

ISSN 2287-4348

제2권 제1호

Vol.2 No.1

한국스마트미디어학회

2013년도 춘계학술대회 학술발표 논문집

Proceedings of KISM Spring Conference 2013  
2013 순천정원엑스포 ICT 합동학술대회

일시 : 2013년 5월 31(금) ~ 6월 1일(토)

장소 : 순천대학교 70주년 기념관

주최 : (사)한국스마트미디어학회

(사)한국인터넷정보학회

주관 : 순천대학교, 2013순천만국제정원박람회조직위원회

<http://www.kism.or.kr>



KOREAN INSTITUTE OF SMART MEDIA  
한국스마트미디어학회

13:30-15:30 제목 : 계층적 디리쉴레 확률과정 주제모형과 국부적 패치특징들을 이용한 영상분류 방법 /346

저자 : 조완현(전남대), 나인섭(전남대), 서성채(전남대)

---

13:30-15:30 제목 : 유스 케이스 기반의 리스크 결정 매트릭스 계산 자동화 도구 개발 /350

저자 : 김보연(홍익대), 김영철(홍익대), 이재협(한국기술교육대)

---

13:30-15:30 제목 : 기능점수를 이용한 Goal 지향 유스케이스 기반의 요구사항 추출에 관한 연구 /353

저자 : 박보경(홍익대), 문소영(홍익대), 김기두(한국정보통신기술협회), 김영철(홍익대)

---

13:30-15:30 제목 : 클라우드 서비스 기반 로봇 컴파일링 환경을 위한 스크립트 언어 모델 설계 /355

저자 : 장우성(한백전자), 김영철(홍익대), 이재협(한국기술교육대)

---

13:30-15:30 제목 : 스펙트럴 매팅을 위한 효과적인 불투명 변수 추정 /358

저자 : 리아즈 시드라(조선대), 이상웅(조선대)

---

13:30-15:30 제목 : 색상 정보 및 비율을 이용한 차량 번호판 영역 검출 /361

저자 : 박세동(전북대), 김형석(전북대)

---

13:30-15:30 제목 : 동영상의 평균밝기 유지를 위한 히스토그램 평활화 방법 /364

저자 : 김종인(전남대), 이재원(전남대), 홍성훈(전남대)

---

13:30-15:30 제목 : 연결 구성 요소 레이블을 사용한 동골 자동차 번호판 영역검출 /368

저자 : 락차(전남대), 김수형(전남대), 나인섭(전남대)

---

13:30-15:30 제목 : Score Sleep Stages by Using Deep Belief Network and Hidden Markov Model /372

저자 : Nguyen Hong Quy(전남대), Luu-Ngoc Do(전남대), Hyung-Jeong Yang(전남대)

---

## 유스 케이스 기반의 리스크 결정 매트릭스 계산 자동화 도구 개발

김보연, 김영철, 이재협\*

홍익대학교 소프트웨어공학연구실, 한국기술교육대학교 컴퓨터공학부\*

e-mail: {yeon, bob}@selab.hongik.ac.kr, jae@kut.ac.kr\*

### Automatic Computing Tool Development for Risk Decision Matrix Based on Use Case

Boyeon Kim, R.YoungChul Kim, Jae H. Kim\*

Dept. of CIC, Hongik University, Korea University of Technology &  
Education\*

#### 요약

새로운 리스크 관점에서 기존의 리스크 테스팅은 조직, 프로젝트, 제품에만 초점이 맞추어져 있어 소프트웨어 개발에 관련된 리스크 활동은 잘 고려되지 않는다. 이전 논문에서는 소프트웨어 개발에 리스크를 적용하고자 유스 케이스 기반 리스크 위험도 측정과 우선순위화를 제안 했다[1]. 이 논문에서는 소프트웨어 제품에서도 요구사항 단계에서의 유스 케이스 레벨 리스크를 추출하고 리스크 결정 매트릭스[1]를 통하여 리스크의 우선순위를 측정하는 프로세스 메커니즘의 자동화 계산 도구 개발을 하고자 한다.

#### 1. 서 론

요즘은 많은 소프트웨어 개발팀들이 개발 전 단계에서 소프트웨어 리스크에 대해 예측을 하고 있으며, 계속적으로 많은 연구가 이루어지고 있다[3,4,5,6,7,8,9]. 그럼에도 불구하고 리스크를 완벽하게 제거하기 어렵다. 하지만 개발 전 단계에서 개발자가 더 많은 문제가 발생할 수 있는 부분을 중점적으로 설계하고 구현하여 리스크를 줄임으로써 불확실함과 손실을 줄이려 한다. 설령 모든 리스크를 전부 피할 수 없지만 제한된 시간 내에 잠재적인 리스크들을 식별하여, 리스크에 대한 확률과 영향도에 따라 중요도를 측정을 통해 우선순위화 함으로써 줄이는 과정이다.

본 논문은 소프트웨어 제품에서도 요구사항 단계에서의 유스 케이스 레벨 기반 리스크를 추출하고 리스크 결정 매트릭스를 통하여 리스크 우선순위를 측정하는 프로세스 메커니즘의 계산 자동화 도구 개발을 제안하고자 한다. 논문의 구성은 다음과 같다. 2장은 관련 연구로 유스케이스 기반의 리스크 결정 매트릭스 대하여 설명한다. 3장은 유스 케이스 포인트 기반의 리스크 결정 매트릭스 자동화 도구를 제안한다. 마지막 4장은 결론 및 향후 연구이다.

#### 2. 유스케이스 기반의 리스크 결정 매트릭스

기존 리스크 기반의 테스팅과는 다르게 소프트웨어 개발에 관련된 리스크 활동이 잘 고려되기 위해 제안하였다. 더욱 정확한 요구사항 추출을 위해 유스케이스 별로 리스크 요구사항을 식별한다. 리스크 지향 유스 케이스 명세서

를 작성하고 유스 케이스 별로 리스크의 요구사항과의 관련정도를 분석 후, 리스크의 우선순위(Priority)와 리스크 포인트(Risk Point, RP)를 식별한다. ViRE 및 GoRE 논문의 매트릭스[3]를 참조하여, 리스크 결정 매트릭스 제안한다. 리스크 포인트는 위험 없음, 위험정도가 낮음, 위험정도가 보통, 위험정도가 높음으로 1부터 5까지이다. 그다음 리스크 지향 리스크 결정 매트릭스에 적용한다.

	UC <sub>1</sub>	UC <sub>2</sub>	UC <sub>3</sub>	UC <sub>4</sub>	UC <sub>5</sub>	RP
RR <sub>1</sub>						
RR <sub>2</sub>						
RR <sub>3</sub>		R <sub>ij</sub>				
RR <sub>4</sub>						
RR <sub>5</sub>						

UC: 단위 Use Case  
 RR: Risk Requirements  
 RP: Risk Point  
 R<sub>ij</sub>: Correlation  
 (강험(9), 보통(3), 약험(1))  
 RI : Risk Impact

(그림 1) 리스크 결정 매트릭스

그림1은 리스크 결정 매트릭스이다. 가로축은 요구사항 추출 과정 중에 식별된 유스케이스이고, 세로축은 각각의 유스케이스에서 식별된 리스크 요구사항이다. 리스크 요구사항에 따른 리스크 포인트(RP)와 리스크 요구사항과 유스케이스의 연관관계정도를 상, 중, 하로 나누어 9점, 3점, 1점으로 매트릭스에 적용한다. 이와같이 리스크 결정 매트릭스를 통하여 다음 식을 계산하여 리스크 영향도(Risk

\* 본 연구는 지식경제부 및 한국산업기술평가관리원의 산업원천기술개발사업의 일환으로 수행하였음. [10035708, 고신뢰 자율제어 SW를 위한 CPS(Cyber-Physical Systems) 핵심 기술 개발]

Impact)를 계산할 수 있다. 리스크 영향도 수식은 다음과 같다.

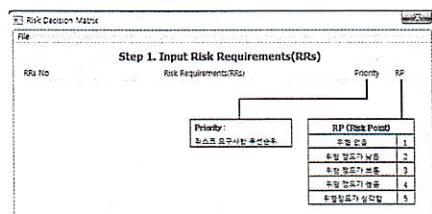
$$\begin{aligned} RI &= (CI_1 \times R_{i,j}) + (CI_2 \times R_{i,j}) + \dots + (CI_i \times R_{i,j}) \\ &= \sum_{i,j=1}^n (CI_i \times R_{i,j}) \end{aligned}$$

이처럼 리스크 결정 매트릭스를 통하여 리스크 영향도를 계산하고 각각의 유스 케이스에 대한 리스크 우선순위를 추출해 낼 수 있다.

### 3. 유스 케이스 기반의 리스크 결정 매트릭스 자동화 도구

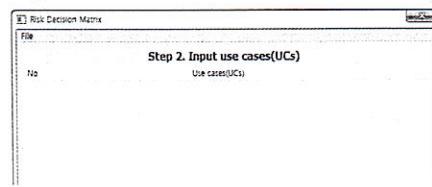
#### 3.1 리스크 결정 매트릭스 자동화 도구

##### Step1: Input Risk Requirements(RRs)



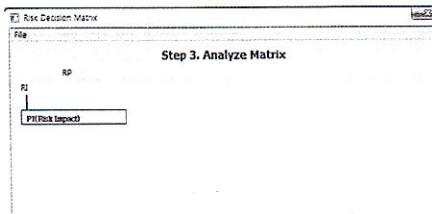
(그림 2) 리스크 요구사항 입력

##### Step2: Input Use Cases(UCs)



(그림 3) 유스 케이스 식별

##### Step3: Risk Decision Matrix



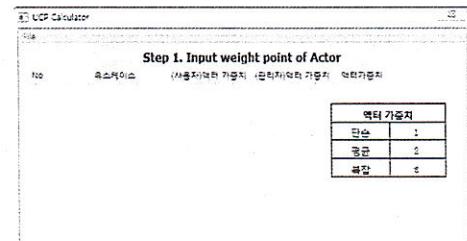
(그림 4) 리스크 결정 매트릭스

#### 3.2 유스 케이스 포인트 계산

##### Step1: 액터의 분류에 따른 규모 계산

액터의 형태에 따라 단순, 평균, 복잡으로 나눈다. 조정 전 액터 가중치(Unadjusted Actor Weight, UAW)의 계산식은 다음과 같다.

$$\Sigma(\# \text{ of Actors} \times \text{WF})$$

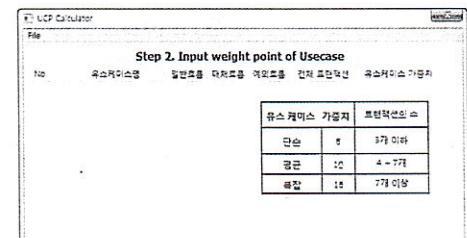


(그림5) 액터 가중치 계산

##### Step2: 유스 케이스 분류에 따른 규모 계산

유스 케이스는 트랜잭션 수를 통해서 간단, 평균, 복잡으로 나눈다. 조정 전 유스케이스 가중치(Unadjusted Use Case Weights, UUCW)의 계산식은 다음과 같다.

$$\Sigma(\# \text{ of Use Cases} \times \text{WF})$$



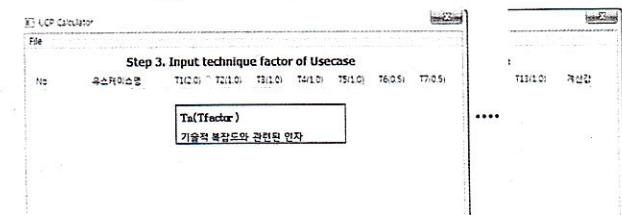
(그림6) 유스케이스 가중치 계산

##### Step3: 조정 인자 계산

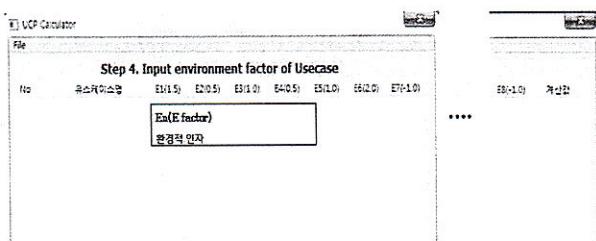
그림 5에서 입력한 액터 가중치를 기술적 복잡도 인자(TCF)는 문자시스템, 응답시간 또는 처리성능 요구사항, 최종 사용자 능률, 복잡한 내부 연산, 재사용성, 설치 용이성, 사용성, 이식성, 수정 용이성, 병행성, 특수 보안 요구사항 포함, 제 3자에 의한 직접적 접근 제공, 사용자 교육을 위한 특수 시설이 요구됨은 총 13개 항목으로 자동 계산된다. 그림 6은 프로젝트의 프로세스 및 자원의 특성과 관련된 8개의 항목으로 되어 있으며, 환경 인자(EF)를 자동 계산해준다. 그림 7에선 기술적 복잡도 인자 13개와 그림 8에선 환경 인자 8개에 0부터 5까지의 값을 할당한다. 할당된 값에 -1부터 2까지의 가중치를 곱한다. 기술적 복잡도 인자와 환경 인자를 계산한다. 계산식은 다음과 같다.

$$TCF = 0.6 + (0.01 \times TFactor)$$

$$EF = 1.4 + (-0.03 \times EFactor)$$



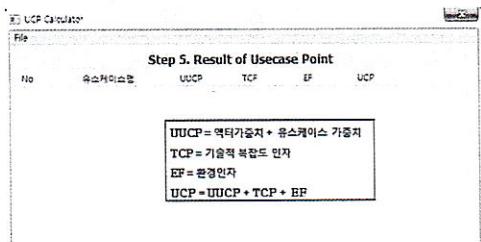
(그림 7) 기술적 복잡도 인자 계산(TCF)



(그림 8) 환경인자 계산(EF)

**Step4: 조정 인자를 반영하여 유스케이스 포인트 계산**  
전 단계의 인자들을 통하여 값을 계산한다. 계산식은 다음과 같다.

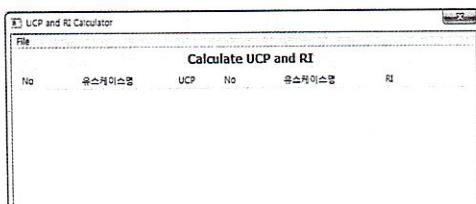
$$UCP = UUCP \times TCF \times EF$$



(그림 9) 유스케이스 포인트 계산

### 3.3 UCP and RI 상관관계 분석

기존 리스크 결정 매트릭스 자동화 도구와 유스 케이스 포인트(UCP) 계산기를 통하여 상관관계를 분석해 주는 도구이다



(그림 8) UCP와 RI 상관관계 분석

## 4. 결론

기존 논문의 유스 케이스 기반 요구사항 분석을 통한 리스크 추출 및 우선순위화 메카니즘을 계산 자동화 도구 개발을 제안하고 있다. 기존에 프로젝트, 제품에만 맞추어져 있는 것이 아닌 소프트웨어 개발에 맞추어 리스크 추출 및 우선순위화하고자 한다. 이는 '불확실성'과 '손실'을 내포하고 있는 리스크를 개발 전에 추출하여 더 많은 문제가 발생할 수 있는 부분을 중점적으로 설계하고 구현하여 리스크를 줄이고자 함이다.

본 논문에서 제시한 리스크의 우선순위를 측정하는 프로세스 메카니즘 자동화 도구는 3가지 단계를 가지고 있다. 첫 번째 단계는 기존의 리스크 결정 매트릭스 계산 자동화 도구를 통해 요구사항을 추출과 리스크를 식별하고 리

스크와 유스 케이스의 연관관계를 통해 리스크 영향도를 계산한다. 두 번째 단계는 액터의 가중치와 유스 케이스의 가중치를 주어 기술적 복잡도 인자와 환경인자를 통한 조정인자를 반영한 유스 케이스 포인트를 계산하였다. 마지막으로 첫 번째 단계에서 계산된 리스크 영향도와 두 번째 단계에서 계산된 유스 케이스 포인트를 비교분석한다.

## 참고문헌

- [1] 김보연, 김재승, 박보경, 손현승, 김영철, 김우열 “유스 케이스 기반 요구사항 분석을 통한 리스크 추출 및 우선순위화 연구”, 한국정보처리학회, 2012.11.
- [2] 이선경, 강동원, 배두환, “유스케이스 트랜잭션 기반의 소프트웨어 공수 예측 기법”, 한국정보과학회 논문지, Vol16, No.5, pp566-570, 2010.05.
- [3] 박보경, “Goal 지향 요구사항 기반 테스팅을 위한 요구사항 추출 및 우선순위화에 관한 연구”, 석사학위논문, 홍익대학교, 2012.
- [4] Boehm, B.W., "Software Risk Management: Principles and Practices", Software, IEEE, Vol8, No.1 , pp.32-41, 1991.01.
- [5] Software Engineering A Practitioner's Approach, Seventh Edition
- [6] Exploring risk-based testing and its implications Felix Redmill, Redmill Consultancy, 22 Onslow Gardens, London N10 3JU, U.K.
- [7] C.R, Synmons, Software Sizing and Estimationg MK II FPA. Wiley-Inter-science, 1991.
- [8] [http://en.wikipedia.org/wiki/Use\\_Case\\_Points](http://en.wikipedia.org/wiki/Use_Case_Points)
- [9] Stale Amland, "Risk-based testing: Risk analysis fundamentals and metrics for software testing including a financial application case study", Journal of Systems and software, Vol53, pp.287-295, 2000.