

An Integrated Method of Iterative and Incremental Requirement Analysis for Large-Scale Systems

Jisung Park[†] · Jaeho Lee^{††}

ABSTRACT

Development of Intelligent Systems involves effective integration of large-scaled knowledge processing and understanding, human-machine interaction, and intelligent services. Especially, in our project for development of a self-growing knowledge-based system with inference methodologies utilizing the big data technology, we are building a platform called WiseKB as the central knowledge base for storing massive amount of knowledge and enabling question-answering by inferences. WiseKB thus requires an effective methodology to analyze diverse requirements convoluted with the integration of various components of knowledge representation, resource management, knowledge storing, complex hybrid inference, and knowledge learning. In this paper, we propose an integrated requirement analysis method that blends the traditional sequential method and the iterative-incremental method to achieve an efficient requirement analysis for large-scale systems.

Keywords : Requirement Analysis, Large-Scale System, Big Data

시스템 요구사항 분석을 위한 순환적-점진적 복합 분석방법

박지성[†] · 이재호^{††}

요약

인공지능 기반 지능형 시스템의 개발에는 일반적으로 신뢰성 높은 대규모 지식처리, 지식의 통합과 인간 수준의 이해, 지식기반 인간-기계 협업, 전문가 수준의 지능 서비스 등의 효과적 통합이 요구된다. 특히 빅데이터 이해 기반 자가학습형 지식베이스 및 추론 기술 개발을 목표로 하고 있는 과제의 일환으로 개발 중인 WiseKB 통합 플랫폼은 대용량 지식을 저장하여 추론과정을 통한 질의 및 응답이 가능한 대규모 지식 베이스 역할을 수행하며 이를 위하여 지식표현, 자원통합, 지식저장소, 지식베이스, 복합추론, 지식학습 등의 요소기술들의 효과적 통합이 필수적이다. 통합 플랫폼의 효율적 통합을 위해서는 정확한 요구사항 분석이 중요하며, 이는 시스템의 특성을 고려한 적절한 요구사항 분석 방법론의 적용이 필요하다. 대표적인 요구사항 분석 방법인 순차적 방법론과 순환-점진적 방법론은 WiseKB와 같은 시스템의 대규모 복합적 개발 특성을 고려할 때 다양한 요구사항을 체계적으로 파악하기에 한계가 있다. 본 논문에서는 이러한 한계를 개선하고자 순차적 방법과 순환-점진적 방법론을 결합해 각 단점을 보완하고 대규모 복합적 특성을 갖는 시스템의 요구사항 분석을 효율적으로 진행할 수 있는 통합 방법론을 제시하고, 실제 적용을 통해 그 효과를 보인다.

키워드 : 요구사항 분석, 대규모 시스템, 빅데이터

1. 서론

인류 사회가 정보화 사회로 변화하면서 정보의 생성, 가공과 유통은 인간의 모든 활동의 중심이 되었으며, 이에 따라 기업은 정보화 사회에서 경쟁우위를 확보하고 수익을 창출하기 위해 물품의 기본적인 데이터를 통해 추출한 정보나

지식을 중요한 요소로 활용하기 시작했다. 기존에는 기업 활동을 통해 생성되는 데이터가 주를 이루었지만, 21세기에 접어들면서 관계형 데이터베이스로 구축된 데이터가 ERP 등의 시스템을 이용해 가치 있는 정보를 추출하는 마이닝 과정을 거치며, 의미 있는 지식정보로 가공되어 기업의 의사결정에 이용되고 있다[1].

나아가 최근의 IT기술의 발전은 데이터의 가공 및 활용 뿐 아니라 생산경로의 확산이라는 측면에서도 큰 변화를 가져오고 있다. 이와 같은 변화로 인해 기업뿐만 아니라 개인을 통해서도 데이터가 생성/공유되면서 텍스트와 소리, 사진과 영상 등 데이터의 종류가 다양화됨과 동시에 생성되는

* 이 논문은 2014년도 서울시립대학교 교내학술연구비에 의하여 지원되었음.

[†] 비회원 : 삼성SDS 연구원

^{††} 종신회원 : 서울시립대학교 전자전기컴퓨터공학부 교수

Manuscript Received : April 25, 2016

First Revision : December 28, 2016

Accepted : December 28, 2016

* Corresponding Author : Jaeho Lee(jaeho@uos.ac.kr)

데이터의 양과 속도 또한 이전과 비교할 수 없을 만큼 커졌다. 통계에 의하면 2009년 전 세계의 데이터 생성량은 약 0.79제타바이트에 불과했지만, 이는 점진적으로 확장되어 2020년에는 44배가 증가될 것이며, 또한 이 중 70퍼센트 이상의 데이터가 개인을 통해 생성될 것이라고 예측했다[2].

이와 같은 요구에 따라 최근의 기술은 대용량의 다양한 데이터를 처리할 수 있는 기술로서 빅데이터를 제안하고 있다. 빅데이터는 위와 같이 폭발적으로 증가하는 다양한 정형 및 비정형 데이터와 이 데이터를 처리하는 기술을 모두 포괄하는 하나의 개념으로 등장하였다. 가트너는 빅데이터를 큰 용량(Volume), 빠른 속도(Velocity), 데이터 소스와 형태의 다양성(Variety)을 갖는 정보자산으로, 이를 통해 의사결정 등의 프로세스 최적화를 이루기 위해 새로운 형태의 처리방식이 필요하다고 정의하였다[3].

위의 정의와 같이 데이터의 양과 속도, 다양성의 변화에 따라 데이터의 활용 방법에도 변화가 필요하다. 대량의 정형 및 비정형 데이터를 수집 및 저장하고, 추론을 통해 지식정보로 변환할 수 있는, 기존의 데이터베이스를 대체하는 새로운 시스템이 필요하게 된 것이다. 이 대안으로 등장한 것이 바로 지식베이스(Knowledge Base)이다.

지식베이스란 특정 분야의 전문 지식이나, 문제 해결에 필요한 사실과 규칙을 저장하는 기술로, 전문가 시스템의 구성요소로서 처음 사용되었다. 여기서 지식이란, 정형화된 관계형 데이터베이스 등의 데이터가 아닌 컴퓨터가 이해할 수 있는 세상의 모든 사실들을 의미한다. 지식베이스 시스템(Knowledge-based System)은 이 지식베이스와 추론엔진으로 구성되어, 지식의 추론을 통해 문제를 해결하는 컴퓨터 시스템을 말한다.

초기 지식베이스 시스템은 DB-Pedia, NELL, YAGO 등 학술연구 목적의 지식베이스 활용이 대부분이었다. 하지만 빅데이터 기술이 발전함에 따라 지식베이스 시스템과 같은 인공지능 시장 또한 폭발적으로 성장하고 있다[4]. 여러 발표 자료에 따르면, 빅데이터 시장은 연 평균 40% 이상 성장하면서 2015년에는 170억 달러 규모에 이를 것으로 예상되었으며, 인공지능 시장도 빅데이터 시장의 성장에 따라 2013년 8억 달러 규모에서 이후 폭발적으로 성장해 2015년에는 330억 달러 규모에 이를 것으로 예상되었다[5, 6].

기업들도 새로운 수익원으로서의 지식베이스 시스템 활용에 집중하고 있다. IBM은 지식베이스 시스템과 질의응답 기술을 결합한 Watson 컴퓨터를 의료분야 등에 이용하기 시작했고[7], Google은 지식베이스 시스템을 검색 엔진에 응용한 Knowledge Graph 서비스를 선보였으며[8], 이 외에도 애플의 Siri, 페이스북의 Graph Search 등 지식베이스 시스템을 기반으로 한 서비스/플랫폼의 글로벌 경쟁력은 점점 더 치열해지고 있다. 우리나라도 지식베이스 및 추론 원천 기술을 확보하고, 경쟁력을 확보하기 위해 빅데이터 이해 기반 자가학습형 지식베이스 시스템인 WiseKB 개발 연구를 진행하고 있다.

WiseKB는 세 가지 차별성을 목표로 삼고 개발을 진행한

다. 첫 번째로 지식정보의 표현에 있어 First Order Logic과 같은 논리언어, 인간의 자연어, 통계 지식 등을 통합하고, 기계의 추론뿐만 아니라 인간의 큐레이션을 함께 진행하는 '다형지식', 두 번째로 인간에 의한 일괄적인 지식정보의 획득에서 벗어나 기계학습을 통한 지식 획득의 자동화와 획득된 지식을 자동으로 검증하고 교정하는 '자기진화', 세 번째로 시맨틱 추론뿐만 아니라 시공간, 확률 추론을 통합적으로 수행하며, 추론/분석 기술의 경제성을 위해 인메모리 분산 추론과 GPGPU 등을 활용하는 '복합추론'이 그것이다. 이러한 세 가지 차별성을 확보하기 위해서는 다양한 정형/비정형 데이터의 수집 시스템, 수집된 지식의 학습, 추론 및 확장, 검증된 지식의 저장과 같은 기능들이 필수적이며, 또한 지식의 큐레이션과 학습된 지식의 피드백, 서로 다른 기반기술로 개발되는 각 기능의 협업과 분산 자원의 효율적인 이용 등 지식베이스 시스템의 통합적인 관리를 위한 통합 플랫폼 또한 WiseKB의 핵심적인 요소라 할 수 있다.

위에서 서술한 것과 같이 통합 플랫폼은 WiseKB의 통합적인 관리를 담당하는 핵심적인 요소이기 때문에, WiseKB 개발 연구의 성공을 위해서는 바로 이 통합 플랫폼의 성공적인 개발 여부가 중요하며, 이를 위해서는 가장 먼저 정확한 요구사항 분석이 선행되어야 한다.

정확한 요구사항 분석에는 효율적인 요구사항 분석 방법론의 선택과 적용이 중요한 역할을 차지한다. 소프트웨어 개발 방법론에서 요구사항 분석은 크게 두 가지 방법으로 이루어진다. 첫 번째로는 각 단계를 설정하고 이전 단계를 모두 완료한 후 다음 단계를 진행하는 순차적 방법, 두 번째로는 요구사항 분석과 개발 단계를 순환적으로 수행하며 점진적으로 요구사항을 수정/확장해 나가는 순환-점진적 방법이 그것이다. 각 방법은 각자의 장점과 단점, 적용했을 때 더 효율적인 요구사항 분석이 가능한 시스템이 다르기 때문에 개발하고자 하는 시스템의 특성을 고려한 후 두 가지 방법 중에 더 유리한 요구사항 분석 방법을 선택해야 한다.

하지만 통합 플랫폼의 개발 특성을 고려했을 때 두 요구사항 분석 방법 모두 효율적인 요구사항 분석이 어렵다는 문제가 발생한다. 요구사항 분석 방법 선택을 어렵게 하는 통합 플랫폼의 특성은 크게 두 가지이다. 첫 번째는 통합 플랫폼의 개발이 통합의 대상인 각 기능의 개발과 동시에 이루어진다는 것이다. 플랫폼의 요구사항을 정확히 파악하기 위해서는 각 어플리케이션의 입력/출력 자료형, 사용하는 기반기술 등이 모두 확정되고 이를 파악할 수 있어야 하지만 플랫폼과 각 어플리케이션의 개발이 동시에 진행된다면 플랫폼 요구사항 분석에 필요한 사항들이 전부 정의되어 있지 않은 상태에서 요구사항 분석을 진행해야 한다. 따라서 순차적 방법론을 적용할 경우 각 정보가 확정된 이후에야 요구사항 도출과 분석을 진행할 수 있다는 문제가 발생한다. 두 번째는 각 어플리케이션이 수많은 개발주체에 의해 개발된다는 것이다. WiseKB 개발 프로젝트는 크게는 8개의 Work Package, 세부적으로는 약 10개 이상의 개발주체가 참여하는 대규모 프로젝트이다. 이 경우 각 개발주체에 각각 피드백

을 받기 위해서는 많은 시간이 소요되며, 초기 개발 이후 피드백과 요구사항 수정이 빠른 주기로 이루어져야 효율적인 분석이 가능한 순환-점진적 요구사항 분석 방법론을 선택할 수 없게 된다.

이러한 문제를 해결하기 위해서는 위의 두 요구사항 분석 방법론이 아닌 새로운 요구사항 분석 방법론이 필요하다. 본 논문에서는 WiseKB 통합 플랫폼과 같은 개발 특성을 갖는 시스템의 효율적인 요구사항 분석을 위한 요구사항 분석 방법론을 제안하고, 이를 실제 통합 플랫폼의 요구사항 분석에 적용함으로써 그 효과를 검증하고자 한다.

2. 관련 연구

본 논문에서 제안하는 내용은 기존 요구사항 분석 방법의 단점을 보완할 수 있는 복합 요구사항 분석방법이다. 이 장에서는 서론에서 언급한 순차적 방법과 순환-점진적 방법의 특징과 절차, 장/단점을 각 방법을 대표하는 Waterfall Model과 Agile Process 방법론을 통해 파악하고, 복합 요구사항 분석 방법을 위해 참고한 기타 요구사항 분석 방법들에 대해 설명한다.

2.1 Waterfall Model에서의 요구사항 분석

Waterfall Model은 소프트웨어 개발 방법론 중 하나로, 소프트웨어의 요구사항 분석, 설계, 구현, 테스트 및 유지보수가 순차적인 단계를 가지고 진행되며 앞의 단계가 모두 완료된 후에 다음 단계를 진행하는 방법을 말한다. 요구사항 분석에서도 이 Waterfall Model의 순차적 진행방법이 적용되는데, 이 경우 요구사항 분석을 도출 → 분석 → 명세화 → 검증의 네 단계로 나누고 각 단계를 순차적으로 진행하게 된다. 순차적 요구사항 분석 방법은 Waterfall Model에서 볼 수 있듯 계획에 따른(Plan-driven) 요구사항 분석이며, 요구사항 분석은 각 단계의 산출물인 문서를 통해 이루어진다[9].

개발규모가 큰 시스템의 요구사항 분석은 요구사항 수정에 필요한 비용이 많이 들기 때문에, 단계에 따라 순차적으로 요구사항 분석을 진행한 후 설계를 진행하는 것이 효율적이다. 따라서 이 경우 효율적인 요구사항 분석을 위해서는 순차적으로 요구사항 분석을 진행하는 것이 좋다.

반면 순차적 분석 방법을 따르게 되면 요구사항 분석을 위해서는 필요한 모든 요구사항이 도출되어야 한다. 따라서 요구사항을 초기에 전부 정의할 수 없는 경우에는 모든 요구사항이 도출 가능한 시점까지 분석과 검증 등 이후의 절차를 미루어야 하고, 이로 인해 시스템의 설계에도 영향을 미치게 된다. 또한 설계 이후 단계에서 요구사항의 수정사항이 발생할 경우 요구사항 도출부터 다시 모든 단계를 밟아나가야 하는 단점 또한 존재한다.

2.2 Agile Process에서의 요구사항 분석

Agile Process는 Waterfall Model과 반대되는 특징을 가

진다. 개발의 각 단계를 순차적으로 진행하는 Waterfall Model과 달리 Agile Process는 여러 단계가 짧은 주기를 가지고 반복된다. 주기의 순환을 거치면서 요구사항의 추가/수정과 함께 시스템의 개발 또한 완성되어 가는 것이다.

이처럼 Agile Process에서의 요구사항 분석은 설계 및 구현 과정과 분리되어 있지 않고, 맞물려 진행된다. 우선 이해관계자로부터 초기 요구사항을 수집하고 정의하는 과정은 Waterfall Model에서의 요구사항 분석과 같다. 하지만 모든 요구사항을 도출 및 분석한 후에 설계를 진행하는 Waterfall Model과는 달리, Agile Process에서는 초기에 수집된 요구사항들을 분석한 후 분석된 요구사항만을 가지고 시스템의 초기 설계 및 구현을 진행한다. 이렇게 구현된 시스템을 통해 초기 요구사항에서 부족한 점, 혹은 수정되어야 할 부분이 있는지를 피드백을 통해 파악하고, 이를 토대로 초기 요구사항에서 요구사항을 추가/수정한 후, 다시 구현과 피드백, 요구사항 분석이 짧은 주기로 반복되면서 최종적인 요구사항이 완성되어 나간다. Agile Process는 설계 및 구현이 중심이 되는 방법론이기 때문에 문서의 중요도는 상대적으로 낮다[10, 11].

Agile Process의 가장 큰 장점은 요구사항 변화에 유연한 대응이 가능하다는 것이다. 초기에 요구사항을 완벽하게 도출/정의할 수 없음을 가정하고 설계/구현을 먼저 진행하면서 요구사항을 점진적으로 구체화 해 가기 때문에, 초기 요구사항 분석 이후에도 끊임없이 요구사항을 보완할 수 있게 된다. 또한 설계 및 구현을 중심으로 개발을 진행하기 때문에 각 단계의 문서 등 산출물 제작에 대한 부담을 줄일 수 있다.

하지만 개발규모가 큰 시스템의 개발에서는 Agile Process의 적용이 불리하다. Agile Process는 요구사항 분석과 설계/구현의 빠른 반복을 통한 점진적 확장을 가장 큰 특징으로 갖는데, 개발규모가 큰 경우 초기 요구사항 수집 및 구현에도 상당한 시간이 소요되며, 구현된 시스템을 통한 피드백과 요구사항의 수정 또한 오랜 시간이 소요될 수밖에 없으며, 오히려 Waterfall Model에 비해 비효율적인 요구사항 분석이 이루어지게 된다.

2.3 기타 요구사항 분석 방법

본 단락에서는 요구사항 분석 방법을 제안함에 있어 참고한 Use Case Driven Requirement Engineering, Twin Peaks Model, Prototyping과 그 변형인 Paper Prototyping에 대해 설명한다.

Use Case Driven requirement는 요구사항 분석 과정에 유즈케이스를 이용하는 요구사항 분석 방법의 하나로, 초기에 요구사항을 직접 도출하는 것이 아닌 Use Case를 도출한 후 이를 통해 요구사항을 도출 및 정의하는 방법론이다. 여기서 유즈케이스 명세는 단순히 Actor와 유즈케이스를 표시할 뿐만 아니라, 유즈케이스를 테이블 형태로 재구성 하고, 유즈케이스 트리를 통해 유즈케이스와 유즈케이스 간의 연결을 명시하여 요구사항을 구체화한다는 특징을 갖는다[12].

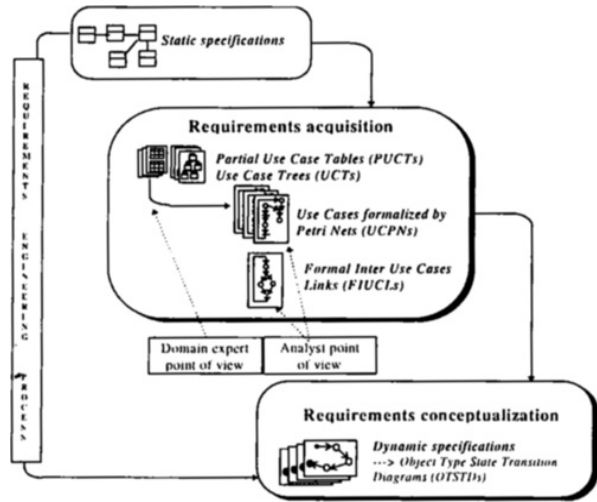


Fig. 1. Use Case Driven Requirement Engineering Process [12]

Twin Peaks의 경우 아키텍처 설계에서의 요구사항 분석 및 개발 방법으로 제안된 방법으로, 소프트웨어 개발 방법론 중 Spiral life-cycle model을 차용한 방법이다. 요구사항 분석이 완료된 이후 아키텍처의 설계를 진행하는 방식에서 벗어나 아키텍처 설계와 요구사항 분석이라는 개발의 ‘두 봉우리’를 동시에 오가면서 주기를 반복해나감에 아키텍처와 요구사항을 모두 구체화해나간다. Twin Peaks 방법론은 초기 아키텍처 설계 시 현재 요구사항을 만족하는 상용화된 제품들을 확인하고 구체화에 따라 그 범위를 지속적으로 줄여갈 수 있다는 장점을 갖는다[13].

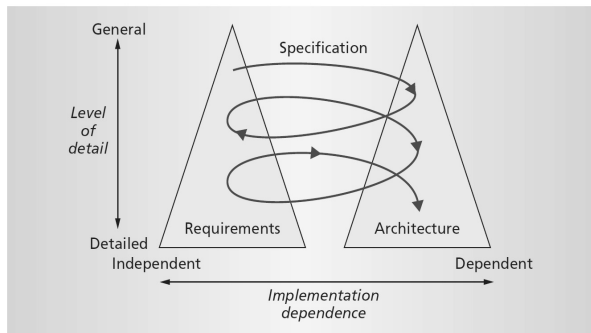


Fig. 2. The Twin Peaks Model [13]

Prototyping 방법론은 Agile Process의 한 예로, 요구사항을 토대로 한 프로토타입의 구현이 중심이 되는 방법론이다. 이 방법론은 먼저 핵심적인 사용자 요구사항을 도출한 후 이 요구사항을 만족하는 프로토타입을 구현하고, 이 프로토타입을 사용자에게 제공해 새로운 요구사항에 대한 피드백을 받아 요구사항을 수정하고, 이를 토대로 프로토타입 구현과 요구사항 수정을 반복하는 절차로 구성되어 있는데, 초기 요구사항을 토대로 완전한 시스템이 아닌 개략적인 프로토타입을 먼저 구현하고, 이를 테스트하고 요구사항을 수정하는 과정에 사용자가 적극적으로 개입한다는 특징을 갖는다[14].

Paper Prototype은 UI 요구사항 수집에 Prototype 방법론을 이용하는 변형된 요구사항 분석 방법으로 제안되었는데, 피드백을 위해 실제 개발환경을 통한 UI 구현을 하지 않고 종이나 그래픽 툴을 이용해 개략적인 Paper Prototype을 작성하는 특징을 갖는다.

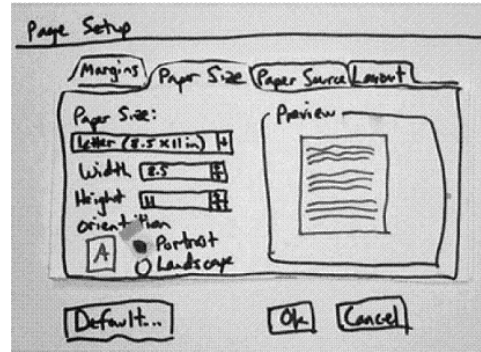


Fig. 3. Paper Prototype Example - Page Setup Dialogue [15]

절차는 크게 다섯 단계로 이루어지는데, 먼저 요구사항 분석가가 해당 도메인에 대한 연구를 진행해 기본적인 프로세스를 파악한 후, 요구사항 도출을 위해 시스템의 세부적인 사항을 이해한다. 이후 사용자와의 인터뷰를 통해 UI 구성에 대한 시스템의 주요 요구사항을 도출하면서 사용자와 함께 paper prototype을 작성한다. 이후 수집된 요구사항을 정리하고, 이 요구사항에 맞추어 프로토타입을 검증하는 과정을 거친 후 다시 피드백을 통해 요구사항의 구체화가 이루어진다.

이 방법론의 장점은 사용자가 UI와 관련된 시스템적인 부분을 고려하지 않고 자유롭게 생각하는 바를 이야기하고 수정할 수 있게 되어 요구사항 수집을 보다 친밀하게 할 수 있다는 것이다[15].

3. 연구 내용

앞서 서술한 것과 같이, 요구사항 분석은 크게 순차적 방법과 순환-점진적 방법의 두 가지 방법으로 이루어질 수 있으며, 효율적인 요구사항 분석을 위해서는 개발하는 시스템의 특성을 고려한 선택과 적용이 이루어져야 한다.

그러나 WiseKB 통합 플랫폼의 요구사항 분석은 두가지 방법으로는 요구사항분석에 한계가 존재한다. 통합 플랫폼을 설계하기 위해서는 크게 두 가지 사항을 고려해야 하는데, 첫 번째로는 플랫폼과 각 어플리케이션의 개발이 분산된 여러 개발주체를 통해 진행되기 때문에 요구사항 추가/수정의 반복에 따른 비용적인 부담을 최소화해야 한다는 것이며, 두 번째로는 플랫폼과 각 어플리케이션의 개발이 동시에 이루어지기 때문에 초기 수집 요구사항을 통해 요구사항 분석을 진행할 수 있으면서, 이후 추가/수정되는 요구사항에 유연하게 대응할 수 있어야 한다는 것이다.

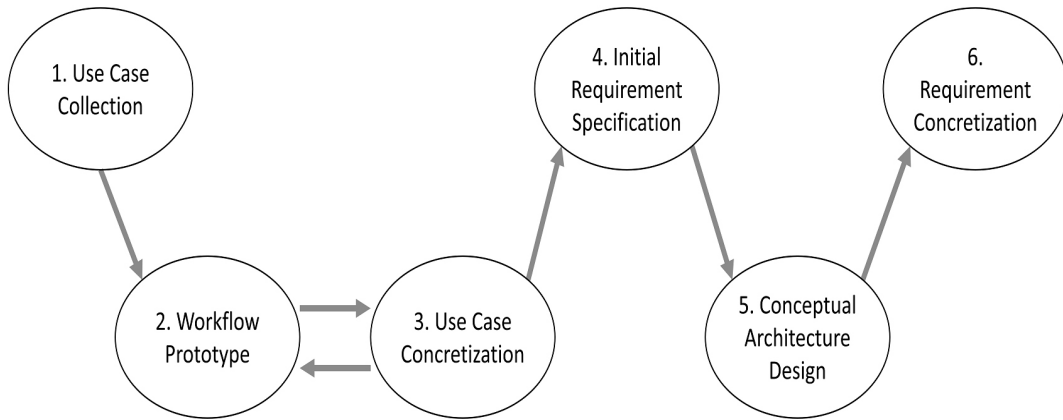


Fig. 4. Analysis Process for Complex Requirement Analysis

하지만 기존의 요구사항 분석 방법은 이 고려사항을 만족하지 못한다. 우선 순차적 방법론의 경우 수많은 개발주체를 통해 개발되는 대규모 시스템이라는 점에서는 적합한 분석 방법이지만, 완벽히 정의되지 않은 어플리케이션 요구사항으로는 플랫폼 요구사항을 제대로 도출할 수 없다는 문제점이 있다. 반면 순환-점진적 방법론은 요구사항의 점진적 확장과 변화에 따른 유연한 대응이 가능하다는 점에서는 본 시스템의 설계에 적합한 면이 있지만, 통합 플랫폼과 같은 대규모 시스템에서 빠른 피드백을 기대할 수 없다는 점에서 부적합하다.

효율적인 요구사항 분석을 위해서는 통합 플랫폼의 개발 특성을 만족하는 새로운 요구사항 분석 방법이 필요하다. 본 논문에서는 위에서 언급한 순차적 방법론과 순환-점진적 방법론의 절차적 결합을 통해 각각의 단점을 보완하여 효율적인 통합 플랫폼 요구사항 분석 및 설계가 가능하도록 하는 복합 분석 방법을 제안한다.

3.1 복합 분석 방법론의 정의

본 논문에서 제안하는 내용은 여러 요구사항 분석 방법의 특징을 결합해 효율적인 시스템 요구사항 분석이 가능하도록 하는 방법론으로, 이 방법론은 크게 네 가지 세부 방법론들을 결합하여 요구사항을 분석한다.

첫 번째로, 대규모 시스템에서 계획에 따른 정확한 요구사항 수집이 가능하도록 순차적 방법론의 틀을 차용하여, 일련의 단계를 거친 후에는 시스템 설계가 가능한 완전한 요구사항이 도출될 수 있도록 하며, 각 단계에서 요구사항 명세 등의 문서 작성을 거친다.

두 번째로, 초기 요구사항이 명확하지 않은 점을 고려해 요구사항 분석 방법 중 Use Case Driven Approach를 차용하여, 플랫폼과 각 어플리케이션의 유즈케이스를 먼저 도출/구체화 한 후 이를 통해 요구사항을 도출한다.

세 번째로, 유즈케이스의 구체화에는 순환-점진적 방법론을 차용하여 반복적이고 빠른 피드백을 통해 유즈케이스를 구체화한다. 이 때 Paper Prototyping 방법론을 차용하

여 유즈케이스를 도식화된 워크플로우 형태로 변형시켜 구체화한다.

네 번째로, Twin Peaks 방법론을 차용해 초기 요구사항 정의를 통한 아키텍처 설계를 통해 다시 세부 요구사항을 도출해내는 과정을 거친다.

본 논문에서 제안하는 방법론은 위에서 언급한 바와 같이 4가지 종류의 방법론을 결합하여, 전통적인 방법론의 장점과 반복적이고 점진적인 방법론의 장점을 동시에 취하고자 한다. 이러한 방법론은 WiseKB와 같이 대규모인 동시에 다양한 기능이 함께 개발되는 시스템에 대해 순차적으로 요구사항을 분석하고 점진적인 확장을 가능케 함으로써, 요구사항 분석 절차의 효율을 높여줄 것으로 기대된다.

3.2 요구사항 분석 절차

요구사항 분석 절차는 총 여섯 단계로 이루어진다.

1. 초기 유즈케이스 수집 : 먼저 각 어플리케이션 개발주체로부터 각 어플리케이션의 동작을 정의하는 초기 유즈케이스 명세를 수집한다. 이 과정을 통해 유즈케이스 명세서가 완성되는데, 이 유즈케이스 명세서에는 어플리케이션의 유즈케이스 정의에 필요한 시나리오와 Actor, 그리고 수집 가능한 유즈케이스의 세부 설명이 포함된다.
2. 워크플로우 프로토타입 작성 : 각 어플리케이션을 통해 초기 유즈케이스가 수집되면, 플랫폼 요구사항 분석가는 수집된 초기 유즈케이스를 토대로 모든 어플리케이션의 유즈케이스를 개괄적으로 연결해 워크플로우 프로토타입을 작성한다. 여기서 워크플로우는 각 Actor간의 연결과 전체적 흐름을 표현한 하나의 흐름도를 말한다.
3. 유즈케이스 구체화 : 워크플로우 프로토타입이 작성되면, 이를 각 어플리케이션 개발주체에 배포하고 각 어플리케이션 별 피드백을 통해 워크플로우를 구체화한다. 이렇게 구체화된 워크플로우는 각 개발주체와의 세부 질의응답 등을 통해 점진적으로 구체화되며, 워크플로우의 구체화와 함께 유즈케이스 또한 추가/수정을 반복하며 점차 구체화된다.

4. 초기 요구사항 도출 및 명세화 : 여러 번의 반복을 통해 유즈케이스 구체화가 완료되면, 완성 된 유즈케이스 명세를 토대로 플랫폼의 개괄적인 초기 기능 및 비기능 요구사항을 도출하고 이를 문서로 명세화한다. 작성되는 요구사항 명세서에는 플랫폼의 요구사항 내용과 기능/비기능 구분, 요구사항의 우선순위를 상/중/하 등으로 나누어 명시한다.
5. 개념 아키텍처 설계 : 이 단계에서는 이전 단계에서 도출된 초기 요구사항에서 플랫폼에 포함되어야 하는 개별적 구성요소를 도출한 후, 이를 토대로 한 개념적 아키텍처 설계 안을 작성한다. 아키텍처 설계도에는 초기 요구사항을 통해 도출한 구성요소가 개별적으로 배치되거나, 비슷한 항목의 경우 묶여 그룹 별로 표현된다. 아키텍처 설계도가 완성되면, 각 구성요소들의 사용 가능 기술 후보를 미리 파악하고 추가 요구사항 도출 과정에 반영할 수 있게 된다.
6. 세부 요구사항 도출 : 이 단계에서는 이전에 작성된 유즈케이스 명세와 초기 요구사항 명세, 개념 아키텍처 설계도 등을 통해 세부 요구사항 양식을 확정하고 각 어플리케이션 개발 주체에 이를 배포해 세부 요구사항을 수집한다. 수집된 요구사항은 세부 요구사항 명세서로 정리되는데, 이 문서에는 각 프로세스 혹은 컴포넌트 별 입/출력 흐름과 자료형, 전달방식 등의 입/출력 요구사항, 실행의 독립/의존성 및 외부 연결 등 실행과 관련된 요구사항, 또한 각 프로세스 실행에 필요한 프로그램 등의 실행 환경이 명시된다.

4. 요구사항 분석 방법론 적용

본 논문에서 제안한 순환-점진적 통합 요구사항 분석 방법의 특징을 설명하고, 분석 방법론의 각 절차를 세부적으로 제안하였다. 본 장에서는 실제 WiseKB 통합 플랫폼의 요구사항 분석 과정에 위에서 설명한 요구사항 분석 방법을 적용하여, 각 단계별 진행 과정, 결과와 산출물을 설명하고, 또한 이에 따른 효과를 검증한다.

4.1 적용 결과

WiseKB 통합 플랫폼 요구사항 분석에 통합 요구사항 분석 방법을 적용했을 때 각 단계별 적용 과정과 결과는 다음과 같다.

- 초기 유즈케이스 수집 : 초기 유즈케이스 수집을 위해 WiseKB의 어플리케이션 개발을 담당하는 각 Work Package(WP) 별로 유즈케이스 명세 양식을 작성/배포 후 시나리오와 Actor, 시나리오에 기초한 세부 유즈케이스 명세를 수집하는 과정을 거쳤다. 이 과정에서 25개의 초기 유즈케이스를 명세화 할 수 있었다.
- 워크플로우 프로토타입 작성 : 이 단계에서는 워크플로우 작성을 위해 수집한 유즈케이스를 하나의 흐름으로 잇고, 각 WP별로 이메일/전화 등을 통한 질의응답을 거쳐 각 WP별 3~6개의 박스들의 흐름으로 이루어진 초기 워크플로우를 작성하였다.

Table 1. Initial Specifications of Use Cases

유즈케이스 ID	WP7_UC_001		
유즈케이스 명	지식 저장		
유즈케이스	개요	플랫폼은 지식 저장 요청을 처리할 수 있어야 하며, 그 지식은 저장소에 저장되어야 한다.	
	세부내용 / 흐름	플랫폼은 지식 저장 요청을 처리할 수 있어야 하며, 그 지식은 저장소에 저장되어야 한다. 지식 저장 요청은 저장하고자 하는 지식(ID, 지식 내용)과 대상 저장소에 대한 정보를 담고 있다. 플랫폼은 대상 저장소에 대하여 지식 저장 요청을 처리한 후 처리 결과를 반환한다.	
관련 시나리오	WP7_SC_001	관련 WP	3, 4, 7

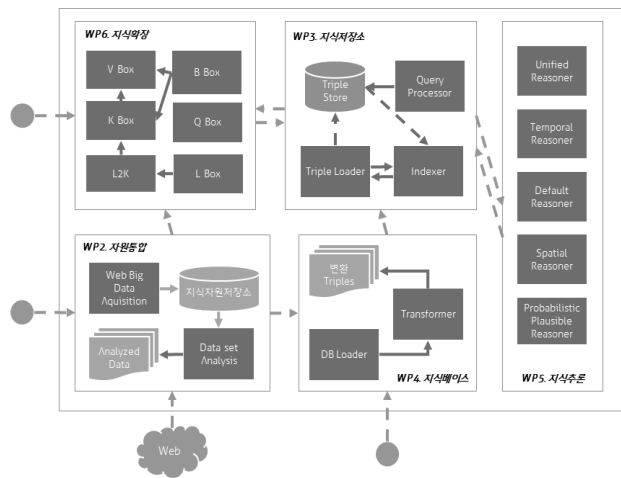


Fig. 5. Initial Workflow

- 유즈케이스 구체화 : 이 단계에서는 우선 각 WP 실무자 회의를 통해 작성한 워크플로우 프로토타입을 공개하고, WiseKB의 전체 흐름을 공유함과 동시에 각 WP 별로 피드백을 통해 워크플로우를 구체화하는 과정을 거쳤다. 각 WP별로 2~3회의 피드백을 수행하였으며, 세부 이해를 위한 질의응답을 통해 각 WP당 박스의 수를 적게는 2배에서 많게는 5배 이상 구체화할 수 있었다. 이렇게 구체화 된 워크플로우는 초기 유즈케이스의 추가/삭제 및 수정에 반영되어 최종적으로 유즈케이스의 수를 50개로 확장하고, 분량 또한 3배로 확장할 수 있었다.
- 초기 요구사항 도출 및 명세화 : 이 단계에서는 앞서 구체화 한 유즈케이스 명세를 종합한 후, 통합 플랫폼의 개발 목적 등을 고려해 20개의 개괄적인 초기 요구사항(기능 요구사항 15개, 비기능 요구사항 5개)을 도출하였다. 도출된 요구사항은 개략적인 설명과 함께 요구사항의 중요도를 파악하는 과정을 거쳐 표로 명세화되었다.

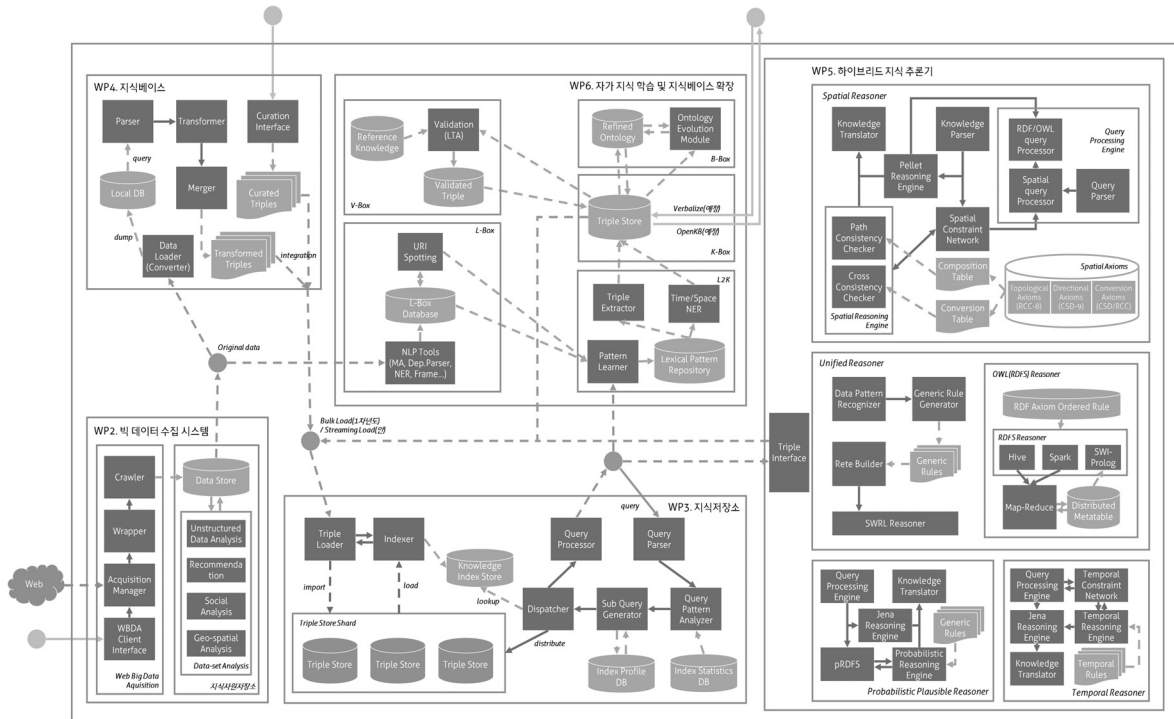


Fig. 6. Elaboration of the Workflow

Table 2. Final Specification of Use Cases

Use CaseID	F-01
Use Case 명	작업 스케줄 등록
개요	워크플로우 및 병렬 태스크 등의 작업을 주기적으로 실행할 수 있도록 스케줄 등록한다.
Use Case 세부 내용	사용자 - 작업을 작성하여 스케줄을 관리하는 사람 스케줄러 매니저 UI - 사용자가 스케줄러를 업로드 및 관리 할 수 있도록 지원하는 UI 컴포넌트 스케줄러 - 다수의 스케줄 작업을 관리하는 컴포넌트
	1. 사용자는 주기적으로 수행해야 하는 작업을 작성한다.
	2. 사용자는 WiseKB 통합 UI를 통해서 스케줄러 매니저 UI 화면을 띄운다.
	3. 사용자는 자신이 작성한 작업을 UI를 통해 업로드 한다.
	4. UI는 사용자가 업로드 한 스케줄 아이템을 스케줄러에 전달한다.
	5. 스케줄러는 UI로부터 전달받은 스케줄을 자신의 스케줄 리스트에 추가하고 스케줄의 변화를 UI에 알린다.
	6. 스케줄러 매니저 UI는 전달받은 이벤트를 통해 사용자에게 정보가 갱신된 UI를 표시한다. 7. 사용자는 자신이 업로드 한 스케줄이 스케줄러에 등록된 것을 UI를 통해 확인한다.

Table 3. Initial Specification of Requirements

요구사항 ID	요구사항명	요구사항	구분	우선 순위	관련 WP
FR-1	작업 스케줄링	사용자가 등록한 시간에 지정한 작업을 실행하는 스케줄링 기능을 제공해야 한다.	기능	2	
FR-2	리소스 모니터링	사용자는 공용 리소스(파일 시스템, 분산 처리 클러스터, 시스템 정보 등등)의 사용 현황을 모니터링 할 수 있어야 한다.	기능	2	
FR-3	병렬 클러스터 관리	사용자가 공용으로 사용할 가능한 병렬 분산 처리 클러스터의 각종 환경설정 정보를 변경하고 적용할 수 있어야 한다.	기능	3	
FR-4	분산파일 시스템 관리	사용자는 분산 파일 시스템에 저장되어 있는 파일 및 디렉토리를 관리할 수 있어야 한다.	기능	2	
FR-5	분산 로깅	다양한 응용 프로그램에서 발생하는 로그를 기록하고 수집할 수 있어야 한다.	기능	3	

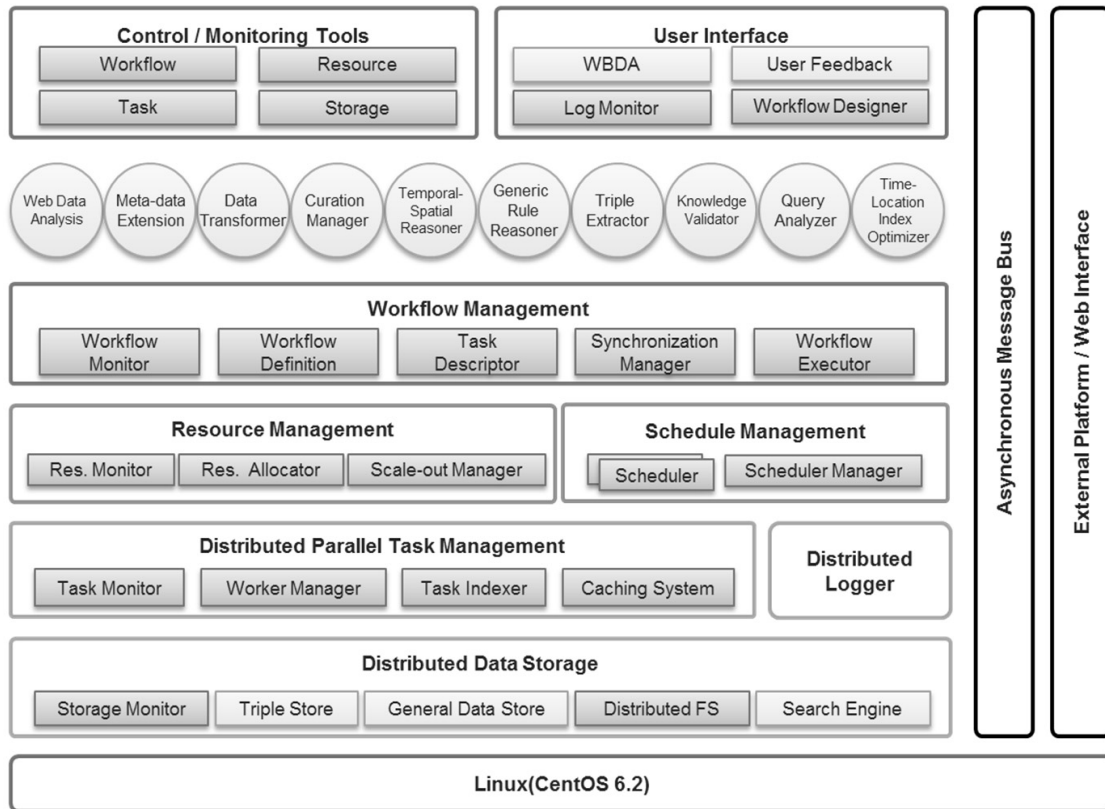


Fig. 7. WiseKB Integrated Platform - Draft Design of the Conceptual Architecture

- 개념 아키텍처 설계 : 이 단계에서는 각 기능 및 비기능 요구사항을 통해 통합 플랫폼에 필요한 항목을 도출하고, 역으로 요구사항의 반영을 검토한 후 각 항목과 요구사항을 연결하는 과정을 거쳤다. 이 과정을 통

해 크기는 10개, 세부적으로는 33개의 항목으로 구성된 하나의 설계도로 작성하였다. 또한 각 기능을 지원하는 실제 기술들의 후보군을 함께 도출해 이후 세부 요구사항 분석 및 설계 과정에 반영할 수 있도록 했다.

- 세부 요구사항 수집 : 실제 시스템 설계를 위해 워크플로우의 각 박스(프로세스)를 단위로 한 세부 요구사항 양식을 작성한 후 각 WP에 배포하고 수집하는 과정을 거쳤다. 수집이 완료된 후에는 총 80항목 이상의 프로세스 별 세부 요구사항 명세를 도출 후 정의할 수 있을 것으로 예상된다.

Table 4. Detailed Specification of Requirements

ID	WP6 - 19		
이름	Constraint Enforcement		
설명	구조화된 외부 지식을 활용하여 지식들간의 다양한 pattern들을 통해 한글 triplet에 적합한 pattern을 KB의 triplet에 적용하여 일치하지 않는 triplet을 대상으로 triplet의 Trustfulness 자동화 검증함		
I/O	입력	자료형	입력 프로세스
		Triplet	Knowledge filter module Enforcement module
	Extra Knowledge		
	출력	자료형	출력 프로세스
Confidence score		V-line	
전달방식	Retrieve&Store		
실행	독립성	독립 컴포넌트	
	의존성	WP6-18 와 knowledge filter module 공유	
	외부연결	WP6-18 과 knowledge filter module 공유, DBpedia 검색 결과 검증에 반영	
플랫폼 지원사항	미정		

4.2 효과

제안한 요구사항 분석 방법을 WiseKB 통합 플랫폼의 요구사항 분석 과정에 적용해 요구사항을 분석하는 과정에서 다음의 효과를 확인할 수 있었다.

첫 번째로, 앞서 언급한 플랫폼 요구사항 분석 과정에서의 고려사항을 만족할 수 있었다. 순차적 방법과 순환-점진적 방법을 결합함으로써 수많은 개발주체에 의해 개발되는 대규모 시스템에서 순차적 방법론의 장점을 취함과 동시에 플랫폼-어플리케이션 동시 개발에 의해 발생하는 순차적 방법 적용의 어려움은 순환-점진적 방법의 장점을 이용해 극복할 수 있게 되었다.

두 번째로, 요구사항 도출에 유즈케이스 도출 과정을 도

입해서 각 어플리케이션의 개발주체에게 별도의 플랫폼 요구사항 도출 및 정의에 대한 부담을 주지 않고, 개발 주체가 잘 이해하고 있는 유즈케이스만을 쉽게 수집할 수 있었으며, 정확한 요구사항 수집은 각 어플리케이션의 개발이 일정 부분 완료 된 이후에 보다 편하게 진행할 수 있었다.

세 번째로, 유즈케이스의 수정에 도식화 된 워크플로우를 사용하여 각 개발주체가 보다 직관적으로 유즈케이스를 이해하고, 쉽게 피드백을 할 수 있게 되었다. 또한 워크플로우의 공유를 통해 WiseKB 내에서 협업이 필요한 각 어플리케이션의 개발 주체 간 이해도를 높일 수 있었다.

네 번째로, 요구사항 분석 이후 개념 아키텍처 설계를 진행함으로써 요구사항이 실제로 아키텍처에 반영될 수 있는 것인지를 검증하고 세부 요구사항 수집의 방향을 설정할 수 있었다. 또한 적용이 가능한 기술들을 미리 파악함으로써 각 기술을 비교분석할 수 있었으며, 분석된 자료는 이후 실제 플랫폼을 설계에 필요한 소요를 줄일 수 있을 것으로 기대된다.

5. 결론 및 향후 연구 방향

본 논문에서는 대규모 시스템의 요구사항 분석에 기존의 방법론을 적용했을 때의 어려움을 해결하기 위해 순차적 방법론과 순환-점진적 방법론을 결합한 복합 요구사항 분석 방법론을 제안하였고, 이를 WiseKB 통합 플랫폼 요구사항 분석 과정에 실제로 적용하여 그 결과를 확인하였다.

적용 결과 두 방법론의 결합을 통해 기존 요구사항 분석 방법론 적용의 난점을 해소함과 동시에 기타 요구사항 분석 방법론의 특징들을 도입하여 각 어플리케이션 개발주체의 요구사항 도출 부담 감소와 각 어플리케이션 간 협업 유도 등 여러 긍정적인 효과 또한 얻을 수 있었다. 또한 WiseKB의 통합 플랫폼뿐만 아니라, 이와 유사한 개발 특성을 갖는 다른 시스템의 요구사항 분석 과정에서도 제안한 요구사항 분석 방법이 이와 동일한 효과를 얻을 수 있을 것이라고 본다.

하지만 이러한 긍정적인 효과에도 불구하고 현재의 연구 결과에도 한계점이 존재한다. 가장 큰 한계점은 본 논문에서 제안한 분석 방법론의 적용을 통해 확인한 긍정적인 효과의 검증과 별개로 본 논문에서 제시된 요구사항 분석 방법론이 다른 요구사항 분석 방법을 적용했을 때와 비교해 어느 정도의 개선 효과를 보였는지에 대한 정량적 비교 평가가 어렵다는 것이다. 향후 연구는 시스템 개발에 있어 보다 효과적인 요구사항 분석 방법을 적용할 수 있도록 요구사항 분석 방법론을 정량적으로 평가할 수 있는 평가 지표를 고안해야 할 것이다. 또한 기존의 지식베이스 플랫폼 등 대규모 시스템에서의 요구사항 분석은 어떻게 이루어졌는지, 또한 어떤 부분에서 차이를 보이는지에 대한 비교분석 또한 평가지표를 가지고 보다 면밀히 이루어진다면 현재의 연구결과보다 발전된 요구사항 분석 모델을 도출할 수 있을 것이다.

References

- [1] G. Press, A Very Short History Of Big Data, Forbes, 2013.
- [2] W. Koff et al., DATA rEVOLUTION, Computer Science Corporation, 2011.
- [3] D. Laney, 3D Data Management: Controlling Data Volume, Velocity, and Variety, META Group, 2001.
- [4] F. M. Suchanek et al., "Knowledge Bases in the Age of Big Data Analytics," *Proceedings of the VLDB Endowment*, Vol. 7, 2014.
- [5] D. Vesset et al., Worldwide Big Data Technology and Services 2012-2015 Forecast, IDC, 2012.
- [6] K. Dervojeda et al., Business Innovation Observatory: Case study - Big Data - Artificial Intelligence, European Union, 2013.
- [7] IBM Systems and Technology Group, Watson: A System Designed for Answers: The future of workload optimized systems design. IBM White Paper, IBM Corporation, 2011.
- [8] A. Singhal, "Introducing the Knowledge Graph: things, not strings," Google Blog, 2012.
- [9] I. Sommerville, "Software Engineering Ninth Edition," Pearson, pp.30-32, 2011.
- [10] I. Sommerville, "Software Engineering Ninth Edition," Pearson, pp.57-62, 2011.
- [11] C. Lan et al., "Agile Requirements Engineering Practices: An Empirical Study," *IEEE, Software*, Vol.25, pp.60-67, 2008.
- [12] B. Dano et al., "A use case driven requirements engineering process," *Requirements Engineering*, Vol.2, pp.79-91, 1997.
- [13] B. Nuseibeh, "Weaving together requirements and architectures," *Computer*, Vol.34, pp.115-119, 2001.
- [14] Hickey, Ann. et al., Prototyping for requirements elicitation and validation: A participative prototype evaluation methodology, 1998.
- [15] J. Vijayan et al., "A New approach to Requirements Elicitation Using Paper Prototype," *International Journal of Advanced Science and Technology*, Vol.28, pp.9-16, 2011.



박 지 성

e-mail : podracer87@gmail.com

2013년 서울시립대학교

전자전기컴퓨터공학부(학사)

2015년 서울시립대학교

전자전기컴퓨터공학과(석사)

2015년~현 재 삼성SDS 연구원

관심분야 : 인공지능, 지능형 시스템



이 재 호

e-mail : jaeho@uos.ac.kr

1985년 서울대학교 계산통계학과(학사)

1987년 서울대학교 계산통계학과(석사)

1997년 University of Michigan(박사)

1998년~현재 서울시립대학교

전자전기컴퓨터공학부 교수

관심분야: 인공지능, 지능 로봇