 인스턴스만을 대변한다. 유스케이스 모델은 이러한 유스케이스 인스턴스들로부터 제공해야 할 서비스를 통합함으로써 유스케이스 클래스를 구성할 수 있을 때 완성된다. 그림 4는 이러한 요구사항 중복관계로부터 유스케이스 집합이 어떻게 유스케이스 모델로 전환되는지를 보여준다.

유스케이스 모델로 전환하기 위해서는 일관성 검사를 통한 유스케이스 클래스로의 통합과 유스케이스 모델링 구분규칙에 따른 유스케이스 클래스간 상호관계의 설정이 필요하다. 관심영역 중복구조의 규명은 유스케이스 인스턴스를 유스케이스 클래스로 통합하고 일관성 검사를 하기 위한 필수적인 전제조건이 된다. 그러나 유스케이스 시나리오의 의미적 중복관계에서 발생하는 논리적 불일치를 유스케이스 모델링 규칙으로 검사하는데는 한계가 있다. 따라서 관심영역에 중복된 유스케이스 시나리오들로부터 기술된 명제들의 어휘 중복관계를 통해 일관성 검사를 실시하는 과정이 제시되어야 한다.

### 3.4 유스케이스의 관심영역 중복구조 분석

일관성 검사는 유스케이스가 의미적으로 상호 중복되는 영역에 대해서만 유효한 일관성 규칙을 추출하여 관계된 유스케이스의 요구사항에 대해서만 제한적으로 적용하는 전략을 사용한다. 그림 5는 이러한 일관성 검사 전략을 적용하기 위하여 유스케이스의 의미구조에서 관심영역 중복관계를 도식화한 다이어그램을 보여준다.

문맥목표와 시스템 기능영역과의 중복에서 "Primary" 기능은 문맥목표를 달성하기 위하여 시스템을 직접 사용하는 액터들에 의해 수행되는 기본기능을 의미한다. "Extends" 기능은 기본기능으로부터 조건적으로 파생된 부가적인 기능의 유스케이스와 연계된다. "Secondary" 기능은 기본기능에 의해 수행된 작업을 제약하거나, 중

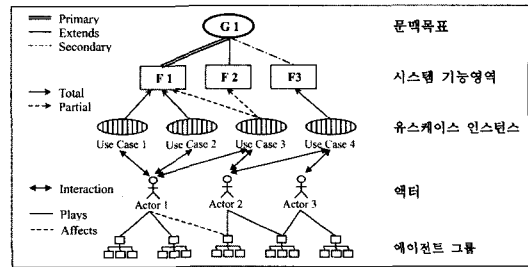


그림 5 유스케이스 관심영역 중복관계 다이어그램

합하는 관리적 목적에 의해 파생되는 지원기능이 된다. 유스케이스 인스턴스는 시스템 기능영역에서 독립적으로 수행이 가능한 "Total"로 표시된 완전중복과 연관된 시스템 기능영역들의 부분적 요구사항을 기술한 "Partial"로 표시된 부분중복으로 식별된다. 동일 기능영역에 중복된 유스케이스들은 그 의미적 유사성에 따라 하나의 유스케이스 클래스로 통합될 수 있으며, 다수의 기능영역으로 분산된 부분중복 유스케이스는 요구사항의 유사성에 따라서 다수의 유스케이스로 분리될 수 있다. 사용자는 실제 그 액터의 역할을 수행할 때 "Plays" 관계가 설정되며, 사용자가 그 액터의 역할을 수행하지 않지만, 사용자의 유스케이스로부터 이 액터의 유스케이스에 영향을 미칠 수 있는 요구사항이 존재할 때 "Affects" 관계를 설정한다.

그러나 많은 수의 유스케이스 인스턴스가 생성되는 경우 유스케이스 중복구조의 복잡도를 다루기 위해서는 체계적인 분석이 필요하다. 중복구조의 체계적 분석을 위한 그림 6의 유스케이스 기능전개도는 그림 5의 유스케이스 관심영역 중복관계 다이어그램을 매트릭스 구조로 전환함으로써 시스템의 기능영역에 중복되는 유스케이스 인스턴스들의 부분집합을 명확하게 식별하게 한다.

이 기능전개도를 통하여 액터와 액터의 인스턴스인 사용자를 분류하고 유스케이스의 시스템 기능영역에 대한 중복여부를 "TO"로 표시된 완전중복과 "PO"로 표시된 부분중복으로 식별하게 되며, 중복관계 상에 있는 시나리오의 의미적 유사성에 따라 동일한 유스케이스 클래스로 통합되거나 서로 다른 유스케이스 클래스로 분리하게 된다. 그림 6의 기능전개도에 대한 또 다른 접근방식은 유스케이스의 통합과 일관성 검사의 전개방향을 설정하는 것이다. 시스템의 구조는 시스템의 기본기능을 수행하는 액터에 의해 지배된다. 따라서 기본기능의 유스케이스로부터 시작하여 조건적 부가기능과 관리적 지원기능으로 요구사항 통합과 일관성 검사를 확대함으로써 시스템의 아키텍처가 가장 중요한 사용자인 기본기능 액터에게 초점을 맞출 수 있도록 할 수 있다.

Context Goal			문맥목표를 단문으로 기술한다.					
Relation To Goal			Primary	Extends		Extends	2ndary	
Function Deployment			Function A	Function B	Function C	Function D	Function E	Function F
Actor	Agent	Use Case						
A1	Usr 1	UC 1						
	Usr 2	UC 2		FO				
	Usr 4	UC 3						
A2	Usr 1	UC 4		FO		PO		
	Usr 3	UC 5					TO	
	Usr 5	UC 6		PO			PO	

유스케이스 정제 및 통합, 일관성 검사의 전개방향

그림 6 유스케이스 기능전개도

3.5 유스케이스 시나리오 어휘 중복관계 분석

유스케이스 시나리오의 불일치 분석은 일차 술어논리 (first-order predicate logic)를 사용한다. 요구사항 불일치 분석의 첫번째 단계는 유스케이스 기능전개도를 통해 의미적인 중복관계가 식별된 유스케이스들의 부분 집합에 대하여 관점 명세에서 스타일을 “일차 술어함수 전환”으로 설정함으로써 이루어진다. 요구사항의 논리적 불일치는 요구사항 어휘가 의미적으로 중복되었다는 전제조건 하에서 발생할 수 있는 결과이다. 따라서 정형식으로 전환된 요구사항 명세의 어휘에서 객체상수와 술어기호 간의 의미적 중복관계를 규명하고, 이를 일차 술어논리로 표현할 때 중복관계의 관점에서 시나리오간의 불일치를 검사할 수 있게 된다. 본 연구는 일관성 검사를 위해서 한 쌍의 시나리오 간에 유효한 관계 또는 일관성 규칙을 이용하는 정리증명 전략을 사용한다. 각각의 시나리오상에 존재하는 어휘는 각 어휘를 제한한 사용자의 관점과 인식을 대변하기 때문에 그 표현이 동일하다 하더라도 의미적으로 상이할 수 있다는 가정이 성립된다. 이러한 관점에서 요구사항 술어명세의 어휘간에 존재하는 중복관계는 “완전중복”, “비중복”, “포함중복”, “부분중복”의 4가지로 분석되며 다음과 같이 일차술어논리로 전환한다.

- 유스케이스 시나리오의 식별 : 동일한 표현기호의 경우 유스케이스 식별자에 의해  $U_{1p}(x)$ ,  $U_{2p}(x)$ 의 술어함수로 재기술함으로써 완전중복으로 해석되지 않도록 식별한다.

$$U_1 c_1 = U_2 c_2 \quad (1)$$

- 상수기호 완전중복 : 해석상 동치관계의 술어기호 완전중복은 식 (2)로 표현된다.

$$(\forall x)(U_1 p(x) \leftrightarrow U_2 q(x)) \quad (2)$$

- 상수기호 비중복 : 해석상 의미적 연관관계가 없는 비중복은 식 (3)으로 표현된다.

$$\neg(U_1 c_1 = U_2 c_2) \quad (3)$$

- 술어기호 비중복 : 두개의 술어기호 명제함수에 대하여 두 정형식을 동시에 만족하는 개체들의 튜플이 존재하지 않는 경우 비중복은 식 (4)와 같이 표현된다.

$$(\forall x)(\neg(U_1 p(x) \wedge U_2 q(x))) \quad (4)$$

- 술어기호 포함중복 : 하나의 술어함수를 만족하는 개체들이 다른 술어함수를 만족하고, 그 술어함수의 논역영역에 포함될 때 포함중복은 식 (5)로 표현된다

$$(\forall x)(U_1 q(x) \rightarrow U_2 p(x)) \quad (5)$$

- 술어기호 부분중복 : 두개의 술어함수를 동시에 만족하는 튜플이 존재하며 동시에 하나의 술어함수만을 만족하는 튜플이 존재할 때 부분중복은 식 (6)으로 표현된다.

$$(\exists x)((U_1 p(x) \wedge U_2 q(x)), (\exists x)((U_1 p(x) \wedge \neg U_2 q(x)), (\exists x)(\neg U_1 p(x) \wedge U_2 q(x))) \quad (6)$$

일차 술어논리로 중복관계를 설정하는 방식은 표현 재기술을 통한 단순화방식이 다를 수 없는 포함중복이나 부분중복을 다루게 됨으로써 요구사항의 불일치를 명확하게 유추할 수 있는 근거를 제시한다. 어휘의 중복관계의 규명은 요구사항을 기술한 사용자의 이해와 관점에 기반을 두기 때문에 사용자 그룹과 함께 요구사항 어휘의 중복관계에 대한 체계적인 분석할 수 있도록 그림 7과 같이 유스케이스 어휘 중복관계도를 사용한다.

어휘 중복관계도는 하나의 시나리오 내부의 어휘 중복관계를 규명한 “intra-overlap”과 한 쌍의 시나리오사이의 어휘 중복관계를 규명한 “inter-overlap”을 식별한다. 완전중복, 비중복, 포함중복, 부분중복은 각각 “=”, “≠”, “⊂” (또는 “⊃”), “∩”의 4가지 종류의 기호를 사용하며 어휘가 수학적 기호 또는 숫자로 동일한 도메인의 원소인 경우 상호간의 함수관계를 술어명세의 함수

**Intra-overlaps**

유스케이스 중복관계	목표 : 회사 지급정책 통제 액터 : 바이어 기능 : 대금지급기일 설정		
Use Case ID	U1		
사용자 그룹	A = {팀 1}, B={팀 2, 팀 3}		
$O_{AB}(U_1, U_1)$	어휘 A	어휘 B	어휘 C
어휘 A	=		
어휘 B		=	
어휘 C	$\subset$		=

**Inter-overlaps**

유스케이스 중복관계	목표 : 회사지급 정책 통제 액터 : 바이어 기능 : 대금지급기일 설정		
Use Case ID	Pair (U1, U2)		
사용자 그룹	A = {팀 1}, B={팀 2, 팀 3}		
$O_{AB}(U_1, U_2)$	어휘 A	어휘 D	어휘 E
어휘 A	$\neq$		
어휘 B		$\cap$	
어휘 C			=

그림 7 유스케이스 어휘 중복관계도

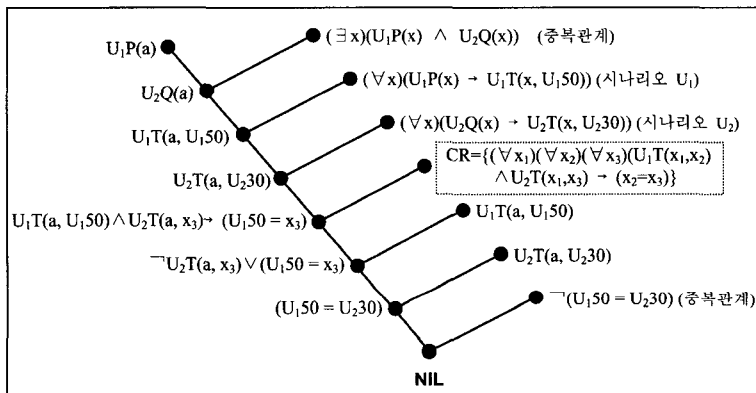


그림 8 증명트리를 통한 일관성 검사

상수로 정의할 수 있다. 일관성 검사가 적용될 범위를 정확히 설정하기 위하여 유스케이스 기능전개도에 유추된 문맥목표 중복, 액터 중복과 시스템 기능영역 중복 및 분석대상 시나리오를 지시하는 유스케이스 식별자를 기술하고, 중복관계를 유추하기 위해 협의해야 할 사용자 대상을 식별하여 시나리오 쌍 비교를 통한 일관성 검사전략을 실행한다.

**3.6 일관성 검사 및 불일치 분석**

본 연구의 일관성 규칙은 특정한 관심영역에서 유효한 지역적인 요구사항간의 관계를 지칭한다. 따라서 규명된 어휘의 부분중복 및 포함중복에 대하여 연관된 시스템 행위의 제약사항을 일관성 규칙으로 설정한다. 유스케이스 시나리오와 세 가지 중복관계를 일차 술어논리로 전환한 정형식 집합이 유추된 일관성 규칙CR의 부정으로 귀결된다면 이를 “모순(NIL)된다”라고 말한다. 일관성 검사는 부분중복 또는 포함중복 관계에 있는 두 개의 술어함수 변수를 두 정형식을 만족하는 객체상수로 치환하여 CR에 대한 모순이 유추되는지 증명함으로써 이루어진다. 그림 8은 술어기호  $P(x)$ 와  $Q(x)$ 가 부분중복관계에 있을 때 CR을  $\{(\forall x_1)(\forall x_2)(\forall x_3)(U_1T(x_1,$

$x_2) \wedge U_2T(x_1, x_3) \rightarrow (x_2 = x_3)\}$ 로 정의하여 적용한 일관성 검사과정을 증명트리로 보여준다.

이러한 일관성 규칙을 근거로 하여  $P(x)$ 의 변수  $x$ 를  $Q(x)$ 와 부분 중복관계가 성립하는 객체상수  $a$ 로 치환하고 증명트리로 일관성을 검사했을 때 사용자 그룹에 의해 식별된 중복관계  $\neg(U_{150} = U_{30})$ 가 일관성 규칙 CR과 모순되는 것을 보여준다. 동일한 객체상수에 대하여 서로 모순된 조건이 적용되고 처리됨으로써 발생하는 요구사항의 불일치는 시나리오에서 규정된 요구사항 명제만으로는 유추되지 않으며 술어기호  $P(x)$ 와  $Q(x)$ 의 중복관계를 규명했을때에만 도출될 수 있다.

**3.7 유스케이스 모델링을 위한 요구사항 분석공정**

유스케이스 모델링을 위한 요구사항 중복 및 일관성 분석은 관점명세로 기술된 문제영역 요구사항의 스타일을 유스케이스 모델로 확장하고, 목표와 관심영역을 중심으로 유스케이스의 의미적 중복구조를 분석하여 요구사항을 구조화하는 것으로부터 출발한다. 그림 9는 유스케이스 모델링을 위한 요구사항 분석공정과 산출물은 보여 준다.

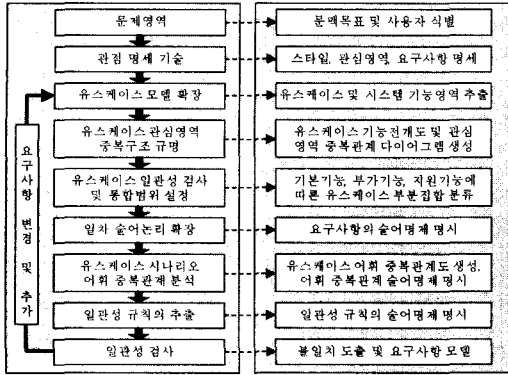


그림 9 유스케이스 모델링 요구사항 분석과정 및 산출물

요구사항 구조화를 통해 유스케이스 시나리오에 대하여 제한적으로 일관성 검사를 적용할 수 있도록 범위를 설정하게 되며, 설정된 범위 내에서 유스케이스 시나리오 어휘의 중복관계를 규명하고 일치 술어함수로 전환한 후 정리증명을 통해 논리적 불일치를 도출하는 분석과정을 거치게 된다. 이 과정을 통해 도출된 불일치 사항은 관점명세의 작업기록에 기록되고, 각 불일치 사항은 불일치 해결 프로세스에 따라 해결되거나, 추후 해결을 위하여 지속적으로 추적된다. 추가 요구사항 또는 사용자의 요구사항 변경은 기 생성된 유스케이스 모델에서 액터와 유스케이스를 적절히 변경하고 그림 9에서 제안된 공정과 같이 반복적 사이클로 일관성 검사 및 요구사항 통합과정을 수행하게 된다.

본 논문에서 제안한 유스케이스 모델링을 위한 요구사항 중복 및 불일치 분석방안은 요구사항 모델로의 통합 근거를 제공할 수 있도록 요구사항을 문맥목표와 관심영역과 연관시켜서 구조화하였다. 또한 이 구조를 기반으로 제한적 범위에서 일관성 검사가 적용될 수 있도록 유스케이스를 분류하고, 요구사항 어휘간에 존재하는 중복관계를 설정함으로써 요구사항 사이에 논리적 불일치를 도출하고 분석할 수 있는 방법을 제안하였다.

#### 4. 구매시스템의 요구사항 분석을 위한 예제 적용

##### 4.1 문제영역

본 논문에서 제시한 방안을 검증하기 위해 UN/CEFACT의 BPAWG가 참조모델로 제공한 공급사슬모델[27]을 선택하였다. 이 공급사슬모델은 공급사슬의 효과적인 관리를 제고하기 위한 업무 프로세스로서 그림 10과 같은 업무영역과 기능에 대한 구조를 갖고 있다. 이 중에서 납품대금 지급조건(Payment Terms) 설정기능의 요구사항 모델링이 본 논문의 적용 도메인이다. 납

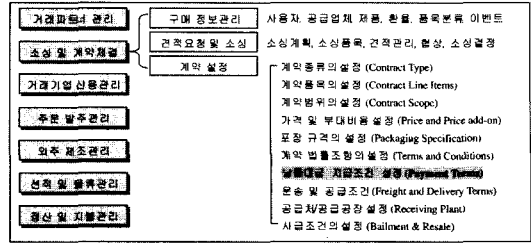


그림 10 공급사슬모델 업무영역 및 계약설정 기능구조

품대금 지급조건은 공급업체가 제품을 제공한 기일로부터 대금을 지급받는 시점까지의 기간으로 필수적인 계약요소가 된다. 납품대금 지급조건 설정기능은 구매기업의 납품대금 지급정책과 일치되어야 하며, 바이어가 공급업체의 종류에 의해 정확한 납품대금 지급기일을 계약서에 설정될 수 있어야 한다.

##### 4.2 유스케이스 모델 확장 및 관심영역 중복관계 분석

납품대금 지급조건 설정 기능에 대하여 관련된 사용자 그룹은 “원부자재팀”, “자재구매팀”, “자금팀”, “법무팀”의 4개 조직으로 식별된다. 이 조직과의 인터뷰를 통해 문맥목표가 “공급업체 종류에 따라 계약서에 정확한 대금지급기일이 결정되어야 하며, 대금지급기일 기준은 회사정책 및 법규에 의해 규제되어야 한다.”는 것으로 설정된다. 각 조직은 회사에서 설정된 제한된 업무영역에서 부분적인 요구사항만을 제시하고 있기 때문에 요구사항들이 서로 중복되거나 일관되지 않을 수 있다. 따라서 제한적인 사용자 중심의 관점을 요구사항 모델의 일관된 관점으로 전환시키기 위하여 부분적인 요구사항들을 구조화할 수 있는 근거를 분명하게 제시하여야 한다. 이러한 근거를 제시하기 위하여 그림 11은 관점명세 형식에 따라 기술된 자재구매팀의 요구사항을 보여준다.

<b>문맥목표</b>	공급업체 종류에 따라 계약서에 정확한 대금지급기일이 결정된다. 대금지급기일 기준은 회사정책 및 법규에 의해 규제된다.
<b>에이전트</b>	자재구매팀
<b>스타일</b>	자연어에 의해 사용자 요구사항 기술
<b>작업계획</b>	유스케이스 생성 (액터 및 유스케이스 추출) 및 세부 시스템 기능영역의 추출
<b>요구사항 명세</b>	자재구매 담당자는 시스템에서 부품입력 및 일반자재입력을 설정할 수 있고, 부품입력은 대금지급기일을 회사 규정에 따라 62 일을 입력하며 일반자재입력은 대금지급기일을 회사 규정에 따라 32 일을 입력한다. 최종 설정된 대금지급기일을 확인하고 계약서의 자재지급 조건을 확정한다.
<b>관심영역</b>	핵심 기능군: 계약설정 모듈 - 대금지급조건 설정 기능
<b>작업기록</b>	N/A

그림 11 “대금지급 조건 설정 서비스” 관점 명세서



액터	유스케이스 이름 (ID)	유스케이스 시나리오 주요 요구사항	사용자 그룹
바이어	원부자재 조건일력 (RawPay)	- 원부자재업체 대금지급 조건 일력 : 32 일 - 회사정책에서 벗어난 대금지급 조건 일력 : 별도의 결제 프로세스를 걸쳐 승인요청	원부자재팀
	부품/일반자재 조건일력 (PartPay)	- 부품업체 대금지급 조건 일력 : 62 일 - 일반자재업체 대금지급 조건 일력 : 32 일	자재구매팀
관리자	대금지급 정책설정 (TermPay)	- 직접자재업체 대금지급 조건 설정 : 62 일 - 간접자재업체 대금지급 조건 설정 : 32 일	자금팀
	대금지급 규제설정 (CtlTerm)	- 하도급업체 대금지급 조건 설정 : 60 일 이하	법무팀
서비스 세부기능 (기능 ID)	지급기일 결정 (PayDecide)	바이어는 공급업체 종류에 의해 대금지급기일 결정	
	지급기일 승인 (PayApprove)	회사정책에 벗어난 지급조건은 별도 승인 요청	
	지급정책 관리 (PayControl)	회사정책에 따른 지급조건의 사전 설정	

그림 12 문제영역의 유스케이스 및 시스템 기능영역 추출

문맥목표		공급업체 종류에 따라 계약서에 정해진 대금지급기일이 결정된다. 대금지급기일 기준은 회사정책 및 법규에 의해 규제된다.		
목표와의 관계		Primary	Extends	Secondary
기능 전체			PayApprove	PayControl
액터	사용자그룹 유스케이스		PP	PO
바이어	원부자재팀 RawPay			
	자재구매팀 PartPay			
관리자	자금팀 TermPay			
	법무팀 CtlTerm			

그림 13 문제영역의 유스케이스 1차 기능전개도

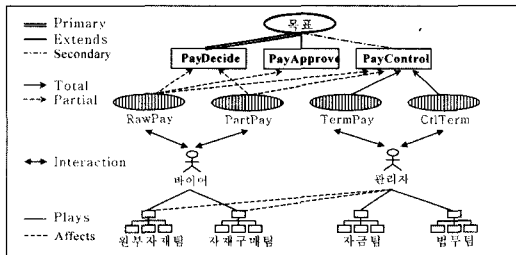


그림 14 문제영역의 Use Case 중복관계 다이어그램

설정된 각 사용자그룹의 관점명세서를 근거로 유스케이스를 추출한다. 그림 12는 개별 사용자그룹의 부분적 요구사항에서 추출한 액터, 유스케이스 시나리오의 핵심 요구사항, 각 유스케이스 시나리오를 통하여 시스템에서 사용해야 할 시스템 기능영역을 세분화한 결과를 보여준다. 유스케이스 모델 확장은 요구사항의 구조를 보다 명확하게 보여준다. 개별 사용자의 관점은 시스템과 상호 작용하는 공통된 역할로서 액터의 관점으로 전환된다. 4개의 조직의 요구사항은 동일한 액터의 관점에서 서로 일관되어야 하며, 업무의 유사성에 따라 단일 요구사항 모델로 통합될 수 있는지 검토되어야 한다. 그림 12에서 추출된 액터, 유스케이스 및 시스템 기능영역은 요구사항 모델에서 구조적 관계를 규명하기 위하여 유스케이스 기능전개도의 입력물로 사용된다.

그림 13과 그림 14는 “대금지급조건 설정 서비스” 요

구사항 모델링의 기반으로 사용하기 위해 분석된 유스케이스의 기능전개도와 관심영역 중복관계 다이어그램을 보여준다. 기능전개도 분석의 결과로 추출된 유스케이스 요구사항이 다수의 시스템 기능영역에 분산되어 있으며, 동일한 시스템 기능영역에 다수의 유스케이스가 중복되어 있음을 알 수 있다. 이러한 요구사항의 중복구조를 통하여 모델링 관점에서 요구사항이 어떻게 통합되고, 구조화되어야 하는지를 보여준다.

문제영역의 기본기능은 대금지급기일 결정 “Pay-Decide” 기능이다. 이 기능은 “대금지급조건 설정 서비스”의 기본적인 행위를 결정하기 때문에 우선적으로 정제되고 통합되어야 한다. 정제된 기본기능의 요구사항을 바탕으로 추가적인 기능인 “PayApprove” 기능과 “Pay-Control” 기능으로 일관성 검사 및 요구사항 통합을 확대해 나간다.

### 4.3 유스케이스 어휘의 중복관계 분석 및 일관성 검사

4.2에서 시스템 기능영역을 기준으로 식별된 유스케이스 부분집합은 유스케이스 어휘의 중복관계 분석 및 일관성 검사의 입력물로 사용된다. 첫번째 분석대상으로 기본기능 액터인 바이어에 의해 사용되는 기본기능 “PayDecide”에 중복되는 유스케이스 부분집합 “RawPay”와 “PartPay”가 추출된다. 이 유스케이스는 관점명세 형식의 스타일을 일차 술어논리로 정의함으로써 술어함수의 정형식 집합으로 전환된다. 그림 15는 이러한 일차 술어논리의 전환결과를 보여준다.

RawPay (RP)	- 원부자재 업체이면 대금지급기일을 32일로 일력한다. → $\{ ( \forall x )( RP \text{ RawSupplier}(x) \rightarrow RP \text{ PayTerm}(x, RP32) ) \}$
PartPay (PP)	- 부품업체이면 대금지급기일을 62일로 일력한다. → $\{ ( \forall x )( PP \text{ PartSupplier}(x) \rightarrow PP \text{ PayTerm}(x, PP62) ) \}$ - 일반자재업체이면 대금지급기일을 32일로 일력한다. → $\{ ( \forall x )( PP \text{ GenSupplier}(x) \rightarrow PP \text{ PayTerm}(x, PP32) ) \}$

그림 15 “PayDecide”기능의 유스케이스 일차 술어논리 전환

유스케이스 시나리오에 사용된 어휘간에는 중복관계가 존재한다. 중복관계는 단일 요구사항 명세 내부에서도 발생하게 때문에 완전한 일관성 검사를 위해서는 단일 요구사항 명세의 “intra-ovlap”분석이 요구사항 명세 쌍의 “inter-overlap” 분석에 선행되어야 시행되어야 한다. 그림 16은 유스케이스 어휘 중복관계도를 이용하여 “PayDecide” 세부기능에 중복된 유스케이스들의 중복관계 분석결과를 보여준다.

“intra-ovlap”분석에서 서로 다른 어휘간에는 비중복관계만 존재하지만, 요구사항 명세 쌍(RawPay, PartPay)의 “inter-overlap” 분석에서는 원부자재 업체가 부품업체 및 일반자재 업체와 부분중복 관계에 있음을 식별할 수 있다. 이것은 원부자재 업체이면서 동시에 부품

Intra-overlaps									
유스케이스 중복관계	목표: 대금지급기일의 정확한 설정 및 회사정책/법규에 의한 규제 액터: 바이어 / 기능: PayDecide								
Use Case ID	RawPay (RP)			Use Case ID	PartPay (PP)				
사용자그룹	A = {원부자재팀}, B = {자재구매팀}			사용자그룹	A = {원부자재팀}, B = {자재구매팀}				
$O_{AB}(RP, RP)$	RawSupplier	PayTerm	32	$O_{AB}(PP, PP)$	PartSupplier	GenSupplier	PayTerm	62	32
RawSupplier	=			PartSupplier	=	=			
PayTerm		=		GenSupplier			=		
32			=	PayTerm				=	
				62				=	
				32					=

Inter-overlaps					
Use Case ID	Pair (RawPay, PartPay)				
사용자그룹	A = {원부자재팀}, B = {자재구매팀}				
$O_{AB}(RP, PP)$	PartSupplier	GenSupplier	PayTerm	62	32
RawSupplier	$\cap$	$\cap$			
PayTerm			=		
32				*	=

그림 16 "PayDecide" 어휘 중복관계도

요구사항 정형식의 전체집합	f(1) $(\forall x)(RP.RawSupplier(x) \rightarrow RP.PayTerm(x, RP.32))$ f(2) $(\forall x)(PP.PartSupplier(x) \rightarrow PP.PayTerm(x, PP.62))$ f(3) $(\forall x)(PP.GenSupplier(x) \rightarrow PP.PayTerm(x, PP.32))$ f(4) $(\exists x)(RP.RawSupplier(x) \wedge PP.PartSupplier(x))$ f(5) $(\exists x)(RP.RawSupplier(x) \wedge PP.GenSupplier(x))$ f(6) $(\exists x)(\neg RP.RawSupplier(x) \wedge \neg PP.PartSupplier(x))$ f(7) $(\exists x)(\neg RP.RawSupplier(x) \wedge \neg PP.GenSupplier(x))$ f(8) $(\exists x)(\neg RP.GenSupplier(x) \wedge \neg PP.GenSupplier(x))$ f(9) $(\exists x)(\neg RP.RawSupplier(x) \wedge \neg PP.GenSupplier(x))$ f(10) $(\forall x_1)(\forall x_2)(RP.PayTerm(x_1, x_2) \rightarrow PP.PayTerm(x_1, x_2))$ f(11) $\neg(RP.32 = PP.62)$ f(12) $(RP.32 = PP.32)$
일관성 규칙	$CR = \{(\forall x_1)(\forall x_2)(\forall x_3)(RP.PayTerm(x_1, x_2) \wedge PP.PayTerm(x_1, x_3) \rightarrow (x_2 = x_3))\}$

그림 17 "PayDecide"기능의 정형식 집합과 일관성 규칙의 추출

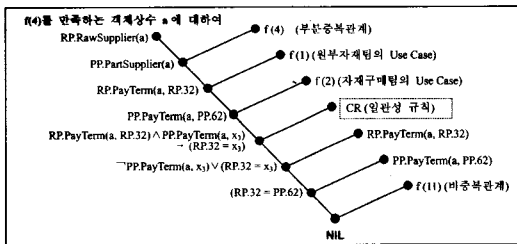


그림 18 "PayDecide" 일관성 검사

업체 또는 일반자재업체인 객체상수에 대하여 적용되는 요구사항 간에 불일치가 존재할 가능성이 있음을 의미한다.

그림 17과 그림 18은 "PayDecide"기능의 유스케이스 요구사항을 정형식의 집합으로 전환되고, 부분중복관계에 따라 추출된 일관성 규칙 CR을 통해 자동화되는 일관성 검사과정을 보여준다. 부품업체와 원부자재업체의 부분중복관계에 따라 부품업체이면서 동시에 원부자재

업체인 객체상수의 경우에는 반드시 대금지급조건이 동일해야 한다는 일관성 규칙이 설정된다. 일관성 검사는 부분중복관계의 정형식을 참으로 하는 객체상수로 변수를 사례화함으로써 이루어진다. 이 일관성 검사결과 부품업체이면서 동시에 원부자재 업체인 객체 "a"에 대하여 서로 다른 대금지급기일이 결정되는 모순 "NIL"이 귀결됨으로써 불일치를 도출하고 있다.

요구사항의 불일치는 중복되는 도메인을 갖는 용어의 정의로부터 발생한다. 따라서 요구사항 분석에 참여한 사용자 그룹과 함께 공통어휘 정의를 통하여 원부자재 업체를 상위 분류 개념인 부품업체를 포함한 직접자재 업체와 일반자재업체로 구분함으로써 그림 19와 같이 해결된다. 이 결과로 요구사항 명세 쌍(RawPay, PartPay)의 대금지급조건 요구사항은 유스케이스 "MatlPay"로 통합되고, "RawPay"의 예외조건 승인요청 요구사항은 "MatlPay"의 대안적인 부가기능으로서 유스케이스 "DeviPay"로 분리된다.

#### 4.4 "대금지급조건 설정 서비스" 유스케이스 모델링

그림 19에서 보는 바와 같이 요구사항 어휘의 중복관계에 근거한 일관성 검사는 요구사항 통합의 수단으로 사용되며, 요구사항 통합의 결과는 기능전개도를 통하여 검토된다. 그림 20은 문제영역이 최종적으로 일관되고 완결된 유스케이스 모델로 통합되는 과정을 보여준다.

4.3에서 보여준 일관성 검사와 요구사항 통합과정은 그림 20의 "1차 기능 통합"에서 "2차 기능 통합"으로 전개되는 과정을 기술한 것이다. 4.3에서 전개된 동일한 방법에 따라 순차적으로 "PayControl"기능에 중복된 유스케이스 "TermPay"와 "CtdTerm"가 통합되어 회사의 대금지급정책과 법규에 따라 지급조건을 규제하는 유스

액터	사용자그룹	유스케이스 시나리오 주요 요구사항	유스케이스 이름 (ID)
바이어	원부자제된 자제구버터	직접자제업체 대금지급 조건 : 62 일 일반자제업체 대금지급 조건 : 32 일	대금지급 조건일력 (MatlPay)
		회사정책에서 벗어난 대금지급 조건 : 별도 결제프로세스를 거쳐 승인 요청	예외조건 승인요청 (DeviPay)

그림 19 "PayDecide" 불일치 해결 및 통합

케이스 "RulePay"가 생성된다. 다음 단계로 "Pay-Decide"기능과 "PayControl"기능 사이의 일관성 분석 및 통합과정을 통해 "MatlPay"가 2차 기능으로 중복된 부분적인 요구사항에 대하여 발생할 수 있는 불일치를 해결하게 되면 "대금지급조건 설정 서비스"의 유스케이스 요구사항 모델을 획득하게 된다. 최종 유스케이스 요구사항 모델은 대금지급조건 설정의 기본기능을 규정하는 "MatlPay"와 이 유스케이스가 대금지급조건을 입력 시 사전 규제된 제약조건을 사용하도록 규정하는 "RulePay" 그리고 규정된 제약조건에 충족하는 대금지급조건이 없는 경우 대안적인 별도승인과정의 기능을 규정하는 "DeviPay"로 구성된다. 최종 유스케이스 기능 전개도를 통해 최초에는 분산된 유스케이스 인스턴스들이 하나의 기능으로 완전 중복된 개별 유스케이스 클래스로 통합되고, 각 기능은 문맥목표를 달성하기 위하여 기본기능, 부가기능, 관리기능으로 구조화되어 있음을 알 수 있다. 이러한 유스케이스 모델은 그 구조 측면과 요구사항에서의 논리적 일관성 측면에서도 완전한 요구사항 모델 역할을 하게 된다.

4.5 비교분석 및 평가

유스케이스 다이어그램을 적용한 UML(Unified Modeling Language)은 객체지향 개발프로세스에 있어서

산업표준으로 인정되는 모델링 언어를 정의하고 있다 [28]. 그러나 UML 표준은 구체적인 모델링 프로세스를 제공하지 않고 있기 때문에 개발자의 임의적 추상화 레벨 설정과 자의적인 유스케이스 기술방식으로 인하여 유스케이스 다이어그램을 생성하는데 많은 어려움을 갖고 있다. 일반적인 유스케이스 모델링 접근방식은 Jacobson에 의해 일반화된 "시나리오 근간"의 접근방식이다 [10]. "시나리오 근간"의 접근방식은 사용자의 입장에서 시스템의 행위를 선형적 시퀀스로 기술하고, 이를 통하여 개별 유스케이스를 식별한다는 측면에서 고객중심의 기능적 요구사항을 추출하는데 강력한 방법으로 사용되고 있다. 시나리오 근간의 정형화된 유스케이스 다이어그램 추출규칙[29]은 시나리오를 근간으로 규정된 시스템 행위의 중복성을 분석하여 유스케이스 추출, 통합, 일반화 및 구체화를 다루는데 매우 유용하다. 그러나 시나리오 근간의 접근방식은 사용자에 의해 제시된 시스템 행위의 제한된 경로에 집중되어 있기 때문에 시스템 행위의 융통성이 제약되어 요구사항이 불안정하고, 복수의 사용자 관점 충돌로 인하여 요구사항을 구조화하는데 한계를 갖고 있다. 또한 요구사항 통합을 정당화 할 수 있는 요구사항 중복에 대한 명백한 근거 및 기준을 제공하지 않기 때문에 통합과정에서 요구사항의 일관성을 달성하기 어렵다는 단점을 갖고 있다. 요구사항이 목표로 잘 구조화될 때 갖게 되는 구조적 완전성[8]을 지원하기 위하여 강화된 다른 방식이 "목표지향" 접근방식이다[9,14]. 이 접근방식은 개별 요구사항보다 안정적이며 복수의 사용자의 관점을 통합할 수 있는 목표의 계층구조를 이용하여 시나리오를 근간으로 유스케이스 집합을 구조화함으로써 보다 안정적인 요구사항 구조를

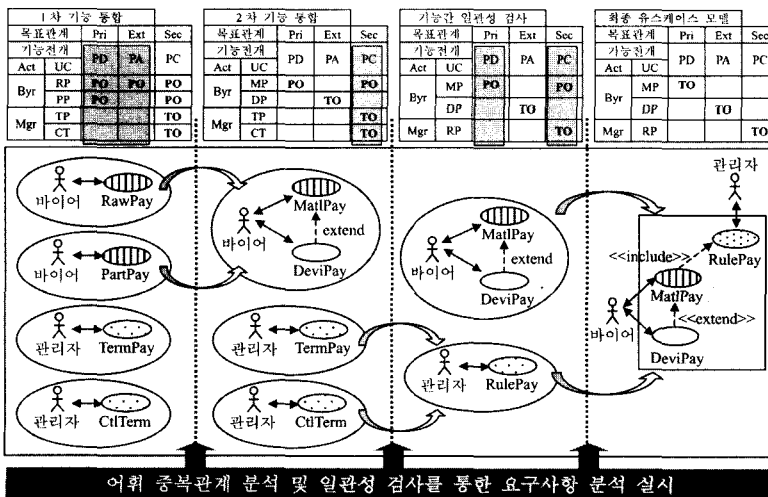


그림 20 요구사항 일관성 검사 및 유스케이스 모델 통합 과정

표 1 요구사항 모델의 효과성에 근거한 기존 연구의 비교

접근방식	모델링 근거	구조적 완전성	논리적 일관성
시나리오 근간	사용자 관점에서 시스템 행위의 선형적 시퀀스에 근거한 요구사항 추출	시나리오 문맥에 제약된 유스케이스 관계 설정, 다수의 경로 존재시 요구사항 불안정화	유스케이스 통합을 위한 요구사항 중복 근거 제시 취약, 일관성 관리방안 결여
목표 지향	비즈니스 도메인의 계층목표에 근거한 요구사항의 구조화 및 추출	계층목표와 대응되는 유스케이스 구조 설정, 상위수준 목표에 근거하여 요구사항이 안정적	목표중복 근거의 불일치 분석, 사용자 해석 근거의 요구사항 문맥 일관성 분석 취약
공통용어 근간	사용자 해석에 근거한 요구사항 어휘의 표준화 및 요구사항 정제	적용되는 요구사항 모델링 기법을 지원하는 부가적 수단, 모델링 기법의 효과에 의존적	어휘 완전중복 및 비중복 식별, 단순 사용자 의사소통에 의존, 문맥 일관성 분석 취약
관점 지향	다중 관점의 메서드 관계규칙에 근거한 일관성 검사 및 요구사항 통합	구조적 완전성은 메서드 관계규칙에 적용되는 모델링 기법의 효과에 의존적, 불안정화	표현 구문규칙에 근거한 일관성 검사, 요구사항 문맥에 대한 불일치 분석은 제한적
중복 근간	요구사항 어휘 중복관계에 근거한 일관성 감사 및 요구사항 통합	요구사항 구조화 방안 미제시, 시스템의 규모 및 범위, 복잡성에 따른 적용효과 유동적	사용자 해석에 근거한 문맥의 일관성 규칙 정형화, 정리증명의 일관성 검사 자동화

생성할 수 있다. 이 접근방식은 목표중복에 근거하여 유스케이스간의 목표 불일치를 다룰 수 있는데 반하여, Anton이 지적한 바와 같이 개별 유스케이스의 상세한 문맥정보의 결여로 인하여 요구사항의 논리적 불일치를 적절히 다루지 못함으로써 요구사항의 유효성을 상실할 위험을 갖고 있다[30].

“시나리오 근간” 및 “목표 지향”의 두 가지 방식 모두는 액터, 유스케이스 및 유스케이스간의 관계를 추출하기 위한 구조적 관점에 집중되어 있기 때문에 다중 사용자관점 하에서 요구사항 통합에 따른 논리적 일관성을 보장하지 못하는 단점을 갖고 있다. 논리적 일관성을 확보하기 위하여 이 두 가지 접근방식을 보완하는 전통적인 수단은 “공통용어(Glossory) 근간”의 요구사항 정제이다[25,31]. Kurt Bittner는 시스템 행위를 명확하게 도출하기 위해 세부적인 시퀀스를 기술할 때 발생하는 추상화 수준의 복잡성 및 어휘중복에 따른 불일치를 다루는데 “공통용어(Glossary)”의 유용성을 강조하고 있다[31]. 그러나 공통용어에 근거한 요구사항 재기술은 동일한 표현기호를 갖는 어휘를 완전중복만으로 간주하기 때문에 동음이의어, 유사어가 존재하는 이질적인 요구사항 명세의 단순한 재기술에서조차 그 요구사항 중복 및 불일치를 적절하게 다룰 수 없는 취약점이 발생한다[16]. 요구사항 모델링에 있어서 다중사용자 관점의 일관성 검사방법으로 본 논문에서 검토된 접근방식은 Finkelstine이 제안한 “관점(Viewpoint)지향”의 접근방식[24]과 Spanoudakis가 제안한 “중복(Overlap)근간”의 접근방식[16]이다. 관점지향의 접근방식은 다중 사용자 관점의 통합을 위하여 요구사항 표현기법과 모델링언어를 재사용이 가능한 메서드로 정의하고 메서드간의 관계를 표현하는 구문규칙을 일관성 검사방법으로 사용한

다. 그러나 이 방법의 일관성 보장은 메서드로 규정된 구문규칙의 불일치 해결능력에 의존하고 있기 때문에, 일관성 규칙이 불안정하며 구문규칙에 의존한 일관성 검사는 사용자의 상이한 해석에 의해 발생하는 요구사항 문맥의 논리적 불일치를 다루는데 한계를 갖고 있다 [24]. 반면에 “중복근간”의 접근방식은 요구사항 어휘의 중복관계를 사용자의 해석에 근거하여 분석하고 일치시킬 수 없는 어휘의 정형방법론을 일관성 검사방법으로 사용하기 때문에 요구사항 문맥의 논리적 일관성을 확보하는데 유용하다. 그러나 이 방식은 상위관점에서 요구사항의 구조화를 다루지 않고 있기 때문에 큰 규모의 소프트웨어 개발에 적용될 때 지역적인 정리증명 방식의 일관성 검사범위가 광범위하게 증가하여 시스템 전체 요구사항의 일관성을 효과적으로 검사하는 것이 불가능해지는 단점을 보여준다.

표 1의 비교분석에서 보여주는 바와 같이 기존 연구는 요구사항 모델링의 구조적 완전성과 논리적 일관성의 두 속성 중에 어느 하나에 중심을 두고 있다. 반면에 본 논문에서 제안된 유스케이스 모델링 접근방식은 목표지향의 기능 구조분석을 사용자의 해석에 근거한 일관성 검사과정으로 연계함으로써 일관성 검사를 다중 사용자의 관점 통합의 수단으로 사용한다. 특히 기능 구조분석이 일관성 검사 범위와 요구사항 통합 우선순위를 설정하는 수단으로 사용됨으로써 많은 시간과 노력을 요하는 요구사항 쌍간의 정리증명 방식을 효율적으로 사용토록 제안하고 있다. 유스케이스 모델링 과정에서 다중 사용자 관점의 논리적 모순을 제거하는 것은 개발 초기 단계의 요구사항 불일치가 설계 및 구현단계로 이전되지 않도록 안정적인 요구사항 모델을 획득하는데 중요한 역할을 한다.

5. 결론

일차술어논리와 요구사항 중복관계를 중심으로 한 정형방법론은 수학적 증명을 통해 요구사항의 논리적 불일치를 식별하고, 일관성 검사를 자동화할 수 있는 수단을 제공한다. 그러나, 소프트웨어 아키텍처를 구성하는데 중요한 요구사항 모델을 구조화할 수 있는 방안을 명확하게 제시하지 못하고 있다. 반면에 유스케이스 중심의 분석은 사용자 관점을 중심으로 한 요구사항 분석을 통하여 요구사항 모델링의 복잡성을 해소할 수 있다는 장점을 갖고 있다. 그러나 사용자의 제한적 관점에 의해 중복되는 요구사항 간의 논리적 충돌을 도출하기 어렵다는 단점을 갖고 있다. 이러한 문제점을 보완하기 위해 본 논문에서는 요구사항 중복 관계에 근거한 요구사항 일관성 검사를 요구사항 통합 수단으로 사용하여 유스케이스 모델로 구조화할 있는 방안을 제안하였다. 본 논문에서 제시된 방안은 다중 사용자의 관점을 문맥목표와 시스템 관심영역의 중복개념을 사용하여 구조화 할 수 있는 근거로서 관점명세 형식을 제안하였다. 또한 유스케이스 기능전개도를 제시함으로써 요구사항의 중복구조를 명확하게 분석하고, 요구사항 일관성 및 통합을 전개해야 할 방향을 도출할 수 있는 수단을 제공한다. 그리고 어휘 중복관계의 규명과 일관성 검사공정은 유스케이스 기능전개도 분석과 결합하여 요구사항 모델로 통합할 수 있는 방안을 제시한다. 본 논문의 제안방안을 적용한 사례분석은 다중 사용자 환경에서 서로 상충되는 요구사항들이 요구사항 모델 통합을 위한 구조분석을 통하여 논리적 일관성 검사과정으로 연계됨으로써 요구사항 모델 구조화와 요구사항 일관성 확보라는 요구사항 모델 개발목표를 동시에 지원할 수 있음을 보여준다.

본 연구에서 제시된 술어논리의 정리증명 방안은 다수의 복잡한 요구사항 명세에 적용될 때 많은 비용과 시간을 소요하는 작업이다[26]. 따라서 요구사항의 중요도 및 비용 효과분석을 통하여 효율적인 일관성 검사전략을 선택할 수 있는 방안이 연구되어야 한다. 이러한 향후과제의 연구는 소프트웨어 개발 초기의 요구사항 추출 및 분석단계에서 비용 효율적으로 요구사항 불일치를 관리하고, 일관성을 획득하는 필수적인 요소가 될 것이다.

참고 문헌

[1] The Standish Group, "The CHAOS Report (1994)," [http://www.standishgroup.com/sample\\_research/chaos\\_1994\\_1.php](http://www.standishgroup.com/sample_research/chaos_1994_1.php), 1994.  
 [2] F. Brooks, "No Silver Bullet ; Essence and Acci-

dents of Software Engineering," *Computer*, Vol. 20, No. 4, pp. 10-19, 1987.  
 [3] R.M. Tong, L.A. Appelbaum, V. N. Askman, and J. F. Cunningham, "Conceptual Information Retrieval Using RUBIC" in *Proc. of 10th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, Louisiana, pp.247-253, 1987.  
 [4] B. Mynatt, *Software Engineering with Student Project Guidance*, Prentice Hall : Englewood Cliffs, NJ, 1990.  
 [5] S. Park, H. Kim, Y. Ko, and J. Seo, "Implementation of an Efficient Requirements-Analysis Supporting System Using Similarity Measure Techniques," *Information and Software Technology*, Vol. 42, No. 6, pp. 429-438, 2000.  
 [6] S. Jones, D. Till, and A. Wrightson, "Formal Methods and Requirements Engineering ; Challenges and Synergies," *Journal of Systems and Software*, Vol. 40, No. 3, pp. 263-273, 1998.  
 [7] A. Davis and D. Leffingwell, "Using Requirements Management to Speed Delivery of Higher Quality Applications," Rational Software Corp., Technical Report, <http://www.rational.com/media/whitepapers/696wp.pdf>, 1995.  
 [8] K. Yue, "What Dose It Mean to Say that a Specification is Complete?" in *Proc. IWSSD-4, 4th Int. Workshop on Software Specification and Design*, Monterey, CA, IEEE Computer Society Press, pp.42-49, 1987.  
 [9] 이재호, 김재선, 박수용, "유스케이스 모델링을 위한 시나리오 근간의 목표지향 분석 방안", *정보과학회논문지 소프트웨어 및 응용*, 29권, 4호, pp. 211-224, 2002.  
 [10] I. Jacobson, *Object-Oriented Software Engineering : A Use Case Driven Approach*, Addison-Wesley : Reading, MA, 1992.  
 [11] G. Booch, *Object-Oriented Analysis And Design with Applications*, Benjamin/Cummings : Redwood City, CA, 1994.  
 [12] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-Oriented Modeling and Design*, Prentice-Hall : Englewood Cliffs, NJ, 1991.  
 [13] A. Hunter and B. Nuseibeh, "Analyzing Inconsistent Specifications," in *Proc. of Third IEEE International Symposium on Requirements Engineering (RE'97)*, Annapolis, MD, IEEE Computer Society, pp.78-86, 1997.  
 [14] A. Dardenne, A. van Lamsweerde and S. Fickas, "Goal-directed Requirements Acquisition," *Science of Computer Programming*, Vol. 20, No. 1-2, pp. 3-50, 1993.  
 [15] G. Spanoudakis and A. Finkelstein, "Reconciling Requirements : A Method for Managing Interference, Inconsistency and Conflict," *Annals of Soft-*

- ware Engineering, Vol. 3, No. 1, pp. 433-457, 1997.
- [16] G. Spanoudakis, A. Finkelstein, and D. Till, "Overlaps in Requirements Engineering," *Automated Software Engineering*, Vol. 6, No. 2, pp. 171-198, 1999.
- [17] M. Heimdahl and N. Leveson, "Completeness and Consistency Analysis of State-Based Requirements," *IEEE Trans. on Software Engineering*, Vol. 22, No. 6, pp. 363-377, 1996.
- [18] M. Fowler, *UML Distilled: Applying the Standard Object Modeling Language*, Addison-Wesley : Reading, MA, 1998.
- [19] 서정훈, 강교철, "소프트웨어 요구공학에서 취척 기반의 운영 모델 타당성 검사", 석사논문, 포항공과대학교 정보통신대학원, 1998.
- [20] A. Finkelstein, G. Spanoudakis, and D. Till, "Managing Interference," in *Joint Proc. of Viewpoints 96 : An International Workshop on Multiple Perspectives in Software Development*, ACM Press : San Francisco, pp. 172-174, 1996.
- [21] A. Finkelstein, D. Gabbay, A. Hunter, J. Kramer, and B. Nuseibeh "Inconsistency Handling in Multi-perspective Specifications," *IEEE Trans. on Software Engineering*, Vol. 20, No. 8, pp. 569-578, 1994.
- [22] 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.
- [23] G. Spanoudakis and H. Kim "Diagnosis of the Significance of Inconsistencies in Object-Oriented Designs : A Framework and its Experimental Evaluation," *Journal of Systems and Software*, Vol. 64, No. 1, pp.3-22, 2002.
- [24] B. Neuseibeh, A. Finkelstein, and J. Kramer, "Method Engineering for Multi-perspective Software Development," *Information and Software Technology*, Vol. 38, No. 4, pp. 267-274, 1996.
- [25] 황만수, 이원우, 류성렬, "CSCW환경에 기반한 요구공학 프로세스모델 설계", *한국정보처리학회 논문지*, 7 권, 10호, pp. 3075-3085, 2000.
- [26] B. Nuseibeh, S. Easterbrook, and A. Russo, "Making Inconsistency Respectable in Software Development," *Journal of Systems and Software*, Vol. 58, No. 2, pp. 171-180, 2001.
- [27] United Nations Center for Trade Facilitation and Electronic Business (UN/CEFACT), Business Process Analysis Working Group, "International Supplier Chain Reference Model (BP044)," [http://www.unece.org/cefact/docum/sessdocs/bawg\\_0398.htm](http://www.unece.org/cefact/docum/sessdocs/bawg_0398.htm), 2001.
- [28] OMG, "OMG Unified Modeling Language Specification Version 1.5," OMG formal document, <http://www.omg.org/technology/documents/modeli>  
*ng\_spec\_catalog.htm*, 2003.
- [29] 유철중, 정소영, "요구사항 기술서로부터 유스케이스 다이어그램 추출기법", *한국정보처리학회 논문지 D*, 9-D권, 4호, pp.639~650, 2002.
- [30] A. Antón, R. Carter, A. Dagnino, J. Dempster and D. Siege, "Deriving Goals from a Use Case Based Requirements Specification," *Requirements Engineering Journal*, Springer-Verlag, Vol. 6, pp. 63-73, 2001.
- [31] K. Bittner, "Managing Use-Case Details," in the Rational Edge of IBM, white paper, [http://www.therationaledge.com/content/apr\\_01/t\\_usecase\\_kb.html](http://www.therationaledge.com/content/apr_01/t_usecase_kb.html), 2001.

#### 최진재

1996년 2월 서강대학교 화학과 졸업  
2004년 2월 서강대학교 정보통신대학원 공학석사 취득. 1996년~2002년 대우자동차주식회사 구매본부 대리. 2002년 10월~현재 지엠대우오토엔테크놀로지 Worldwide Purchasing 차장. 관심분야는 소프트웨어공학, Business Process Modeling and Design, Supply Chain Management 등임

#### 황선영

1976년 2월 서울대학교 전자공학과 졸업. 1978년 2월 한국과학원 전기 및 전자공학과 공학석사 취득. 1986년 10월 미국 Stanford 대학 전자전산공학 박사 학위 취득. 1976년~1981년 삼성반도체 주식회사 연구원, 팀장. 1986년~1989년 Stanford 대학 Center for Integrated Systems 연구소 책임연구원. Fairchild Semiconductor Palo Alto Research Center 기술자문. 1989년 1992년 삼성전자(주) 반도체 기술자문. 2002년 4월~2004년 2월 서강대학교 정보통신대학원장. 1989년 3월~현재 서강대학교 전자공학과 교수. 관심분야는 컴퓨터 시스템, SoC 설계 및 framework 구성, Embedded System 설계 등임