

# 요구사항 기술서로부터 유스케이스 다이어그램의 추출기법

유철중<sup>†</sup> · 정소영<sup>††</sup>

## 요약

소프트웨어의 성공적인 개발을 위해서는 체계적이고 명확한 요구분석이 이루어져야 한다. UML에서는 유스케이스 모델링을 통해 사용자 또는 고객의 요구사항을 파악하고 업무 시스템의 범위를 결정하는 방법을 제공하고 있다. 본 논문에서는 효율적이며 정확한 유스케이스 모델링을 위한 연구의 일환으로, 요구사항 기술서로부터 정형화된 규칙을 적용해 가면서 단계적으로 유스케이스 다이어그램을 추출하는 기법을 제시하였다. 요구사항 기술서 관련규칙( $R_{RS}$ ; Rules for Requirements Specification)을 적용하여 요구사항 기술서를 변경하고, 변경된 요구사항 기술서에 액터 추출 규칙( $R_A$ ; Rules for Actors), 유스케이스 추출 규칙( $R_U$ ; Rules for Use Cases), 관계 추출 규칙( $R_R$ ; Rules for Relationships)을 각각 적용하여 액터, 유스케이스, 관계를 추출하여 최종적으로 유스케이스 다이어그램(Use Case diagram)을 작성하게 된다. 본 논문에서 제시한 규칙을 인사관리 요구사항 기술서에 적용해 본 결과, 정형화된 규칙이 없이 서술적인 몇몇 조건을 바탕으로 유스케이스 다이어그램을 추출해야 하는 기존의 어려움을 줄일 수 있는 효과를 확인하였다.

## Use Case Diagram Extraction Technique from Requirements Specification

Cheol Jung Yoo<sup>†</sup> · So Young Jeong<sup>††</sup>

### ABSTRACT

We have to carry out systematic, definite requirements analysis for the successful development of software. The UML gives the ways to grasp user or customer requirements and decide the boundary of business systems from the use case modeling. This paper presents how to extract use case diagram from the requirements specification systematically by applying the standardized rules as a part of the study for use case modeling. We modify requirements specification by applying  $R_{RS}$ (Rules for Requirements Specification) and extract actor, use case, relationship by applying  $R_A$ (Rules for Actors),  $R_U$ (Rules for Use Cases) and  $R_R$ (Rules for Relationships) to the modified requirements specification separately and then become to make out use case diagram in the end. By applying the rules presented in this paper to the requirements specification for personnel management, we can reduce the existing difficulties of extracting use case diagram based on the narrative descriptions without any standardized rules.

**키워드:** UML(Unified Modeling Language), 요구사항 기술서(Requirements Specification), 액터(Actor), 유스케이스(Use Case), 유스케이스 다이어그램(Use Case Diagram)

### 1. 서론

소프트웨어 개발에 있어서 가장 기본적인 과정은 고객과의 면담을 통하여 고객의 요구사항을 분명하게 명세하는 것이다. 이때 복잡한 현실 세계의 요구사항을 보다 정확히 파악하기 위하여 반복적인 분석과 설계과정을 수행하여 모델링을 하게되는데, 많은 모델을 만들어 낼수록 상대적으로 많은 뷰를 나타내며 요구사항에 더욱 근접해간다고 할 수 있다. UML(Unified Modeling Language)은 이러한 다양한 관점을 다룰 수 있는 효과적인 모델링 도구로서 일반적으로

표준으로 제정된 UML을 가지고 개발 프로세스에 적용하고 있는데, 요즘 가장 많이 사용되고 있는 개발 프로세스가 RUP(Rational Unified Process)이다. RUP는 '유스케이스 중심' 개념을 가지고 있어 개발자는 시스템의 기능적 요구사항들을 결정하고 시스템이 무엇을 할 것인가에 대한 상세한 설계로서 유스케이스 모델링을 수행하게 된다[1]. 이 과정에서 유스케이스는 시스템의 행위를 결정하는 것으로, 시스템의 기능을 정의하고 범위를 결정함으로써[2] 시스템과 외부 환경 변수를 구분하고, 상호 관계를 정립하는 기능을 가지고 있다. 유스케이스와 액터와의 관계를 UML로 시각화하여 유스케이스 다이어그램이 작성되고 사용자 및 고객에게 구체적인 정보를 빠르게 전달할 수 있다[3]. 따라서 고객이 요구하는 시스템의 기능적인 측면을 정확히 기술하기 위하여 잘 정제된 유스케이스 다이어그램 작성은 필수적이라

\* 본 연구는 전북대학교 공학연구원 영상·정보신기술연구센터의 지원으로 이루어졌음.

† 종신회원: 전북대학교 컴퓨터과학과 교수

†† 준회원: 전북대학교 대학원 컴퓨터통계정보학과

논문접수: 2002년 1월 29일, 심사완료: 2002년 5월 27일

할 수 있다. 그러나 액터와 유스케이스를 추출하고 그들간의 관계를 작성하는 데 있어 어려움에 직면한다. 정형화된 규칙 없이 여러 문헌에서 보여주고 있는 에메모리한 서술적인 개념과 몇몇 단편적인 조언을 바탕으로 각 요소를 추출하고 유스케이스 다이어그램을 작성하기란 쉽지 않기 때문이다. 따라서 개발자들이 손쉽게 유스케이스 다이어그램을 작성할 수 있도록 정형화된 규칙이 필요하게 되었다.

본 논문에서는 요구사항 파악 후 최종적으로 작성된 요구사항 기술서로부터 고객의 요구를 정확히 반영하여 명백한 의사소통을 도울 수 있는 정형화된 유스케이스 다이어그램 추출 기법을 제안하고자 한다. 2장에서는 관련연구로서 요구분석의 공학적 접근방법인 요구공학의 공정과 유스케이스 다이어그램 작성에 관련된 각 요소의 개념을 설명한다. 3장에서는 유스케이스 다이어그램 추출을 위해 견지해야할 규칙들을 설명하고, 4장에서 본 논문에서 제안한 기법에 따라 인사관리 시스템에 대한 요구사항 기술서로부터 유스케이스 다이어그램 추출 사례를 살펴본다. 5장에서는 결론 및 향후 연구과제를 제시한다.

## 2. 관련 연구

### 2.1 요구공학 공정

소프트웨어가 사용되는 분야가 복잡해지고 대형화되면서 소프트웨어 개발에 있어서 사용자의 요구분석과 올바른 서술이 점차 복잡하고 어려워지게 되었으며, 그에 따라 요구사항에 관계되는 모든 활동에 대한 좀더 체계적이고 공학적인 접근이 필요하게 되었다. 이에 시스템 개발에 관련된 고객들의 요구사항에 대한 추출, 명세화, 분석, 관리를 일관성 있게 수행하기 위한 체계적이고 반복적인 활동 및 프로세스의 총칭으로서의 요구공학[4,5]에 관한 연구가 활발히 진행되고 있다.

요구공학은 소프트웨어공학 단계에서 사용자가 원하는 시스템을 찾아내고, 설계에 들어가기 전에 문서화하는 모든 공

정을 포함한다. 보통 요구공학에서 사용되는 공정은 (그림 1)과 같이 표현할 수 있으며[6], 본 절에서는 UML 기반 요구공학 프로세서로서 요구사항 추출, 요구사항 분석, 요구사항 명세, 요구사항 검증, 요구사항 관리, 요구사항 변경관리의 여섯 단계로 나누어 각 프로세스가 갖는 특징을 살펴보고자 한다[7].

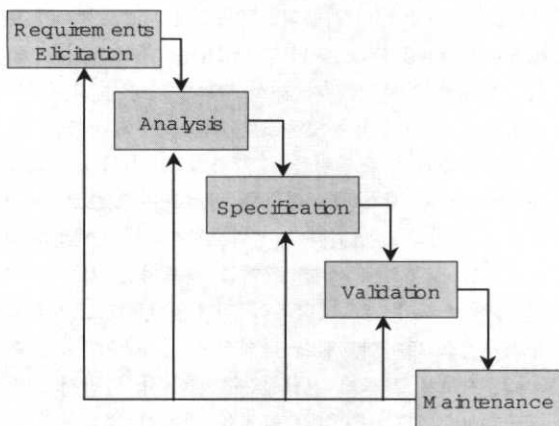
요구사항 추출은 고객, 시스템 사용자, 그리고 시스템 개발에 관련된 사람들과 서로 의견을 교환함으로써 실제로 개발하고자 하는 시스템에 대한 요구들을 찾아내는 공정으로 시스템의 감추어진 정보를 관련된 모든 사람에게 분별하고 정확한 문장으로 얻어내는 과정이다.

요구사항 분석은 객체지향 분석과정에 적합하도록 요구사항을 재명세하기 위하여 분석하는 단계로서, 사용자의 요구를 이해하고 문제 해결의 여러 제약조건들을 정리하여 무슨(what) 시스템을 구현할 것인가를 분석하게 된다. 요구사항 추출단계를 통해 요구사항 획득 후 분류한 요구사항과 추출한 공통어휘를 기반으로 시스템의 영역을 정의하고, UML 도구인 유스케이스와 액터를 사용하여 요구사항을 구조화하고, 활동도(Activity Diagram)를 통해 시스템을 재이해한다. 요구사항에서 시스템의 경계의 정의 및 제한 그리고 시스템에 부과되는 제약사항을 식별함으로써 영역을 분석한 후, 요구사항에서 외부 인터페이스를 찾는 데 인터페이스는 일반적으로 유스케이스 다이어그램에서 액터로 존재하게 된다. 추출한 액터가 시스템에 어떤 기능을 요구하는지를 중심으로 유스케이스를 추출하고, 추출된 유스케이스간의 공통범위와 기능식별 및 관리를 위하여 서로 관련된 유스케이스를 큰 기능군으로 클러스터링한다. 그리고 관련된 유스케이스 안에서 업무들간의 중요도, 위험요소와 복잡성의 포함정도, 새롭거나 개발에 검증되어 있지 않은 기술의 적용 유무에 따라 요구사항 등급을 부여한다. 마지막으로 추출된 유스케이스 별로 활동도를 그려 목표 시스템의 작업흐름을 모형화하고, 시스템을 보다 상세하게 이해한다. 산출물로는 후보 액터와 유스케이스에 대한 명세 또는 이에 대한 다이어그램, 활동도 명세 또는 다이어그램이 있다.

요구사항 명세 단계는 요구사항 분석단계에서 진행한 요구사항의 구조화와 획득한 재이해를 기반으로 기존 요구사항을 재명세한다.

요구사항 검증 단계에서는 재명세 요구사항과 공통어휘를 검증하게 된다. 재명세한 요구사항을 기존 요구사항과 비교, 검토하고 추출된 공통어휘를 검증한다.

요구사항 관리는 요구사항 문서와 후보 액터, 유스케이스, 활동도 분석 결과 등의 형상에 대한 관리가 필수적으로 시행되어야 하고, 분석단계의 형상관리 계획도 수립하는 단계이다. 이와 더불어 각 요구사항 명세가 후보 유스케이스 및 액터와 어떻게 연관되고 추적 되는지를 기술하여야 하며, 분석단계의 검증과 검증계획으로 요구공학 프로세스의 형상과 어떻게 연관할 것인지에 대한 계획을 수립하여야



(그림 1) 요구공학 공정

한다. 또한 소프트웨어 산출물이 완전히 사용자의 요구사항에 부합하도록 검증과 검증 또는 다른 관리 활동을 진행함으로써 품질 보증이 이루어지도록 한다.

요구사항의 변경관리는 식별자인 요구사항 고유의 ID와 요구사항의 상태를 나타내는 상태정보로 관리되는데, 고객으로부터 요구사항에 대한 변경요구가 발생하면 타 형상에 미치는 영향분석 이후 요구사항 간의 연관관계와 타 형상과의 연계를 설정하기 위해 추적 테이블에 관련된 요구사항 간의 정보를 저장한다.

## 2.2 유스케이스 다이어그램

개발자는 고객과의 면담 및 시스템 분석을 거쳐 요구사항 기술서를 작성하게 된다. 요구사항 기술서(명세서)는 시스템 개발을 위해 적절한 기술 용어를 이용하여 요구사항 정의서를 재작성한 것으로[8], 유스케이스 모델링 과정에서 가장 중요한 기초 자료가 된다[9]. 고객의 요구사항이 파악되면 액터와 유스케이스를 찾아 시스템의 경계를 파악하고 [10] 유스케이스 다이어그램을 작성하게 된다. 결국 유스케이스 다이어그램은 시스템의 일부 혹은 전체 액터, 유스케이스, 그리고 이들간의 상호작용을 그림으로 표현한 것[11]으로 볼 수 있다.

### 2.2.1 액터(Actor) 찾기

액터는 시스템의 외부에 존재하면서 시스템과 직접 상호 작용하는 누구 혹은 어떤 것으로 '시스템과 상호작용 한다'는 것은 액터가 시스템으로부터 메시지를 받거나 시스템에게 메시지를 보내는, 혹은 시스템과 정보 교환하는 것을 말한다[12]. 액터를 고려할 때 중요한 것은 사람이나 직위 보다는 '역할'을 먼저 생각해야 한다는 것이다[2]. '역할'은 사용자와 시스템 사이의 관계를 구성하게 되는데 고유의 필요, 흥미, 기대, 행위, 책임 같은 것의 집합에 의해 정의되며[13] 이것은 액터(actor)와 사용자(user)를 구별해야 해야 함을 의미한다. 즉, 사용자(user)는 시스템을 사용하는 실제 사람인데 비해, 액터(actor)는 사용자(user)가 수행하는 어떤 역할을 나타낸다[14]는 것을 유념해야 한다. 다음은 액터를 찾을 때 유용하게 사용되는 질문들을 기술하였다[3, 10].

- 누가 시스템을 사용하는가?
- 누가 시스템을 설치할 것인가?
- 누가 시스템을 시작, 유지관리, 종료할 것인가?
- 다른 어떤 시스템이 이 시스템을 사용할 것인가?
- 누가 이 시스템으로부터 정보를 얻을 것인가?
- 누가 시스템에 정보를 넣을 것인가?
- 현재 어떤 일이 자동적으로 발생하는가?

### 2.2.2 유스케이스(Use Case) 찾기

유스케이스는 액터와 시스템간의 상호작용시 발생하는 것[15]으로 액터가 어떤 특정한 목적을 달성하기 위해 시스

템 내부에서 수행하는 활동의 순서이다[16]. 따라서 유스케이스는 특정 액터에 대하여 관측 가능한 결과를 산출하고 언제나 액터에게 어떤 가치를 제공해야 한다. 요구사항과 유스케이스 간의 관계를 보면 유스케이스는 행위의 단위를 기술한 것, 요구사항은 행위에 적용된 법칙을 기술한 것, 유스케이스는 하나 혹은 그 이상의 기능적 요구사항을 만족할 수 있으며, 기능적 요구사항은 하나 혹은 그 이상의 유스케이스에 의해 만족되어 질 수 있을 것이다[17]. UML에서 유스케이스는 항상 액터에 의해 시작되며[10] 유스케이스를 초기화하는 액터의 입장에서 보통 서술형으로 명명된다. 그러므로 요구사항 기술서에서 액터와 시스템간의 상호작용을 찾아 서술어를 중심으로 하여 유스케이스를 추출한다. 이와 같은 방법을 통해 시스템을 파악하는 방법은 최종적으로 시스템을 사용할 사람의 입장에서 시스템을 파악한다는 중요한 의미를 가진다. 유스케이스를 추출할 때 유용하게 사용될 질문들을 보면 다음과 같다[3, 10].

- 각 액터의 업무(임무)는 무엇인가?
- 이 시스템에서 어떤 액터라도 정보를 작성, 저장, 변경, 제거 또는 읽을 수 있는가?
- 어떤 유스케이스가 정보를 생성, 저장, 변경, 제거 또는 읽을 수 있는가?
- 어떤 액터가 외부변화에 대한 사항들을 시스템에 알릴 필요가 있는가?
- 시스템 내 특정 현상에 대해 어떤 액터에게 알릴 필요가 있는가?
- 어떤 유스케이스가 시스템을 지원하고 유지할 것인가?
- 모든 기능적 요구사항이 유스케이스에 의해 수행되어 질 수 있는가?

### 2.2.3 관계(Relationship) 찾기

액터와 유스케이스를 추출한 후 유스케이스를 상세 기술하는 과정에서 관계를 추출하여 액터와 유스케이스간의 관계를 기술한다. 관계의 유형으로는 '연관관계', '확장관계', '포함관계', '일반화관계'로 구분할 수 있으며 이들 관계는 '액터와 유스케이스간의 관계', '유스케이스와 유스케이스간의 관계', '액터들간의 관계'를 표현할 때 사용된다.

#### (1) 액터와 유스케이스와의 관계

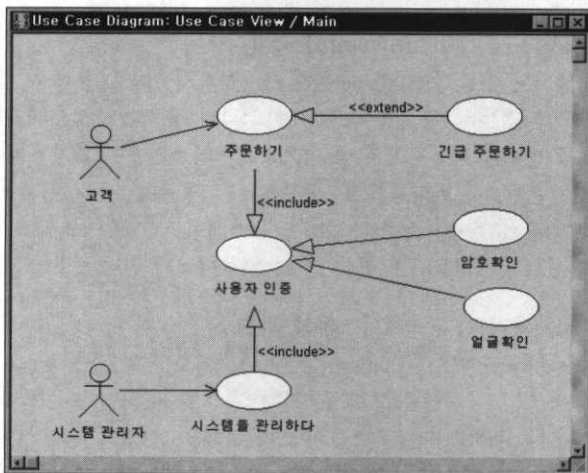
- 연관관계(Association) : 관련된 요소간에 한방향으로 또는 양방향으로 모두 진행 가능한 연관성을 나타낸다. UML에서 요소간의 연관관계는 실선으로 표현하며, 액터에서 유스케이스로 또는 유스케이스에서 액터로의 경우처럼 단지 한방향으로만 진행할 경우 상호작용의 초기화를 담당하는 요소에서 상대 요소로 화살표를 추가한다.

#### (2) 유스케이스와 유스케이스와의 관계

- 포함관계(Include) : 여러 개의 유스케이스들이 동일한 기

능을 공유하는 경우가 있다. 공통 기능을 별도의 독립된 유스케이스로 생성하고 이를 필요로 하는 유스케이스들과 연관성을 형성하도록 한다. UML에서는 공통된 유스케이스 방향으로 색칠안된 삼각형 화살표에 스테레오 타입을 이용하여 «include»로 표현한다.

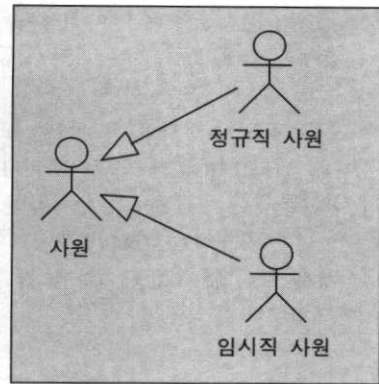
- 확장관계(Extention) : 기존의 유스케이스에 새로운 요구사항을 추가로 표현하고자 할 때 기존의 유스케이스를 수정하는 것 보다 기존의 유스케이스를 그대로 두고 새로운 요구사항을 별도의 유스케이스로 작성하여 기존 유스케이스를 참조하도록 하는 연관성을 '확장관계'로 본다. 포함관계에서는 공통 기능을 표현하는 유스케이스가 없는 경우 포함관계 연관성을 가지는 다른 유스케이스를 독립적으로 설명하기 어렵지만, 확장관계에서는 기존 유스케이스만으로도 독립적으로 설명이 가능하다. 확장관계 역시 포함관계와 마찬가지로 색칠안된 삼각형 화살표에 스테레오 타입을 설정하여 «extend»로 표현한다.
- 일반화관계(Generalization) : 하나의 유스케이스가 다른 유스케이스와 유사하나 그 이상의 더 많은 일을 수행할 때 사용한다[2]. 구체화된 유스케이스(child)는 일반화된 유스케이스(parent)로부터 기능과 의미를 상속 받고, 자신만의 기능을 추가·재정의하여 완전한 유스케이스를 생성한다. UML에서 일반화 관계는 구체화된 유스케이스에서 일반화된 유스케이스 방향으로 색칠안된 삼각형 화살표를 이용하여 표현한다. 다음의 (그림 2)는 액터와 유스케이스 또는 유스케이스와 유스 케이스 간에서 나타날 수 있는 연관관계, 포함관계, 확장관계, 일반화관계를 표현하였다.



(그림 2) 연관, 포함, 확장, 일반화 관계 표현

(3) 액터들간의 관계

액터들간의 관계에는 객체지향의 상속의 의미와 같은 일반화관계가 있다. (그림 3)의 사원과 정규직 사원, 임시직 사원은 일반화관계라고 볼 수 있다.

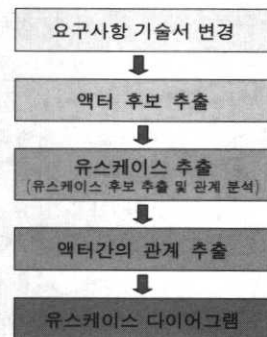


(그림 3) 액터간의 일반화관계

3. 유스케이스 다이어그램 추출 과정 및 규칙

유스케이스 다이어그램을 작성할 때 액터나 유스케이스를 추출하기 위해 일반적으로 사용되는 유용한 질문들을 2.2에서 살펴보았다. 이 장에서는 요구공학적인 접근으로 작성된 요구사항 기술서를 바탕으로 개발자들이 좀 더 쉬운 방법으로 유스케이스 다이어그램을 작성할 수 있도록 유스케이스 다이어그램의 각 요소를 추출할 수 있는 규칙을 제안하고, 유스케이스 다이어그램을 추출하는 과정을 정의하였다.

본 논문에서 제시하고자 하는 유스케이스 다이어그램 추출 과정을 (그림 4)와 같이 나타내었다.



(그림 4) 유스케이스 다이어그램 추출 과정

3.1 요구사항 기술서 변경

요구사항 기술서를 바탕으로 액터, 유스케이스, 관계를 추출하기 위해 요구사항 기술서를 문장 단위로 분석한 후 필요에 따라 제시된 규칙에 의해 문장을 변경한다. 이때, 각 문장 단위로 '문장번호'를 부여하며 적용되는 규칙은 RRS (Rules for Requirements Specification)라고 정의하고 그 내용은 다음과 같다.

규칙 1 : 요구사항 기술서의 수동적 문장은 능동적 문장으로 변경한다[Rrs 1].

면담이나 요구사항 정의서를 통해 사용자의 요구사항을

파악한 후 개발자의 입장에서 요구사항 기술서를 작성할 때 시스템을 사용할 업무담당자 및 사용자를 중심으로 실제 사용하고 있는 용어를 이용하여 기술하고, 수동적 문장 보다는 한글로 작성된 보통 평서문의 형태인 능동적 문장으로 변경하는 것이 합리적이다.

**규칙 2: 주어**가 생략된 부분이나 지시어로 기술된 부분은 의미를 파악하여 주어(동작주체)를 기술한다[R<sub>RS</sub> 2].

역할(액터)과 유스케이스를 추출하기 위해서는 동작의 주체 및 동작행위를 아는 것이 관건이므로 주어(동작주체)가 생략된 경우 문장을 읽고 문장의 주어를 유추하여 기록한다.

**규칙 3: 요구사항 기술서의 문장을 「주어+목적어(+직접목적어+간접목적어)+서술어」 형태로 변경한다**[R<sub>RS</sub> 3].

동작주체와 행위가 핵심 내용이므로, 동작주체는 주어로서 문장의 앞부분에 기술하고 동작행위는 서술어의 형태로 문장의 뒤에 기술하는 형태로 변경한다.

**규칙 4: 동일한 기능을 담당하는 역할의 명칭은 통일한다**[R<sub>RS</sub> 4].

요구사항 기술서를 작성할 때 각 고객마다 같은 기능을 수행하는 것에 대한 명칭을 다르게 표현할 경우 한가지 명칭으로 통일해 준다. 예를 들어 사원 인사 기록을 담당하는 역할을 표현할 때, 요구사항 기술서에서 인사부 담당자, 인사담당자, 인사기록 담당자 등으로 표현되었다면 이들 3개의 명칭 모두 하나의 기능을 나타내므로 '인사담당자'로 통일하여 기술한다.

3.2 액터 후보 추출

위 3.1의 요구사항 기술서 관련 규칙의 적용 결과로 변경된 요구사항 기술서가 작성되면 문장번호 순으로 다음의 액터 추출 규칙을 적용한다. <표 1>의 액터 후보 목록 작성 양식에 그 과정을 기술하는데 결과적으로 선택여부 항목에서 액터 후보가 추출된다. 이때 적용되는 규칙을 R<sub>A</sub>(Rules for Actor)라고 하며 정의된 내용은 다음과 같다.

<표 1> 액터 후보 목록 작성 양식

일련번호	문장번호	액터후보	역할기술	적용규칙	선택여부
:	:	:	:	:	:
:	:	:	:	:	:

**규칙 1: 각 문장에서 주어나 간접목적어가 사람인 경우 액터 후보로 간주한다**[R<sub>A</sub> 1].

시스템에 정보를 입력하거나, 시스템으로부터 정보를 얻을 수 있는 역할을 담당하는 시스템 또는 사용자를 UML에서는 액터로 본다. 따라서, 어떤 행위의 주체를 지시하는 주어나 간접목적어에 해당하는 '사람'은 일단 액터 후보로 간주한다.

**규칙 2: 한 문장의 주어로서, 사람은 아니지만 역할을 가지는 의도적 동작 주체(명사, 고유명사로 표현되는)를 액터 후보로 간주한다**[R<sub>A</sub> 2].

'의도적'이라는 말은 특정한 결과를 목적으로 수행하는 것을 의미하므로 그 동작을 수행하여 결과를 얻을 수 있는 경우에 해당하는 동작주체는 규칙 R<sub>A</sub> 1에서 정의하고 있는 사람은 아니더라도 액터 후보로 간주한다.

**규칙 3: 주어**가 동작의 상태를 나타내는 경우 상태에 능동적인 역할을 할 수 있는 주체는 잠재적 액터로 간주한다[R<sub>A</sub> 3].

요구사항 기술서에는 직접 나타나지 않더라도 시스템의 상태에 변화를 줄 수 있는 액터를 잠재적 액터로 보고 액터 후보로 간주한다.

**규칙 4: 동작주체의 행위에 의해 결과를 얻거나 이익을 얻는 대상을 액터 후보로 간주한다**[R<sub>A</sub> 4].

동작을 수행하는 주체는 아니지만 다른 동작 주체의 행위에 의해 생성되는 결과를 입력받거나, 그 결과로 자신의 상태에 변화를 가져올 수 있는 대상은 액터 후보로 간주한다.

**규칙 5: 동작의 주체이지만, 의도적이지 않은(우연한, 의미없는) 동작을 수행하는 경우는 액터 후보로 간주하지 않는다**[R<sub>A</sub> 5].

'의도적이지 않은'에는 목적을 가지지 않아도 수행되거나 목적과 관계없는 동작이 수행되었을 경우를 의미하므로 액터 후보에서 제외시킨다.

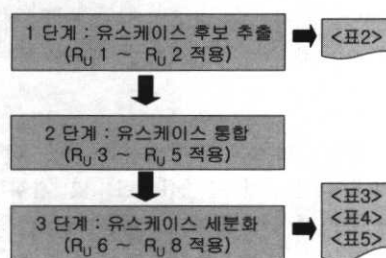
**규칙 6: 동작의 주체 중 이벤트를 발생시키지 않는 주어는 액터 후보에서 제외시킨다**[R<sub>A</sub> 6].

동작을 수행하는 주체이기는 하지만 동작 자체가 어떤 결과를 나타내는 것이 아니라 단순한 상태 설명인 경우에 해당한다면 액터 후보에서 제외시킨다.

**규칙 7: R<sub>A</sub> 1~R<sub>A</sub> 4 중 하나 이상에 해당되면 액터 후보로 결정하되 R<sub>A</sub> 5~R<sub>A</sub> 6에 해당한다면 액터 후보에서 제외시킨다**[R<sub>A</sub> 7].

3.3 유스케이스 추출

액터 후보로부터 유스케이스를 추출하는 과정을 다음의 (그림 5)와 같이 3단계로 나타내었다.



(그림 5) 유스케이스 추출 과정



1단계에서 <표 2>의 유스케이스 후보 목록을 작성한 후, 2단계와 3단계를 거치는 동안 해당사항이 있는 경우에 <표 3>, <표 4>, <표 5>의 양식을 작성한다. 유스케이스 추출 규칙은 Ru(Rules for Use Case)로 정의하고 그 내용은 다음과 같다.

<표 2> 유스케이스 후보 목록 작성 양식

일련 번호	액터후보	문장 번호	활동	유스케이스	결과를 넘길대상이 있는가?	적용 규칙	선택 여부
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:

**규칙 1:** 액터 후보와 관련된 문장에서 동사구를 분석하여 액터 후보가 수행할 수 있는 활동들의 순서를 찾는다. 액터 후보가 수행하는 이벤트에 따라 임시 유스케이스를 추출한다[Ru 1].

유스케이스는 액터의 목적을 달성하기 위해 시스템 내부에서 수행하는 활동의 순서를 가리키므로 액터 후보와 관련있는 동작성을 의미하는 '동사구'를 분석하고, 순서를 부여하여 유스케이스를 추출한다. 확정된 유스케이스가 아니므로 '임시 유스케이스'로 분류한다.

**규칙 2:** 액터 후보의 활동에 따른 유스케이스 중 단순한 동작의 내용을 기술한 경우 액터 후보와 임시 유스케이스를 제거한다[Ru 2].

액터 후보의 목적을 가지지 않는 단순 동작은 임시 유스케이스에서 제거하고, 목적 없는 활동성을 가지는 액터 후보 또한 제거한다.

**규칙 3:** 하나의 액터 혹은 여러 액터에서 같은 행위를 수행하는 유스케이스가 중복되는 경우는 하나만 작성한다[Ru 3].

같은 활동을 의미하는 유스케이스를 모두 나열할 필요 없이 하나만 기술한다.

**규칙 4:** 정의된 유스케이스 중 동일 기능을 수행하는 유스케이스가 구체화되어 있는 경우는 인터페이스를 고려하여 일반화 할 수 있다[Ru 4].

유스케이스 다이어그램을 가능한 단순화하는 것이 더 효율적이므로 동일 기능을 수행한다면 구체화된 여러 유스케이스를 기술하기 보다는 일반화된 유스케이스를 기술하고 필요에 따라 일반화된 유스케이스를 구체화하는 것이 더 합리적이다.

**규칙 4.1:** 유스케이스를 일반화하는 경우 고객의 요구사항을 고려하여 일반화 수준(level) 및 관점(aspect)에 따라 필요한 유스케이스 다이어그램을 작성할 수 있다.

고객의 요구사항 관점에서 단순화된 유스케이스 다이어그램보다는 상세화된 유스케이스 다이어그램이 필요할 수

도 있다. 따라서 모델러는 일반화 수준 및 관점을 조절하고 각 수준 및 관점에 따라 여러 개의 유스케이스 다이어그램을 작성하여 제시할 수 있다.

**규칙 5:** 정의된 유스케이스 중 하나의 유스케이스가 세분화되어 있어 포괄적 의미의 유스케이스와 세분화된 유스케이스가 공존한다면 포괄적 의미를 가지는 유스케이스로 통합할 수 있다[Ru 5].

포괄적 의미의 유스케이스를 기술함으로써 유스케이스 다이어그램을 단순화하고, 필요한 경우에 유스케이스를 세분화 시키는 것이 더 합리적이다.

**규칙 6:** 정의된 유스케이스 중 각 유스케이스에 내용이 공통적으로 사용되는 부분이 있다면 공통된 내용만을 추출하여 <표 3>과 같은 형식으로 별도의 유스케이스를 작성한다[Ru 6].

<표 3> 공통 유스케이스 목록 작성 양식

기존 유스케이스	공통 유스케이스
:	:
:	:

**규칙 7:** 정의된 유스케이스 중 다른 유스케이스를 바탕으로 새로운 요구사항을 추가하여 작성된 새로운 유스케이스를 추출하여 <표 4>와 같은 형식으로 작성한다[Ru 7].

<표 4> 기존 유스케이스에 요구사항이 추가된 유스케이스 목록 작성 양식

기존 유스케이스	기존 유스케이스에 요구사항이 추가된 유스케이스
:	:
:	:

**규칙 8:** 정의된 유스케이스 중 공통된 기능과 속성을 가지지만 더 많은 기능을 포함하고 있는 유스케이스를 추출하여 <표 5>와 같은 형식으로 작성한다[Ru 8].

<표 5> 더 많은 기능을 포함하는 유스케이스 목록 작성 양식

기본 유스케이스	더 많은 기능을 포함하는 유스케이스
:	:
:	:

### 3.4 관계 추출

관계 추출 규칙 R<sub>R</sub>(Rules for Relationships)를 적용하여 최종적으로 추출된 액터와 유스케이스간의 관계를 <표 6>의 형태로 작성한다. 관계 추출을 위한 규칙 R<sub>R</sub>의 내용은 다음과 같다.

〈표 6〉 관계 추출 작성 양식

추출된 액터	유스 케이스	공통 유스케이스	기존 유스케이스에 요구사항이 추가된 유스케이스	더 많은 기능을 포함하는 유스케이스	적용 규칙
:	:	:	:	:	:
:	:	:	:	:	:

규칙 1 : <표 2>의 액터 후보 중 공통된 속성과 기능을 갖는 경우 공통된 속성과 기능만을 갖는 액터와 그 공통 속성에 새로운 속성을 재정의한 나머지 액터들로 구분한다. 이때, 공통된 속성과 기능만을 갖는 액터를 상위레벨에 놓고 나머지 액터들은 하위 액터로 보고 하위 액터에서 상위 액터로 일반화 관계로 정의한다[R<sub>R</sub> 1].

규칙 2 : <표 2>에서 선택된 유스케이스와 액터와의 관계를 분석한다. 이때, 유스케이스를 초기화하는 액터 쪽에서 유스케이스 방향으로 열린 화살표를 그리고, 액터가 유스케이스에 의해 결과를 받는다면 유스케이스에서 액터 방향으로 열린 화살표를 그린다[R<sub>R</sub> 2].

규칙 3 : R<sub>U</sub> 3~R<sub>U</sub> 6의 내용은 이벤트 흐름이 두 개 이상의 유스케이스에 걸쳐 진행되는 경우에 해당되므로 유스케이스와 유스케이스간의 관계로서 표현한다[R<sub>R</sub> 3].

규칙 3.1 : R<sub>U</sub> 7에 의해 추출된 <표 3>의 내용대로 유스케이스를 작성하고 기존 유스케이스 방향에서 공통 유스케이스 방향으로 포함관계를 정의한다[R<sub>R</sub> 3.1].

규칙 3.2 : R<sub>U</sub> 8에 의해 추출된 <표 4>의 내용대로 유스케이스를 작성하고 기존 유스케이스 방향에서 기존 유스케이스에 요구사항이 추가된 유스케이스 방향으로 확장관계를 정의한다[R<sub>R</sub> 3.2].

규칙 3.3 : R<sub>U</sub> 9에 의해 추출된 <표 5>의 내용대로 유스케이스를 작성하고 기존 유스케이스 방향에서 더 많은 기능을 포함하는 유스케이스 방향으로 일반화 관계를 정의한다[R<sub>R</sub> 3.3].

#### 4. 적용 사례

지금까지 제시한 요구사항 기술서 관련 규칙, 액터 추출 규칙, 유스케이스 추출 규칙, 관계 추출 규칙을 인사기본 정보 관리 시스템 구축 사례[9]에 적용하여 유스케이스 다이어그램을 작성하고자 한다.

##### 4.1 요구사항 기술서 관련규칙 적용

요구사항을 파악하기 위해 『인사기본 정보 관리』 요구사항 기술서의 각 문장에 ‘문장 번호’를 부여하고 문장 번호 순으로 R<sub>RS</sub> 규칙 적용 여부를 분석하였다.

##### 4.1.1 요구사항 기술서에 문장번호 부여

인사기본 정보관리 요구사항 기술서의 각 문장에 문장번호를 다음과 같이 부여하였다.

- (1) 인사기록은 사원의 개인신상, 학력, 가족, 병역, 경력 사항 등을 등록, 변경, 유지하는 업무이다.
- (2) 사원의 종류에는 정규직, 임시직 사원이 있으며 인사 기록은 각 직군(정규직/임시직)에 따라 각각 다른 형태로 관리된다.
- (3) 정규직 사원(신입(대졸/고졸)/경력)의 경우에는 인사부 인사담당자가 인사부 채용담당자로부터 최종합격자 명단과 입사지원서 자료, 입사구비서류를 넘겨받아 시스템에 등록하고, 인사부 발령 담당자에게 인사명령을 의뢰하며, 임시직 사원의 경우에는 인사부 인사담당자가 인사부 인력수급담당자로부터 해당자 명단과 입사구비서류 등을 넘겨 받아 별도로 인사기록에 등록한다.
- (4) 정규직 사원의 인사기본정보 관리를 위한 세부절차는 아래와 같다.
- (5) 인사부 인사담당자는 제출된 입사구비서류를 근거로 개인 신상 정보(성명, 주민등록번호, 생년월일, 주소, 연락처 등), 학력사항, 가족사항, 병역사항 및 경력사항(주로 경력사원의 경우)을 입력한다.
- (6) 경력사원의 경우에는 인사명령서를 근거로 직위, 연봉, 부서를 등록하고, 신입사원의 경우에는 별도의 전임 또는 기타 인사명령이 발효될 때까지 공란으로 처리한다.
- (7) 임시직 사원의 인사기본정보관리를 위한 세부절차는 아래와 같다.
- (8) 인사부 인사담당자가 개인 신상 정보(성명, 주민등록번호, 주소, 연락처등)와 필요할 경우 경력사항을 입력하고, 반드시 계약만료일과 계약연장여부를 표시하도록 하며, 임시직 사원에게도 사번을 부여한다.
- (9) 인사기본정보를 등록한 후, 등록된 인사사항을 변경할 경우에 대한 절차는 아래와 같다.
- (10) 새로운 인사명령이 발효하여 인사기록사항을 변경해야 할 경우, 인사부 인사담당자는 관련 인사명령서류를 확인하고, 인사부장의 결재를 온라인으로 취득한 후에 변경하고, 사원이 자신의 인사관련 사항의 변경에 의해 인사기록사항의 변경을 요청할 경우(가족사항, 학력사항, 병역사항 등) 해당사원으로부터 관련 증빙서류를 제출 받아 확인한 후 변경한다.
- (11) 또한, 인사부 인사담당자는 분기별로 부서별 인원현황, 각 부서내 직위별 인원현황, 부서별 임시직 인원현황 등을 작성하여 인사부장에게 보고한다.

##### 4.1.2 R<sub>RS</sub> 규칙 적용 후 변경 결과

위의 문장번호를 부여한 요구사항 기술서에 R<sub>RS</sub> 규칙을 적용한 결과는 다음과 같다.

**결과 1 : 문장 (3)번에 RRS 1과 RRS 3을 적용하여 다음과 같이 변경되었다.**

- (3.1) 정규직 사원(신입(대졸/고졸)/경력)의 경우에는 인사부 채용담당자가 인사부 인사담당자에게 최종합격자 명단과 입사 지원서 자료, 입사구비서류를 넘겨준다.
- (3.2) 3.1에 의해 인사부 인사담당자는 시스템에 등록하고, 인사부 발령 담당자에게 인사명령을 의뢰한다
- (3.3) 임시직 사원의 경우에는 인사부 인력수급담당자가 해당자 명단과 입사구비서류 등을 인사 담당자에게 넘겨준다
- (3.4) 3.3에 의해 인사부 인사담당자는 별도로 인사기록에 등록한다

**결과 2 : 문장 (6)번에 RRS 2를 적용하여 다음과 같이 변경되었다.**

등록 및 처리 주체가 '인사부 인사담당자'임을 유추할 수 있다. 따라서 『인사부 인사담당자가 경력사원의 경우에는 인사명령서를 근거로 직위, 연봉, 부서를 등록하고, 신입사원의 경우에는 별도의 전임 또는 기타 인사명령이 발효될 때까지 공란으로 처리한다.』로 문장을 변경한다.

**결과 3 : 문장 (10)번에 RRS 3을 적용하여 다음과 같이 변경되었다.**

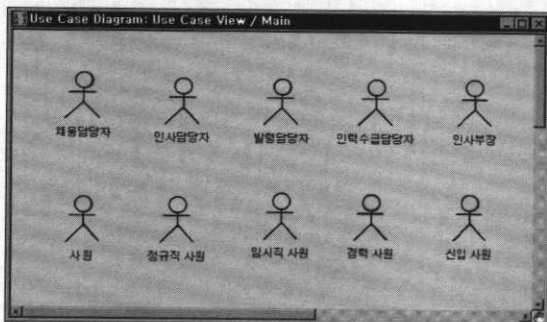
- (10.1) 새로운 인사명령이 발효하여 인사기록사항을 변경해야 할 경우, 인사부 인사담당자는 관련 인사명령서류를 확인하고, 인사부장은 온라인으로 결재를 한다.
- (10.2) 사원이 자신의 인사관련 사항의 변경에 의해 인사기록사항의 변경을 요청할 경우 인사부 인사담당자에게 관련 증빙서류를 제출한다.
- (10.3) 10.2에 의해 인사부 인사담당자가 확인 후 변경한다

4.2 유스케이스 다이어그램

4.2.1 액터 추출 규칙 적용

변경된 요구사항 기술서를 바탕으로 RA를 적용하여 액터 후보 목록을 추출하였다.

<표 7>의 액터 추출 결과를 (그림 6)과 같이 표현하였다.



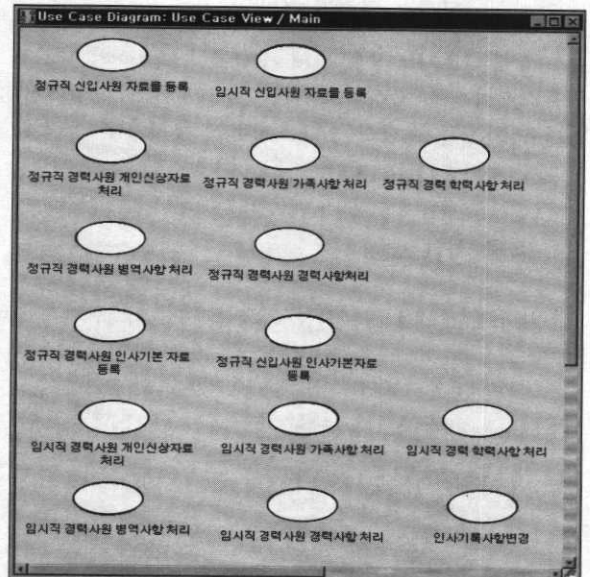
(그림 6) 액터 추출 결과

<표 7> 액터 후보 목록 작성 결과

일련 번호	문장 번호	액터후보	역활기술	적용규칙	선택여부
1	1	인사담당자	사원의 개인신상, 학력, 가족, 병역, 경력사항을 등록, 변경, 유지	RA3 RA6	×
2	2	인사담당자	사원의 직군에 따라 다른 형태로 관리함(사원-정규직,임시직)	RA3 RA6	×
3	3.1	채용담당자	인사담당자에게 자료를 넘겨줌	RA1	○
4	3.2	신입사원 정규직 인사담당자	채용 담당자로부터 자료를 넘겨받아 시스템에 등록, 발령 담당자에게 인사명령 의뢰	RA1 RA4	○
5	3.3	발령 담당자	인사담당자로부터 발령을 의뢰 받음	RA4	○
6	3.3	인력수급 담당자	자료를 인사담당자에게 넘겨줌	RA1	○
7	3.4	인사담당자	인사기록에 등록함	RA1	○
8	5	경력사원 정규직 인사담당자	서류를 근거로 개인신상, 가족, 학력, 병역, 경력사항등을 입력	RA1	○
9	6	인사담당자	인사명령서에 따라 직위, 연봉, 부서등록(경력사원, 신입사원 따로 처리)	RA1	○
10	8	임시직 인사담당자	개인신상정보 입력, 기타사항 입력	RA1	○
11	10.1 10.3	인사담당자	인사기록사항 변경 · 인사부장의 결재 후 변경 · 사원의 증빙서류 제출 후 변경	RA1 RA4	○
12	10.1	인사부장	인사담당자에게 변경에 따른 결재를 함	RA1	○
13	10.2	사원 (정규직,임시직)	변경 요청 후 서류제출	RA1	○
14	11	인사담당자	분기별 서류를 작성하여 인사부장에게 보고	RA1	○
15	11	인사부장	인사담당자에게 보고 받음	RA4	○

4.2.2 유스케이스 추출 규칙 적용

<표 7>에서 선택된 액터 후보만을 대상으로 하여 활동을 기술하고 유스케이스를 생성한다. 생성된 각 유스케이스가 수행한 결과를 넘길 대상이 있는지를 결정하고 적용규칙 및 규칙 적용 결과에 따른 선택여부를 기술하였다. 유스케이스 다이어그램은 순차적, 반복적 과정을 거치므로 일련번호를 부여하여 추출과정을 순서적으로 볼 수 있게 하였다.



(그림 7) 유스케이스 추출 결과 1



1 단계 : 유스케이스를 추출한 결과는 다음의 <표 8>과 같은 양식으로 작성된다. <표 8>에서 선택된 유스케이스만을 작성한 결과는 (그림 7)과 같다.

<표 8> 유스케이스 후보 목록 작성 결과

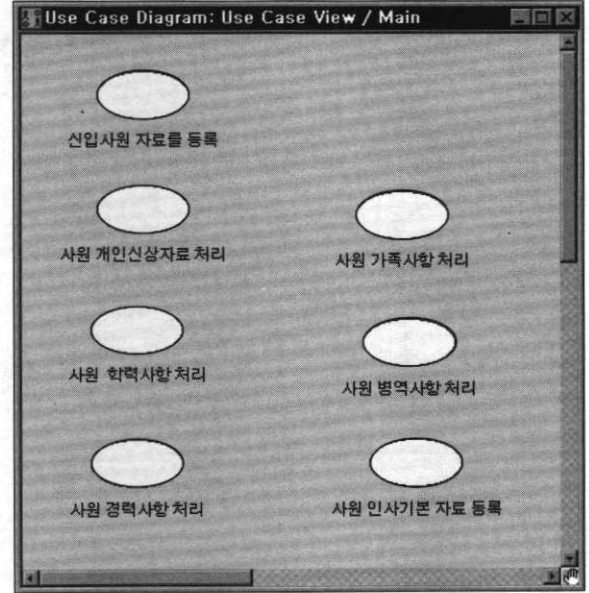
일련 번호	액터 후보	문장 번호	활 동	유스케이스	결과를 넘길 대상이 있는가?	적용 규칙	선택 여부
1	채용 담당자	3.1	인사담당자에게 자료를 넘긴다	1) 자료를 넘긴다	인사담당자	Ru 2	×
2	인사 담당자	3.2	신입 정규직 자료를 시스템에 등록	1) 신입 정규직 자료를 등록한다		Ru 1	○
3	"	3.2	발령담당자에게 인사명령 의뢰	1) 인사명령 의뢰	발령담당자	Ru 2	×
4	"	3.4	신입 임시직 자료를 등록함	1) 신입 임시직 자료를 등록한다		Ru 1	○
5	"	5	정규직 경력사원 개인신상, 가족, 학력, 병역, 경력사항 처리	1) 개인신상자료 처리 2) 가족사항 처리 3) 학력 사항 처리 4) 병역사항 처리 5) 경력사항 처리		Ru 1	○
6	"	6	정규직 경력사원과 정규직 신입사원의 인사명령서에 따라 직위, 연봉, 부서 처리	1) 정규직 경력사원의 인사 기본 자료 처리 2) 정규직 신입사원의 인사 기본 자료 처리		Ru 1	○
7	"	8	임시직 경력사원의 개인신상 정보 및 기타 사항 처리	1) 개인신상자료 처리 2) 가족사항 처리 3) 학력 사항 처리 4) 병역사항 처리 5) 경력사항 처리		Ru 1	○
8	"	10.1	인사부장의 결재후 인사기록사항 변경	1) 개인신상자료 처리 2) 가족사항 처리 3) 학력 사항 처리 4) 병역사항 처리 5) 경력사항 처리		Ru 1	○
9	"	10.3	사원의 요청으로 인사기록 사항 변경	1) 개인신상자료 처리 2) 가족사항 처리 3) 학력 사항 처리 4) 병역사항 처리 5) 경력사항 처리	사원	Ru 1	○
10	"	11	분기별 서류를 작성하여 인사부장에게 보고	1) 서류를 작성하여 인사부장에게 보고	인사부장	Ru 2	×
11	발령 담당자	3.3	발령의뢰를 받음	발령 지시		Ru 2	×
12	인력수급 담당자	3.3	인사담당자에게 임시직 신입사원의 자료를 넘긴다	1) 임시직 신입사원의 자료를 넘긴다	인사담당자	Ru 2	×
13	인사부장	10.1	인사담당자에게 변경에 따른 결재를 함	1) 변경 결재를 함	인사담당자	Ru 2	×
14	"	11	분기별 자료를 인사부담당자로부터 보고 받음	분기별 자료를 보고 받는다		Ru 2	×
15	사원	10.2	변경요청후 서류를 제출한다	1) 인사기록에 대한 변경을 요청함 2) 서류를 제출함	인사담당자	Ru 2	×

2 단계 : (그림 7)과 같이 작성된 유스케이스를 대상으로 Ru 3~Ru 5의 규칙을 적용하여 유스케이스를 통합하면 (그림 8) 및 (그림 9)와 같은 결과를 얻을 수 있다.

결과 1 : 반복되는 유스케이스를 정리하기 위해 Ru 3을 적용하였다.

정규직 경력사원 개인 신상자료 처리 유스케이스, 정규직 경력사원 가족사항 처리 유스케이스, 정규직 경력사원 학력 사항 처리 유스케이스, 정규직 경력사원 병역사항 처리 유스케이스, 정규직 경력사원 경력사항 처리 유스케이스, 임

시직 경력사원 개인신상자료처리 유스케이스, 임시직 경력사원 가족사항처리 유스케이스, 임시직 경력 학력사항처리 유스케이스, 임시직 경력사원 병역사항 처리 유스케이스, 임시직 경력사원 경력사항 처리 유스케이스는 한번만 작성한다.



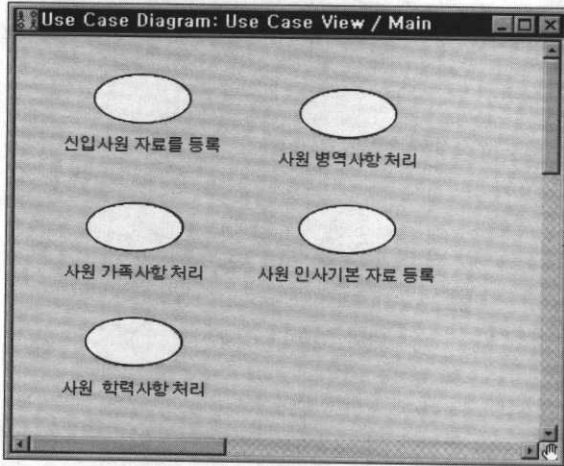
(그림 8) 유스케이스 추출 결과 2

결과 2 : 구체화된 유스케이스를 일반화하기 위해 Ru 4를 적용하였다.

- ① 정규직 경력사원 개인 신상 자료 입력 유스케이스와 임시직 경력사원 개인 신상 자료 입력 유스케이스를 사원 개인 신상 자료 입력 유스케이스로 일반화하였다.
- ② 정규직 경력사원 가족 사항 입력 유스케이스와 임시직 경력사원 가족 사항 입력 유스케이스를 사원 가족 사항 입력 유스케이스로 일반화하였다.
- ③ 정규직 경력사원 학력 사항 입력 유스케이스와 임시직 경력사원 학력 사항 입력 유스케이스를 사원 학력 사항 입력 유스케이스로 일반화하였다.
- ④ 정규직 경력사원 경력 사항 입력 유스케이스와 임시직 경력사원 경력 사항 입력 유스케이스를 사원 경력 사항 입력 유스케이스로 일반화하였다.
- ⑤ 정규직 경력사원 병역 사항 입력 유스케이스와 임시직 경력사원 병역 사항 입력 유스케이스를 사원 병역 사항 입력 유스케이스로 일반화하였다.
- ⑥ 정규직 신입사원 자료를 등록 유스케이스와 임시직 신입사원 자료를 등록 유스케이스를 신입사원 자료 등록 유스케이스로 일반화하였다.
- ⑦ 정규직 경력사원 인사 기본 자료 등록 유스케이스와 정규직 신입사원 인사 기본 자료 등록 유스케이스를 사원 인사 기본 자료 등록 유스케이스로 일반화하였다.

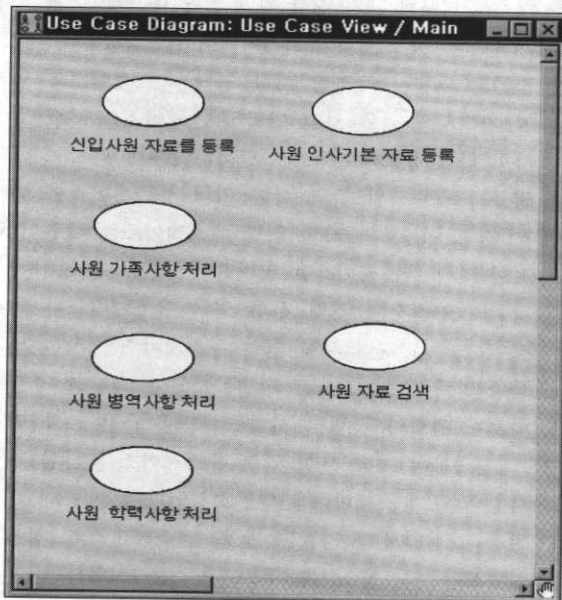
**결과 3: 세분화된 유스케이스를 포괄적 의미의 유스케이스로 통합하기 위해 Ru 5를 적용하였다.**

경력사원 개인 신상 자료 입력 유스케이스와 경력사원 경력 사항 입력 유스케이스가 사원 인사기본 자료 등록 유스케이스로 통합되어 (그림 9)와 같이 반영되었다.



(그림 9) 유스케이스 추출 결과 3

**3단계:** 추출된 유스케이스가 통합된 단계까지의 결과 (그림 9)에 Ru 6~Ru 8의 규칙을 적용하여 유스케이스를 세분화한다. <표 9>, <표 10>, <표 11>의 결과를 (그림 9)에 적용하여 새로운 유스케이스를 추출한 결과는 (그림 10)과 같이 작성된다.



(그림 10) 유스케이스 추출 결과 4

**결과 4: ① 공통 유스케이스를 생성하기 위해 Ru 6을 적용하여 <표 9>를 작성하였다.**

사원 가족사항 처리 유스케이스, 사원 병역사항 처리 유스케이스, 사원 학력사항 처리 유스케이스는 사원 자료 검색 과정을 통해 수행될 수 있으므로 이들 3개의 유스케이스에서 사원 자료 검색 유스케이스를 공통 유스케이스로 생성할 수 있다.

<표 9> 공통 유스케이스 생성 결과

기존 유스케이스	공통 유스케이스
사원 가족사항 처리	사원 자료 검색
사원 병역사항 처리	"
사원 학력사항 처리	"

**② 기존 유스케이스에 요구 사항이 추가된 유스케이스를 추출하기 위해 Ru 7을 적용하였다. 그 결과는 <표 10>과 같다.**

<표 10> 기존 유스케이스에 요구사항이 추가된 유스케이스 추출 결과

기존 유스케이스	기존유스케이스에 요구사항이 추가된 유스케이스
없 음	없 음

**③ 기존 유스케이스를 기본 유스케이스로 하여 그보다 더 많은 기능을 포함하는 유스케이스를 추출하기 위해 Ru 8을 적용하였다. 그 결과는 <표 11>과 같다.**

<표 11> 더 많은 기능을 포함하는 유스케이스 추출

기본 유스케이스	더 많은 기능을 포함하는 유스케이스
없 음	없 음

4.2.3 관계 추출 규칙 적용

액터와 유스케이스를 추출한 뒤 액터간, 액터와 유스케이스간 또는 유스케이스와 유스케이스간의 관계를 추출하여 다이어그램을 완성하게 된다. 추출된 액터와 유스케이스에 R<sub>RR</sub>을 적용하여 <표 12>와 같은 결과를 얻을 수 있다.

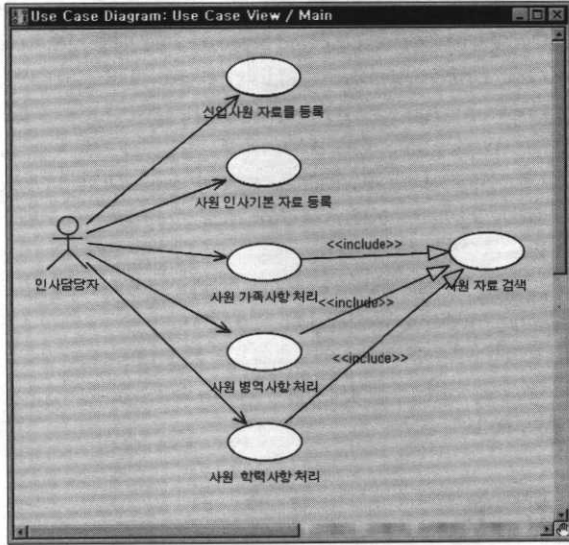
<표 12> 관계 추출 작성 결과

추출된 액터	유스케이스	공통 유스케이스	기존 유스케이스에 요구사항이 추가된 유스케이스	더 많은 기능을 포함하는 유스케이스	적용 규칙
인사 담당자	신입사원자료를 등록	×	×	×	R <sub>RR</sub> 2
	사원 인사 기본자료 등록	×	×	×	R <sub>RR</sub> 2
	사원 가족 사항 처리	사원자료검색	×	×	R <sub>RR</sub> 2 R <sub>RR</sub> 3.1
	사원 학력 사항 처리	"	×	×	R <sub>RR</sub> 2 R <sub>RR</sub> 3.1
	사원 병역 사항 처리	"	×	×	R <sub>RR</sub> 2 R <sub>RR</sub> 3.1

4.2.4 유스케이스 다이어그램 작성

지금까지의 과정에서 추출된 액터, 유스케이스, 관계를

종합하여 유스케이스 다이어그램을 완성한 결과는 (그림 11)과 같다.



(그림 11) 인사 기본 정보 관리 유스케이스 다이어그램

### 4.3 의의 및 평가

유스케이스 모델은 복잡한 표기법 없이 액터와 유스케이스 간의 관계를 보여줌으로써 유스케이스 다이어그램을 산출하여 간결하게 시스템의 기능을 파악할 수 있도록 지원하지만, UML의 분석모델이나 설계모델에 비하여 정형화 정도가 낮고 정해진 규칙의 부재로 모델링이 모호하다는 단점을 가지고 있다[18]. 따라서 인터뷰 결과 혹은 이미 작성된 요구사항 기술서의 내용을 몇몇 유용한 질문들을 사용하여 액터와 유스케이스 및 관계를 추출하여 유스케이스 다이어그램을 작성하는 데는 어려움이 따른다. 이에 본 논문에서는 작성된 요구사항 기술서의 문장을 분석 단위로 하여 액터, 유스케이스 및 관계를 추출하는데 필요한 규칙을 제안하고, 그 규칙을 적용하여 유스케이스 다이어그램을 추출하는 과정을 단계별로 제시하였다. 또한 유스케이스 다이어그램의 추출과정의 단계별 결과를 제시된 양식에 맞게 기술하여 각 요소들의 선택여부 및 다음 단계의 기초를 제공하게 하였다. 현재 사용되고 있는 방법들과는 달리 본 논문에서 제안하고 있는 규칙들의 구체적 특징들은 다음과 같다.

첫째, 유스케이스 다이어그램을 추출하는 과정을 (그림 4)와 같이 단계별로 제시하여 각 단계의 핵심행위와 단계별 연관성을 알 수 있게 하였다. 뿐만 아니라 초보자들도 각 과정에서 제안한 규칙에 따라 순서적으로 유스케이스 다이어그램을 작성할 수 있다.

둘째, 요구사항 기술서로부터 액터, 유스케이스, 관계를 추출함에 있어서 문장을 분석 단위로 취급하여 각 문장마다 문장번호를 부여하고 요구사항 기술서 관련규칙을 적용

함으로써 액터와 유스케이스를 쉽게 추출할 수 있다.

셋째, 액터 추출 규칙을 적용한 결과를 <표 1>과 같이 정리하고 각 액터후보별 역할기술과 선택여부 항목을 통해 객관적으로 액터를 추출할 수 있다.

넷째, 유스케이스를 추출하는 과정을 (그림 5)와 같이 유스케이스 후보추출, 통합, 세분화의 3단계로 정의하였다. 단계별 추출을 통해 <표 3>, <표 4>, <표 5>를 작성하여 정제된 유스케이스를 추출할 수 있다. 이 과정에서 필요하다면 규칙 4.1에 따라 모델러가 고객과의 관계를 고려하여 통합되거나 세분화된 유스케이스를 여러 개의 다이어그램 모델로 표현할 수 있다. 예를들어 통합된 유스케이스를 포함한 다이어그램을 통합 전의 유스케이스를 포함한 하위 다이어그램과 함께 표현함으로써 고객의 이해를 도울 수 있을 것이다.

다섯째, 관계 추출 과정에서는 <표 6>과 같은 양식을 통해 액터 추출 단계와 유스케이스 추출 단계의 결과를 이용하여 관련성을 쉽게 추출할 수 있다.

여섯째, 기존의 각 방법들은 유스케이스를 추출하고 그 유스케이스를 상세화하여 시나리오를 작성하는 과정에서 관계가 추출되는데 본 논문에서는 유스케이스와 관계를 추출하는 과정을 분리하였다.

그러나, 유스케이스에 대한 상세한 시나리오를 작성하여 상세 수정 과정을 통해 잠재적 액터를 찾아내고 그 외의 부수적인 결과들로부터 요구사항 기술서를 수정하는 작업까지는 포함하고 있지 않다. 따라서 유스케이스를 기반으로 하는 다른 다이어그램을 작성하기 위해서는 추가작업이 필수적이며 더불어 유스케이스의 문서화에 관한 부분이 결여되어 해결해야 할 과제로 남아 있다.

## 5. 결론 및 향후 연구과제

본 논문에서는 요구사항 기술서로부터 유스케이스 다이어그램을 추출하는 과정을 4단계로 구성하고 각 단계에서 유용하게 사용될 규칙을 제안하였다. 1 단계에서는 요구사항 기술서에  $R_{RS}$ 를 적용하여 문장 단위를 기준으로 분석 및 정제된 요구사항 기술서로 변경할 수 있었다. 2 단계에서는  $R_A$ 를 적용하여 후보 액터를 추출하였는데, 각 후보 액터의 역할이 기술됨으로써 다음 단계의 유스케이스 추출에 기초 자료로 사용되었다. 3 단계에서는  $R_U$ 를 적용함으로써 유스케이스가 추출되고, 추출된 유스케이스로부터 유스케이스들 간의 연속된 흐름이 있는지를 판단하기 위해 공통 유스케이스, 기존 유스케이스에 요구사항이 추가된 유스케이스, 더 많은 기능을 포함하는 유스케이스가 별도로 작성되었다. 4 단계에서는 관계를 추출하는 과정으로  $R_R$ 를 적용하였다. 액터간의 관계, 액터와 유스케이스간의 관계, 작성된 유스케이스들 중 이벤트 흐름이 두 개 이상의 유스케



이스에 걸쳐 진행되는 경우 유스케이스와 유스케이스간의 관계를 설정할 수 있었다.

향후과제로는 액터를 추출함에 있어 역할의 범위를 어디 까지 설정할 것인가에 대한 자세한 기준과 유스케이스를 추출함에 있어 활동의 세분화 정도나 비슷한 기능을 어떻게 묶을 것인가에 대한 정밀한 기준을 제공하는 것이 필요하다. 더 나아가 유스케이스 다이어그램의 패키징화, 유스케이스의 시나리오를 작성하고 시나리오에 따라 액티비티 다이어그램을 작성하는 기법 등에 대한 연구가 필요할 것으로 보인다.

### 참 고 문 헌

[1] Leszek A. Maciaszek, *Requirements analysis and system design Developing information systems with UML*, Addison-Wesley Pub. Co., 2001.

[2] Martin Fowler, Kendall Scott, *UML Distilled Second Edition : A Brief Guide to the Standard Object Modeling Language*, Addison-Wesley Pub. Co., 2000.

[3] Joseph Schmuller, *Teach Yourself UML in 24 Hours*, SAMS Pub. Co., 1999.

[4] Ian Sommerville & Pete Sawyer, *Requirements Engineering*, John Wiley & Sons, Inc., 1997.

[5] Klaus Pohl, *Process-Centered Requirements Engineering*, John Wiley & Sons, Inc., 1996.

[6] 강기선, 김진태, 박병철, 박수용, "요구공학개요", 소프트웨어 공학회지, 제11권 제4호, pp.5-11, 1998.

[7] 이원우, 박수용, 류성열, "객체지향 어플리케이션 개발을 위한 UML 기반의 요구공학 프로세스", 제1회 한국소프트웨어공학 학술대회, pp.165-172, 1999.

[8] 장옥배, 유철중, 이병걸, 김지홍, 양해술, 김병기, *소프트웨어공학*, 도서출판 한산, 2001.

[9] 류형규, 이순천, 류시원, 신성호, *UML 기반 객체지향 클라이언트/서버 구축*, 홍릉과학출판사, 2000.

[10] Geri Schneider, Jason P.Winters, *Applying Use Case : A Practical Guide*, Addison-Wesley Pub. Co., 1998.

[11] Terry Quatrani, *Visual Modeling With Rational Rose 2000 and UML*, Addison-Wesley Pub. Co., 2000.

[12] Hans-Erik Eriksson, Magnus Penker, *UML Toolkit*, John

Wiley & Sons, Inc., 1998.

[13] Wirfs-Brock Rebecca, Wilkerson Brian, Wiener Lauren, *Designing Object-Oriented Software*, Prentice Hall, 1990.

[14] Ivar Jacobson, Magnus Christerson, Patrik Jonsson, Gunnar Övergaard, *Object-Oriented Software Engineering-A Use Case Driven Approach*, Addison-Wesley Pub. Co., 1998.

[15] Cockburn Alistair, *Writing Effective Use Case*, Addison-Wesley Pub. Co., 2000.

[16] Doug Rosenberg, Kendall Scott, *Use Case Driven Object Modeling with UML : A Practical Approach*, Addison-Wesley Pub. Co., 1999.

[17] Armour Frank, Miller Granville, *Advanced Use Case Modeling*, Addison-Wesley Pub. Co., 2001.

[18] 김민선, 김수동, "전사적 프로젝트의 Use Case 모델링을 위한 실무 지침", 정보처리학회 추계학술발표논문집, 제6권 제2호, pp.1-4, 1999.



### 유 철 중

e-mail : cjyoo@moak.chonbuk.ac.kr

1982년 전북대학교 전산통계학과 졸업  
(이학사)

1985년 전남대학교 대학원 계산통계학과  
졸업(이학석사)

1994년 전북대학교 대학원 전산통계학과  
졸업(이학박사)

1982년~1985년 전북대학교 전자계산소 조교

1985년~1996년 전주기전여자대학 전자계산과 부교수

1997년~현재 전북대학교 자연과학대학 컴퓨터학과 조교수

관심분야 : 소프트웨어공학, 에이전트공학, 컴포넌트기술, 분산  
객체기술, GNSS(GPS), GIS, 멀티미디어, 인지과학 등



### 정 소 영

e-mail : syjeong@cs.chonbuk.ac.kr

1995년 전주대학교 상업교육학과 졸업

2001년 전북대학교 교육대학원 전자계산  
교육 졸업(석사)

2002년~현재 전북대학교 대학원 컴퓨터  
통계정보학과 박사과정

관심분야 : 소프트웨어공학, 요구공학, 에이전트공학, 분산객체  
기술