

---

# 유스케이스 기술서 작성 도구의 개발

최환복\* · 김윤호\*\*

## An Implementation of Writing Tool for Use Case Description

Hwan-bok Choi\* · Yun-ho Kim\*\*

### 요 약

본 논문에서는 유스케이스 기술서 작성 및 편집을 체계적으로 하기 위한 도구를 제시하고자 한다. 또한, 시나리오의 기술에 있어서 1컬럼 방식과 2컬럼 방식 모두를 지원하도록 하였으며, 시나리오 각 단계별 작성과 편집의 처리에 있어서 자동화된 단계 번호 매기기를 지원하여 작성의 편의성을 높였다. 본 논문에서 제시하는 유스케이스 기술서 작성 도구는 편리하고 효과적으로 유스케이스 기술서를 작성하게 해 줄 뿐만 아니라, 유스케이스 다이어그램과 통합된 유스케이스 모델링 도구의 개발에 기여할 것으로 기대된다.

### ABSTRACT

This paper presents an implementing of Use Case Description writing tool to support systematically writing and editing Use Case Description. It also supports both 1 column and 2 column formats of use case scenarios and increases convenience for editing scenario steps with automated step numbering function we presented. It results in not only user-friendly and effective description for use case but also contributing to development of integrated use case modeling tool with use case diagram.

### 키워드

UML, UseCase CASE Tool, OOP, Object-oriented Software System

## I. 서 론

유스케이스 (Use Case)는 Ivar Jacobson[1]이 제시한 이래 시스템 분석의 주요한 기법으로 널리 사용되어 왔다. 유스케이스는 액터와 사용사례들 간의 연관관계를 시각적으로 나타내는 유스케이스 다이어그램과 실질적인 사용사례의 내용을 기술한 유스케이스 기술서 (description)로 제시되어 있다 [1]. UML[2]에서는 유스케이스 다이어그램에 대한 정의만을 제공하므로 대부분의 UML 도구에서는 유스케이스 다이어그램의 지원을

기본으로 하고 있다. 그러나 유스케이스에서 실질적으로 중요한 것은 사용사례의 세부적인 내용을 표현한 유스케이스 기술서로써 유스케이스 다이어그램보다는 유스케이스의 내용을 작성을 하는데 집중하는 것이 바람직하다[3]. Constantine[4]은 사용자의 의도에 초점을 둔 필수 유스케이스 (Essential UseCase) 기법을 제안했다. 유스케이스 시나리오 작성의 실재를 다룬 대표적인 내용으로는 Cockburn[5]이 제안한 1컬럼 방식과 Rebecca[6]가 제안한 2컬럼 방식이 있다. 1컬럼 방식은 하나의 영역에 사용자와 시스템간의 상호 작용 시나리오를 기술하

---

\* 안동대학교 컴퓨터공학과 석사과정  
\*\* 안동대학교 전자정보산업학부 정교수

는 방법으로써 시스템 내부에 대해 세부적으로 기술할 때 유용한 방식이다. 이에 반해 2컬럼 방식은 사용자와 시스템을 각기 다른 영역으로 나눠서 상호 작용 시나리오를 기술하는 방법으로써 사용자와 시스템 간의 상호 작용을 간략하게 기술할 수 있어 시스템 개발의 초기에 많이 사용된다[7]. 그림 1은 물건 판매에 대한 유스케이스 시나리오를 1컬럼 형식으로 기술한 예를 보인 것이며 [5], 그림 2는 ATM에서 출금을 2컬럼 형식으로 기술한 예를 보인 것이다[6].

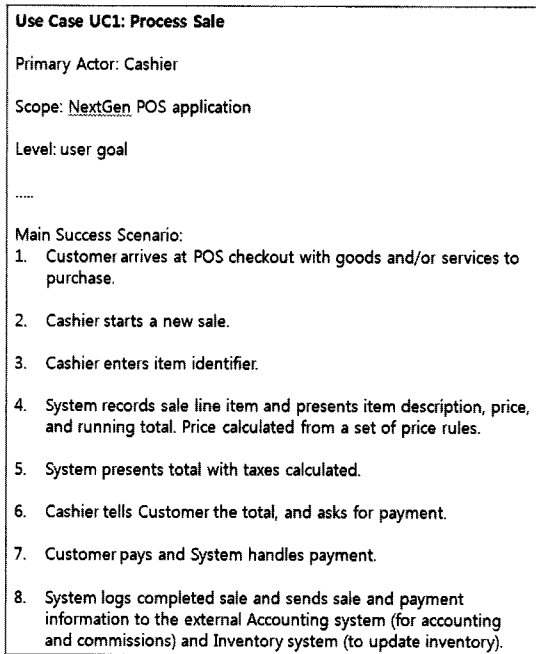


그림 1. 1컬럼 방식의 시나리오  
Figure 1. Scenario with 1 column

현재 유스케이스 모델링을 지원하는 대부분의 도구들은 UML에 기초하여 유스케이스 다이어그램만 지원하고 있다. 따라서 본 논문에서는 유스케이스 기술서를 체계적으로 작성하기 위한 도구를 제시하고자 한다. 또한, 시나리오 작성에 있어서 1컬럼과 2컬럼 형식 모두에 대한 시나리오 작성과 관리를 위해 시나리오 단계를 유지하는 방법을 제시하고자 한다.

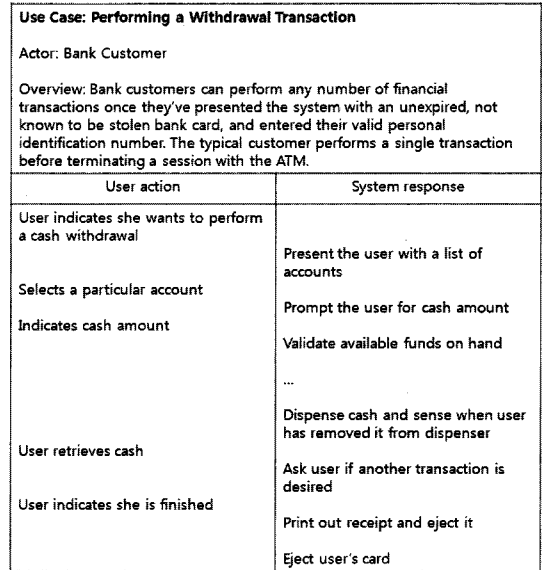


그림 2. 2컬럼 방식의 시나리오  
Figure 2. Scenario with 2 column

## II. 유스케이스 기술서 작성 도구

### 2.1. 유스케이스 기술서 시스템의 구성

유스케이스 기술서 작성 도구는 기존의 유스케이스 기술을 문서화 도구에서 작성하던 것을 통합된 형태로 작성하고 관리하기 위한 도구이다. 본 논문에서 제시하는 유스케이스 기술서 작성 도구의 전체적인 시스템 구성을 그림 3에서 보인다.

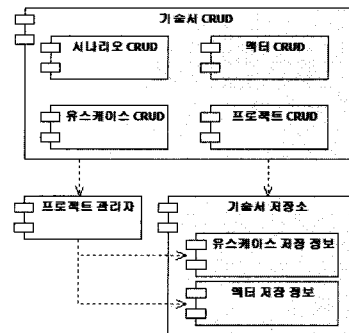


그림 3. 유스케이스 기술서 작성 시스템  
컴포넌트 다이어그램  
Figure 3. The components diagram of use case  
description writing tool

유스케이스 기술서 작성 도구는 역할에 따라 3개의 컴포넌트로 구성되는데, 유스케이스 기술서 작성에 필요한 액터에 대한 정보와 작성된 유스케이스 기술서를 저장하는 ‘기술서 저장소’ 컴포넌트, 액터와 유스케이스를 각 프로젝트별로 구분하는 ‘프로젝트 관리자’ 컴포넌트, 유스케이스 작성기의 모든 CRUD(Create, Read, Update, Delete)[8]를 처리하는 ‘CRUD’ 컴포넌트들로 구성되어 있다. 특히, CRUD 컴포넌트는 유스케이스 기술서상에서 항목에 대한 CRUD가 일어났을 경우 프로젝트 관리자가 어떤 프로젝트에 소속된 항목인지 판단하고 항목에 대한 정보를 CRUD로 넘겨준다. CRUD는 항목에 대한 처리를 마친 후 기술서 저장소에 처리 결과를 반영하는 역할을 한다.

유스케이스 기술서를 작성하기 위해서는 기술에 필요한 유스케이스 항목이 필요하다. 유스케이스 작성하는데 사용되는 항목은 다양하므로, 본 논문에서는 이들 중에서 가장 널리 사용되고 있는 Cockburn이 제안한 항목[9]을 유스케이스 항목들로 제공한다. 또한 다수개의 유스케이스가 있을 때 각각의 유스케이스를 구별하여 처리하기 위해서 ID항목을 추가하였다. 본 논문에서 기술서를 처리하기 위하여 설정한 유스케이스 항목들은 다음과 같다.

- ID: 유스케이스를 구분하기 위한 식별자
- 이름: 유스케이스의 이름
- 범위: 설계중인 시스템
- 수준: 사용자 목적 또는 세부기능
- 주요액터: 목적을 수행하는 시스템의 서비스를 호출하는 주된 액터
- 이해관계자 및 관심사항: 해당 유스케이스를 중요하게 생각하는 사용자와 이들이 원하는 사항
- 사전조건: 유스케이스가 시작할 때 만족해야 하는 조건
- 성공보증: 유스케이스가 성공적으로 끝났을 때 만족해야 하는 조건
- 주요성공 시나리오: 성공하기 위한 전형적이고 조건이 없는 경로의 시나리오
- 확장: 성공이나 실패에 대한 대안 시나리오

기술서 항목들 중에서 주요 성공 시나리오와 시나리오에 대한 확장을 처리하기 위하여 설정한 시나리오의

한 단계(step)를 구성하는 항목은 다음과 같다.

- 단계번호: 단계가 시나리오에서 몇 번째에 해당하는지에 대한 번호
- 단계에서 수행하는 활동

유스케이스 기술서를 작성할 때 유스케이스에 관계 없는 액터가 적용되거나 존재하지 않았던 액터가 추가되는 것과 같은 경우를 방지하기 위해서는 유스케이스의 액터를 관리할 필요가 있다. 본 논문에서는 이를 위해 예상되는 액터를 등록하고 등록된 액터들을 한 곳에서 관리하고 등록된 액터에 대해서만 유스케이스에서 선택할 수 있도록 한다. 액터의 등록과 등록된 액터를 유지하기 위해서는 액터에 관한 정보가 필요하다.

일반적으로 액터는 이름으로 표현되지만 여러 개의 액터가 있을 경우 액터간의 구별을 위해 ID 항목을 추가하였다. 액터의 등록과 등록된 액터를 유지를 위해 설정한 항목은 다음과 같다. (등록된 액터는 유스케이스 기술서 작성에서 주요액터 항목에서 참조하도록 한다.)

- ID: 액터를 구분하기 위한 식별자
- 이름: 액터의 이름

## 2.2. 유스케이스 시나리오 작성 처리

주요성공 시나리오는 일반적으로 다수개의 시나리오 단계로 구성됨으로 단순히 하나의 항목으로 처리하기에는 복잡하다. 따라서 주요성공 시나리오를 유스케이스 항목과 따로 유지하고 유스케이스 항목에서는 따로 구성된 시나리오를 참조(reference)하는 형태를 가지도록 하여 시나리오 단계 처리의 복잡도를 줄일 수 있도록 하였다.

일반적으로 주요성공 시나리오를 구성하는 단계는 하나의 단계에 대한 하위 단계를 기술함으로써 좀 더 세부적인 활동을 표현을 할 수 있다. 이렇게 하나가 다른 하나를 포함하는 형태는 트리에서 상위 노드가 하위 노드를 가지는 것과 유사하다. 따라서 본 논문에서는 주요성공 시나리오 구성을 위해 트리구조를 사용하고 시나리오에 기술되는 각 단계는 시나리오 유지를 위해 트리의 노드형태로 표현한다.

1컬럼 방식의 처리는 하나의 영역에 액터와 시스템간의 상호작용을 기술하는 것으로 그림 4는 단계번호만을

이용해서 주요성공 시나리오의 처리 구조를 보인 것이다. 각 단계들을 통합하는 루트노드를 두고 각 시나리오의 각 단계들을 루트노드의 자식 노드로 구성하고 시나리오 단계의 하위 단계가 있을 경우 해당 단계를 하위 노드가 되도록 처리한다.

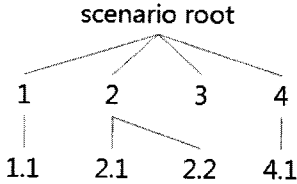


그림 4. 1컬럼에서 시나리오 단계 구성  
Figure 4. Scenario step configuration in 1 column

2컬럼 방식의 처리는 시나리오 작성 영역을 2개로 구분하여 액터 영역은 액터의 행위를, 시스템 영역은 액터 행위에 대한 시스템의 응답을 기술하도록 처리한다. 2개의 영역에 대한 시나리오 구성을 위해 1컬럼에서 구성했던 트리와 같은 구조를 각 영역별로 사용한다. 2컬럼에서 중요한 것은 트리가 2개로 구성되어 상호간의 연결이 단절된다는 것이다. 즉 시나리오에서 액터의 행위와 시스템의 응답을 연관된 순차성을 유지하도록 처리하는 방법이 필요하다. 이를 위해서 액터 영역 트리와 시스템 영역 트리간 서로의 경계를 넘는 연결이 필요할 경우에는 이전 단계에서 다음 단계에 대한 참조를 유지하도록 한다. 그림 5는 2컬럼에서 시나리오 단계를 구성한 예를 보인 것이다. 단계 1에서 단계 2로 시나리오가 이어지지만 다른 영역이기 때문에 단계 1에서 다음 단계인 2의 참조를 가지도록 한다. 단계 4와 5, 단계 6과 7 역시 서로 다른 영역이기 때문에 다음에 오는 단계의 참조를 가진다.

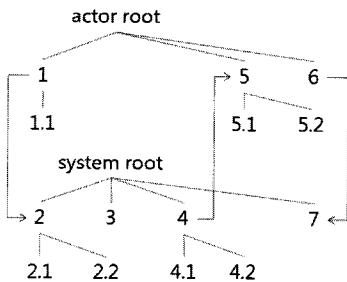


그림 5. 2컬럼의 시나리오 단계구성  
Figure 5. Scenario step configuration in 2 column

주요성공 시나리오를 트리형태로 유지하기 위해서 시나리오 각 단계를 표현하는 노드에 단계에 대한 정보를 정의하는 것이 필요하다. 유지할 정보로는 시나리오 단계의 항목에서 단계번호와 내용, 시나리오 단계간의 순서를 위한 이전 단계와 다음 단계의 참조, 해당 단계의 하위 단계를 유지하는 요소, 대안에 해당하는 확장을 참조하는 정보를 이용하여 처리한다. 시나리오 단계 저장을 위해 필요로 하는 정보는 다음과 같다.

- 단계번호: 단계가 시나리오에서 몇 번째에 해당하는지에 대한 번호
- 내용: 단계에서 수행하는 활동
- 이전단계: 현재 단계 이전에 수행되는 단계의 참조
- 다음단계: 현재 단계 이후에 수행되는 단계의 참조
- 하위단계: 현재 단계의 하위 단계
- 확장: 단계의 대안에 해당하는 확장의 참조

시나리오의 대안 흐름인 확장은 하나의 시나리오 단계에 여러 개가 있을 수 있는데 이 모두를 하나로 다루기에는 개수가 많아 관리가 힘들다. 따라서 시나리오 단계 구성과 유사하게 하나의 확장을 의미하는 트리를 구성하고 해당 시나리오 단계에서 확장을 참조하도록 한다. 다음은 확장 단계와 시나리오 단계의 유사점을 바탕으로 확장을 저장하기 위해 설정한 요소들을 보인다.

- 단계번호: 단계가 확장에서 몇 번째에 해당하는지에 대한 번호
- 내용: 확장 단계에서 수행하는 활동
- 이전확장: 현재 단계 이전에 수행되는 확장의 참조
- 다음확장: 현재 단계 이후에 수행되는 확장의 참조
- 하위확장: 현재 확장의 하위확장

단계번호는 현재의 단계 또는 확장이 전체에서 몇 번째에 해당하는지에 대한 정보를 표현한다. 단계번호를 구성하는 방법에는 여러 가지가 있지만 일반적으로 많이 사용하는 형식을 선정하였다. 시나리오 단계에 대한 단계번호 형식은 구분자(.)를 기준으로 왼쪽은 상위 단계의 단계번호를 사용하고 오른쪽은 현재 단계의 번호를 사용한다.

확장은 주요 성공시나리오의 하나의 단계에 여러 개가 있을 수 있어 시나리오에서 사용하는 형식의 단계번호

호로는 모든 확장을 표현할 수 없다. 따라서 각각의 확장을 구분할 수 있는 방법이 필요한데 확장의 단계번호 앞쪽은 주요 성공 시나리오 단계의 번호를 사용하고 연속해서 확장의 개수에 따라 알파벳을 부여하여 각각의 확장을 구분하도록 한다. 이후 하위 확장에 대해서는 시나리오 단계와 같은 형식으로 단계번호를 부여한다. 그림 6은 본 논문에서 제시하는 유스케이스 기술서 작성 도구에서 사용하는 시나리오와 확장의 단계번호 형식을 나타낸 것이다.

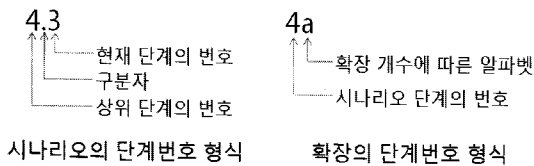


그림 6. 시나리오와 확장의 단계번호 형식  
Figure 6. Step numbering format for scenario and extension

그림 7은 유스케이스 기술서를 처리하기 위한 항목들 간의 연관관계를 전체적으로 보인 것이다. 액터와 유스케이스를 프로젝트 별로 구분하기 위해 프로젝트를 설정하고 액터와 유스케이스는 해당 프로젝트가 관리하며 하나의 프로젝트는 여러 개의 액터와 유스케이스를 포함한다. 유스케이스의 복잡도를 낮추기 위해 유스케이스와 시나리오를 분리하고 유스케이스에서는 분리된 시나리오를 참조하도록 한다.

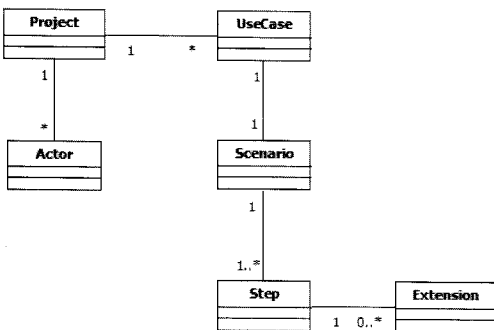


그림 7. 유스케이스 기술서 작성 도구 전체 항목간 연관 관계  
Figure 7. Relationships of use case description writing tool component

시나리오에는 하나 이상의 시나리오 단계로 구성하며 시나리오 단계는 확장을 포함한다.

### 2.3. 유스케이스 시나리오 편집 처리

시나리오의 갱신은 시나리오 단계에서 변화가 생겼을 경우 변화와 관련된 시나리오 단계에 대해서 새로운 정보를 할당하고 갱신하는 것을 의미한다. 시나리오 전체에 대해서 갱신이 가능하지만 이는 비효율적이기 때문에 본 논문에서는 시나리오 갱신을 위하여 갱신범위를 설정하여 그 범위 안에서만 갱신이 수행되도록 하여 효율성을 높인다. 변경이 일어나는 시점에 변경되는 단계를 ‘변경점’으로 부르기로 하고 변경점을 포함하여 변경점과 같은 수준의 모든 단계에 대해서 시나리오를 갱신한다. 그림 8은 시나리오에 대해 변경점과 갱신범위를 표현한 것이다. 시나리오의 2번째 단계에서 변경이 일어날 경우 2번째 단계를 변경점으로 설정하고 이후의 모든 단계에 대해서 시나리오 정보의 갱신을 수행한다.

Actor	System
1. User indicates she wants to perform a cash withdrawal	변경점
3. Selects a particular account 3.1 3.2 .....	2. Present the user with a list of accounts 2.1 2.2  갱신범위

그림 8. 변경점과 갱신범위  
Figure 8. Changing point and updating scope

2컬럼의 경우 액터와 시스템 영역이 나누어져있어 2개의 트리가 존재한다. 비록 트리가 2개로 이루어져 있지만 시나리오 각 단계의 정보에 다음 단계에 대한 참조를 가지고 있기 때문에 변경점을 기준으로 다음 단계에 대한 참조를 이용하여 시나리오 정보를 갱신한다.

유스케이스 기술서에서 시나리오에는 사용자의 요청과 이에 대한 시스템의 응답이 순차적으로 표현되는 부분이다. 시나리오의 순서를 나타내기 위해서 순서번호를 표기하지만 사용자가 직접 표기할 때 잘못된 번호를 표기하여 혼란을 야기할 수 있다. 본 논문에서는 시나리오 단계에 대해서 사용자가 직접 번호를 매기는 것을 시스템 상에서 시나리오 단계 추가, 삭제, 수준 올리기, 수

준 내리기에 대해 자동화된 시나리오 단계 번호 매기기를 제공하여 실수로 인한 오류를 사전에 방지하도록 한다.

2.3.1. 시나리오 단계와 확장의 단계번호 갱신

시나리오 단계와 확장의 단계번호 갱신을 한 곳에서 갱신할 수 있지만 이러한 방법은 복잡도를 높일 수 있다. 따라서 본 논문에서는 시나리오 단계가 하위 단계와 확장을 포함하고 있을 때 이들의 갱신을 해당 시나리오 단계가 담당하도록 책임을 분산함으로써 복잡도를 낮추도록 한다. 시나리오 단계번호 갱신은 유스케이스의 시나리오에서 하나의 단계에 대해서 단계번호를 갱신하는 것으로 갱신 절차는 그림 9와 같다.

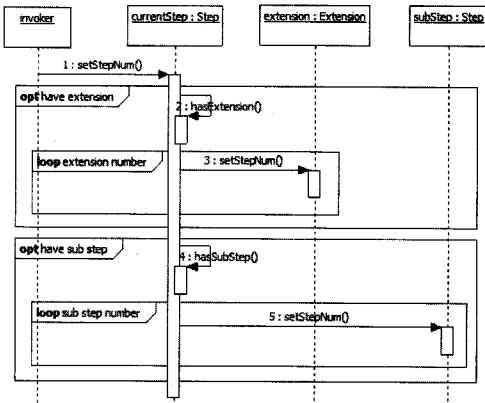


그림 9. 시나리오 단계번호 갱신 시퀀스 다이어그램  
Figure 9. The sequence diagram of updating scenario step number

그림 9에서 보는 바와 같이, 시나리오 단계가 추가, 삭제 등 변경이 일어났을 경우 새로운 단계번호가 시나리오 단계에 할당되고 해당 단계는 기존의 단계번호를 새로운 단계번호로 변경되는 과정을 표현한다. 확장과 하위 단계를 검사하고 확장 또는 하위 단계가 존재하면 이들에 대해서 변경된 단계번호를 할당하게 된다. 그림에서 invoker는 시나리오 단계번호 갱신을 시작하는 객체를, currentStep은 현재 단계 객체를 의미한다. extension은 currentStep 객체의 확장 객체이며, subStep은 currentStep의 하위 단계 객체를 의미한다.

시나리오의 대안 흐름인 확장 또한 시나리오 단계와

같이 하위 확장을 포함할 수 있기 때문에 확장 또한 하위 확장에 대한 단계번호 갱신 책임을 해당 확장에 부여한다. 확장에서 단계번호를 갱신하는 수행절차는 그림 10과 같다.

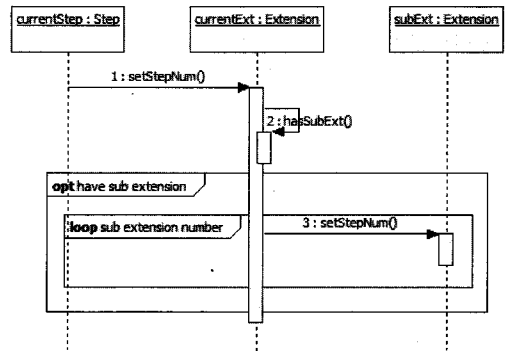


그림 10. 확장의 단계번호 갱신 시퀀스 다이어그램  
Figure 10. The sequence diagram of updating extension step number

그림 10은 시나리오 단계의 단계번호가 갱신되었을 때 확장이 있을 경우 확장에 새로운 단계번호가 할당하고 기존의 단계번호를 새로운 단계번호로 변경하는 절차를 나타낸다. 이때 하위 확장이 있는지를 검사하고 하위 확장에 대해서도 기존의 단계번호를 새로운 단계번호로 변경한다. 그림에서 currentStep은 현재 단계의 객체를, currentExt는 currentStep 객체에서 현재 확장을, subExt는 currentExt 객체의 하위 확장 단계 객체를 의미한다.

2.3.2. 시나리오 단계 추가

시나리오 단계추가는 새로운 시나리오 단계를 생성하고 기존의 시나리오에 단계를 추가하는 것으로 이 기능을 수행하는 절차는 그림 11과 같다. 그림에서 보는 바와 같이, 단계가 추가될 때 액터 영역인지 시스템 영역인지를 판단 후 새로운 단계를 생성하고 기존 단계의 참조를 추가된 단계의 참조로 변경하여 새로운 시나리오 단계가 추가되는 과정을 나타낸다. 그림에서 invoker는 시나리오 단계 추가를 시작하는 객체를, scenario는 유스케이스에서 따로 유지되는 시나리오 객체를 의미한다. currentStep은 시나리오에서 현재 단계를, newStep은 추가되는 새로운 시나리오 단계를 의미한다.

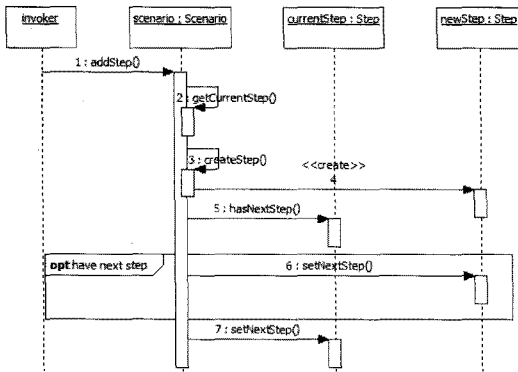


그림 11. 시나리오 단계 추가 시퀀스 다이어그램  
Figure 11. The sequence diagram of adding new scenario step

### 2.3.3. 시나리오 단계 삭제

시나리오 단계 삭제는 시나리오에서 기존에 존재하는 단계를 삭제하는 것으로 단계 삭제 순서는 그림 12와 같다.

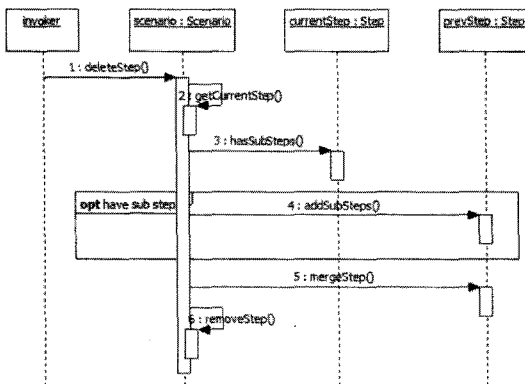


그림 12. 시나리오 단계 삭제 시퀀스 다이어그램  
Figure 12. The sequence diagram of deleting scenario step

그림 12와 같이 시나리오 단계를 삭제할 때 삭제하고자 하는 단계에 하위 단계가 있을 경우 하위 단계를 어떻게 처리해야 되는지를 고려해야 한다. 본 논문에서는 삭제를 원하는 단계 이전 단계의 하위 단계로 이동으로 처리한다. 그림에서 invoker, scenario, currentStep은 시나리오 단계 추가와 같으며, newStep은 추가되는 새로운 시나리오 단계를 의미한다.

### 2.3.4. 시나리오 단계 수준 내리기

시나리오 단계 수준 내리기는 시나리오에서 기존에 존재하는 단계에 대해서 수준을 내려 특정 단계의 하위 단계로 만드는 것으로 수행 절차는 그림 13과 같다. 그림에서 보는 바와 같이 시나리오 단계의 수준을 내릴 때 이에 대한 처리를 확인할 수 있다. 이때 하위 단계가 있다면 수준을 함께 내린다. 수준을 내리는 것은 기존 상위 단계와의 연결이 끊어지는 것을 의미하며 이 경우 새로운 상위 단계가 필요하다. 본 논문에서는 수준을 내리는 단계 이전의 단계를 새로운 상위 단계로 설정한다. 그림에서 나타난 객체는 시나리오 단계 삭제와 같은 역할을 수행한다.

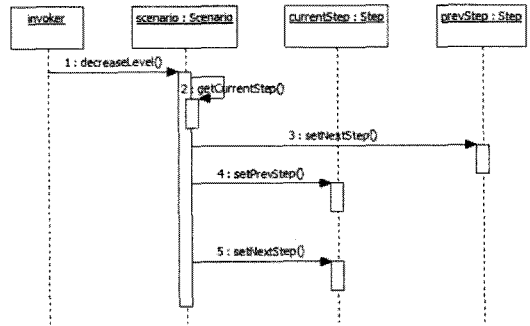


그림 13. 시나리오 단계 수준 내리기 시퀀스 다이어그램  
Figure 13. The sequence diagram of decreasing scenario step level

### 2.3.5. 시나리오 단계 수준 올리기

시나리오 단계 수준 올리는 것은 시나리오에서 기존에 존재하는 단계에 대해서 수준을 올리는 것으로 이를 처리하는 과정은 그림 14와 같다.

그림 14와 같이 시나리오 단계의 수준을 올릴 때 현재 단계 이후에 같은 수준의 연속되는 단계가 있을 경우 중간 수준의 올라갔을 때 이후의 단계는 기존의 상위 단계와 연결이 끊어지기 때문에 이에 대한 처리를 고려해야 한다. 본 논문에서는 수준이 올라가는 단계 이후의 단계들을 수준이 올라가는 단계의 하위 단계로 이동으로 처리하여 연결이 유지되도록 한다.

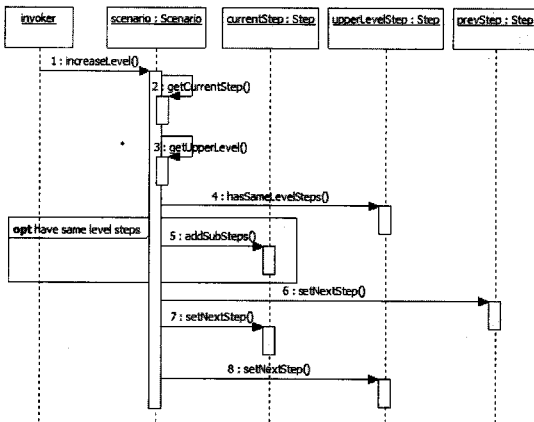


그림 14. 시나리오 단계 수준 올리기 시퀀스 다이어그램  
Figure 14. The sequence diagram of increasing scenario step level

그림 14에서 나타난 invoker, scenario, currentStep, prevStep 객체는 시나리오 단계 삭제와 같은 역할을 수행하며, upperLevelStep은 현재 단계의 상위 단계를 의미한다.

### III. 유스케이스 기술서 작성 도구의 구현

본 장에서는 유스케이스 기술서 작성 도구를 구현한 내용을 기술하며, 그림 15는 시스템의 구현 내용을 클래스 다이어그램으로 표현한 것이다.

그림 15의 각 클래스의 역할을 설명하면 다음과 같다. Project 클래스는 유스케이스 기술서 작성기에서 각각의 프로젝트를 관리하기 위한 클래스로써 각 프로젝트에 해당하는 Actor와 UseCase를 관리한다. Actor 클래스는 유스케이스 기술서에서 주요액터로 사용되는 각각의 액터에 대한 정보를 저장하는 클래스이다. UseCase 클래스는 유스케이스 기술서에 대한 정보를 저장하는 클래스이다. UseCase 클래스는 앞의 유스케이스 기술에 필요한 항목을 기초로 하여 각 항목들을 속성으로 추출하였으며, 복잡도가 높은 시나리오는 따로 구성된 클래스를 참조한다. Scenario 클래스는 UseCase의 주요 성공 시나리오를 저장하는 클래스로 actorAreaSteps와 system

AreaSteps는 시나리오에서 각각 액터 영역과 시스템 영역의 시나리오 단계 유지를 위한 루트 역할을 한다. Step 클래스는 시나리오에서 각 단계에 대한 정보를 저장하는 클래스이며 시나리오 저장을 위한 요소를 기초로 하여 각 속성을 추출하였다. Extension 클래스는 시나리오 단계의 확장에 대한 정보를 저장하는 역할을 하는 클래스이다.

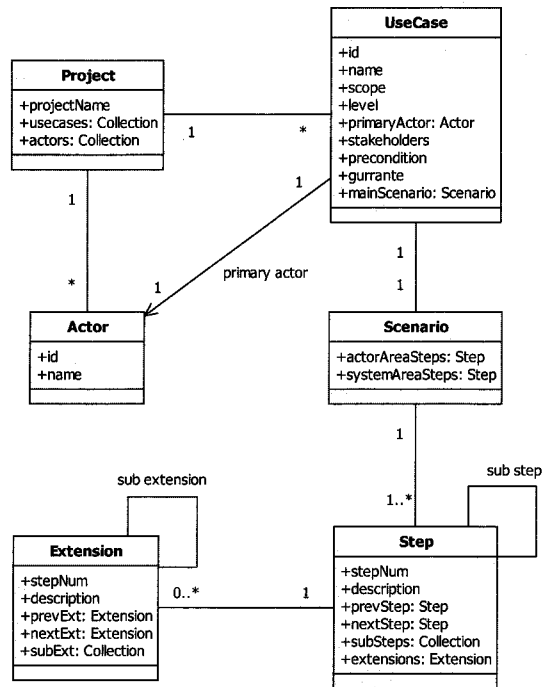


그림 15. 기술서 작성 도구의 클래스 다이어그램  
Figure 15. Class diagram of use case description writing tool

그림 16은 유스케이스 명세서 작성기의 화면을 Swing[10]을 이용해서 구현한 내용을 보인다. 액터와 유스케이스를 프로젝트별로 구분하기 위한 프로젝트의 리스트를 보여주기 위해 A영역을 구성하였으며, A영역에서 선택된 프로젝트에 등록된 액터의 리스트를 보여주기 위해 B영역을 구성하였다. C영역은 A영역에서 선택된 프로젝트에 등록된 유스케이스 리스트를 보여주기 위해 구성하였다. D영역은 유스케이스 명세서 작성 영역으로서 이름, 액터, 주요 성공 시나리오, 확장과 같은 유스케이스 기술에 필요한 항목들을 작성할 수 있도록



록 구현하였다.

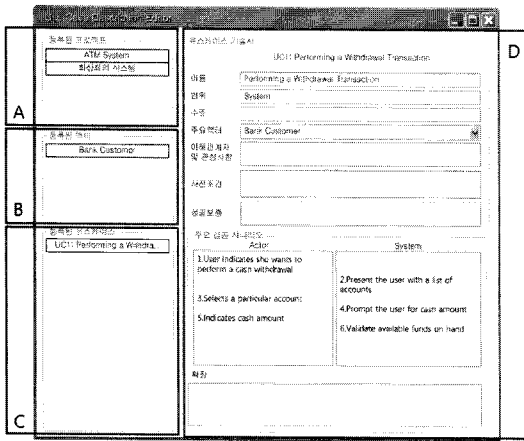


그림 16. 명세서 작성기의 화면구성 및 유스케이스 작성 예

Figure 16. UI of the use case description writing tool and example of use case description

#### IV. 결 론

유스케이스가 제안된 이후 현재까지 시스템의 기능적인 요구사항을 추출하고 시스템이 어떻게 사용되는지를 표현하는 분석기술로서 많이 사용되고 있다. 유스케이스 내용의 중요성이 부각되지만 현재 대부분의 유스케이스 모델링을 지원하는 도구는 유스케이스의 다이어그램에 집중되어 있다. 따라서 유스케이스의 내용을 기술할 수 도구가 요구된다. 본 논문에서는 유스케이스 기술서 작성 도구의 개발을 제시하였다. 유스케이스 기술에 필요한 항목들을 선정하였고, 시나리오 작성에서 단계들을 유지하는 구조를 제시하였다. 또한 2컬럼에서 시나리오 단계들 간의 연관을 위한 형식과 시나리오 단계와 확장을 저장하기 위한 요소를 정의하였다. 각 항목들 간의 복잡한 연관 관계를 줄이기 위해 정보를 저장하는 부분을 따로 두고 필요한 항목을 참조하도록 하였으며, 시나리오 편집에 대한 CRUD를 제시하였다. 마지막으로 작성 도구 구현에 필요한 클래스를 식별하고 이들 간의 연관관계와 유스케이스 기술서 작성 도구의 사용자 UI 화면과 도구를 사용한 예를 보였다. 향후 과제로는 유스케이스의 시각적 목차와

이들 간의 연관성이 표현되는 유스케이스 다이어그램 작성과 유스케이스 명세 항목의 수가 많은 만큼 사용자가 필요로 하는 항목을 선택할 수 있도록 사용자 항목 정의가 필요하다.

#### 참고문헌

- [ 1 ] Ivar Jacobson, Magnus Christerson, Patrik Jonsson, Gunnar Overgaard, *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, 1992,
- [ 2 ] OMG Group, *UML Specification*, [www.omg.org](http://www.omg.org).
- [ 3 ] Martin Fowler, *UML Distilled*, 3rd Ed., Addison-Wesley, 2004
- [ 4 ] Larry L. Constantine, "Essential Modeling: Use Case for User Interfaces," *ACM Interactions*, 1995
- [ 5 ] Alstair Cockburn, *Structuring Use Case with Goals*, HaT.Technical Report, 1995
- [ 6 ] Rebecca Wirfs-Brock, "Designing Scenarios," *The Smalltalk Report*, vol3, no.3, 1993.
- [ 7 ] Craig Larman, *Applying UML and Patterns*, 3rd Ed., Prentice-Hall, 2005
- [ 8 ] Kilov. H, "From semantic to object-oriented data modeling," *Proceedings of the First International Conference on System Integration*, 385-393, 1990
- [ 9 ] Alstair Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2001
- [ 10 ] Sun Microsystems, *Java Standard Edition*, [java.sun.com](http://java.sun.com)

#### 저자소개



최 환 복(Hwan-Bok Choi)

2008.2 안동대학교 컴퓨터공학과  
공학사

2008.3-현재 안동대학교  
컴퓨터공학과 석사과정

※ 관심분야 : 객체지향 분석/설계/프로그래밍



김 윤 호(Yun-Ho Kim)

1983. 2. 경북대학교 전자공학과  
공학사

1993. 2. 경북대학교 대학원  
컴퓨터공학과 공학석사

1997. 2. 경북대학교 대학원 컴퓨터공학과 공학박사

1997. 8. 현재 안동대학교 전자정보산업학부 교수

※관심분야: 인터넷 컴퓨팅, 객체지향 분석/설계/프로  
그래밍, 분산객체시스템, 병렬처리