

Software and Data Engineering

■ Software Engineering

- Reconfigurable Test Execution Machine for Embedded System
..... Kim Kyoung Jin · Chung Ki Hyun · Choi Kyung He 243
- Use Case Identification Method based on Goal oriented Requirements Engineering(GoRE)
..... Bokyung Park · R. Youngchul Kim 255

■ Data Engineering

- Predicting Movie Success based on Machine Learning Using Twitter
..... Junyeob Yim · Byung-Yeon Hwang 263
- A Normalization Method of Distorted Korean SMS Sentences for Spam Message Filtering
..... Kang Seung-Shik 271

■ Multimedia Processing

- A Curve Lane Detection Method using Lane Variation Vector and Cardinal Spline
..... Hwan Heo · Gi-Tae Han 277

■ Web Science

- Named Entity Recognition and Dictionary Construction for Korean Title: Books, Movies, Music and TV Programs
..... Yongmin Park · Jae Sung Lee 285

소프트웨어 공학

- 재구성이 가능한 임베디드 시스템 테스트 실행기
김경진 · 정기현 · 최경희 243
- Goal 지향 요구공학 기반의 유스케이스 식별 방법
박보경 · 김영철 255

데이터 공학

- 트위터를 이용한 기계학습 기반의 영화흥행 예측
임준엽 · 황병연 263
- 스팸 문자 필터링을 위한 변형된 한글 SMS 문장의 정규화 기법
강승식 271

멀티미디어 처리

- 차선 변화벡터와 카디널 스플라인을 이용한 곡선 차선 검출방법
허환 · 한기태 277

웹 사이언스

- 한국어 제목 개체명 인식 및 사전 구축: 도서, 영화, 음악, TV프로그램
박용민 · 이재성 285

Use Case Identification Method based on Goal oriented Requirements Engineering(GoRE)

Bokyung Park^{*} · R. Youngchul Kim^{**}

ABSTRACT

Our previous research[1] suggested object extraction and modeling method based on Fillmore's case grammar. This approach had not considered of use case extraction and method. To solve this problem, we adopt Fillmore's semantic method as linguistic approach into requirement engineering, which refine fillmore's case grammar for extracting and modeling use cases from customer requirements. This Refined mechanism includes the definition of a structured procedure and the representation of visual notations for 'case' modeling. This paper also proposes the use case decision matrix to identify use case size from extracted use cases based on goal oriented requirement engineering(GoRE), which related with the complexity of use case, and also prioritizes the use cases with this matrix. It demonstrates our proposal with the bank ATM system.

Keywords : Fillmore Case Grammar, User Behavior Analysis, Goal Oriented Analysis, Use Case Size Decision, Goal Modeling

Goal 지향 요구공학 기반의 유스케이스 식별 방법

박보경^{*} · 김영철^{**}

요약

기존 논문[1]에서는 Fillmore의 Case Grammar를 기반으로 객체 추출 및 모델링 방법을 제안하였다. 이 방법은 유스케이스 추출 및 결정 방법을 고려하지 않았다. 이러한 문제를 해결하기 위해, 본 논문에서는 요구공학에서 자연어 처리를 위해 Fillmore의 의미적 방법을 채택하였다. 즉, 고객 요구사항으로부터 유스케이스를 모델링하고 추출하기 위해 Fillmore의 Case Grammar를 개선한다. 개선된 메커니즘은 구조화된 절차를 정의하고 시각적 표기법을 수행한다. 또한 유스케이스의 복잡성과 관련된 Goal 지향 요구공학(GoRE)을 기반으로 추출된 유스케이스에서 유스케이스 크기를 식별하는 유스케이스 결정 매트릭스(Use Case Decision Matrix)를 제안한다. 이 매트릭스에서 유스케이스를 우선순위화 한다. 사례연구로 은행 ATM 시스템에 적용하였다.

키워드 : 필모어의 격문법, 사용자 행위 분석, 목표 지향 분석, 유스케이스 크기 결정, Goal 모델링

1. 서론

소프트웨어 시스템의 성공적인 개발을 위해서는 정확한 요구사항 분석이 매우 중요하다. 자연어로 된 요구사항을 분석하기 위해서, [1]에서는 Fillmore의 메커니즘을 통해 사용자의 요구를 자연어로 변환한다. 이를 시나리오 기반의

분석 방법을 이용해 요구사항으로 식별하고, 객체 추출 및 모델링한다. 하지만 이 방법은 사용자의 요구만 분석하기 때문에, 고객의 요구는 반영되지 못한다. 따라서 개발자는 고객과 사용자(End-User)의 요구를 동시에 정확히 분석하기 어렵다.

요구사항을 분석하고 추출하는 또 다른 방법은 Goal과 시나리오를 사용하는 것이다. [2]에서는 Goal 모델링과 시나리오 제작 기술을 이용하여 요구사항 추출 방법을 제안하였다. 이 방법은 요구사항 추출에 효과적이지만, 유스케이스와 연결되지 않는다[3]. 유스케이스는 자연어 개념을 기반으로 유스케이스 명세서를 작성하기 때문에, 고객과 최종 사용자의 요구를 정확히 분석할 수 있다. [3]에서는 자연어 개념을 이용한 유스케이스 중심의 분석과 Goal&Scenario 기반의

※ 이 논문은 본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업원천기술개발사업(10035708, 고신뢰 자율제어 SW를 위한 CPS(Cyber-Physical Systems) 핵심 기술 개발)과 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2013R1A1A2011601).

* 준회원: 홍익대학교 컴퓨터정보통신공학과 박사과정

** 정회원: 홍익대학교 컴퓨터정보통신공학과 정교수

Manuscript Received: April 15, 2014

First Revision: April 22, 2014

Accepted: April 22, 2014

* Corresponding Author: R. Youngchul Kim(bob@hongik.ac.kr)

요구사항 추출 기법을 결합하고자 하였다. Goal 모델링은 요구사항을 식별할 수 있는 효과적인 방법이며, 시나리오는 실제 상황을 설명하기 때문에 실제 요구사항을 수집하기 용이하다. 하지만 이 방법은 유스케이스의 크기(Size)는 고려되지 않는다. 소프트웨어 개발 시, 프로젝트의 노력, 비용 및 시간을 도출하기 위한 방법으로 유스케이스의 크기는 매우 중요하다[4]. 정확히 소프트웨어의 크기를 추정하기 위해서 Karner는 Use Case Point를 제안하였다. 이 방법은 프로젝트 추정에 많이 활용되고 있지만, 정확성 손실이 발생할 수 있다[5]. 또한 트랜잭션의 수는 분석가에 따라 달라질 수 있으며, 유스케이스 작성 및 세부방법에 따라 복잡성 분류가 달라질 수 있다. 이를 해결하기 위해 [4]에서는 Use Case Point를 개선하여 프로젝트의 크기 추정 방법을 제안하였다. 이 방법은 초기 프로젝트에 적용될 수 있으며, 소프트웨어 개발이 반복 수행되거나 새로운 요구사항이 발견될 때 재적용될 수 있다. 하지만 요구사항으로부터 유스케이스 도출 방법은 제시되지 않았다.

본 논문에서는 자연어 요구사항으로부터 유스케이스 추출 및 결정 방법을 제안한다. 이를 위해, Fillmore의 Case Grammar를 적용하여 Goal 기반의 유스케이스를 식별한다. 또한 유스케이스 결정 매트릭스를 통해 요구사항과 유스케이스 간의 연관관계를 분석하여 유스케이스의 크기를 결정한다. 유스케이스 크기는 연관관계가 강할수록, 요구사항의 수가 많을수록 증가한다. 이것은 유스케이스가 복잡하다는 것을 의미한다. 따라서 유스케이스의 복잡도에 따라 유스케이스의 우선순위를 결정할 수 있다. 우선순위를 수행하는 이유는 모든 유스케이스를 우선순위화하여 유스케이스 기반의 테스트 케이스를 추출하고자 함이다. 즉, 테스트 케이스 Coverage의 정량화를 위해 테스트 케이스 우선순위를 수행한다. 본 논문에서 추출된 결과는 유스케이스 기반의 테스트 단계로 확장하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 Fillmore의 격 문법과 사용자 행위 분석 기반 요구추출 방법에 대해 소개한다. 3장은 개선된 Fillmore의 메커니즘과 Goal 모델링을 통한 Use Case 추출 및 크기 결정 방법을 설명한다. 4장에서는 사례연구로 은행 ATM 시스템에 적용하였으며, 5장에서는 결론 및 향후 연구를 기술한다.

2. 관련 연구

2.1 Fillmore의 Case Grammar

기존 Fillmore의 Case Grammar는 술어(동사)를 중심에 두고, 각 단어의 의미적인 관계를 기술하여 의미구조를 표현한다[7,8]. 문장은 동사와 보편적 어휘(Universal Vocabulary)에서 추출된 격(Case)이다. 격은 동사의 의미적 수행에 필요한 의미 관계를 말한다[8]. 격에 대한 종류와 정의는 수행된 연구마다 조금씩 다르다. Fillmore(1971)가 정의한 9개의 격

Table 1. The Existing Fillmore's Case [6]

Case	Definition
Agent	the instigator of the event
Counter-agent	the force or resistance against which he action is carried out
Object	the entity that moves or changes or whose position or existence is in consideration
Experiencer	the entity which receives or accepts or experiences or undergoes the effect of an action
Source	the place from which something moves
Goal	the place to which something moves
Locative	location or spatial orientation
Instrument	the stimulus or immediate physical cause of the event
Result	the entity that comes into existence as a result of the action

은 각 단어의 의미적 관계를 기술하여 의미구조를 표현한다. 또한 동사와 관련된 동일한 의미를 가진 명사구를 식별할 수 있다. 표 1은 Fillmore가 제시한 격의 정의를 나타낸다.

2.2 사용자 행위 분석 기반 요구추출 방법

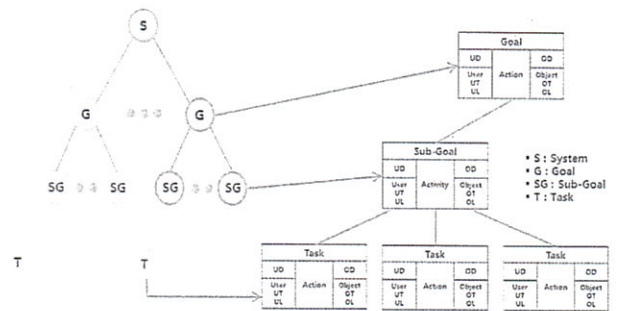


Fig. 1. User Behavior Structuring

기존 논문에서는 사용자의 행위를 Goal 중심의 분석을 통해 요구사항 추출 방법을 제안하였다[9]. 이 방법은 사용자 정보(User Data)를 비교·분석하기 위해 육하원칙(5W1H) 기반으로 정리하여 사용자 정황 정보(User Context Data)를 생성한다. 이 데이터는 사용목적과 사용자, 그리고 대상물 관계 중심의 관점으로 분석하여 사용자행태 정보(User Behavior Data)를 구조화한다. 구조화된 데이터는 사용자의 행동, 행위 간 연관관계를 분석하여 요구사항을 추출한다. 여기서, Goal 기반의 분석 방법은 Goal 중심의 사용자행태 정보(Goal Oriented User Behavior Data)를 만드는 것이다[9]. 사용자의 행동(Action)을 각각의 Sub-Goal로 연계하여 행위(Activity)를 도출하고, 이를 행태(Behavior) 별로 정의한다.

그림 1은 [9]에서 제안한 사용자행태 구조화이다. Goal은 다수의 Sub-Goal로 구성되어 있으며, 각각의 Sub-Goal은 다수의 Task로 구성된다. 여기서 Task는 User Data(UD), Action,

Object Data(OD)로 구성된다. 사용자의 행동(Action)을 최소 단위로 규정하였다. 행위(Activity)는 한 명의 사용자가 자신의 고유한 방법과 절차를 가지고 태스크를 수행한다[9]. 사용자 행태는 각각의 태스크를 수행하기 위해 행하는 행위의 유형(Pattern)이다. Goal에 따라 Sub_Goal이 다르게 되어 이를 수행하는 행위(Activity)의 조합이 달라진다. 즉, 태스크에서 User Data와 Object Data는 Action을 중심으로 연결된다. Action을 중심으로 연결된 Task들이 모여 Activity가 된다. 그런 다음 Activity들을 Behavior별로 정의한다.

3. Goal 지향 요구공학 기반의 Use Case 추출

3.1 개선된 Fillmore의 표기 정의

Goal을 기반으로 자연어로 된 요구사항을 식별하기 위해 Fillmore의 Case Grammar를 이용한다. 이 방법은 동사를 중심으로 각 단어의 의미 관계를 분석하기 때문에 자연어 처리에 용이하다. 다른 문법형태로 작성된 요구사항은 의미 관계 분석을 통해 같은 분석 결과를 얻을 수 있다. 또한 동사와 연관된 동일한 의미를 지닌 명사구 식별과 처리가 가능하다[10]. 따라서 본 논문에서는 자연어로 된 요구사항으로부터 유스케이스를 구조화하기 위해 6개의 격을 정의한다. 표 2는 개선된 Fillmore의 격이다.

Table 2. Refined Fillmore's Case

Fillmore's Case	Definition
Agent	the instigator of the event
Object	the entity which receives or accepts or experiences or undergoes the effect of an action
Source	location or spatial orientation
Instrument	the stimulus or immediate physical cause of the event
Verb	the entity that comes into existence as a result of the action
Goal	The system goals that achieve from the needs of Users

Table 3. The Notation for Use case Modeling

Notation	Definition	Notation	Definition
	Goal		Verb
	Actor		Object
	Source		Instrument
	Semantic Relation		Semantic Relation

각각의 격을 식별한 다음, 유스케이스 표기법을 이용하여 모델링한다. 표 3은 유스케이스 모델링을 위한 Fillmore의 개선된 표기법이며, 이를 기반으로 그림 2와 같이 구조화한다.

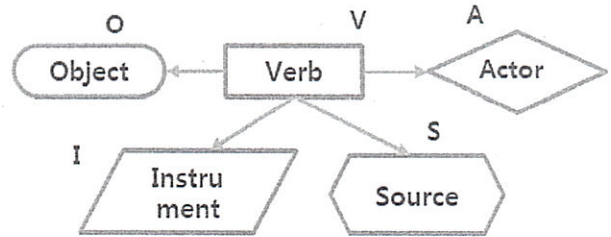


Fig. 2. Refined Structured Requirements Method

3.2 개선된 Fillmore Case Grammar 기반의 Use Case 식별 방법

그림 3은 Fillmore 격문법 기반의 Goal 지향 유스케이스 추출 및 결정 과정이다. 이 과정은 Goal Modeling과 유스케이스 우선순위화 단계로 구성된다. Goal Modeling 단계는 구조화된 요구사항을 Goal 기반으로 구조화하여 행동(동사, Verb) 간의 연계성을 도출한다. Goal 모델링은 요구사항을 식별하는 데 효율적이기 때문에[3], 요구사항 분석을 통한 유스케이스 구조화 및 Goal 기반의 요구사항 분류가 가능하다. 본 논문에서는 사용자행태 분석을 위한 Task 구조 대신 격(Case) 기반의 요구사항 구조화 방법(Step 1)을 적용한다. 그림 4에서 Goal은 개발 중인 시스템의 Goal을 말하며, 이는 다수의 Sub-Goal과 연관된다. 시스템은 다수의 유스케이스로 구성되기 때문에, Sub-Goal은 유스케이스의 Goal이다. 각각의 유스케이스는 다수의 요구사항(Requirements)들로 구성되며, 1단계에서 구조화된 요구사항을 적용한다.

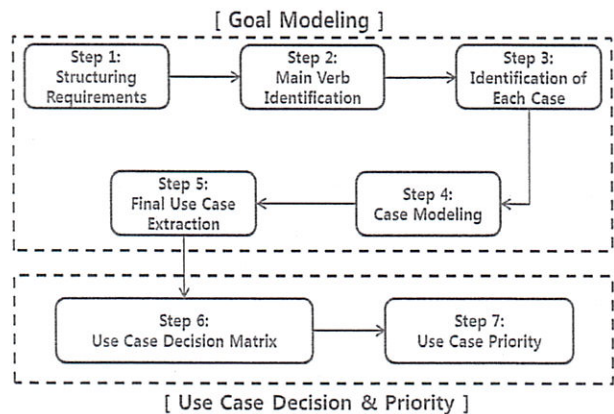


Fig. 3. Goal Oriented Use Case Extraction and Size Decision Process

■ Step 1: Structuring Requirements

요구사항을 구조화하기 위해, 자연어 문장을 제시한 문장 데이터로 전환하는 과정이 필요하다. 표 2의 개선된 Fillmore의 Case를 기반으로 자연어 문장을 분석한다. 분석한 격들은 그림 5 형식에 맞게 작성한다.

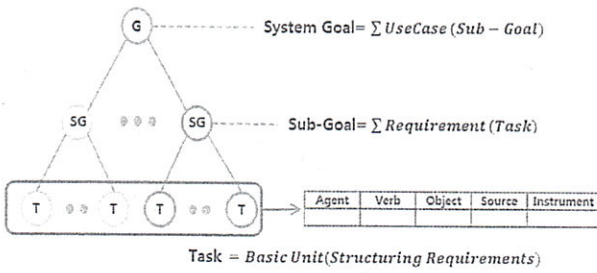


Fig. 4. Refined Goal Hierarchy

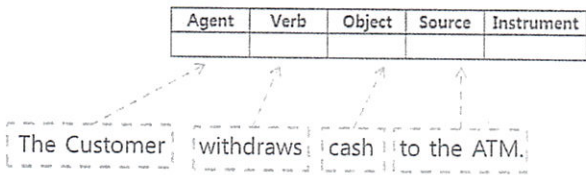


Fig. 5. Structuring Requirements Template

■ Step 2: Main Verb Identification

이 단계에서는 구조화된 요구사항에서 동사들을 리스트화 하고, 문단 단위로 Main Verb를 식별한다. 식별된 Main Verb는 개선된 요구사항 구조화 방법으로 모델링한다. 표 4 는 구조화된 요구사항을 리스트화하고, Main Verb를 식별 한 것이다.

Table 4. Main Verb List

Agent	Verb	Object	Source	Instrument
ATM	Display	Error Message	Customer	
Customer	Select	Withdraw Service	Customer	
ATM	Display	Account	Customer	
Customer	Select	Account	ATM	Button
ATM	Display	Account	Customer	
Customer	Check	Information	ATM	
Customer	Withdraw	Cash	ATM	
ATM	Stop	Transaction	Bank System	

그림 6은 식별된 Main Verb를 중심으로 해당 문장을 개 선된 요구사항 표기법으로 변환한 것이다.

■ Step 3: Case Identification based on Main Verb

격 식별 단계는 Main Verb에 따라 의미적으로 영향을 받 는 각각의 격을 식별한다. 식별된 격들과 연관된 다른 문장 의 동사를 식별한다. 단, 이 과정은 식별된 격들이 포함되어 있는 문단 내에서 수행해야 한다. 즉, 문장별로 연관된 격을 Verb->Actor->Object->Source->Instrument 순으로 식별 한다. 더 이상 비교할 격이 없을 때까지 반복해서 식별한다. 그림 6에서 Main Verb는 withdraws이고, 의미적으로 영향

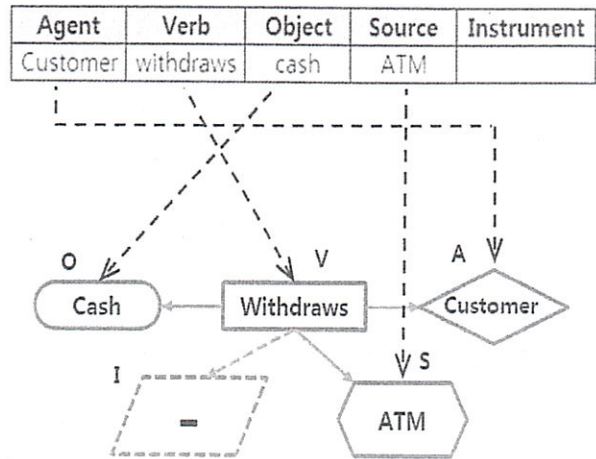


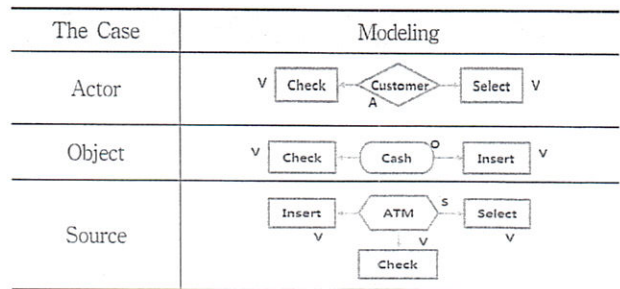
Fig. 6. Refined Structured Requirements of Main Verb

을 받는 격은 Actor(Customer), Object(Cash), Source (ATM) 이다. 각각의 격과 연관된 Verb를 식별한다. Actor (Customer) 와 연관된 Verb는 Select, Check이다. 이를 연결하고, Object (Cash)와 연관된 Verb를 분석한다. Object와 연관된 Verb는 Insert, Check이며, Source(ATM)와 연관된 Verb 는 Select, Insert, Check이다.

■ Step 4: Case Modeling

격 모델링 단계는 의미적으로 연관된 격들을 모델링한다. Main Verb에 따라 식별된 각각의 격을 개선된 요구사항 구 조화 방법으로 모델링한다. 표 5는 Main Verb에 따라 식별 된 격 모델링 과정이다.

Table 5. Case Modeling



■ Step 5: The Final Use Case Extraction

각각의 격을 모델링 한 다음, 의미적으로 연관된 다른 격 을 식별한다. 연관된 격끼리 연결한다. 이때, 격 비교를 통해 서 동일한 관계인 격들을 분석하고, 하나로 통합한다. 이러 한 과정을 통해 Main Verb와 의미적으로 연관된 요구사항 을 추출한다. Main Verb를 중심으로 추출된 하나의 요구사 항 덩어리는 그림 4의 Sub-Goal이 되며, 이는 하나의 유스 케이스로 볼 수 있다. 5단계까지의 과정이 Goal Modeling 과정이다.

	UC ₁	UC ₂	UC ₃	UC ₄	UC ₅	
CR ₁	●			○	○	UC Use Case
CR ₂		●	○			CR Customer Requirements
CR ₃		○	R _{ij}	●	○	R _{ij} Correlation
CR ₄		○			○	CP Correlation Point(Strong●: 9, Average○: 3, Weak○: 1)
CR ₅		○		○		UCSP Use Case Size Point
UCSP	9	16	3	13	3	

Fig. 7. Use Case Decision Matrix

■ Step 6: Use Case Decision Matrix

이번 단계에서는 요구사항과 추출된 유스케이스를 기반으로 유스케이스 결정 방법을 제안한다. 그림 7은 유스케이스의 크기를 식별하기 위한 결정 매트릭스이다.

유스케이스의 크기는 유스케이스와 요구사항 간의 연관관계가 강할수록, 유스케이스에 포함되는 요구사항의 수가 많을수록 증가한다. 그림 7에서 가로축은 Fillmore의 메커니즘을 통해 추출된 유스케이스이며, 세로축은 식별된 고객 요구사항이다. 요구사항과 유스케이스 간의 연관관계 점수(R_{ij})의 Correlation Point: CP)와 요구사항의 개수를 계산하여 유스케이스의 크기(UCSDP: Use Case Size Decision Point)를 결정한다.

$$UCSDP = (CP_1 \times l) + (CP_2 \times m) + (CP_3 \times n)$$

$$= 9l + 3m + n \quad (1)$$

(단, l: CP₁(9)의 개수, m: CP₂(3)의 개수, n: CP₃(1)의 개수)

■ Step 7: Use Case Priority

이전 단계에서 추출한 유스케이스 크기 점수를 비교하여 유스케이스의 우선순위를 결정한다. 유스케이스 크기 점수가 동일할 때에는, 각각의 조건을 고려한다. 그림 7에서, UC5의 크기 점수는 9l+3m+n = 9×0+3×0+1×3 = 3이다. UC4는 9l+3m+n = 9×1+3×1+1×1 = 13이다. 여기서 두 유스케이스에 연관된 요구사항의 개수는 같지만, 연관관계가 UC4가 더 강하기 때문에 UC4의 크기 점수가 더 크다. 또한 UC2와 UC4는 같은 연관관계(9, 3, 1점)를 각각 가지고 있다. 하지만 유스케이스에 포함된 요구사항의 개수가 UC2가 더 많기 때문에 유스케이스 크기 점수가 더 크다(UC2: 9l+3m+n = 9×1+3×2+1×1 = 16, UC4: 9l+3m+n = 9×1+3×1+1×1 = 13). 따라서 유스케이스 크기는 연관관계가 강할수록, 요구사항의 수가 많을수록 증가한다는 것을 알 수 있다.

4. Case Study

사례연구로 은행 ATM 시스템에 적용하였다. 요구사항 구조화 단계(Step 1)에서는 자연어로 된 ATM 요구사항을 그림 8과 같이 구조화한다. 하나의 문장을 각각 분석하고,

Agent	Verb	Object	Source	Instrument

The customer selects deposit service using the button in the ATM. ATM safely transmits account transaction to bank system. ATM requests password to customer. ATM displays "Error Messages" to the customer.

Fig. 8. Structuring Requirements of ATM

해당되는 격에 배치한다.

요구사항을 구조화한 후, 문단단위로 동사들을 리스트화하여 Main Verb를 식별한다(Step 2). 표 4는 구조화된 ATM 요구사항에서 Main Verb 중 Deposit를 식별한 것이다.

Table 6. Main Verb Identification(ATM)

Agent	Verb	Object	Source	Instrument
Customer	Select	Deposit Service	ATM	Button
Customer	Insert	Card	ATM	
ATM	Request	Card	Customer	
ATM	Request	Password	Customer	
Customer	Select	Check Service	ATM	Button
ATM	Display	Error Message	Customer	
Customer	Input	Account	ATM	
Customer	Select	Transfer Service	ATM	Button
Customer	deposit	Cash	ATM	
ATM	Request	Account	Customer	
Customer	Select	Account	ATM	Button
ATM	Transmit	Trnsaction	Bank System	

그림 9는 식별된 Main Verb(Deposit)와 연관된 해당 문장을 개선된 요구사항 표기법으로 변환한 것이다.

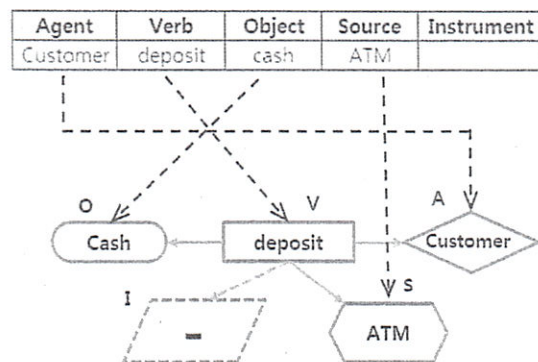


Fig. 9. Refined Structured Requirements of "Deposit" Main Verb

Main Verb(Deposit)에 따라 의미적으로 영향을 받는 각각의 격을 식별한다(Step 3). 그림 8에서 Main Verb는 Deposit이고, 의미적으로 영향을 받는 격은 Actor(Customer), Object (Cash), Source(ATM)이다. 각각의 격과 연관된 Verb를 식

별한다. 표 7은 "Deposit" Main Verb에 따라 식별된 각각의 격 모델링 과정이다.

Table 7. Case Modeling of "Deposit" Verb

The Case	Modeling
Actor	
Object	
Source	

각각의 격을 모델링한 다음, Main Verb와 의미적으로 연관된 격끼리 연결한다(Step 5). 격 비교를 통해서 동일한 관계인 격들을 분석하고 그림 10과 같이 설정하였다.

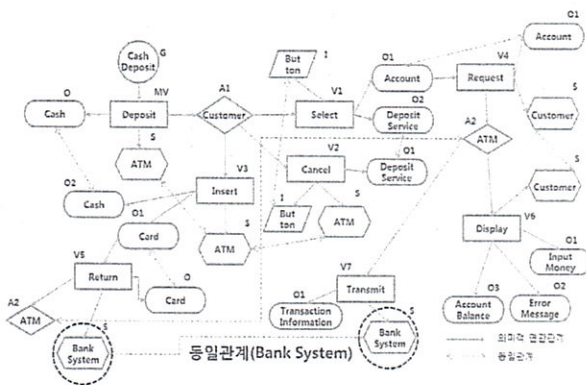


Fig. 10. Case Relation

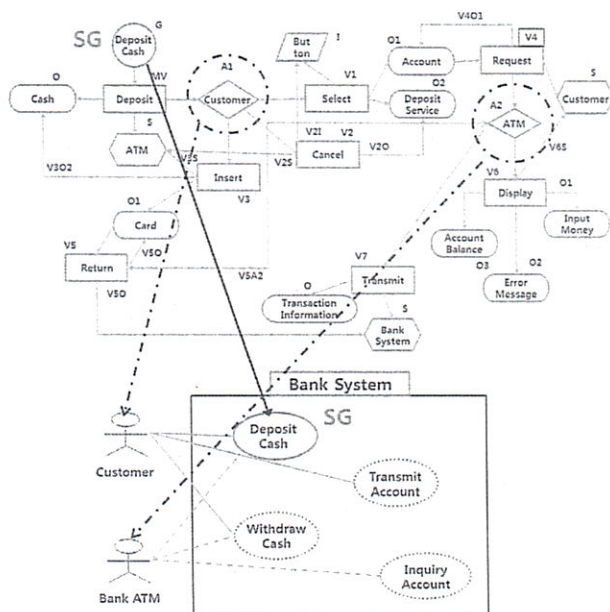


Fig. 11. "Deposit" Use Case

의미적 관계는 \longrightarrow 화살표로 표시하였으며, 동일관계는 \longleftarrow 화살표로 표시하였다. 동일 관계일 경우, 각각의 격들은 하나로 통합한다. 그림 10에서 Return 동사의 Source 격 Bank System과 Transmit 동사의 Source 격 Bank System은 동일관계이다. 따라서, 각각 표시된 격을 하나로 통합한다. Deposit Cash 유스케이스에서는 총 11개의 동일관계를 식별하였다. 그림 11에서는 이를 통합하여 하나로 표시하였다.

그림 10에서 각각의 격들을 분석하고, 하나로 통합한다. 이러한 과정을 통해 최종 유스케이스를 추출한다. 그림 11은 최종적으로 추출된 'Deposit Cash' 유스케이스이며, SG는 최종적으로 추출된 Sub_Goal이다.

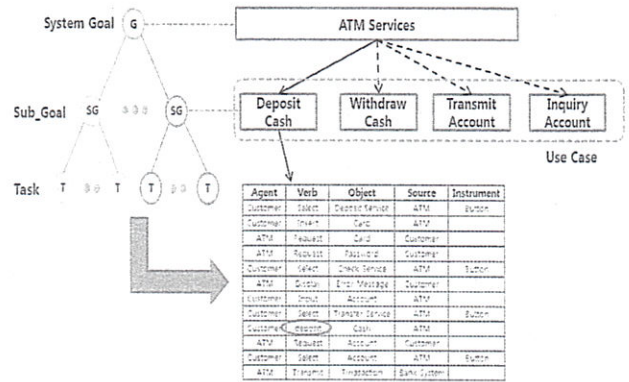


Fig. 12. Goal Hierarchy for ATM

ATM의 요구사항에서는 최종적으로 총 4가지 유스케이스(Deposit Cash, Withdraw Cash, Transmit Account, Inquiry Account)가 추출되었다(그림 12). 유스케이스 이름은 MainVerb와 Noun으로 구성하였다. 그림 11을 통해서 각각의 유스케이스에 포함된 격들의 연관관계는 식별 가능하나, 유스케이스의 크기가 어느 정도인지 알 수 없다. 따라서 유스케이스 결정 매트릭스를 이용하여 유스케이스의 크기를 산출한다(Step 6). 가로축은 추출된 유스케이스를, 세로축은 식별된 요구사항을 적용한다. 본 논문에서는 지면 관계상 ATM 유스케이스 결정 매트릭스는 생략한다. 표 8은 유스케이스별 요구사항의 개수와 UCSDP(Use Case Size Decision Point)를 나타낸 것이다. 식별된 ATM의 요구사항은 총 52개가 추출되었다.

표 8에서 CP는 연관관계 점수(Correlation Point)를 나타내며, CP1은 9점, CP2는 3점, CP3는 1점을 나타낸다. Deposit Cash 유스케이스와 연관된 요구사항은 26개(Number of Requirements)이며, CP1인 요구사항의 개수는 12개, CP2인 요구사항은 9개, CP3인 요구사항은 5개였다.

$$\text{Deposit Cash} = (9 \times 12) + (3 \times 9) + (1 \times 5) = 140$$

$$\text{Transfer Account} = (9 \times 20) + (3 \times 14) + (1 \times 6) = 228$$

$$\text{Inquiry Account} = (9 \times 2) + (3 \times 6) + (1 \times 6) = 42$$

$$\text{Withdraw Cash} = (9 \times 12) + (3 \times 7) + (1 \times 1) = 130$$

식(1)을 이용하여 유스케이스의 UCSDP를 계산한다.

Table 8. The Number of Requirements & UCSDP

	Deposit Cash	Transfer Account	Inquiry Account	Withdraw Cash
CP1(9)	12	20	2	12
CP2(3)	9	14	6	7
CP3(1)	5	6	6	1
Number of Requirements	26	40	14	20
UCSDP	140	228	42	130

표 8의 결과에 따라, 유스케이스는 Transfer Account > Deposit Cash > Withdraw Cash > Inquiry Account 순으로 우선순위가 된다. 또한 유스케이스 크기는 연관관계가 강할수록, 요구사항의 수가 많을수록 크다는 것을 알 수 있다.

5. 결 론

본 논문에서는 자연어 처리 기법과 Goal 기반의 분석을 통해 유스케이스를 추출하고, 유스케이스의 결정 및 우선순위가 방법을 제안한다. 자연어로 된 요구사항을 식별하기 위해 Fillmore의 Case Grammar를 개선하였다. 이 방법은 동사를 중심으로 각 단어의 의미관계를 분석하여 구조화하기 때문에 자연어 처리에 용이하다. 구조화된 요구사항은 Goal 기반의 분석을 통해 유스케이스를 추출한다. Goal 모델링은 Goal 기반으로 요구사항을 구조화하여 행동 간의 연계성을 도출하기 때문에, 유스케이스 구조화가 가능하다. 추출된 유스케이스는 연관된 요구사항을 식별하여, 유스케이스 결정 매트릭스에 적용한다. 유스케이스 크기는 연관관계가 강할수록, 요구사항의 수가 많을수록 증가한다. 하지만 제안한 방법은 각각의 문장을 분석해야 하므로, 많은 시간과 노력이 든다. 향후에는 다양한 사례 연구가 더 필요하며, 구조화된 요구사항의 효율적인 분석 방법을 연구하고자 한다. 또한 제안한 방법의 최종 목적은 유스케이스 기반의 테스트 케이스 추출이다. 따라서 추출된 결과를 유스케이스 기반의 테스트 단계로 확장할 것이다.

Reference

- [1] SungBin Ahn, Dongho Kim, Chaeyun Seo, R. Youngchul Kim, "Object Extraction and Modeling Method from the User Requirements with Fillmore's Case Grammar," KSEJW 2010, Vol.8, No.1, pp.98-99, 2010.
- [2] A. I. Anton, "Goal-based requirements analysis," in Proceeding of the Second International Conference on Requirements Engineering(ICRE '96), Colorado, 1996, pp.136-144.
- [3] Jintae Kim, Sooyong Park, and Vijayan Sugumaran, "A Linguistics-Based Approach for Use Case Driven Analysis Using Goal and Scenario Authoring," Natural Language Processing and Information Systems Lecture Notes in Computer Science, Vol.3136, pp.159-170, 2004.
- [4] Gustavo Bestetti Ibarra, Patricia Vilain, "Software Estimation Based on Use Case Size," 2010 Brazilian Symposium on Software Engineering(SBES), 2010, pp.178-187.
- [5] G. Karner, "Resource Estimation for Objectory Projects," Objectory Systems, 1993.
- [6] C. J. Fillmore, "The Case for Case," Universals in Linguistic Theory, Bach&Harms Eds. NewYork: Holt, Rinehart& Winston, 1967.
- [7] Byungmi Lim, "A Study on Fillmore's Case Grammar," Ph.D. dissertation, University of Sangmyung Women's, Korea, 1987.
- [8] Bokyoung Park, Soyoung Moon, Kidu Kim, Boyeon Kim, R. Youngchul Kim, "Use Case Oriented Requirements Engineering for improving the pervious ViRE's Process", The 38th KIPS Fall Conference, Vol.19, No.2, pp.1497-1499, 2012.
- [9] Jihong Jung, "User behavior analysis framework on ubiquitous environment," Ph.D. Dissertation of Hongik, 2009.
- [10] Boyeon Kim, "Use Case Extraction Method of Customer Requirements Based on Refined Fillmore Case Grammar Mechanism," M. D. Dissertation, University of Hongik, 2013.

박 보 경



e-mail : bkpark@mail.hongik.ac.kr
2003년 홍익대학교 컴퓨터정보통신공학과
(학사)
2011년 홍익대학교 소프트웨어공학(석사)
2010년~현 재 홍익대학교 컴퓨터정보통신
공학과 박사과정

관심분야: 소프트웨어공학, 요구공학, 테스트 프로세스, 모델 기반
테스팅

김 영 철



e-mail : bob@hongik.ac.kr
2000년 일리노이공대 전산과(공학박사)
2000년~2001년 LG산전 중앙연구소
Embedded system 부장
2001년~현 재 홍익대학교 컴퓨터정보통신
공학과 정교수

관심분야: 테스트 성숙도 모델(TMM), 임베디드 소프트웨어 개발
방법론, 모델 기반 테스트, 메타모델, 비즈니스 프로
세스 모델, 사용자 행위 분석 방법론 등