

모바일 앱의 사용자 의견으로부터 소프트웨어 및 시스템 요구사항을 추출하기 위한 프로세스와 방법

오동석* · 김선빈* · 류성열**

Processes and Methods for Eliciting Software and System Requirements from Users' Opinions in Mobile App

Dong-Seok Oh* · Sun-Bin Kim* · Sung-Yul Rhew**

■ Abstract ■

For mobile service organizations, it is one of the most important tasks to reflect users' opinions rapidly and accurately. In this study, the process is defined to elicit requirements of software/system improvement for mobile application by extracting and refining from users' opinion in mobile app, and detailed activities procession method in this processing are also proposed. The process consists of 3 activities to get requirements of software/system improvement for mobile app. First activity is to transform mobile app to software structure and define term dictionary. Second activity is to elicit simple sentences based on software from users' opinion and refine them. The last activity is to integrate and adjust refined requirements. To verify the usability and validity of the proposed process and the methods, the outputs of manual processing and semi-automated processing were compared. As a result, efficiency and improvement possibility of the process were confirmed through extraction ratio of requirements, comparison of execution time, and analysis of agreement ratio.

Keyword : Users' Opinion, Mobile App, Requirement Elicitation, Requirement Refinement,
Requirement Integration

1. 서 론

소프트웨어의 진화적 관점에서 변경에 대한 고려는 필수적이며(Pressman, 2008), 모바일 앱(App)을 개발하여 서비스하는 조직에게도 소프트웨어와 시스템의 개선을 위한 변경을 필수적으로 고려해야 한다. 사용자 의견을 소프트웨어와 서비스 개선에 반영하는 것은, 사용자 중심의 요구공학에서 중요한 과제로 인식되고 있으며, 이를 통해 소프트웨어의 성공과 지속적인 운영이 가능하다(Kujala, 2012). 모바일 앱을 통한 서비스의 성공을 위해서도 사용자 의견을 신속하고 정확하게 소프트웨어 및 시스템에 반영해야 한다. 특히, 사용자 의견에 민감한 모바일 앱의 경우, 이러한 사용자 의견에서 소프트웨어 및 시스템 개선과 관련된 요구사항을 추출하고 반영하는 것은 모바일 앱의 개선 및 성공적 운영에 기여할 수 있다.

그러나 모바일 앱을 사용한 결과로서 기술한 사용자 의견은 사용자만의 기호와 어휘 등에 따라 비구조적이고 비정형적이며, 불분명한 의도와 개인적인 상황 및 경험, 소프트웨어에 대한 지식의 부족 등으로 인해 해석하기에 애매모호하다. 이렇게 비정형적인 언어로 기술된 사용자 의견을 소프트웨어와 시스템 개선 요구사항으로 직접적으로 반영하기에는 어려움이 있으며(Byun, 2012), 이에 대해서는 통상적으로 인간의 지성을 이용해 분석하고 검증하는 것이 가장 정확하고 효과적이다(Fraser et al., 1991).

하지만, 소량의 사용자 의견에 대해서는 직접적으로 수집하거나 전체를 수작업 처리할 수 있으나, 대량으로 축적된 사용자 의견들을 처리하기 위해서는 상당한 시간과 노력이 필요하여 비효율적이다. 또한, 변화가 빈번한 모바일 앱 시장의 지배력과 지속적인 사용자 관심을 유지하기 위해서는, 보다 효율적이고 자동화된 방식으로 사용자 의견을 반영할 수 있는 방안이 필요하다.

본 연구의 목적은 모바일 앱에 대해 대량으로 축적된 사용자 의견으로부터, 유의미한 개선 요구사

항을 추출할 수 있도록, 비효율적인 수작업(manual) 처리 방식 대신 보다 적은 노력으로 처리할 수 있도록 정형화된 프로세스와 세부 방법을 제시하는 것이다.

본 연구에서 제시하는 프로세스는, 정형화와 자동화를 위해 크게 3개의 활동(Activity)으로 구성하고, 각 활동별로 3개의 태스크(Task)를 구성하여, 총 9개의 태스크와 태스크에서 사용할 기법들을 제시하였다. 이에 대해 3개의 모바일 앱을 대상으로 사용자 의견에 대한 요구사항 추출 업무를 수작업으로 처리한 결과와 비교 분석하였다. 제안한 기법을 적용하여 산출한 결과에 대해서 요구사항 추출 비율, 소요된 수행시간, 그리고 추출한 요구사항 내용 일치율의 3가지 항목을 기준으로 평가하고 검증하였다.

이를 통해 제안한 프로세스의 효율적인 자동화 가능성과 효과적 운영 가능성을 확인하였고, 도출된 문제점을 통해 향후 개선 방향을 도출하였다.

본 연구는 모바일 서비스 운영자에게 대량의 사용자 의견으로부터 보다 적은 노력으로 소프트웨어 및 시스템 개선에 반영할 수 있는 요구사항을 추출하는 방안으로 사용할 수 있다.

2. 관련 연구

2.1 요구사항 개발

Karl Wiegers는 요구공학을 크게 요구사항 개발과 요구사항 관리로 나누고, 요구사항 개발은 크게 추출, 분석, 명세, 검증의 절차를 제시하는 요구공학 프레임워크를 제시하였다(Weigers, 2003). 이후 수많은 연구자들에 의해 요구사항을 개발하는 다양한 관점의 연구들이 시도되었고, 사용자 중심의 요구공학 관점으로 요구사항을 추출하기 위해 실용적이고 다양한 인터뷰 방법이 제시되었다(Alistair, 2002). 요구사항을 추출하기 위해 다양한 기법이 연구되고 제시되고 있으나, 이는 사람의 언어를 문서로 표현한 정형적이고 명확한 문구를 대

상으로 하고 있어, 비정형적이고 애매모호한 사용자의 의견을 대상으로 하는 연구는 최근에 이루어지고 있다(Byun, 2012; Kim, 2010).

본 연구에서는 특히, 모바일 앱에 대한 개선 요구사항 추출에 집중하여 이를 자동화하기 위한 효율적이고 효과적인 프로세스와 방법에 대해 제시하고자 한다.

2.2 사용자 의견의 요구사항 추출 프로세스

요구사항 개발을 위한 기본적인 절차를 기반으로, 보다 실용적이고 실무에 적용 가능하도록 구체적인 액티비티, 태스크 등을 정의하는 연구가 지속적으로 시도되고 있다(Dieste and Juristo, 2011; Soares et al., 2011).

특히, 변정원은 최근의 모바일 환경과 소셜 네트워크 환경에서 사용자 의견으로부터 소프트웨어 요구사항을 추출하기 위한 전반적인 프로세스와 기법들을 제시하고 있다(Byun, 2012). 사용자 의견 분류 및 요구사항 추출, 요구사항 정제 및 계층화, 요구사항 기여 정도 결정, 요구사항 타당성 분석, 요구사항 우선순위화 및 배정 계획 수립의 총 5개의 프로세스, 14개의 태스크, 11개의 추출 기법을 소개하고 있고, 연구 대상인 사용자 의견은 이메일, 사용자 리뷰, 댓글 등이다. 이 연구는 요구사항을 추출하기 위한 광범위한 절차를 제시하고 있으나, 이를 실무에 적용하기에는 다소 부담스럽다는 실무자의 의견이 있고, 이를 전반적으로 실행하기 위한 자동화 지원 부분에 대해서도 개선과 발전의 여지가 많다.

본 연구에서는 다양한 사용자 의견 중 사용자 리뷰와 댓글을 대상으로 하며, 이미 수집된 사용자 의견으로부터 개선 요구사항을 추출하기 위한 절차에 집중하며, 특히 자동화 처리 부분에 대해서 이전 연구에 비해 보다 실용적으로 접근하고자 한다.

2.3 한국어 문장 정형화

본 연구에서 대상으로 하는 사용자 의견은 한국

어로 작성되어 있으며, 요구사항을 자동으로 추출하고 처리하기 위해서는 한국어에 대한 특성을 고려해야 한다. 기존, 한국어에 대한 정형화된 표현에 관한 연구는 말뭉치의 계량적 분석을 통하여 빈번한 어휘의 연속단위, 즉 형태소의 연속체를 추출하여 정형화된 표현의 특성을 고찰하였다(Choi et al., 2010).

말뭉치를 대상으로 형태소의 연속을 분석하는 방법으로 N-gram의 개념을 이용하여, 동시에 나타나는 일련의 연속된 형태소의 집합을 나타냈다. 이를 통하여 길이, 빈도, 그리고 분포의 3가지의 기준으로 한국어의 정형화된 표현에 대하여 연구를 수행하였다. 이 중, 길이의 기준에서 5개 형태소가 연속되는 경우의 분석 효율성이 더 높음을 확인하였다. 본 연구에서는 한국어 문장의 정형화된 표현에 대한 기존의 연구 성과를 반영하여, 사용자 의견 문장의 정형성과 의미성 여부를 판단하는데 적용한다. 이를 위해 오픈소스인 한국어 형태소 분석기를 사용하여 문장을 걸러냈다(KLDP.net, 2009).

기존의 연구에서는 학술 교양서와 연설, 강의, 강연 및 일상대화 등 학술문어, 학술구어, 자유대화의 세 가지의 말뭉치를 중심으로 연구를 수행하였다. 그러나 본 연구에서는 문장의 분석이 용이하지 않은 사용자 리뷰와 댓글 의견을 대상으로 분석을 수행한다.

2.4 감정적 요소의 배제

모바일 앱에 대한 사용자 의견의 특성상, 사용자의 정성적 또는 감정적 표현을 나타내는 단어나 패턴이 많이 포함되어 있다. 예를 들면, 사용자의 감정 표현(좋아요, 편하네요, 불편해요), 감정 기호(^^;;, *^^*), 연속 자음(ㅋㅋ, ㅎㅎ), 연속 모음(ㅠ ㅠ, ——) 등이 있다. 이렇게 문장 안에 포함되어 있는 감성적인 요소는 글로 제시하는 의견을 좀 더 강조하는 효과를 줄 수 있어, 분석 기법에 따라 문장 의미의 우선순위를 결정할 때 활용할 수 있다.

기존 연구에서는 대량의 웹포럼 게시글을 자동으로 수집하여 텍스트의 의미, 구조, 어휘 등의 특

성을 도출하고, 정보적 지지 글과 감성적 지지 글의 유형을 자동적으로 탐지하고 분류하는 모델을 제시하였는데, 이 때 감성적 지지 글을 분류하는 변수로 감성적 표현 및 단어가 중요한 변수로 작용하였음을 제시하였다(Woo et al., 2012).

하지만, 본 연구에서는 사용자 의견의 추출 및 정제 시에 정형적인 부분에 대해서 집중하기로 하고, 문장 내에 포함되어 있는 감성적 요소에 대한 처리는 연구 범위에서 제외한다.

2.5 요구사항의 유형 분류

요구사항의 유형에는 비즈니스, 사용자, 시스템, 기능 요구사항 등이 있으며, 이러한 요구사항들은 일련의 계층적 구조 및 관계를 가진다. 또한 다양한 요구사항의 유형과 이에 수반되는 정보들의 관계에 대하여 언급하고 있다(Weigers, 2003). 이런 요구사항의 유형 분류 및 분석을 위해서는 요구사항 분석가의 다양한 경험적인 접근이 매우 필요하다(Fraser et al., 1991).

이런 경험적 접근으로 인한 정성적 요소를 감소하고 체계적인 분류를 위해, 요구사항의 다양한 유형을 고려하고 이를 분류하기 위한 방법으로 언어학(Linguistics) 기반 접근방식의 연구가 진행되어 왔다(Hussain et al., 2008, Fliedl et al., 2000). 이와 관련되는 대부분의 연구에서는 정형화된 소프트웨어 요구사항 명세서를 기반으로, 명세된 각 요구사항의 유형을 분류하거나 계층구조를 확인하였고, 연구 중 일부는 일부 규칙과 분류기준을 제시하였으나, 분석가의 경험적 판단에 의해 분석방법이 사용되기도 하였다. 또한 각 요구사항을 분류하기 위해 자동화 방안과 도구를 연구 개발하기도 하였다(Cleland-Huang et al., 2007).

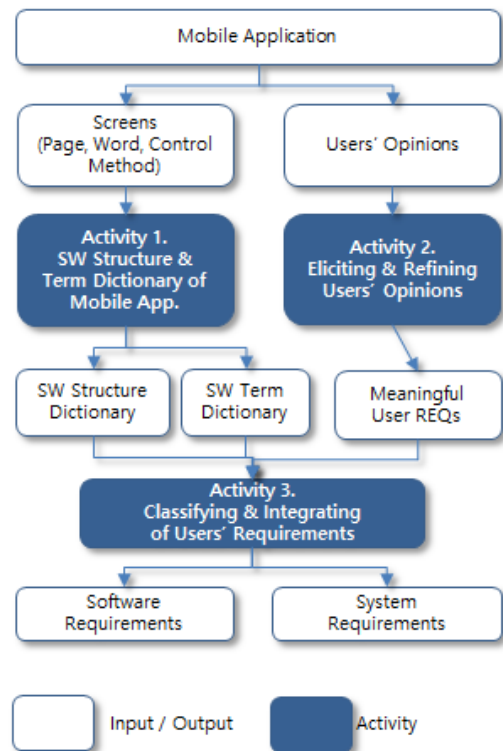
그러나 기존 연구는, 한국어의 어문 구조 및 특성과 상이한 영어를 기반으로 수행되어 적용에 어려움이 있다. 또한 근본적으로 본 연구의 분석 대상인 사용자 의견은 정형화된 유형의 소프트웨어 요구사항 명세서가 아니므로, 기존 연구의 절차와

기법을 적용함에 어려움이 있다.

본 연구에서는 주어진 사용자 의견을 추출 및 정제된 문장에 대해 소프트웨어 구조 사전과 용어 사전을 기준으로 소프트웨어 요구사항과 시스템 요구사항으로 분류하고자 하며, 사용자 의견을 분석하고 분류하는데 발생하는 노력을 절감하도록 절차와 태스크를 제안한다.

3. 모바일 앱에 표현된 사용자 의견 추출 및 정제 프로세스

모바일 앱에 표현된 사용자 의견을 추출하고 정제하여 소프트웨어와 시스템에 대한 개선 요구사항으로 정리하는 프로세스는 <Figure 1>과 같다. 제안하는 프로세스는 크게 3개의 액티비티로 구성된다.



<Figure 1> Processes for Eliciting Software and System Requirements from Users' Opinions in Mobile App

첫째 액티비티는 ‘모바일 앱의 소프트웨어 구조 사전 및 용어사전 구축’으로 모바일 앱 화면에 대한 정적인 구조와 동적인 소프트웨어 구조를 매핑한 소프트웨어 구조사전 구축과, 구축된 소프트웨어 구조 사건의 용어를 핵심어로 삼고 이 핵심어와 유사어, 동의어를 가진 용어사전을 구축한다.

둘째 액티비티는 ‘사용자 의견 추출 및 정제’로 모바일 앱에 표현된 사용자의 의견으로부터 소프트웨어 구조사전과 용어사전을 참조하여 개선요구로 의미있는 문장을 추출 정제한다.

셋째 액티비티는 ‘사용자 요구사항 분류 및 통합’으로 추출된 문장을 핵심어를 기반으로 한 단문장의 형태인 요구사항으로 재정리 및 분리하고 중복된 요구사항을 제거한다. 정리된 요구사항이 핵심어에 포함하는지 여부에 따라 소프트웨어 요구사항인지 시스템 요구사항인지를 결정한다.

3.1 모바일 앱의 소프트웨어 구조 사전 및 용어사전 구축(Act. 1)

본 액티비티에서는 모바일 앱의 구조를 분석하여 사용자의 의견이 모바일 소프트웨어 또는 시스템의 개선 요구사항인지를 판단하기 위한 기준을 수립한다. 이를 위해, 모바일 앱의 구조를 표현하는 소프트웨어 구조사전과, 모바일 앱에서 사용된 용어사전을 구축한다.

3.1.1 화면 분석(Task 1-1)

모바일 앱의 화면은 사용자와의 인터페이스 창구(GUI)이면서 소프트웨어 구조의 일부분이다. 또한 화면과 화면, 객체와 객체의 연결 관계 및 연결 기법에 대해서 정의하고 분석한다.

이를 위해 다음의 절차를 통해 화면 분석을 실시한다.

첫째, 모바일 앱 화면의 정적인 구성 요소들을 객체로 인식하고 이를 계층적 구조로 표현한다. 모바일 앱 화면의 분석은 기존에 작성한 문서가 있으면 이를 활용하되, 없으면 수작업으로 문서화 또는

자료화 한다. 각 화면에 존재하는 버튼, 또는 특정 동작을 통하여 사용자의 입력이나 서비스를 제공하는 부분도 화면 내의 객체로 식별 처리한다. 둘째, 모바일 앱 화면의 동적인 구성, 즉 UI나 UX를 포함한 인터페이스 방식을 UML을 이용하여 기술한다. 또 각 화면 내에서의 단위 기능 및 화면 간 이동과 관련된 단어도 관련된 용어로 인식한다. 또한, 화면을 구성하는 객체를 제어하는 방식(탭, 드래그, 더블탭, 확대, 축소 등)을 기술한다. 셋째, 소프트웨어 기능과 무관하거나 불필요한 단어는 제거한다.

3.1.2 모바일 앱 소프트웨어 구조사전 구축 (Task 1-2)

사용자가 모바일 앱의 장·단점과 사용의 용이성과 불편함 등을 표현할 때 명사로 사용할 수 있는 소프트웨어 구조 관련 핵심어를 기반으로 소프트웨어 구조사전을 구축한다. 이를 위해 다음의 절차를 실시한다.

첫째, 모바일 앱 화면을 분석한 자료를 이용하여 핵심어를 추출하여 소프트웨어 구조로 표현한다. 식별된 화면 객체 용어와, 이를 제어하는 방식, 인터페이스 방식 및 페이지의 전환 관계 등을 고려하여 소프트웨어 구조사전을 구축한다. 둘째, 모바일 앱을 구성하는 화면 내의 프레임, 페이지 및 GUI에 대한 핵심어도 구조사전에 포함한다.

소프트웨어 구조 사건의 구축은 모바일 앱을 구성하는 화면 내의 프레임, 페이지 및 GUI를 소프트웨어 계층도로 작성하거나 액셀을 활용해 정리한다.

3.1.3 모바일 앱 소프트웨어 용어사전 구축 (Task 1-3)

모바일 앱 사용자들은 화면의 단어를 핵심어로 사용하지만, 때로는 자기의 관습이나 방식대로 방언과 비표준어 및 유사어를 사용하여 표현할 수 있으므로, 이를 보완 처리하기 위하여 용어사전을 구축한다. 이를 위해 다음의 절차를 실시한다.

첫째, Task 1-1을 통해 생성된 각 화면 당 용어

목록에 대한 동의어 및 유사어를 검색한다. 특정 어휘와 관련된 동의어 및 유사어를 검색하기 위해 21세기 세종계획에 의거 개발된 말뭉치와 말뭉치 활용도구를 이용한다(The National Institute of Korean Language, 2007). 둘째, 확인된 동의어 및 유사어 목록을 보고 본 연구 목적에 해당하지 않는 경우를 제외한다. 셋째, 추출된 동의어 및 유사어를 용어목록에 추가하여 각 화면에 대한 용어사전을 구축한다.

3.2 사용자 의견의 추출과 정제(Act. 2)

본 액티비티는 모바일 앱에 게시된 고객의 의견 및 게시판 등에 나타난 다양한 사용자 의견을 추출 정제하여 사용자 요구사항으로 1차 정리하는 활동이다.

3.2.1 사용자 의견의 수집 정리 및 단문화 (Task 2-1)

댓글 형태의 사용자 의견은 구성이 체계적이지 못하고 사용자마다 표현하는 형식이 다양하므로 이를 정형화된 단문으로 정리하기 위해서 다음의 절차를 통한다.

첫째, 모바일 앱에 나타난 사용자 의견을 수집하여 자료화 한다. 둘째, 수집된 자료 중에 한 개의 단어, 자음이나 모음만으로 구성된 것과 같은 문장의 기본요건을 갖추지 않은 자료들을 삭제한다. 셋째, 복합 문장이거나 중문은 단문으로 분할 정리한다. 이를 위하여 한국어 문장을 종결하는 어말어미 유형(예 : -다, -느냐, -구나 등)과 문장을 종결하는 기호인 마침표, 느낌표 및 물음표 등을 기준으로 문장을 단문화한다.

3.2.2 불필요한 의견 제거(Task 2-2)

Task 2-1을 통해 단문화된 결과에 대해 분석에 불필요한 요소들의 패턴을 미리 준비하고 이를 단문과 비교하여 등록된 패턴이 포함되어 있으면 제거하는 식으로 분석 대상을 걸러낸다. 관련 연구에서 언급했듯이, 문장 내의 감정 요소는 추출한

요구사항의 의미적 분석에 영향을 주는 요소이기는 하나, 본 연구 범위를 벗어나므로 다음과 같은 절차를 통해 불필요한 표현을 제거한다.

첫째, 단문 중 사용자의 감정 표현(예 : 좋아요, 편하네요 등)과 같은 불필요한 요소들을 걸러낸다. 둘째, 불필요한 기호와 이모티콘 등이 사용된 것을 제거한다. 셋째, 연속 자음을 사용한 경우(예 : ㅋㅋㅋ, ㅎㅎ 등)와 연속 모음을 사용한 경우(예 : 一一, ㅍㅍ 등)를 제거한다. 넷째, 익명성이 있는 댓글의 특성상 욕설도 제거한다.

3.2.3 유의미한 사용자 요구사항 추출 (Task 2-3)

불필요한 요소가 제거된 단문 결과 중에서, 분석이 가능한 수준의 단문을 추출한다. 한국어의 정형화된 표현 연구(Choi et al., 2010)에서는 5개 이상의 문장 성분이 있으면 분석의 효율성이 높다고 하였다. 그러나, 비정형적인 사용자 의견이라는 특성상 실제로는 2개 이상의 문장 성분이 존재하면 분석이 가능한 문장이라고 판단하고 이를 유효한 요구사항 문장이라고 볼 수 있다.

이를 위해, 대상 단문에 대해 형태소 분석을 실시하여 한국어 문장 성분을 분리한 후 2개 이상의 문장 성분으로 구성된 단문을 사용자 요구사항으로 추출한다

3.3 사용자 요구사항 분류 및 통합(Act. 3)

본 액티비티에서는 액티비티 2를 통해 1차 정리된 사용자의 요구사항을 소프트웨어와 시스템 개선 요구사항으로 분류한다. 이를 위해, 체언이나 용언을 구조 사전에서 정리한 핵심어를 기준으로 비교하여 분류한다. 분류된 요구사항에 대해서 유사 중복된 요구사항은 제거하고 최종 요구사항 목록을 작성하여 관련 이해당사자에게 전달한다.

3.3.1 핵심어 추출(Task 3-1)

액티비티 2를 통해 정리된 요구사항은 문장을 한국어 8품사로 구분하여 단문으로 정형화한 것이

다. 본 태스크에서는 1차 정리된 문장 중에서 체언과 용언을 소프트웨어 구조사전과 용어사전을 참고로 하여 핵심어 기반 요구사항 명세로 전환하여 재정리 한다. 정의된 요구사항 명세 중에 핵심어가 포함되면 소프트웨어 관련 요구사항이고 아니면 시스템 관련 사항으로 분류한다.

핵심어를 추출하기 위하여 다음과 같은 절차를 실시한다. 첫째, 체언에 연결된 관계언을 삭제함으로써 명사 성분을 식별한다. 둘째, 용언과 접사 문장 성분이 분석된 결과 중 받침이 분리된 경우에 대하여 이를 정제한다. 셋째, 단어를 수식하는 것으로 그 기능이 한정되는 수식언을 삭제한다. 이를 통해 핵심어 분석의 용이성을 확보한다.

3.3.2 사용자 요구사항 분류(Task 3-2)

핵심어를 기반으로 정리된 요구사항 명세를 정렬하여 사용자 요구사항을 분류한다. 이를 위해 다음과 같은 절차를 실시한다.

첫째, 유사어 또는 동의어가 포함된 문장을 제거하고 재정리한다. 둘째, 정리된 문장은 다시 수식언과 독립언에 대해 용어사전을 참조하여 표준화하여 유사 또는 중복을 배제한다. 셋째, 소프트웨어 요구사항으로 분류되지 않은 요구사항을 직접 재검토하여 누락 여부를 확인하여 처리한다. 넷째, 그 이외의 경우에는 소프트웨어 범위 이외의 요구사항으로 판단하여 시스템 개선과 관련된 요구사항으로 분류 처리한다.

3.3.3 사용자 요구사항 통합 저장(Task 3-3)

분류된 사용자 요구사항은 다음의 절차를 통해 최종적으로 관련 이해당사자에게 전달한다.

첫째, 분류된 사용자 요구사항에서 중복을 제거하고, 유사한 요구사항인 경우는 통합한다. 이를 위하여 사용자 요구사항의 핵심어를 기준으로 각 사용자 요구사항을 정렬하고 그룹화(grouping)한 후, 분석가의 직관적 판단과 분석을 통해 중복 및 유사 요구사항 여부를 결정하고 처리한다. 둘째, 정리 분류된 사용자 요구사항은 중요성을 점검하

고 통합 저장관리를 실시한다. 셋째, 명세화된 사용자 의견에 대하여 이상 유무를 검토한다. 넷째, 확정된 요구사항은 해당 부서 및 관련 이해관계자에게 이관하여 실제 개선활동에 활용할 수 있도록 한다.

4. 제안 프로세스의 적용

본 연구에서 제안한 프로세스의 유효성을 검증하기 위하여 분석대상 모바일 앱을, 전자도서 콘텐츠를 열람할 수 있는 I사 B전자도서 앱, 금융거래서비스를 제공하는 E금융 앱 및 K사 금융 앱, 3가지로 선정하였다.

본 연구 주제에 대해 실무적으로 인식이 많이 부족하고, 트래픽 양과 사용자 의견 축적량에 대한 객관적인 기준이 수립되지 않은 관계로, 일 평균 대량 트래픽 발생 여부(약 10만 건 이상), 운영자의 본 연구 주제에 대한 공감과 지원 여부, 사용자 리뷰 및 댓글의 축적된 양 등을 고려하여 해당 모바일 서비스를 선정하였다. 이를 대상으로 제안한 방법을 적용하였고, 자동화 가능성과 운영의 효율성을 검토하였다.

4.1 Activity 1 적용 결과 : 모바일 앱의 SW

구조사전 및 용어사전 구축

사례 대상 모바일 앱의 전체 화면 중 외부 연계(페이스북, 트위터 등)의 경우나, 웹 페이지 호출(프로모션 관련)의 경우는 제외하였다. 실제로 대상이 된 화면들에 나타난 전체 단어들을 추출하고, 화면의 기능 및 동작에 관련되지 않은 단어들이 인식되어 육안 검사를 통해 불필요한 단어를 제거하였고(Task 1-1 결과), 일부 누락된 용어를 보완하는 등 화면 단위의 용어를 보완하여 추출하였다(Task 1-2 결과). 이를 기반으로 말뭉치 사전을 이용하여 동의어 및 유사어를 추가하여(Task 1-3 결과), 모바일 앱 소프트웨어 구조사전 및 소프트웨어 용어사전을 구축하였다.

<Table 1> Construction Result of Mobile App's SW Structure and Term Dictionary

	Screen	Terms	Synonyms
Case 1	15	44	32
Case 2	22	112	6
Case 3	18	52	24

구축한 결과는 <Table 1>과 같다. 첫 번째 사례의 경우, 15개의 화면에 대해 44개의 화면 용어(핵심어)와 32개의 동의어 및 유사어를 포함하여 76건의 소프트웨어 용어사전을 구축하였고, 두 번째 사례의 경우 22개 화면에 대해 112개의 용어와 6개의 유사어를 포함하여 118개의 용어사전을 구축하였고, 세 번째 사례의 경우, 18개 화면에 대해 52개의 용어와 24개의 유사어를 포함하여 76건의 소프트웨어 용어사전을 구축하였다.

4.2 Activity 2 적용 결과 : 사용자 의견 추출 및 정제

4.2.1 Task 2-1 적용 결과 : 사용자 의견 수집 및 단문화

태스크의 적용 과정은, 우선 사용자 의견 데이터베이스를 쿼리하여 그 결과를 엑셀 파일로 저장하고, 엑셀 목록을 실험자가 직접 확인하여 문장의 기본요건을 갖추지 않은 자료들을 목록에서 삭제했다. 복합 문장이거나 중문들을 단문으로 분할 정리하기 위해, 엑셀의 기능을 이용하여 문장 내의 어말어미 유형과 문장을 종결하는 기호 등을 기준으로 문장을 분리하여 단문화한 목록을 생성하였다.

비정형적인 문장 구성으로 인해, 기본적인 문장 종결 어말어미와 종결기호를 기준으로 한 단문화는 만족할 만한 수준은 도달하지 못했다. 이는 다양한 형태의 자연어적인 처리를 위한 규칙이 부족한 것으로 판단하여 문장 종결 어말어미와 종결 기호의 추가 보완하여 제시도 한 결과 만족할 만한 수준으로 <Table 2>와 같은 결과를 얻었다.

<Table 2> Transforming to Simple Sentences from User's Opinions

	Raw Data	Simple Sentence
Case 1	40	110
Case 2	104	247
Case 3	62	149

첫 번째 사례의 경우, 수집한 40건의 문장으로부터 서로 다른 의미를 지닌 문장으로 분할하여 110건의 단문이 생성되었다. 두 번째 사례는 수집된 104건의 문장을 단문화한 결과 247건의 단문을 생성하였고, 세 번째 사례는 62건에서 149건의 단문을 생성하였다.

4.2.2 Task 2-2 적용 결과 : 불필요한 의견 제거

태스크의 적용은 단문화 된 사용자 의견을 대상으로 불필요한 요소를 삭제하도록 제거 규칙을 적용하고 새로운 문장으로 저장하였다. 단문 중 사용자의 감정 표현과 같은 불필요한 요소들은 문장 내에서 삭제하고 기호와 이모티콘 등을 제거했다. 연속 자음/모음을 사용한 부분도 제거하고 익명성이 있는 댓글의 특성상 욕설도 제거했다.

실제 사례에서는 불필요한 표현이라고 판단할 수 있는 규칙이 충분하지 않아 제거 결과는 만족할 만한 수준은 아니었다. 불필요한 표현에 대해서는 지속적인 추가를 통해 규칙의 보완이 이루어질 수 있지만, 불필요한 의견에 대해서는 자연어적인 처리를 위한 추가적인 연구가 필요하다. 이에, 대상이 되는 단문 내용 중 규칙이 될 만한 요소들을 추가하여 다시 진행하였다. 이에 대한 결과는 <Table 3>와 같다. 첫 번째 사례에서는 110개의 단문화 의견 중 64건이 불필요한 의견으로 분류되었고, 두 번째 사례에서는 247건의 단문화 의견 중 176건이 불필요한 의견으로 분류되었고, 세 번째 사례에서는 149건의 단문화 의견 중 72건이 불필요한 의견으로 분류되었다. 표 안의 표시한 비율은 전체 단문화 건수 대비 각 케이스에 대한 비율을 의미한다.

<Table 3> Removal of Unnecessary Opinions

Case	Simple Emotional Expression	Simple Postscript	Simple Request	Etc	Sum
1	35	15	3	11	64
	32%	14%	3%	10%	59%
2	87	58	6	25	176
	35%	23%	2%	10%	70%
3	25	10	4	33	72
	17%	7%	3%	22%	49%

4.2.3 Task 2-3 적용 결과 : 유의미한 사용자 요구사항 추출

불필요한 의견을 제외하여 필요한 의견이라고 분류한 단문화 의견에 대해서 형태소 분석을 통한 문장 성분을 분리하고, 2개 미만의 문장 성분으로 구성된 사용자 의견은 제거하는 기본 규칙을 적용하여 유의미한 사용자 요구사항으로 추출한다. 추출된 결과를 분석한 결과, 기존 연구 결과의 기준 적용을 통해 필터링 된 단문 중에서도 의미가 있는 것으로 판단되는 경우도 있었다.

<Table 4> Acquisition of Meaningful Requirements

Case	Procedure of Requirement Eliciting (REQ : Requirement)	Result	Ratio of Decrease
1	Simple Sentences	110	-
	Necessary Opinions	46	58.2%
	Meaningful Users' REQs	29	37.0%
2	Simple Sentences	247	-
	Necessary Opinions	71	71.3%
	Meaningful Users' REQs	55	22.5%
3	Simple Sentences	149	-
	Necessary Opinions	77	48.3%
	Meaningful Users' REQs	34	55.8%

현재 시점에서는 대상 데이터가 많지 않은 관계로 이런 경우는 필터링 된 것으로 판단하고 진행하였다. <Table 4>에서 표현한 두 가지 감소 비율

의 의미는, 첫째는, 단문화된 사용자 의견 개수 대비 불필요한 의견 개수의 비율을 의미하고, 둘째는, 필요한 의견 개수 대비 무의미한 사용자 요구사항 개수의 비율을 의미한다.

4.3 Activity 3 적용 결과 : 사용자 요구사항 분류 및 통합 저장

유의미한 사용자 요구사항을 대상으로 핵심어 추출(Task 3-1 결과)과 소프트웨어 구조사전 및 소프트웨어 용어사전과의 비교를 통한 분류 과정을 거쳤고(Task 3-2 결과), 유사하고 중복된 요구사항을 통합하였다(Task 3-3결과). 비교 분류 과정은 정형화된 단어 간의 비교이므로 자동화가 가능한 부분이지만, 요구사항의 유사성과 중복성에 대한 판단은 사람의 직관적인 판단에 의존할 수 밖에 없다. 그 결과는 <Table 5>과 같다. 최종 결과 중, 자연어 처리의 불완전함으로 인하여 사전에 제거하지 못한 무의미한 의견이 포함된 경우가 발견되어 이 부분은 제거하였다.

<Table 5> Classifying and Integrating of Requirements

Case	Procedure of Classification and Integration (REQ : Requirement)	Result	Ratio of Decrease
1	Meaningful User REQs	29	-
	Final REQs(SW+System)	22	24.1%
2	Meaningful User REQs	55	-
	Final REQs(SW+System)	22	60.0%
3	Meaningful User REQs	34	-
	Final REQs(SW+System)	27	20.6%

5. 제안한 방식과 수동 방식과의 비교 검증

본 연구에서 제안한 프로세스 및 방법에 대한 유효성을 검증하기 위하여, <Table 2>의 각 사례별로 단문화된 결과를 사용하여 수동 방식을 통한

요구사항 추출 과정과 제안한 프로세스와 방법을 이용한 반자동화 방식의 추출 과정을 비교 수행하였다.

수동 방식을 수행한 실험자는 본 연구에서 사용한 모바일 애플리케이션에 대해 사전 지식이 없는 상태였다. 또한, 사용자의 댓글 의견으로부터 모바일 앱의 소프트웨어 및 시스템 개선 요구사항을 추출하는 본 연구의 제안 프로세스에 대해서도 사전 지식이 없는 상태였으며, 타 모바일 앱의 유지보수 업무 관련 담당자를 대상으로 선정하였다.

실험 시작과 함께, 수동방식 실험자에게 제안 프로세스의 활동, 태스크 및 산출물 양식 등을 설명하여 분석에 활용할 수 있도록 하였으며, 본 연구에서 사용한 모바일 앱에 대해 모바일 앱의 구성, 주요기능 등에 관한 기본사항을 교육하였다. 그 후, 수동 방식 실험자는 모바일 앱의 화면 분석서를 작성하고, 소프트웨어 구조사전과 용어사전을 작성하였다. 그리고, 주어진 사용자 의견 문장을 단문화하였고, 불필요한 의견을 제거하고 유의미한 사용자 요구사항 추출하였다. 추출한 요구사항에 대해 다시 요구사항을 분류하고, 중복 요구사항 제거하거나 유사 요구사항 통합하는 과정을 실시하였다.

본 비교 검증 과정은, 프로세스 및 과정 산출물에 대한 정확한 측정이 목적이 아니라, 사전 지식이 없는 사람에 의한 요구사항 추출 방식에 비해 반자동화된(일부는 수동, 일부는 툴을 사용) 프로세스의 수행 방식이 얼마만큼 효과가 있는지를 확인하는 수준으로만 진행하였다. 실험은 수동 방식 대비 제안 프로세스 적용시 요구사항 추출 및 분류 비율 비교, 수동 방식 대비 프로세스 적용 시 소요 수행시간 비교, 수동 방식 대비 프로세스 적용 결과의 내용의 일치성 비교를 실시하였다.

실험자에 의한 요구사항 추출 결과와 본 연구에서 제시한 프로세스를 적용한 결과를 비교해 보면, 사례마다 편차가 존재하나, <Table 6>에서 보듯이, 소프트웨어 신규 요구사항의 경우 수동 방식 대비 62.5%에서 87.5%의 식별이 가능하고, 소프트

웨어 변경 요구사항의 경우 수동 방식 대비 66.7%에서 83.3%의 식별이 가능하였다. 그 외 시스템 개선과 관련된 요구사항의 식별 비율 결과는 50%~80% 수준이다.

이 결과는, 제한된 수행 환경과 소량의 사례를 대상으로 한 결과이므로 정확한 효율이라고 단정하기는 어려우나, 다수의 모바일 애플리케이션과 대량의 사용자 의견을 대상으로 제안한 방식의 업무 처리 절차를 적용하여 추출된 요구사항이 모바일 서비스 운영자의 의사결정과 서비스의 질적인 향상에 보다 효율적으로 기여할 수 있다고 판단할 수 있다.

<Table 6> Ratio of Requirement Eliciting

Case	Classification (REQ : Requirement)	Manual	Proposed	Ratio of Identification
1	Input	110	110	-
	SW Change REQs	12	10	83.3%
	SW New REQs	8	7	87.5%
	System REQs	2	1	50.0%
2	Input	247	247	-
	SW Change REQs	9	6	66.7%
	SW New REQs	8	5	62.5%
	System REQs	5	4	80.0%
3	Input	149	149	-
	SW Change REQs	17	13	76.5%
	SW New REQs	7	6	85.7%
	System REQs	3	2	66.7%

실험자의 수동 방식의 수행 시간 대비 제안한 절차 및 기법을 적용한 반자동화 수행 시간의 비교 결과는 <Table 7>과 같다. 3가지 사례를 통해 수동 방식의 수행시간 대비 제안한 프로세스의 반자동화 수행한 시간이 약 52%에서 70% 감소함을 알 수 있었다.

이 결과는, 사람의 숙련도와 경험에 따라 많은

편차를 보일 여지가 있는 부분이다. 그러나, 본 연구는 사람의 업무 처리 능력 수준에 상관없이 다수의 애플리케이션과 대량의 사용자 의견으로부터 의미있는 사용자 요구사항을 추출하는 과정을 자동화하기 위한 연구이므로, 이러한 시간 감소는 마찬가지로 서비스의 효율적인 운영에 기여할 수 있다고 판단할 수 있다.

<Table 7> Comparison of Elapsed Execution Time

Case	Manual	Proposed	Ratio of Decrease
1	186min.	57min.	69.4%
2	153min.	72min.	52.9%
3	143min.	62min.	56.6%

여기에 실질적으로 실험자가 식별한 요구사항과 제안한 프로세스를 통해 식별한 요구사항이 내용적인 일치성을 비교해 본 결과는 <Table 8>와 같다.

<Table 8> Semantic Analysis of Requirements

Case	Classification	Human Agreement	Ratio of Agreement	
1	SW Change REQs	12	7	58.3%
	SW New REQs	8	4	50.0%
	System REQs	2	0	0%
2	SW Change REQs	9	4	44.4%
	SW New REQs	8	3	37.5%
	System REQs	5	2	40.0%
3	SW Change REQs	17	6	35.3%
	SW New REQs	7	1	14.3%
	System REQs	3	1	33.3%

사람이 추출하고 식별한 요구사항이 정확하든지 여부를 판단하기 어려운 상황에서 이 결과를 통해 일치율이 좋다 나쁘다의 판단은 무의미하다. 다만, 숙련된 전문가가 요구사항을 추출하는 수준에 이르도록 프로세스를 더욱 정교하게 만들고, 방법상의 규칙 및 기법을 보완을 통해 자동화 업무의 효과성을 향상시킬 수 있을 것으로 판단된다.

6. 결 론

6.1 연구의 요약 및 의의

모바일 애플리케이션의 다양한 사용자 의견을 사용하여 소프트웨어 및 시스템이 개선된다면 사용자에게 보다 양질의 서비스를 제공할 수 있다. 본 연구에서는 대량의 사용자 의견을 소프트웨어 및 시스템 개선에 반영하기 위한 요구사항을 추출하는 프로세스 및 방법을 제안하였다.

제안한 프로세스 및 방법의 자동화 가능성과 유효성을 검증하기 위하여 제한된 실험 환경 하에서 각 과정을 반자동화 방식과 수작업에 의한 방법으로 처리해 보았다.

그 결과, 요청되는 정밀도에 따라 자동화의 수준은 차이가 있겠지만, 운영의 효과성 측면에서도 수동 방식 대비 최대 87% 정도로 추출이 가능하며 최대 70%의 수행시간이 감소하여 효율적인 요구사항의 추출이 가능함을 보였으며, 사람이 직접 추출한 요구사항과의 내용 일치성 비교 측면에서도 현재 수준에서의 효과성을 확인할 수 있었다.

본 연구는 사람의 업무 숙련도와 경험에 관계없이, 다수의 애플리케이션과 대량의 사용자 의견으로부터 소프트웨어 및 시스템 요구사항을 추출하는 과정을 자동화하기 위한 기반 연구로서 그 의미가 있다.

6.2 연구의 시사점

본 연구의 결과를 통해 다음과 같은 시사점을 제시할 수 있다.

첫째, 본 연구에서 제시하는 프로세스 및 방법을 기반으로, 보다 정확하고 체계적인 요구사항 추출 기법을 연구하기 위해서 응용 아키텍처에 대한 근본적인 설계 반영이 필요하다. 본 연구에서는 서비스 개시 이후에 축적되는 비정형적인 사용자의 의견을 중심으로 개선 요구사항을 추출하는 시도를 했지만, 이와 더불어, 응용 아키텍처 내에 서비

스 개시 이후 발생하는 개선사항을 정형적이고 지속적으로 수집하고, 이를 실시간으로 분석할 수 있도록 근본적인 아키텍처 설계가 필요할 것으로 판단된다.

둘째, 현업 실무자들에게도 여러 가지 접근 방식으로 도출된 개선 요구사항을, 실제 운영 시스템에 반영하기 위한 방안이 필요하다. 현업 실무자들은 도출된 요구사항에 대한 직접적인 이해당사자로서 실무에 반영하기 위한 요구사항의 우선순위화, 각 요구사항의 성격에 따른 접근 방식의 선택, 서비스의 운영관리 영향도에 따른 비용적인 가치 부여 등 여러 고려사항을 종합하여 합리적인 의사결정을 할 수 있는 방안이 필요하다. 또한 서비스 개시 이후에 나타나는 운영상의 여러 가지 증상들을 수집하고 이를 통계적 분석이 용이하도록 정확한 서비스 및 시스템 아키텍처 요구사항을 제시하는 것도 필요하다.

6.3 향후 연구

본 연구에서 수행한 사례는, 제한된 수행 환경과 자료를 대상으로 하였기 때문에 명쾌하게 문제를 해결했다고 볼 수 없으며, 향후 추가적이고 집중적인 보완 연구가 많이 필요하다.

우선, 대상 모바일 서비스의 선정에 있어서 실험의 자료가 되는 사용자 의견의 축적량을 기본적인 선정 기준으로 삼았으나, 사용자, 활용 도메인, 분류 체계 등 선정 기준을 보다 다양화하고 객관화할 필요가 있다.

비정형적인 한국어 문장을 다루는 연구이므로, 자연어 특히 한국어 처리에 대한 보다 심도 있는 연구가 필요하며 관련 처리 방법을 위한 규칙 및 기준에 대한 관련 연구가 추가 되어야 한다.

제안한 절차 중에 감정 표현이나 기호, 이모티콘, 연속자음/모음 등은 불필요한 표현으로 제거하였으나, 사용자들이 해당 모바일 서비스를 사용하면서 느끼는 정성적 또는 감성적 반응일 수 있으므로, 문장 내에 포함되어 있는 감정적인 요소들을 추출한 요구사항의 의미에 부가적으로 부여하

는 추가 연구가 필요하다.

마지막으로, 대상 앱에는 해당하지 않았으나, 모바일 앱에 대한 의견을 제시하는 방법 중에, 별표(Star-Rating)를 통해 호감도를 입력하는 기능을 제공하는 앱의 경우에 대해서, 본 연구에서 제시하는 요구사항 추출 프로세스와 함께, 추출된 요구사항에 대한 민감도와 우선순위를 시스템적으로 부여하는 연구를 추가한다면, 모바일 서비스 실무자들이 보다 정확한 의사결정을 할 수 있는 지원체계를 제공할 것으로 기대한다.

References

- Alistair, S., *User-Centered Requirements Engineering-Theory and Practice*, Springer, 2002.
- Byun, J.W., *A Process and Methods for Eliciting the Requirements from Users Opinion*, Doctoral Dissertation, Soongsil University, 2012.
- Choi, J., H.J. Song, and K.I. Nam, "Formulaic Expressions in Korean", *Journal of The Discourse and Cognitive Linguistics Society of Korea*, Vol.17, No.2, 2010, 163-190.
- Cleland-Huang, J., R. Settimi, X. Zou, and P. Solc, "Automated classification of non-functional requirement", *Requirements Engineering Journal*, Vol.12, 2007, 103-120.
- Dieste, O. and N. Juristo, "Systematic Review and Aggregation of Empirical Studies on Elicitation Techniques", *IEEE Transactions on Software Engineering*, Vol.37, No.2, 2011, 283-304.
- Fliedl, G., C. Kop, H.C. Mayr, W. Mayerthaler, and C. Winkler, "Linguistically based requirements engineering-The NIBA project", *Data and Knowledge Engineering*, Vol.35, No.2, 2000, 111-120.
- Fraser, M.D., K. Kumar, and V.K. Vaishnavi,

- “Informal and Formal Requirements Specification Languages : Bridging the Gap”, *IEEE Transaction on Software Engineering*, Vol.17. No.5, 1991, 454-466.
- Hussain, I., L. Kosseim, and O. Ormandijva, “Using Linguistics Knowledge to Classify Non-functional Requirements in SRS documents”, *Natural Language and Information Systems*, Vol.5039, 2008, 287-298.
- Kim, J.H., *A Process and a Technique for Software Requirements Elicitation*, Doctoral Dissertation, Soongsil University, 2010.
- KLDP.net, Hannanum-Korean Morphemic Analyzer, 2009. Available at <http://kldp.net/projects/hannanum> (Accessed July 19. 2014).
- Kujala, S., “User Studies : A Practical Approach to User Involvement for Gathering User Needs and Requirements”, *Acta Polytechnica Scandinavica, Mathematics and Computing Series*, Vol.116, 2002.
- The National Institute of the Korean Language, “21th Century Sejong Plan”, 2007, Available at <http://www.sejong.or.kr/> (Accessed September 2. 2014)
- Pressman, R.S., *Software Engineering : A Practitioner’s Approach 7th Edition*, McGraw-Hill, 2008.
- Soares M.S., J. Vrancken, and A. Verbraeck, “User requirements modeling and analysis of software-intensive systems”, *Journal of Systems and Software*, Vol.84, No.2, 2011, 328-339.
- Wieggers, K., *Requirements Engineering 2nd Edition*, Microsoft Press, 2003.
- Woo, J.Y., M.J. Lee, and Y.C. Ku, “The informative support and emotional support classification model for medical web forums using text analysis”, *The Journal of Information Technology Services of Korea*, Vol. 1, 2012, 139-152.

◆ About the Authors ◆



Dong-Seok Oh (dsoh@ssu.ac.kr)

Dong-Seok Oh received the B.E degree in Computer Engineering from Kwangwoon University, Seoul, Korea, in 1999 and the M.E degree in Software Engineering from Graduate School of Information Science in Soongsil University, Seoul, Korea, in 2007. He is Ph.D Candidate at Soongsil University, Seoul, Korea. He is working as a IT Projects' Quality Manager at OPEN-SNS & INSIDE-INFO, Korea. His current research interests include Project/Software Quality Management, Project Management Office, IT Project Requirement Engineering, IT Project Architecture, IT Project Engineering and related Tools.



Sun-Bin Kim (sbkim@ssu.ac.kr)

Sun-Bin Kim received the B.E degree in computer science and engineering from Soongsil university, Seoul, Korea, and M.E. degree in Soongsil University, Seoul, Korea in 2013. He is in the Ph.D. program at graduate school of Soongsil university as major in software engineering. His current research interests include software initiative, requirements engineering, software engineering, etc.



Sung-Yul Rhew (syrhew@ssu.ac.kr)

Professor Sung-Yul Rhew received Ph.D degrees in Computer Science from Ajou University and the MS degrees in Computer Science from Yonsei University , Seoul, Korea, in 1996 and 1980, respectively. He is working as a Professor of software engineering in school of computer science and engineering at the Soongsil University. His research interests include Software Requirement Engineering, Software Maintenance, Open Source Software.