

정보과학회논문지 : 컴퓨팅의 실제 및 레터

Journal of KIISE : Computing Practices and Letters

VOLUME 19, NUMBER 4, APRIL 2013

내장형 시스템

- 프로그램 실행 지연시간 단축과 소비전력 감소를 위한 명령어 캐시 이미지 일괄 로딩 기법 서효중, 김태현 163

소프트웨어 공학

- 기존 ViRE와 개선된 GoRE 프로세스 비교 연구 박보경, 김영철 171

정보검색

- POI 데이터 검색을 위한 문자열 유사도 측정 기법 진아연, 이재원, 이종우 177

정보분석 서비스

- 효율적 영어 구문분석을 위한 유한 오토마타를 이용한 문장분할 김성동 186

레터 논문

건강정보시스템

- 3차원 비강 모델 및 유동 가시화를 위한 상호 작용기반 분할 기법 서안나, 허고은, 김지인, 김성균 195
변형 정합기법을 이용한 신원이 은닉된 두뇌 모델의 구성 두약유, 이효종 200

네트워크 프로토콜

- 음성 데이터 전송을 위한 멀티 흡 프로토콜 설계 박정연, 황상호, 안병철 205

무선 이동 인터넷

- 모바일 테더링 기반 협력 이종 네트워크의 효용 최대화 노희준, 이지섭, 소성용, 이원준 209

병렬처리

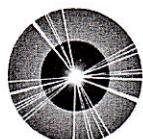
- 집합 I/O를 위한 집합 버퍼 효과 분석 차광호, 조혜영 214

위치 기반 서비스

- WiFi 신호를 활용한 스마트폰 기반 자동출결시스템 신동욱, 김태환, 최중민, 김정선 219

패턴 인식

- 다변량 가우시안 클러스터링 기법을 이용한 비구조화 지면 추출 이홍구, 조성호 224



한국정보과학회

KOREAN INSTITUTE OF INFORMATION SCIENTISTS AND ENGINEERS



기존 ViRE와 개선된 GoRE 프로세스 비교 연구

(A Comparative Study on the ViRE & Refined GoRE's Process)

박 보 경 [†] 김 영 철 ^{‡‡}

(Bokyung Park) (R. Youngchul Kim)

요약 기존 가치혁신 요구공학은 ERRC 분석과 요구사항의 우선순위화를 통해 체계적으로 요구사항을 추출하는 방법이다[1]. 기존 방법은 구조적 개발 방법 기반으로, ERRC 산출 근거 미비와 정량적 수치 표현 보완이 필요하다[2]. 본 연구에서는 현재 대중화된 유스케이스 기반 개발에 적용하도록, 구체적인 시스템의 Value 보다는 Goal을 정의하고, 이에 따른 Goal 지향의 유스케이스 기반 요구공학 메카니즘을 제안한다. 이 절차는 시스템의 Goal을 식별하여, 기본 시스템 단위를 유스케이스로 한다. 또한 Goal 기반의 유스케이스를 식별하여, 요구사항을 추출하고 우선순위화 한다. 이를 위해 기존 ViRE의 ERRC Matrix를 유스케이스 기반의 Goal 중요도 결정 매트릭스로 보완한다. 또한 유스케이스 점수를 이용하여 검증하고, 제안한 방법과 비교, 분석한다.

키워드: 가치혁신 요구공학, Goal 지향 요구공학, 유스케이스 점수, Goal 중요도, 유스케이스

Abstract The existing valued innovative requirement engineering was a way to extract systematically requirements through prioritizing requirement with ERRC[1]. this approach was based on structured method, which needs to complement a quantitatively numerical representation and unclear ERRC based calculation[2]. In this paper, we apply the current popular use case based development with refining ViRE, which is based on Goal instead of value, and also suggest goal oriented requirement engineering mechanism based on modified ViRE. On this process, there is to identify goals with use cases as the basic unit of the system, and prioritizes and extracts requirements within use cases. To do this, we define Use case oriented goal importance decision matrix modified with the ViRE. With this approach, we also compare and analyze our proposed process with the ViRE process, and validate the resulting calculation with the UseCase Point concept.

Keywords: value-innovative requirements engineering, goal oriented requirements engineering, use case point, goal importance, use case

1. 서론

• 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2012-0001845)과 교육과학기술부와 한국연구재단의 지역 혁신인력양성사업으로 수행된 연구결과임

† 학생회원 : 홍익대학교 소프트웨어공학 전공
com02@hongik.ac.kr

‡‡ 정회원 : 홍익대학교 컴퓨터정보통신공학과 교수
bob@hongik.ac.kr
(Corresponding author임)

논문접수 : 2012년 11월 20일
심사완료 : 2013년 1월 25일

Copyright©2013 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨터의 실제 및 레터 제19권 제4호(2013.4)

가치혁신 요구공학(Value-Innovative Requirements Engineering)은 블루오션(Blue-Ocean) 전략을 기반으로 고객의 기대와 요구를 만족시킨다[1]. 이 방법은 가치 기반의 요구사항을 추출함으로써, 기존의 요구사항과 새로운 요구사항을 비교하고 평가한다. ERRC(Eliminate, Reduce, Raise, Create) 분석을 통해 고객의 가치를 부여한다. 또한 공리적 검증으로 고객 가치를 평가하여 새로운 고객 가치를 창출한다.

하지만 이 방법은 구조적 방법을 기반으로 하고 있다. 본 논문은 유스케이스 기반 개발에 맞게 보완·개선하고자 한다. 기존 연구에서 ERRC 결정 매트릭스는 시스템 요소(System Element)를 소프트웨어와 하드웨어 요소로 구분한다. 하지만 분할 기준이 명확하지 않기 때문에

고객 가치 반영 분할 결정(CVPD: Customer Value Based Partitioning Decision) 방법이 제시되었다[3]. 이 방법은 하드웨어와 소프트웨어가 포함되는 시스템에는 적합 하지만 소프트웨어만을 대상으로 하는 시스템에는 적용하기 어렵다. 그리고 ERRC 산출 근거가 미비하기 때문에 정량적인 수치 표현이 어렵다[2].

이런 문제점을 보완하려, Goal 지향 유스케이스 기반의 요구사항 추출 방법을 제안하였다[4-6]. 이 방법은 유스케이스와 요구사항을 추출하고 우선순위화 한다. 이는 시스템의 개발에 맞는 Goal을 식별하고 요구사항에 반영하여 정확한 요구사항을 확보하는데 있다[2]. 기존의 ViRE와 개선한 Goal 지향 유스케이스 기반의 요구공학(Goal Oriented Requirements Engineering) 방법을 비교·분석하고자 한다.

이 논문의 구성은 다음과 같다. 2장에서는 관련연구로 가치혁신 요구공학에 대해서 소개한다. 3장에서는 유스케이스 기반의 Goal 지향 요구공학을 언급한다. 4장에서는 가치혁신 요구공학과 유스케이스 기반의 Goal 지향 요구공학을 비교·분석한다. 마지막으로 5장에서는 결론 및 향후 연구에 대해서 언급한다.

2. 관련 연구

가치혁신 요구공학은 고객 가치를 분석하고, 절차의 명세화와 검증을 통해 새로운 고객 요구사항을 추출한다[1].

그림 1은 5단계로 구성된 가치혁신 요구공학 프로세스이다. 첫 번째 단계에서는 프로젝트의 목표와 범위를 정하고, 시스템 가치 요소를 식별한다. 전략캔버스(Strategy Canvas)로 분석하여 현재의 시장 상태를 파악한다. 새로운 가치요소 식별 단계에서는 과거에 파악하지 못한 새로운 가치요소를 식별하여 우선순위를 정한다. ERRC 분석에서는 고객의 요구사항을 시스템에 할당하

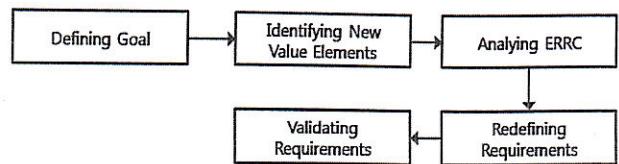


그림 1 가치혁신 요구공학 프로세스

여 새로운 요구사항을 창출한다. ERRC 결정 매트릭스에서는 고객 요구사항(Customer Requirement), 시스템 요소(System Element), 고객 중요성(Customer Importance), 고객 요구사항과 시스템 요소의 상관관계(R_{ij}), 비즈니스 중요성(Business Importance), 상대비용(Relative Cost) 등으로 구성된다. 우선순위화된 요구사항을 1에서 5까지 규모로 고객 중요성을 측정한다. 그리고 고객 요구사항과 시스템 요소의 연관관계를 9, 3, 1점으로 측정하여, 고객 가치를 결정한다. 비즈니스 중요성(BI)과 상대비용(RC)은 식 (1), (2)를 이용하여 계산한다. 상대 비용에서 SE는 시스템 요소를 말한다.

$$BI_j = (CI_i \times R_{1,j}) + (CI_i \times R_{2,j}) + \dots + (CI_i \times R_{i,j}) \quad (1)$$

$$= \sum_{i,j=1}^n (CI_i \times R_{i,j})$$

$$RC_j = (\text{Cost of } SE_j / \text{Cost of } \sum_{j=1}^n SE_j) \times 100 \quad (2)$$

요구사항 재정의 단계에서는 전체 요구사항을 재정의 하여 새로운 시스템에 대한 전략 캔버스를 작성한다. 하이브리드 방법을 통해 ERRC 행동(X, -, +, O)을 결정한다. 마지막으로 요구사항 식별 단계에서는 공리적 접근법을 이용하여 검증한다.

3. 유스케이스 기반 Goal 지향 요구공학

유스케이스 기반의 Goal 지향 요구공학 방법은 가치혁신 요구공학을 개선하려 한 것이다. 유스케이스 단위로

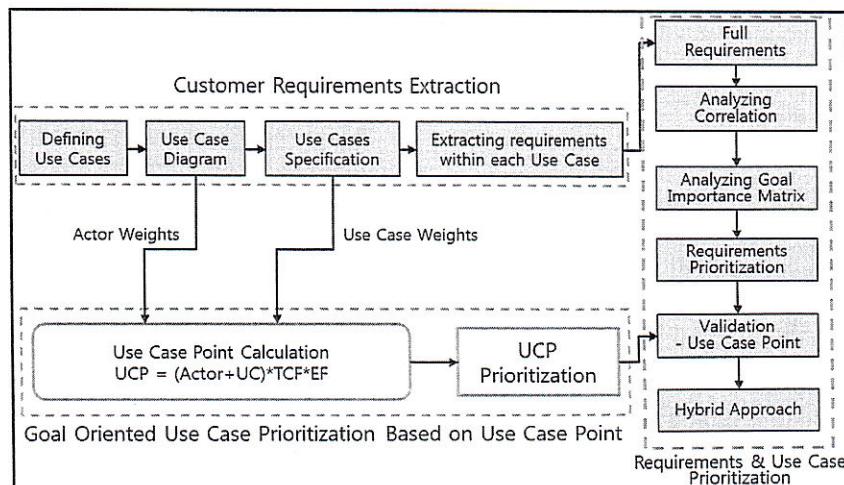


그림 2 유스케이스 기반 Goal 지향 요구공학 프로세스

요구사항을 나누고, Cockburn의 방법[7]을 채택하여 유스케이스의 Goal을 식별한다. 그 이유는 시스템 개발 목적에 맞는 요구사항을 추출하기 위해서이다. 여기서 고객은 시스템 개발을 의뢰한 사람을 말한다. 본 논문에서는 고객의 요구를 반영하는 요구사항을 추출한다. 그럼 2는 유스케이스 기반 Goal 지향 요구공학 프로세스이다.

• Step 1: Customer Requirements

고객 요구사항 추출 과정은 4단계로 구성된다.

1. Defining Use Cases

고객에게서 만들고자 하는 시스템의 요구사항을 수집하고, 정리한다. 문제 기술서를 작성하여 시스템 상황을 분석하고, 유스케이스와 액터를 정의한다. 고객과 개발자간의 의사소통 문제로 동일한 요구사항을 잘못 이해하는 경우가 많다. 따라서 고객의 요구를 정확히 파악하고 수집하는 것이 중요하다.

2. Drawing Use Case Diagrams

이 단계에서는 시스템과 액터를 식별하고, 유스케이스들 간의 관계를 파악한다. 이러한 정보를 토대로 유스케이스 다이어그램을 작성한다. 이때, 각각의 유스케이스에서 Goal을 식별한다.

3. Creating the Goal-Based Use Case Specification

Goal 지향 유스케이스 명세서를 작성하기 위해서 Cockburn[7]이 제시한 Goal 지향 요구사항 방법을 적용한다. 이 방법은 시나리오와 유스케이스를 기술할 수 있다. 유스케이스의 Goal, 액터, 실행 조건, 일반흐름, 대체흐름 등을 기술한다. 여기서 일반흐름, 대체흐름은 Use Case Point 계산에 사용된다.

4. Extracting Requirements within Use Case

Goal 지향 요구사항 명세서를 이용하여 각각의 유스케이스 별로 요구사항을 추출한다. 처음에 정의한 요구사항을 연관된 유스케이스에 매핑한다. 이를 통해 유스케이스 별로 어떠한 요구사항이 적합한지 판단할 수 있고, 보다 정확한 요구사항 추출이 가능하다.

• Step 2: Full Requirements

추출된 고객 요구사항을 정리한다. 고객의 요구사항을 조사, 분석하여 요구사항의 우선순위를 결정한다. 여기서 요구사항의 우선순위란 고객이 중요하게 생각하는 요구사항을 우선순위로 나열한 것을 말한다. 고객 및 사용자들과 협의를 통해 우선순위를 설정하고, 고객 중요도(Customer Importance: CI)를 1에서 5등급으로 부여한다. 중요도는 요구사항의 우선순위에 따라 달라지며, 우선순위가 높을수록 높은 값을 받는다. 고객 중요도 설정 과정은 ViRE의 방법과 같다.

• Step 3: Analyzing Correlation

연관관계 분석은 각각의 요구사항들과 연관된 유스케이스를 매핑한다. 유스케이스들과 연관된 요구사항을 분

석하고, 이들의 관계 정도에 따라 강한관계는 9점, 보통은 3점, 약한관계는 1점으로 부여한다. 이 분석을 통해 어떠한 요구사항이 어떠한 유스케이스에 연관되는지 확인 가능하다.

• Step 4: Analyzing Goal Importance Matrix

연관관계 분석 정보를 Goal 중요도 매트릭스에 적용한다. Goal 중요도 매트릭스는 그림 6의 오른쪽 그림이다. 기존 ViRE 방법과 달리 가로축에는 시스템 요소 대신 유스케이스 단위로 나눠 적용한다. 세로축에는 고객 요구사항을 나열한다. 또한 매트릭스를 분석하고 Goal 중요도를 계산하여, 각각의 유스케이스 우선순위와 요구사항의 우선순위를 도출한다. 매트릭스에는 3단계에서 분석한 연관관계를 적용한다. 9점은 ●, 3점은 ◎, 1점은 ○으로 표시한다. 연관등급($R_{i,j}$)과 고객 중요도(CI_i)를 곱하여 Goal 중요도를 계산한다. 식 (3)은 Goal 중요도 계산 방법이다.

$$GI = [(CI_1 \times R_{1,j}) + (CI_2 \times R_{2,j}) + \dots + (CI_n \times R_{n,j})] \\ = \sum_{i,j=1}^n (CI_i \times R_{i,j}) \quad (3)$$

• Step 5: Requirements Prioritization

이 단계에서는 유스케이스와 요구사항의 우선순위를 비교한다. 매트릭스 분석을 통해 측정된 Goal 중요도를 기반으로 유스케이스에 대한 우선순위를 정한다. 유스케이스마다 연관된 Goal 중요도를 합산하고, 이를 비교한다. 만약 추출된 요구사항이 30개이고, 첫 번째 유스케이스와 연관된 요구사항이 3개이면, 3개의 요구사항에 대한 Goal 중요도를 계산하면 된다. 이렇게 계산된 Goal 중요도를 유스케이스 별로 정리한다. 요구사항 우선순위는 매트릭스 분석에서 Goal 중요도를 각각 계산한다. 이것을 합산하지 않고 각각의 요구사항을 비교한다. 동일한 값이 있을 경우, 제일 큰 값을 제외한 나머지 값들은 고려하지 않는다. 이러한 과정을 통해 요구사항에 대한 우선순위를 도출한다. 유스케이스와 요구사항의 우선순위화를 통해 이해관계자들이 어떤 요구사항이 더 중요한지 판단할 수 있고, 체계적으로 시스템을 개발 할 수 있을 것이다.

• Step 6: Validation - Use Case Point

이 단계에서는 유스케이스의 우선순위를 검증하기 위해 G. Karner가 제안한 유스케이스 점수를 접목하였다 [8]. 이 논문에 제시한 방법은 유스케이스 기반으로 유스케이스의 개수, 크기, 복잡도를 양적으로 평가하는 유스케이스 점수 방식으로 검증한다. 계산 방법은 액터의 형태에 따라 액터 가중치를 결정하고, 트랜잭션 수에 따라 유스케이스 가중치를 결정한다. 유스케이스 가중치는 Goal 지향 요구사항 명세서를 기반으로 결정한다. 또한 두 가중치를 더하여 총 조정되지 않은 유스케이스 가중

표 1 유스케이스 우선순위

UC No	UC	UCP	Priority	UC No	UC	UCP	Priority
UC ₂₀	지출조회	234	1	UC ₁	로그인	79.75	12
UC ₁₇	수입조회	208	2	UC ₉	판매등록	74.25	13
UC ₁₀	판매조회	126.5	3	UC ₁₈	지출등록	72	14
UC ₁₆	재고조회	126.5	3	UC ₂₁	지출삭제	68	15
UC ₁₄	물품조회	121	5	UC ₆	입고등록	63.8	16
UC ₁₉	지출수정	120	6	UC ₂	고객등록	55	17
UC ₇	입고조회	115.5	7	UC ₁₃	물품등록	55	17
UC ₃	고객수정	110	8	UC ₁₅	물품삭제	38.5	18
UC ₁₁	판매수정	104.5	9	UC ₈	입고삭제	33	19
UC ₁₂	판매삭제	84	10	UC ₅	고객삭제	30.8	20
UC ₄	고객조회	82.5	11	UC ₂₂	인쇄	27.5	21

치를 구한다. 시스템에 영향을 미치는 정도에 따라 기술 복잡성 요인(Technical Complexity Factor: TCF)과 생산성에 영향을 미치는 환경요인(Environmental Factor: EF)을 결정한다. 이들을 곱하여 유스케이스 점수를 산출한다(식 (4)).

유스케이스 점수 기반의 우선순위 결과와 이전 단계에서 도출한 결과를 비교한다. 두 결과를 비교하여, 본 논문에서 제안한 방법을 검증한다. 표 1은 유스케이스의 우선순위를 나타낸다.

$$\text{Actor Weights} + \text{Use Case Weights} * \text{TCF} + \text{EF} = \text{Use Case Point} \quad (4)$$

그림 3은 Goal 지향 요구사항 프로세스와 UCP로 도출한 결과를 비교한 것으로서, 두 방법으로 추출한 값들이 거의 일치한다. 따라서 본 논문에서 제안한 Goal 지향의 유스케이스 기반 요구공학 방법으로 추출한 값들이 정확히 추출되었음을 알 수 있다.

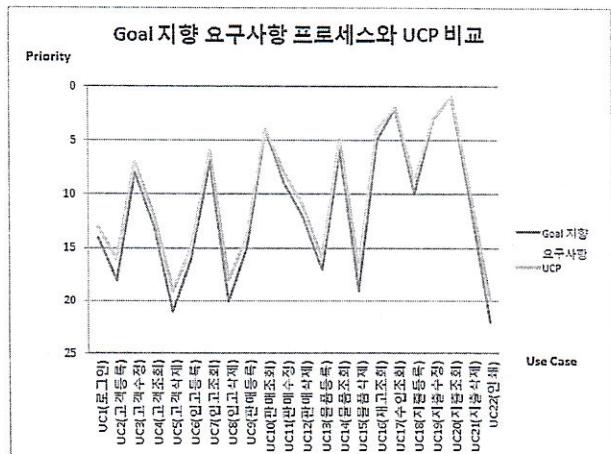


그림 3 GoRE's Process와 UCP 비교[9]

• Step 7: Hybrid Approach

이 단계에서는 유스케이스 점수와 요구사항 중요도를 ViRE의 하이브리드 방법에 적용한다. ViRE에서는 BI

(Business Importance) 값을 RC(Relative Cost) 값으로 나눠 투자수익(ROI)을 결정한다. 이를 기준으로 정렬한다. 정렬된 값을 통해 시스템 요소에 포함하거나 제외할 임계값을 결정한다. 하지만 GoRE에서는 RC 값 추출이 불가능하기 때문에 유스케이스를 점수를 적용한다[10]. 추출된 유스케이스의 임계값을 결정하고, 점선으로 표시한다. 유스케이스가 임계값보다 유스케이스가 크면 증가시키고, 아니면 감소시킨다. 이때, 이해관계자의 협의를 통해 유스케이스 값을 증가시키거나 감소시킬 수 있다. 자동차 물품 관리 시스템 개발 사례에서와 같이 유스케이스의 우선순위 도출이 가능하였다.

그림 4는 Hybrid Approach로 분석한 것이다. 이 예제의 “Reduce or Raise” 영역에서 “Create” 영역을 가리키는 화살표는 고객과의 협의를 통해 유스케이스(UC5, UC8, UC4, UC12, UC10, UC17) 중요도가 증가된다. 하지만 그 반대의 경우(“Reduce or Raise”에서 “Eliminate” 영역을 가리키는 화살표)에는 유스케이스(UC2, UC13, UC6, UC9, UC1)의 중요도가 감소된다.

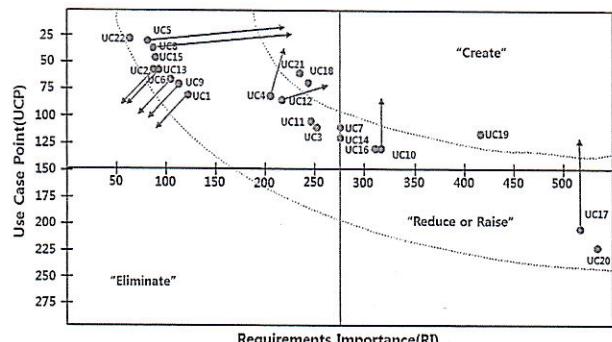


그림 4 Hybrid Approach 분석

4. 기존 ViRE와 보완한 GoRE Process 비교

그림 5는 ViRE와 GoRE의 매트릭스 분석이다. 구조적 방법론을 기반으로 ViRE는 하드웨어와 소프트웨어 요소를 포함하는 시스템 요소를 적용한다. 하지만 GoRE는 객체지향 패러다임의 개발 방법을 기반으로 유스케이스 단위로 나눠 적용한다(❶ ❷). ViRE와 GoRE는 고객 요구사항을 대상으로 한다(❸ ❹). ViRE에서는 ERRC 결정을 통해 새로운 고객 가치를 결정하지만, GoRE는 GI(Goal Importance)를 통해 유스케이스와 요구사항의 우선순위를 결정한다(❺ ❻). ViRE와 GoRE는 고객을 대상으로 하기 때문에 고객 중요도 가치를 부여한다(❻ ❽). 앞으로는 User Needs를 충분히 포함하려는 How to extract requirements from Users 부분도 향후 연구할 예정이다(❻ ❽).

검증 방법으로 ViRE는 ROI, WinWin 방법 등을 통

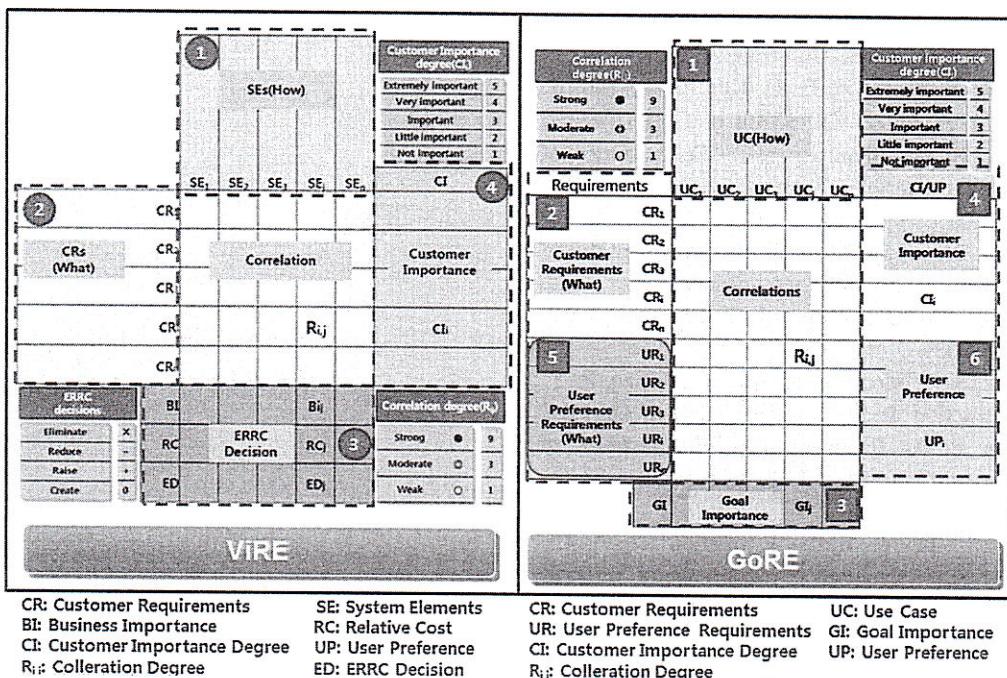


그림 5 ViRE과 GoRE의 Matrix 분석

해 하이브리드 방법으로 ERRC를 결정하고 공리적 방법으로 검증한다. 하지만 GoRE는 유스케이스 점수 기법과 하이브리드 방법을 이용하여 검증한다. ViRE는 기존 값과 추출된 값을 전략캔버스를 통해 비교함으로써, 잠재적인 고객 요구사항을 발견한다. 하지만 GoRE는 Goal 중요도를 통해 유스케이스와 요구사항의 우선순위를 도출함으로써, 개발하고자 하는 시스템의 목적에 맞는 요구사항을 추출한다. 계산 과정에서 ViRE는 상대비용과 비즈니스 중요성을 계산하여 ERRC를 결정한다. GoRE에서는 Goal 중요도를 계산하여 유스케이스와 요구사항의 우선순위를 도출한다. 표 2는 가치혁신 요구공학과 유스케이스 기반의 Goal 지향 요구공학 방법을 비교한 것이다.

5. 결 론

기존 가치혁신 요구공학은 ERRC 결정 매트릭스를 통해 새로운 가치를 부여하고, 고객 요구사항을 추출한

다. 하지만 구조적 방법 기반으로, 현재의 객체지향 및 유스케이스 개발 방법에 적용과 개선이 필요하다. 또한 소프트웨어 만을 대상으로 하는 시스템에는 적합하지 않다. 이를 보완 개선을 위해, Cockburn[8] 방법을 채택하여, 유스케이스 기반의 Goal 지향 요구공학을 제시하였다. 이 방법은 요구사항을 추출하고, matrix 분석을 통해 검증한다. Goal 중요도를 통해 유스케이스와 요구사항을 우선순위화함으로써, 시스템 개발 목적에 맞는 요구사항 추출이 가능하다. 이는 테스트케이스의 우선순위화로 최대 Coverage에 적용하려 한다. 향후에는 다양한 사례를 통한 연구가 필요하다. 실제 사례를 유스케이스 기반의 Goal지향 요구공학 방법에 적용과 결과 분석이 필요하다.

참 고 문 헌

- [1] Sangsoo Kim, "Value-Innovative Requirements Engineering," Dissertation for the Degree of Doctor,

표 2 ViRE와 GoRE 비교 결과

	ViRE	GoRE
적용요소	System Element - SW/HW 구분	Use Case 단위로 적용 Goal 지향 유스케이스
방법론	구조적 방법론.	객체지향 방법론 - Use Case
대상	Customer requirements	Customer Requirements
검증	ROI, WinWin, Hybrid, 공리적 접근법	Use Case Point, Hybrid
요구사항 추출	전략캔버스로 비교하여 잠재적인 고객 요구사항 발견	Goal 중요도를 통해 유스케이스와 요구사항의 우선순위를 도출
계산과정	상대비용(RC), 비즈니스 중요성(BI)을 통해 ERRC를 결정(ED)	Goal 중요도를 통해 유스케이스와 요구사항 우선순위 결정

- Department of Computer Science and Engineering
Graduate School, Korea University (2007).
- [2] B. K. Park, S. Y. Moon, D. H. Kim, C. Y. Seo, R. Y. Kim, "A Study on Extraction of Goal Oriented Use Case Based Requirements," *Proceedings of 2012 Korea Conference on Software Engineering*, 2012.
 - [3] J. Y. Eom, N. K, D. L, H. P. In, "A Quantitative ERRC Analysis Method via Requirements Cost Estimation," *Korea Society of IT Services*, vol.8, No.1, Mar. 2009.
 - [4] B. K. Park, S. Y. Moon, D. H. Kim, C. Y. Seo, R. Y. Kim, "Extraction & Prioritization of Goal Oriented Use Case Based Requirements," *Proceedings of KISM Spring Conference 2012*, vol.1, no.1, pp.62-65, 2012.
 - [5] B. K. Park, R. Y. Kim, "Extraction & Prioritization of User Preference Requirements through User Needs," *The 37th KIPS Spring Conference*, vol.19, no.1, pp.1239-1242, 2012.
 - [6] Karner. G, "Resource Estimate for Objectory Projects," *Objectory System SFAB*, 1993.
 - [7] N. Kim, J. Y. Park, S. Kim, H. P. In, "Customer Value-based HW/SW Partitioning Methodology," *Korea Computer Congress*, vol.34, no.1(A), 2007.
 - [8] Alistar Cockburn, "Goals and Use Cases," *J. Object-Oriented Programming*, vol.10, no.7, Sep. 1997.
 - [9] Sangsoo Kim, Hoh Peter In, "ViRE: Sailing a Blue Ocean with Value-Innovative Requirements," *Software, IEEE*, pp.80-87, vol.25, Jan-Feb. 2008.
 - [10] B. K. Park, S. Y. Moon, K. D. Kim, B. Y. Kim, R. Y. Kim, "Use Case Oriented Requirements Engineering for improving the previous ViRE's Process," *The 38th KIPS Fall Conference*, vol.19, no.2, 2012.



박 보 경

2008년 홍익대학교 컴퓨터정보통신(학사)
2012년 홍익대학교 일반대학원 소프트웨어공학(석사) 2012년~현재 홍익대학교 일반대학원 소프트웨어공학 박사과정 관심분야는 소프트웨어공학, 요구공학, RBT



김 영 철

2000년 Illinois Institute of Technology (공학박사) 2000년~2001년 LG산전 종양연구소 Embedded system 부장 2001년~현재 홍익대학교 컴퓨터정보통신 교수 관심분야는 테스트 성숙도 모델(TMM), 임베디드 소프트웨어 개발 방법론, 모델기반 테스팅, 메타모델, 비즈니스 프로세스 모델, 사용자 행위 분석 방법론 등