

A Study for the Analysis of Domain for the Modeling of User Interface

Lee Eun-Ser[†]

ABSTRACT

User interface is important of programs collaboration. User interface error have a effect in the whole system. As a result, the system reliability will deteriorate. Therefore, we are need to methodology that user interface type is analyze for a reliable analysis in the domain analysis phase. In this paper, we are propose the methodology that extraction and standard of user interface for reliability improvement of domain analysis.

Keywords : Domain Analysis, Domain Modeling, User Interface Modeling

사용자 인터페이스 모델링을 위한 도메인 분석에 관한 연구

이 은 서[†]

요 약

사용자 인터페이스는 프로그램 상호 연동의 중요한 요소가 된다. 사용자 인터페이스에서 발생하는 오류는 전체 시스템에 영향을 주게 되고, 그 결과 고객의 만족도가 낮아진다. 따라서 도메인 단계에서 신뢰성 있는 분석을 위하여 사용자 인터페이스의 형태를 분석할 수 있는 방법이 필요하게 된다. 본 논문에서는 이와 같은 문제를 해결하기 위하여 도메인 분석의 신뢰성 향상을 위한 사용자 인터페이스의 추출 방법과 기준을 제시하고자 한다.

키워드 : 도메인 분석, 도메인 모델링, 사용자 인터페이스 모델링

1. 서 론

현재는 하이테크 시대에 살고 있으며 소형 가전, 산업기계, 회사 시스템, 군사 시스템, 개인용 컴퓨터 소프트웨어 그리고 웹 응용 거의 모든 제품이 사람과의 상호작용이 필요하다. 제품이 성공적이기 위해서는 하이테크 제품에 의해서 제공되는 기능과 특징을 사람이 얼마나 쉽고 효율적으로 다룰 수 있는지를 정량적인 측정치로 보여주는 유용성이 좋아야 한다[1-3]. 이와 같은 상호 작용은 높은 신뢰성을 요구하게 되며, 그 근원은 도메인을 사용하는 사용자가 요구하는 인터페이스를 기반으로 하고 있다.

기술자들이 사람의 상호작용을 연구하면서 두 가지 주요한 문제가 발생했다. 첫째, 황금 규칙이 파악된다. 이들은 기술제품에 모든 사람의 상호작용을 적용한다. 둘째, 소프트웨어 설계자가 황금 규칙을 적절하게 구현한 시스템을 구현할

수 있도록 상호작용 체계가 정의된다. 집합적으로 그래픽 사용자 인터페이스라 불리는 상호작용 체계는 사람 인터페이스와 관련된 가장 엄청난 문제를 해결했다[1, 4, 5].

본 논문에서는 도메인 분석을 정확히 수행하기 위하여 사용자 인터페이스 모델링을 위한 항목 추출하고 그 결과 복잡도를 측정하도록 한다.

2. 기반 연구

2.1 인터페이스 분석과 설계 모델

사용자 인터페이스가 분석되어 설계될 때 네 가지 다른 모델이 등장한다. 인간 공학자(또는 소프트웨어 공학자)는 사용자 모델을 확립하고, 소프트웨어 공학자는 설계 모델을 생성하고, 사용자는 종종 사용자의 정신 모델 또는 시스템 인지라 불리는 정신적 이미지를 개발하고 시스템 개발자는 구현 모델을 생성한다. 불행히도 이들 모델 각각은 상당히 다를 수 있다. 인터페이스 설계자는 이런 차이점을 조화시켜 인터페이스의 일관된 표현을 구해야 한다[6-8].

여러 학자들은 다음과 같은 의견을 제시하였다.

[†] 종신회원 : 안동대학교 컴퓨터공학과 부교수
Manuscript Received : November 19, 2015
First Revision : December 10, 2015
Accepted : December 10, 2015
* Corresponding Author : Lee Eun-Ser(eslee@anu.ac.kr)

진실은 자신을 포함하여 설계자와 개발자는 가끔 사용자를 생각한다는 것이다. 하지만 특정 사용자의 완전한 정신적 이미지 없는 경우 설계자와 개발자의 이미지가 대신한다. 이런 대신은 사용자 중심이 아니라 자기중심적일 뿐이다[11].

효율적인 사용자 인터페이스를 구축하기 위해 “모든 설계는 그 들의 나이, 성별, 신체적 조건, 교육정도, 문화적 인종적 배경, 동기, 목적과 개인 성격 등을 포함하여 의도된 사용자를 이해하는 것에서 시작해야 한다”[12].

- ◆ **초보자.** 시스템에 관한 구문 지식이 없으며 대체적으로 응용이나 컴퓨터에 관한 의미적 지식이 거의 없다.
- ◆ **지식이 있는 간헐적인 사용자.** 응용에 관한 의미적 지식을 어느 정도 가지고 있으나 인터페이스를 사용하는데 필요한 구문 정보를 조금밖에 모른다.
- ◆ **지식이 있는 빈번한 사용자.** 고급 사용자라 할 수 있을 정도의 의미적 구문적 지식을 가지고 있는, 즉 단축키와 축약된 반복모드를 알고자 하는 개인이 이에 해당한다.

사용자 정신 모델(시스템 인지)은 시스템의 이미지이며 사용자가 자신의 머리에 들어있는 지식이다. 예를 들어, 특정 워드 프로세서를 사용하는 사용자에게 어떻게 작동하는지에 대해 물어보면 사용자가 시스템을 어떻게 인지하는지로 대답할 것이다. 이 대답의 정확성은 사용자의 프로파일(예를 들어, 초보자는 기껏해야 스케치 정도의 대답 밖에 못할 것이다)과 애플리케이션 도메인에서의 소프트웨어와의 전반적인 친숙 정도에 달려있다. 워드 프로세서를 완전히 이해하나 특정 워드 프로세서는 단지 한번만 작업한 사용자는 시스템을 알기위해 몇 주간 시간을 보낸 초보자보다는 기능에 대해 보다 완전히 기술할 수 있을 것이다.

구현 모델은 컴퓨터 기반 시스템(인터페이스의 모습과 느낌)의 외부 표현과 인터페이스 구문과 의미를 기술하는 모든 지원 정보(책, 매뉴얼, 비디오테이프 도움말)를 결합한다. 구현 모델과 사용자의 정신 모델이 일치하는 경우 사용자는 소프트웨어에 대해 편안함을 느끼며 효율적으로 사용할 수 있다. 모델의 이런 결합을 위해 설계 모델은 사용자 모델에 포함된 정보를 수용하도록 개발하고 구현 모델은 인터페이스에 관한 구문적 의미적 정보를 정확하게 반영하도록 해야 한다[13].

2.2 도메인 분석

도메인에 대한 분석 작업은 해당 영역에 해당되는 특화된 패턴들을 생성하여 분석 모델을 만들 수 있도록 해준다. 이와 같은 내용은 개발 비용의 감소의 효과를 얻을 수 있게 된다.

소프트웨어 도메인 분석은 특히 애플리케이션 도메인 내의 다수의 프로젝트들의 재사용을 위한 특정 애플리케이션 도메인에서 나온 공통된 요구사항의 구별, 분석, 그리고 명세서이다[1, 9].

특정 애플리케이션 도메인은 항공 전자 공학부터 은행 업무, 멀티미디어 비디오 게임부터 의학 기기에 내장된 소프트웨어에 이르기까지 다양하다. 도메인 분석의 목표는 정확하다: 분석 클래스들이 광범위하게 적용될 수 있어 재사용될 수 있는 분석 패턴들을 발견하거나 창조하는 것이다 [2, 8, 9].

도메인분석은 소프트웨어 프로세스를 위한 상부 액티비티로 보여질 수도 있다. 이를 통해, 도메인분석은 어떤 한 개의 소프트웨어 프로젝트에 연결되지 않은 현재 진행형인 소프트웨어 공학활동이라는 것이다. 어떤 의미에서, 도메인 분석가의 역할은 중공업 환경에서 툴스미스(toolsmith)마스터의 역할과 유사하다. 툴스미스의 업무는 유사한 작업을 하는 많은 사람들이 사용하는 툴들을 설계하고 생성하는 것이다. 도메인 분석가의 역할은 분석 패턴들, 분석 클래스들, 그리고 비슷한 애플리케이션에서 작업하는 대다수의 사람들에 의해 사용될 수 있는 관련 정보를 발견하고 정의하는 것이다.

도메인 분석 프로세스의 주요 입력과 출력물을 나타낸다. 도메인지식의 원천은 도메인에 걸쳐 재사용될 수 있는 객체들을 구별하기 위한 시도로 조사된다. 고객들이 더욱 기술적으로 수준이 높아짐에 따라 무엇에 만큼이나 어떻게 명세서를 작성하느냐에 대한 경향이 있다. 그러나 주요 초점은 무엇에 있어야한다[3, 10].

도메인 분석의 보완적인 관점은 “도메인을 모델링 하는 것과 연관 되어 소프트웨어 공학자들과 다른 주주들이 그것에 대해 더 잘 배울 수 있다[4].

도메인의 이해는 도메인 전문가에 국한된 내용이 아니다. 도메인에 대한 이해를 하지 못한다면 소프트웨어 공학자에 의한 개발 과정들은 사용자를 위한 소프트웨어 개발이 아닌 개발자만의 소프트웨어를 만들게 되는 것이다.

3. 본 론

3.1 사용자 인터페이스 모델링을 위한 항목 추출

사용자 인터페이스는 여러 가지 의미에서 시스템 또는 프로그램에서 중요한 역할을 한다. 사용자 인터페이스를 통하여 사람과 컴퓨터가 의사소통을 하여 효율적인 통신 매개체 역할을 하게 된다. 따라서 사용자 인터페이스의 정확한 추출은 전체 시스템의 문제점을 감소시키고 사용자의 만족감을 증대시키는 수단이 된다. 이와 같은 사항은 설계 단계에서 추출하여 시스템을 구축하고 있다.

도메인 분석은 전문가와 도메인의 특성을 고려하여 분석이 수행된다. 이 과정에서 사용자 인터페이스 추출을 고려하여 도메인 분석을 수행하여 설계단계의 사용자 인터페이스 추출의 완성도를 높이는 것이 전체 시스템의 신뢰성을 높일 수 있게 된다. 도메인 분석 시에 사용자 인터페이스 모델링을 위하여 필요한 항목을 먼저 추출해야 한다. 이 항목들은 사용자 인터페이스 모델링의 완성도와 복잡도를 측

정하는 기준이 된다.

항목의 추출 기준은 사용자 인터페이스의 연동 관점에서 기준을 제시하였다. 이와 같은 항목을 상호작용성으로 정의하였다. 사용자 인터페이스 모델링을 위한 항목은 다음과 같이 추출하였다.

◆ 상호작용성

상호작용성은 사용자 인터페이스의 역할이 다른 기능(모듈 또는 함수, 컴포넌트)과의 연동을 위하여 역할을 수행하는 경우이다. 또한 상호작용성은 유연성이 고려되어야 한다. 유연성은 사용자가 인터페이스를 통하여 시스템을 제어하고 활용할 때 만족도와 편의성을 측정하는 중요한 요소가 된다. 결국 이와 같은 사항은 다른 기능과의 연동을 고려하지 않을 수 없게 된다. 상호작용성의 정도가 많을수록 복잡도 및 위험요소가 증가할 수 있게 된다. 따라서 상호작용성을 측정하기 위하여 정량적인 기준을 제시하여 상호작용성의 정도를 파악하고자 한다.

상호작용성을 파악하기 위한 항목과 기준은 요구사항 분석 단계의 수준이 아닌 도메인 분석 단계의 전문가 지식을 활용하게 된다. 도메인 분석의 결과물은 요구사항의 입력물로 활용되기 때문에 다음 단계에서 활용이 가능하도록 하는 것이 효율적이다. 따라서 상호작용성의 항목과 기준은 기능적인 사항과 비기능적인 관점에서 추출하였다. 상호작용성의 항목과 기준은 다음과 같다.

도메인 전문가의 지식을 추출하는 과정에서 기능들이 서로 상호 작용을 일으키는 경우이다. 도메인의 전문가의 지식 시나리오 분석 과정 중에 사용자와 직접적인 연동이 되는 의사소통 기능인 것을 추출한다. 여기서 의사소통이 되는지의 판단 여부는 입력과 출력, 연산에 필요한 알고리즘의 필요성 유무로 확인할 수 있다. 이 과정에서 입력물과 출력물, 알고리즘을 파악하고 분석하는 것이 아니라 각 요소가 필요한가에 초점을 맞추게 된다. 각 요소를 통하여 입력물과 출력물, 연산 알고리즘의 필요성 유무로 기능의 상호 작용을 판단하게 된다. 각 항목의 도메인의 특성에 따라서 가중치를 정한다. 가중치의 총합은 1로 산정하였다. 상호작용성의 수준을 판단하기 위한 기준은 다음과 같다.

① 입력물의 형태

입력물의 수준을 판단하기 위한 기준으로 입력물이 단순한 수치인 경우가 얼마나 되는지를 판단 기준으로 설정하였다. 도메인 전문가의 지식 시나리오 분석에서 정성적인 부분이 입력으로 처리되는 경우는 상호 작용 과정에서 명확한 의사 전달과 복잡도 측면에서 불리하게 된다. 따라서 입력물의 형태에 의하여 측정하였다. 입력물의 형태가 단순 수치의 입력물로 30% 이내를 차지하면 상, 31~79%를 차지하면 중, 80~100%를 차지하면 하로 측정한다. 상은 0.8, 중은 0.5, 하는 0.2로 수치화 한다. 이와 같은 수치는 상호작용성

의 입력물 형태로 사용자 인터페이스 모델링의 복잡도를 측정하는데 활용이 된다.

Table 1. Standard of Complexity by Input Type

	High	Middle	Low
Simple numerical value	0.8	0.5	0.2

② 출력물의 형태

출력물의 수준을 판단하기 위한 기준으로 출력물이 단순한 수치인 경우가 얼마나 되는지를 판단 기준으로 설정하였다. 도메인 전문가의 지식 시나리오 분석에서 정성적인 부분이 출력으로 생성되는 경우에는 결과물이 다른 기능의 입력물로 활용될 수 있다. 출력물의 형태가 단순 수치의 결과물로 30% 이내를 차지하면 상, 31~79%를 차지하면 중, 80~100%를 차지하면 하로 측정한다. 상은 0.8, 중은 0.5, 하는 0.2로 수치화 한다. 이와 같은 수치는 상호작용성의 출력물 형태로 사용자 인터페이스 모델링의 복잡도를 측정하는데 활용이 된다.

Table 2. Standard of Complexity by Output Type

	High	Middle	Low
Simple numerical value	0.8	0.5	0.2

③ 연산 알고리즘

연산 알고리즘의 수준을 판단하기 위한 기준으로 연산이 단순한 사칙연산으로 수행되는 경우가 얼마나 되는지를 판단 기준으로 설정하였다. 연산 알고리즘의 단순한 사칙연산 비중이 30% 이내를 차지하면 상, 31~79%를 차지하면 중, 80~100%를 차지하면 하로 측정한다. 상은 0.8, 중은 0.5, 하는 0.2로 수치화 한다. 이와 같은 수치는 상호작용성의 연동 관점에서 복잡도 정도를 측정하기 위하여 사용된다.

Table 3. Standard of Complexity by Computational Algorithm

	High	Middle	Low
The four fundamental arithmetic operations	0.8	0.5	0.2

④ 저장소 활용

도메인 전문가의 지식을 분석하는 과정에서 산출된 결과나 과정 등이 최종 산출물이 아닌 경우 중간 결과물을 저장할 저장소가 별도로 필요하게 된다. 도메인 전문가가 시스템으로 구축된 형태가 아닌 본인의 머리를 활용하여 문제를 해결할 때에는 머릿속에서 모든 결과물을 기억하여 최종 결과물을 얻게 된다. 그러나 시스템으로 구축된 경우에는 결

과물의 활용을 위한 연동과 그 중간 결과물의 위치를 시스템에서 제공해야 한다. 따라서 도메인 분석 단계에서 이와 같은 저장소 활용여부를 판별하여 사용자 인터페이스 모델링에 내포되어 있는 복잡도를 측정하고자 한다. 저장소 활용은 시스템 관점에서 자료구조와 그 장소를 접근할 수 있는 링크와 주소가 필요하게 된다. 저장소 활용의 비중이 80% 이상을 차지하면 상, 31~79%를 차지하면 중, 30% 이내를 차지하면 하로 측정한다. 상은 0.8, 중은 0.5, 하는 0.2로 수치화 한다.

Table 4. Standard of Complexity by Use of Repository

	High	Middle	Low
Use of repository	0.5	0.3	0.1

①, ②, ③, ④ 항목에 가중치를 곱하여 전체 산정 값을 결정한다. 이때 각 항목의 가중치의 합은 1을 넘지 않는다. 따라서 복잡도 산정 값도 1을 넘지 않는다. 도메인 산정 값의 수식은 다음과 같다.

$$\begin{aligned} & \text{사용자 인터페이스 복잡도} \\ &= (\text{①항} \times \text{가중치}) + (\text{②항} \times \text{가중치}) \\ &+ (\text{③항} \times \text{가중치}) + (\text{④항} \times \text{가중치}) \end{aligned}$$

사용자 인터페이스 도메인 복잡도의 활용은 각 항목의 산정 값을 기준으로 도메인 분석 과정에서 사용자 인터페이스의 모델링을 최적화하기 위하여 복잡도가 발생하는 부분을 식별하게 된다. 따라서 사용자가 시스템과 의사소통을 하는 방법을 유연하게 구축하고 사용자 관점에서 인터페이스를 설계하여 만족도를 높이고자 한다.

4. 사례 연구

4.1 개발 배경

- ◆ 웹캠이나 CCTV 또는 스마트폰의 영상을 데이터베이스로 저장하고 그 데이터베이스를 웹과 연동시켜 스마트폰으로 해당 웹을 통해 언제든지 그 영상을 볼 수 있도록 하는 어플리케이션을 개발한다.
- ◆ 폰으로 CCTV를 보는 기존의 시스템과 개발하려는 앱들은 접근 방법이 다르다. 또한 기존의 시스템은 고가의 가격으로 이용이 가능하다. ip를 이용하여 CCTV를 볼 수 있는 앱은 ip만 알면 해킹당할 위험이 있다고 알려져 있다.
- ◆ 이러한 앱과 차별성을 두어 무료로, 안전하게 CCTV를 볼 수 있는 앱을 제작하여 많은 사람들이 이용할 것이라는 기대효과를 가져 올 수 있다.

4.2 프로젝트 추진 내용

적용 사례에서는 폭포수 모델을 기반으로 생명주기를 설정하였다. 단계별로 진행되는 폭포수 모델을 사용하기로 하였으며, 각 단계별 산출물이 다음 단계를 입력물로 사용되게 된다.

◆ 요구사항 분석

- 개발하려는 소프트웨어에 대한 요구사항을 명세하고, 이를 이해하고 분석하는 단계
- 요구사항 정의서가 산출물로 나오며, 전반적인 구현 기능에 대한 문서화가 되어 있다.

◆ 설계

- 소프트웨어의 구조에 대해서 프로그램적인 설계가 이루어지는 단계
- 인터페이스 구조, 알고리즘 등에 대한 산출물들이 나온다.

◆ 구현

- 설계된 것을 토대로 하여 소스코드를 작성하여 실제로 구현하는 단계이다.

◆ 테스트

- 형상관리항목에 의해 적절한 기준을 세워 테스트하는 단계
- 핵심 구조물부터 테스트를 하며 그 후 오류발견 및 작동여부에 대한 문서를 작성.

◆ 유지보수

- 개발된 소프트웨어에 대해 수정해야 할 부분에 대해서 수정/보완 작업.

3장에서 제시한 이론을 기반으로 사용자 인터페이스 모델링의 복잡도를 측정하고자 한다.

입력물의 형태는 중으로 추출되었다. 중의 결과는 0.5를 갖는다. 그리고 가중치는 0.2를 부여하였다. 출력물의 형태는 하로 추출되었다. 중의 결과는 0.5를 갖는다. 그리고 가중치는 0.2를 부여하였다. 연산 알고리즘은 상으로 추출되었다. 상의 결과는 0.8을 갖는다. 그리고 가중치는 0.3을 부여하였다. 저장소 활용은 상으로 추출되었다. 상의 결과는 0.8을 갖는다. 그리고 가중치는 0.3을 부여하였다.

추출된 결과를 기반으로 하여 사용자 인터페이스 복잡도를 측정하면 다음과 같다.

$$\begin{aligned} & \text{사용자 인터페이스 복잡도} \\ &= (0.5 \times 0.2) + (0.5 \times 0.2) + (0.8 \times 0.3) + (0.8 \times 0.3) \\ &= 0.1 + 0.1 + 0.24 + 0.24 = 0.68 \end{aligned}$$

산출된 결과에 의하면 사용자 인터페이스 복잡도의 많은 부분이 연산자 알고리즘과 저장소 활용에서 많은 비중을 차



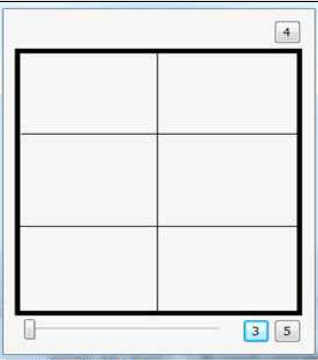
화면이름	웹상에 전송된 화면을 다른 모바일로 확인		
화면			
관련기능	<p>프로그램 실행 화면 촬영 선택 화면 녹화화면 선택 화면</p> <ol style="list-style-type: none"> 1. 확대기능 : 화면을 6개의 부분으로 나뉘어 각 부분을 누를시 그 부분만 확대되어 나타남 2. 축소기능 : 확대된 화면을 다시 손가락으로 누르면 축소되어 6개의 나뉜 부분으로 나타남 3. 녹화기능 : 녹화화면 버튼을 누르면 누른 시점부터 종료시점까지 녹화됨 4. 종료기능 : 종료버튼을 누르면 현재 보이는 화면이 종료됨.0 5. 음량기능 : 음량 버튼을 입력 시 알람기능 on/off 6. 확대/축소 기능 : 슬라이드 바를 이용해 확대/축소 기능 (확대된 상태로 드래그 -> 화면 이동) 7. 촬영기능 : 촬영 버튼을 누르면 모바일의 카메라를 이용해서 촬영가능 8. 촬영종료 : 촬영 버튼을 누르면 모바일의 카메라를 이용해서 촬영가능 		
화면설명	<p>화면은 스마트폰 화면을 나타내며, 각 버튼과 관련기능이 연관. 프로그램 실행 -> 웹에 저장된 화면중 보고자 하는 화면을 touch -> touch한 화면이 모바일에 켜짐 (모바일에서 기능[관련기능])</p>		

Fig. 1. Definition of Screen

지하고 있다. 따라서 연산자 알고리즘과 저장소 활용 분야에서는 영상 파일을 빈번하게 저장하고 탐색이 가능해야 하므로 도메인의 특성상 많은 비중을 차지하였다. 연산자 알고리즘은 단순 수치 연산으로 바꾸기 위하여 사용자 정의 연산을 내장 함수로 대체하였다. 저장소 활용은 별도의 저장소의 자료를 배열에 저장하여 다른 데이터베이스 접근을 최소화 하였다. 이와 같은 개선사항을 기반으로 사용자 인터페이스 복잡도를 재수행하였다.

추출된 결과를 기반으로 하여 사용자 인터페이스 복잡도를 측정하면 다음과 같다.

연산 알고리즘은 중으로 추출되었다. 중의 결과는 0.5를 갖는다. 그리고 가중치는 0.3을 부여하였다. 저장소 활용은 중으로 추출되었다. 중의 결과는 0.5를 갖는다. 그리고 가중치는 0.3을 부여하였다.

사용자 인터페이스 복잡도

$$\begin{aligned}
 &= (0.5 \times 0.2) + (0.5 \times 0.2) + (0.5 \times 0.3) + (0.5 \times 0.3) \\
 &= 0.1 + 0.1 + 0.15 + 0.15 = 0.5
 \end{aligned}$$

산출된 결과는 개선 전에 비하여 0.18이 개선되었다. 이를 기반으로 사용자 인터페이스 모델링을 수행하였다. 화면 정의서는 Fig. 1과 같다.

5. 결론 및 향후 연구 방향

본 논문에서는 도메인 분석 시에 사용자의 만족도와 유연성을 제공하기 위하여 사용자 인터페이스 복잡도를 측정하

였다. 복잡도는 사용자 인터페이스 모델링에서 시스템과 사용자가 의사소통을 하는 화면 정의서를 통하여 결과를 산출하였다.

향후 연구 내용으로는 사용자 인터페이스 복잡도를 산출하기 위한 기준의 측정값에 대하여 많은 사례에 적용하여 기준 값에 대하여 검증이 필요하다. 또한 검증을 통하여 기준의 재 정의가 필요하다. 그리고 본 논문의 이론을 웹 앱을 통하여 연동할 수 있는 도구의 개발이 필요하다. 마지막으로 상, 중, 하에 부여한 가중치에 대하여 여러 프로젝트에 적용하여 값을 보정할 필요가 있다.

References

- [1] Kim sung kyu, "Software engineering," Han san, p.301, 2011.
- [2] D. G. Firesmith, "Object-oriented requirements analysis and logical design: A Software Engineering Approach," John Wiley & Sons, Inc. New York, NY, USA, 1993.
- [3] R. S. Arnold, "Software Restructuring," *Proc. IEEE*, Vol.77, No.4, pp.607-617, 1989.
- [4] T. Lethbridge and R. Laganie, "Object-Oriented Software Engineering Practical Software Development Using UML and Java," McGraw-Hill, 2001.
- [5] Sommerville, Software engineering, Pearson, 2011, p.83.
- [6] S. A. White and M. Alford, et al., "System engineering of computer-based systems," *IEEE Computer*, Vol.26, No.11, pp.54-65, 1993.
- [7] Yoon chung, "Successful Software development methodology," Life power press, 1999.

- [8] Shari lawrence, "Software engineering forth edition," Pearson, p.208, 2010.
- [9] Choi eun man, "Software engineering," Jungik publishing co, 2011, p.156.
- [10] Han huck su, "Introduction of software engineering," Hongrung, 2011, p.117.
- [11] J. Patton, "Understanding user centricity," *IEEE Software*, Vol.24, No.6, pp.9-11, 2007.
- [12] B. Shneiderman and C. Plaisant, "Designing the user interface," 4th ed., Addison-Wesley, 2004.
- [13] A. Monk, (ed.), *Fundamentals of human-computer interaction*, Academic Press, 1984.



이 은 서

e-mail : eslee@anu.ac.kr

2001년~현 재 ISO/IEC 15504 국제 선임
심사원

2004년 중앙대학교 컴퓨터공학과(박사)

2004년~현 재 임베디드 산업협회 전문
위원

2004년~현 재 한국정보통신기술협회 위원

2012년~현 재 안동대학교 컴퓨터공학과 부교수

관심분야: CBD, Formal method, Quality model, SPI(Defect
Analysis)