

# 거시구조 개념을 이용한 요구사항 분석과 Use-case 도출 방법

조 용 균<sup>†</sup> · 박 용 익<sup>\*\*</sup> · 박 용 범<sup>\*\*\*</sup>

## 요 약

소프트웨어의 규모가 커지고 다양해짐에 따라 고객이 원하는 요구사항을 정확하게 이해하여 표현하는 것의 중요성이 강조되고 있다. 특히 개발 초기에 요구사항은 추상적이고 개념적이기 때문에 핵심적인 시스템의 요구사항을 도출해서 명세하는 작업은 쉬운 일이 아니다. 기술적인 측면에서 이론적으로 다양한 분석과 검증방법이 소개되고 있지만 실무적으로는 요구사항을 명확하게 도출하기 힘들다. 본 논문에서는 고객이 원하는 요구사항을 명확하게 도출하기 위해, 텍스트언어학(textlinguistics)의 거시구조 개념을 이용한다. 먼저, 서술적으로 명세 된 요구사항을 텍스트언어학(textlinguistics)의 거시구조로 분석하였고 다음으로 거시규칙을 적용하여 거시명제를 도출하여 요구사항을 명확히 정의하도록 하였다. 마지막으로 도출된 거시명제를 이용하여 Use-case Model로 문서화하는 방법을 제시하도록 한다.

**키워드 :** 요구사항 분석, 요구사항 정의, 요구사항 명세, 거시구조, 유즈케이스 도출, 텍스트 언어학, 거시명제, 거시규칙

## Requirements analysis and Use-case realization using Macrostruktur

Yong Kyun Cho<sup>†</sup> · Yong-Ik Bak<sup>\*\*</sup> · Young Bom Park<sup>\*\*\*</sup>

### ABSTRACT

As software is varied and grew in size, the importance of clear understanding and representing customer's need, has been emphasized. It is not easy to drive and define critical system needs, since requirements in early stage of the development, is abstract and conceptual. Technical perspective view, Several theoretical analysis methods and validation techniques are introduced, but in practice, it is hard to drive clear customer's requirement. In this paper, to drive clear customer's requirements, Macrostruktur concept of textlinguistics is used. First, Macrostruktur of textlinguistics is used to analysis descriptive requirements. Second, Macrostruktur-rule is applied to analyzed Macrostruktur data in order to derive Macrostruktur-propositions. Finally, it is proposed to generating Use-case model using derived Macrostruktur-propositions.

**Keywords :** Requirement Analysis, Requirement Specification, Requirement Definition, Macrostruktur, Use-case Realization, Textlinguistics, Macrostruktur Proposition, Macrostruktur Rules

### 1. 서 론

정보통신 기술의 발전에 따른 소프트웨어의 규모가 커지고 다양해짐에 따라 많은 기업들이 소프트웨어의 품질향상을 위하여 고객이 원하는 요구를 정확하게 이해하고, 관리하는 방법의 중요성이 강조되고 있다[1]. 미국 Standish Group 소프트웨어사업 실패원인 보고서에 따르면, 소프트웨어 프로젝트

의 성공률은 매우 낮은 수준으로, 실패 원인은 49.3%가 사용자와의 비효율적인 의사소통으로 인해서 발생하는 요구사항에 대한 불명확한 정의와, 이로 인한 잦은 요구사항의 변경 때문인 것으로 나타나고 있다.[2] 즉 소프트웨어 프로젝트의 실패가 적용기술이나 방법론의 부족에 기인하기 보다는, 고객의 요구를 이해, 분석, 문서화하고, 변경을 관리 통제하는 기술과 방법의 문제인 것으로 나타났다[3].

특히 개발 초기에 요구사항은 추상적이고 개념적이기 때문에 핵심이 되는 시스템의 요구사항을 명확하게 도출하여 명세하는 작업은 어렵다. 실무적으로 기술적인 측면에서 이론적으로 다양한 분석 및 검증방법이 소개되고 있지만 대부분의 요구사항은 서술적인 명세에 그치기 때문에 핵심이 되는 요구사항을 명확하게 도출하는 기술에는 용이하지 않다[4].

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원 사업의 연구결과로 수행되었음(NIPA-2011-(C1090-1131-0008).  
<sup>†</sup> 준 회 원: 단국대학교 전자계산학과(석사)  
<sup>\*\*</sup> 정 회 원: 서울대학교 인문학연구원 전임연구원  
<sup>\*\*\*</sup> 중신회원: 단국대학교 컴퓨터학과 교수  
 논문접수: 2010년 12월 10일  
 수정일: 1차 2011년 2월 28일, 2차 2011년 5월 2일, 3차 2011년 7월 7일  
 심사완료: 2011년 7월 8일

본 논문에서는 고객이 원하는 핵심적인 요구사항을 명확하게 도출하기 위해, 텍스트언어학(textlinguistics)의 거시구조(Macrostruktur)를 이용하여 요구사항을 분석하여, 의미명제를 도출하고, 도출된 의미명제들에 거시규칙을 적용하여 핵심적인 요구사항을 정의한다. 거시명제와 의미명제들을 Use-case Model로 문서화하는 방법을 제시하여 소프트웨어 개발에 참여하는 고객과 사용자 그리고 개발자들이 시스템의 기능과 행위에 대해서 좋은 의사소통 도구로 사용될 수 있도록 제안한다.

## 2. 관련 연구

### 2.1 텍스트언어학(textlinuistics)의 텍스트 분석

텍스트언어학(textlinuistics)은 약 40년 전부터 독일을 중심으로 발전한 언어학의 한 분야로 한국에서는 10여 년 전부터 많은 발전을 해오고 있다. 연구의 대상은 의사소통과 텍스트와의 관계를 살피고, 텍스트를 텍스트답게 하는 것이 무엇인지를 검토하는 것이다[5].

#### 2.1.1 텍스트의 연속과 의미를 나타내는 명제

본 논문에서 분석의 대상으로 다룰 자료는 텍스트로서 텍스트란 청각과 시각적인 매개에 의하여 표현된 전달내용이다. 의미의 명확성을 기할 필요가 있을 때, 텍스트라는 용어를 사용한다[7]. 많은 언어발화에서 여러 텍스트 사이에 연속을 기술하므로 일정한 연관성이 있다는 것을 확인할 수 있는데 이때, 텍스트의 연속을 기술하기 위해서는 여러 텍스트 자체도 연속적 구조를 가질 수 있는 복합문이다.

복합문과 텍스트 연속 사이에는 일련의 체계적 차이가 있기 때문에 텍스트의 연속에 대한 기술을 복합문에 대한 기술과 동일시할 수 없다. 이러한 차이는 복합문과 문 연속 안에 있는 문장 간의 관계는 의미론적 성격을 나타내고 확대해보면 문장의 일반적이고 개념적 의미뿐만 아니라 지시적 관계와도 관련된다.

텍스트를 의미론적 성격과 지시적 관계에서 의미적 대상 자체로 본다면 텍스트는 의미를 나타내거나 무엇을 지시하기 위해서 이용된다. 이런 텍스트의 의미를 대략적으로 말해서 명제(Proposition)라고 부른다[8].

텍스트의 명제들 간에는 의미적 관계로 구성되어 있다.

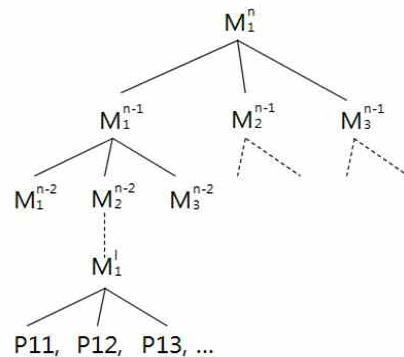
전체로서의 명제 사이의 연관성을 고찰하고 명제간의 연결 가능성에 대한 조건을 살펴보면 명제가 서로 연결되는 경우는 명제 해석 시에 명제에 대응하는 상태가 상호 연결되는 경우이다.

텍스트를 바라보는 관점은 다양한 관점들이 있지만 어떤 관점을 취하든 분명한 것은 텍스트가 형식 단위가 아니라 의미 단위란 것이고, 문장으로 이뤄지지만 문장의 총화 이상의 문법적 단위라는 것이다. 따라서 텍스트를 단일한 형식적 정의로 파악하기 보다는 다양하지만 통합적 요소를 만족 시키는 전략적 정의의 집합으로 보는 관점을 취한다[6].

#### 2.1.2 반 다이크(van Dijk)의 거시구조

텍스트는 명제와 명제 혹은 문장과 문장들이 결합해서 하위 텍스트를 이루고, 이 작은 텍스트들이 결합해서 보다 상위 텍스트를 형성하거나, 언어적 요소와 상황 등의 비언어적 요소들이 결합하여 전체 텍스트를 형성한다. 이와 같은 명제들 혹은 문장들의 연합인 텍스트 전체에 바탕하고 있거나 큰 단위에 기초를 두고 거시적인 관점으로 연관성에 주목하는 총괄적 구조를 거시구조(Macrostruktur)라고 한다 [5][6][7].

거시구조를 가지는 텍스트는 전체 텍스트를 요약한 텍스트 주제를 가지게 된다. 이런 구조는 수많은 명제로 이루어진 문 연속의 전체나 부분은 더 총괄적인 차원에서 하나의 의미단위를 형성하게 된다. 또한 거시구조는 여러 층위를 가질 수 있는데 상위의 거시구조를 가지는 텍스트는 하위의 요약한 텍스트 주제를 가질 수 있고, 결과적으로 여러 차원에서 거시구조는 (그림 1)과 같이 계층적인 구조를 가질 수 있다.



(그림 1) 계층적 구조를 갖는 거시구조

이런 층위 표현은 층위마다 텍스트 주제를 나타내는 거시명제를 도출할 수 있다. 계층적 구조를 나타내는 거시구조는 상대적인 성격을 가지고 있다. 명제들의 총괄적인 텍스트 주제를 가진 상위층의 거시명제는 같은 계층의 여러 거시명제들이 모여 그 상위의 거시명제를 나타낼 수 있다. 전체적 의미에 해당하는 최상위 거시명제의 파악은 아래로부터의 순차적인 거시명제의 파악에 의해 최종적으로 달성된다고 볼 수 있다[8].

#### 2.1.3 거시규칙의 성격과 원리

거시규칙의 기능은 한마디로 정보의 축소와 조직이며, 이러한 거시구조의 형성은 이해에 필수적인 요소가 된다. 텍스트의 생산과 해석에 있어서 거시 규칙을 적용하는 근본적인 목적은 텍스트의 주제를 명확히 파악하는 것이고, 이를 명시적으로 확인할 수 있는 중요한 단서 중의 하나가 요약이기 때문에, 요약은 이를 검증해 주는 한 척도가 된다. 거시규칙을 통해서 거시구조에 포함된 명제들이 바로 거시명제가 되고, 한 편의 글의 개요나 요약은 바로 이 거시명제들에 의해서 이루어진다[7].

반 다이크(van Dijk)는 거시명제의 파악은 바꿔 쓰기 식 환원규칙인 거시규칙에 근거해서 이루어진다고 말한다. 텍스트에서 상위 명제로의 정보 상속에서 거시 규칙이 중요한 이유는 중요한 정보와 덜 중요한 정보를 구분할 수 있는 하나의 기준이 되기도 하고, 논지의 일관성을 유지하여 텍스트 생산자와 수용자의 의사소통 상황을 원활히 해 주기도 하며, 단위 명제들을 구성하는 기능도 하기 때문이다. 거시규칙의 종류는 생략규칙, 선택규칙, 일반화규칙, 구성화 규칙이 있다[8][9].

### 2.2 Use-case Model

UML을 설계언어로 사용하는 방법론에서는 요구사항을 명세하는 도구로는 Use-case Model을 사용하고 있다. Use-case Model은 시스템을 사용하는 역할자인 Actor의 관점에서 시스템을 사용하는 사례들을 정리했다고 할 수 있다[10].

Use-case를 추출하고 분류할 때에 시스템이 개발될 상황을 고려하기 때문에 요구사항 분석이 사용자 입장보다는 개발자의 입장에 가깝게 분석된다. 따라서 요구사항의 초기 단계인 추출공정을 제공하지 않는다. 그리고 Use-case가 Actor와 시스템의 상호작용을 일련의 작업 순서로 표현하기 때문에 기능적인 요구사항을 표현하기에는 알맞으나, 비 기능적인 요구사항을 표현하기에는 부적합한 특징을 가진다[11].

## 3. 거시구조 개념을 이용한 요구사항 분석방법

본 논문에서는 텍스트언어학(textlinguistics)의 거시구조로 요구사항을 분석하여 핵심적인 요구사항을 도출하여 Use-case Model을 이용하여 문서화한다.

### 3.1 거시구조를 이용한 요구사항의 분석

텍스트언어학(textlinguistics)의 거시구조를 이용하여 요구사항 텍스트에 내포되어 있는 의미들 간의 의미적 관계를 분석하여 의미명제들을 도출한다. 도출된 의미명제들은 거시명제를 도출하기 위한 필요정보가 된다.

요구사항은 고객의 요구를 의미하는 하나의 텍스트로서, 문장과 문장들의 결합으로 이루어진다. 각 문장들이 내포하고 있는 다양한 의미들은 고객의 요구를 나타낸다. 본 논문에서는 고객의 요구를 의미하는 하나의 글 또는 텍스트를 요구사항 텍스트라고 정의하고 R로 표현한다. 요구사항 텍스트 R에 결합되어 있는 문장들 중 의미적 관계를 갖는 문장들의 집합을 요구사항 문장이라고 정의하며 S로 표현한다. 요구사항 문장 S가 내포하고 있는 다양한 의미들에서 하나의 개별적인 의미를 명제단위로 정의한 것을 의미명제라고 정의하며 P로 나타낸다. 의미명제 P가 나타내는 개별적인 의미는 요구사항이 가진 기능적 요소라는 특성을 도출하기 위한 것으로 행위나 기능을 나타내는 원자적인 성격을 갖는 하나의 의미로 단문으로 표현된다. 그리고 핵심적인 요구사항으로 정의되는 거시명제는 의미명제들에 거시규칙을 적용하여 도출될 수 있기 때문에 요구사항 텍스트를 의

미명제들로 구성되어야 한다.

요구사항 텍스트 R에서 의미명제 P까지 도출하는 과정을 총 3단계로 정리하면 <표 1>과 같다.

<표 1> 요구사항 의미명제 도출과정 3단계

1단계)	요구사항 텍스트 R에서 각 문장들에서 의미적 관계를 분석하여 같은 의미를 나타내는 문장들의 그룹으로 구성한다. 문장들의 그룹은 요구사항 문장 S가 되고, n개로 분할된 요구사항 문장 S는 의미적으로 요구사항 텍스트 R에 모두 포함된다.
2단계)	분할된 요구사항 문장 Si에서 각 문장들안에서 의미를 나타내거나 무엇을 지시하는 기능이나 행위들을 개별적인 의미들로 분할한다. 분할되는 개별적인 의미들은 대상, 대상의 행위나 기능, 행위나 기능에 대한 특성으로 구성하여 작성한다.
3단계)	분할한 개별적인 의미들 간의 의미적 관계를 분석하여 의미명제 P를 도출하여 대상, 대상의 행위나 기능, 행위나 기능에 대한 특성의 형식으로 정의한다. 도출한 의미명제들 P는 요구사항 문장 S의 구성요소로서의 의미적으로 모두 포함된다.

요구사항 텍스트에서 요구사항의 의미들을 명확히 분석하기 위해서는 문장 간의 의미적 관계를 분석하고, 문장이 내포하고 있는 개별적인 의미들의 의미적 관계를 분석하여, 명제단위로 분할해야 한다. 도출되는 의미들은 의미와 의미적 대상 자체를 지시하고 의미하고 있기 때문에 명제(Proposition)로 표현된다.

<표 1> 요구사항 의미명제 도출과정 3단계에서 의미명제 도출과정 각 단계마다 의미적 관계를 찾아 분석하기 위하여 일련의 조건이 필요하다. 의미적 관계를 찾아 분석할 수 있는 조건은 반 다이크(van Dijk)의 텍스트 학에서 제시된 문의 연속 조건[8]을 이용하여 <표 2>와 같이 의미적 관계 분석의 네 가지 조건을 제시한다.

반 다이크(van Dijk)의 텍스트 학에서 제시된 문의 연속 조건[8]은 단순히 A와 B사이의 의미적 관계에 대하여 연관성의 존재 여부만 분석하기 때문에 명확한 의미적 관계분석을 수행하기 위하여 반 다이크(van Dijk)의 텍스트 학에서 제시된 문의 연속 조건[8]을 이용하여 중복되고 상위개념을 포함하는 조건은 삭제하고, 명제의 선행의미와 후행의미를 명확하게 분석하기 위한 조건으로 재구성하여 <표 2>와 같이 의미적 관계 분석의 네 가지 조건을 제시한다.

<표 2> 명제 의미적 관계 분석의 네 가지 조건[8]

- |  |
|--|
| <ol style="list-style-type: none"> <li>1) A는 B의 원인이다. 또는 A는 B의 이유이다.</li> <li>2) A는 B의 결과이다. A는 일종의 행위이거나 행위의 결과이다.</li> <li>3) A와 B는 동일한 상황에서 일어난다. A, B는 동일개념 영역에 속한다. 이때 다음 몇 가지가 허용된다. <ul style="list-style-type: none"> <li>- A는 B와 동시적이다.</li> <li>- A는 B의 또는 B는 A의 부분기간에 일어난다.</li> <li>- A와 B는 서로 중복된다.</li> </ul> </li> <li>4) A는 필연적으로 B의 부분이 되거나 그 역이다. A가 일어나면 반드시 B가 일어나거나 그 역이다.</li> </ol> |
|--|

<표 2> 의미적 관계 분석의 네 가지 조건에서 A와 B는 의미적 관계를 분석하기 위해서 선택되는 요구사항 문장 S 또는 의미명제 P를 나타낸다. A는 선행되는 요구사항 문장 S 또는 의미명제 P를 나타내고, B는 후행되는 요구사항 문장 S 또는 의미명제 P를 나타낸다.

요구사항 의미명제 도출과정 3단계를 통해 의미명제를 도출하는 과정을 자세히 살펴보고자 한다.

요구사항 의미명제 도출과정 1단계는 요구사항 텍스트 R을 구성하고 있는 각 문장들 간의 의미적 관계를 분석하여 같은 의미를 나타내는 문장들의 그룹화를 통해 요구사항 문장 S를 구성하는 단계이다.

요구사항 텍스트는 하나 이상의 일반적인 문장들로 구성된 복합문이다. 요구사항 텍스트에서 각 문장들을 같은 의미로 구성된 요구사항 문장 S로 구성하기 위해서는 문장들 간의 의미적 관계 분석 조건을 통해 의미적 관계를 분석하고 같은 의미를 나타내는 문장들을 선정해야 한다.

요구사항 텍스트 R을 구성하는 문장들 중에서 분석을 위해 선택된 선행되는 문장을  $s_i$ (where  $0 \leq i \leq R_{total}$ )라고 정의하고, 선택된 문장  $s_i$ 와 분석하기 위해 선택되는 후행되는 문장을  $s_{ij}$ (where  $0 \leq j \leq R_{total}$ ,  $i \neq j$ )라고 정의한다. 문장  $s_i$ 와  $s_{ij}$ 의 의미적 관계를 분석하여 구성된 요구사항 문장을  $S_k$ (where  $0 \leq k \leq R_{total}$ )라고 정의한다.  $R_{total}$ 은 요구사항 텍스트를 구성하는 일반적인 문장들의 총 개수이다.

의미적 관계를 분석하기 전에 먼저 그룹화 될 요구사항 문장  $S_k$ 를 준비한다.  $s_i$ 를 포함하는 요구사항 문장  $S_k$ 가 존재하면  $s_i$ 와 의미적 관계로 분석되는 모든  $s_{ij}$ 를 포함하는 요구사항 문장  $S_k$ 를 선택한다.  $s_i$ 를 포함하는 요구사항 문장  $S_k$ 가 존재하지 않으면  $s_i$ 를 초기 구성요소로 포함하는 요구사항 문장  $S_k$ 를 생성한다.

$s_i$ 와  $s_{ij}$ 의 관계가 의미적 관계 분석의 네 가지 조건에 의해 분석되면 문장  $s_i$ 와  $s_{ij}$ 은 내포하고 있는 의미가 같은 의미를 나타내고 있기 때문에  $s_{ij}$ 를 요구사항 문장  $S_k$ 의 구성요소로 그룹화 한다. 요구사항 문장  $S_k$ 로 그룹화 된 문장  $s_i$ 와  $s_{ij}$ 는 요구사항 문장  $S_k$ 의 구성요소가 된다.

문장  $s_i$ 와  $s_{ij}$ 의 관계가 의미적 관계 분석의 네 가지 조건에 의해 분석되지 않으면  $s_i$ 와  $s_{ij}$ 는 내포하고 있는 의미가 서로 다른 의미를 나타내고 있기 때문에  $s_i$ 와 의미적 관계 분석을 수행할 다음 문장  $s_{ij+1}$ (where  $0 \leq j+1 \leq R_{total}$ )과 의미적 관계가 있는지 분석한다.

이와 같은 방법으로 모든 문장들을 의미적 관계 분석을 위해 선택된 선행되는 문장  $s_i$ 로 선정하여 다른 문장들  $s_{ij}$ 와 의미적 관계 분석을 통해 요구사항 문장  $S_k$ 를 구성한다.

$$\begin{aligned}
 R &= \{ s_0, s_1, \dots, s_l, 0 \leq l \leq R_{total} \} \\
 S_k &= \{ s_0, s_1, \dots, s_m, 0 \leq m \leq R_{total} \} \\
 R &= \{ S_0, S_1, \dots, S_k, 0 \leq k \leq R_{total} \} \quad (1)
 \end{aligned}$$

(1)과 같이 의미적 관계 분석에 의해 구성된  $S_k$ 는 요구사항 텍스트 R을 구성하는 문장들의 그룹으로 구성되기 때문에 도출된 모든  $S_k$ 는 요구사항 텍스트 R에 포함된다.

요구사항 의미명제 도출과정 2단계는 일반적인 문장들 로 구성된 요구사항 문장에서 개별적인 의미들을 모두 도출하는 단계이다.

$$\begin{aligned}
 S_k &= \{ s_0, s_1, \dots, s_i, 0 \leq i \leq R_{total} \} \\
 s_i &= \{ p_0, p_1, \dots, p_l, 0 \leq l \leq S_{total} \} \\
 S_k &= \{ p_0, p_1, \dots, p_i, 0 \leq i \leq S_{total} \} \quad (2)
 \end{aligned}$$

(2)와 같이 요구사항 문장은 요구사항 텍스트와 마찬가지로 하나 이상의 개별적인 의미들로 구성된 복합문이다. 요구사항 문장을 구성하는 일반적인 문장들은 하나 이상의 개별적인 의미들로 구성된다.

요구사항 문장은 같은 의미를 내포하고 있는 문장들의 집합으로 구성되기 때문에 요구사항 문장을 나타내는 개별적인 의미들을 도출할 수 있다. 도출되는 개별적인 의미를  $p_i$ (where  $0 \leq i \leq S_{total}$ )이라고 정의한다.  $s_{Total}$ 은 일반적인 문장들에서 도출한 개별적인 의미들의 총 개수이고,  $S_{Total}$ 은 일반적인 문장들의 집합인 요구사항 문장에서 도출한 개별적인 의미들의 총 개수이다.

요구사항 문장에서 도출되는 개별적인 의미  $p_i$ 는 대상, 행위나 기능, 행위나 기능의 특성으로 구성된다.

대상은 시스템에서 제공하는 행위나 기능들을 통해 자신의 목적을 달성하기 위하여 시스템과 상호작용하는 존재이다. 대상의 행위나 기능은 대상이 원하는 목적을 달성하기 위하여 시스템이 제공하고 있는 기능들을 말한다. 행위나 기능의 특성은 대상과 행위나 기능으로 표현되지 못하는 시스템의 기능을 수행하는데 있어 필요한 부가적인 정보들을 말한다.

이 세 가지 구성요소를 이용하여 개별적인 의미를 표현하는 이유는 최소의 정보를 이용하여 원자적인 성격을 갖는 단문으로 의미들을 표현할 수 있기 때문이다. 개별적인 의미에서 대상은 주어로 대상의 행위나 기능은 동사, 목적어 및 보어로 나타내어 최소의 정보를 이용하여 의미를 표현하면 애매모호한 표현들을 줄이고 명확하게 의미를 표현할 수 있다. 그리고 대상과 대상의 행위나 기능으로 표현되지 못한 시스템 기능에 대한 필요한 부가적인 정보들을 행위나 기능의 특성에 작성하므로 최소의 정보로 표현되지 못하는 의미들을 보조해 줄 수 있다.

개별적인 의미에서 대상을 주어로 표현되는 모든 명사와 대상의 행위나 기능을 주어를 꾸며주는 동사, 목적어 및 보어 형태들이 모두 대상과 행위나 기능이 되는 것은 아니다.

대상이 너무 큰 의미를 나타내는 명사로 선정되면 그 밖에 작은 의미를 구성하는 명사들이 모두 큰 의미를 나타내는 명사에 추상적으로 구성되기 때문이다. 너무 큰 의미를 나타내는 명사는 바로 개발자고자 하는 시스템이다. 개발하고자 하는 시스템이 대상으로 선택되면 시스템은 모든 행위나 기능을 포함할 수 있기 때문에 개발하고자 하는 시스템은 대상에서 제외된다. 행위나 기능도 대상과 마찬가지로 추상적이고 개념적인 행위나 기능은 행위나 기능에서 제외된다. 지금까지 살펴본 정보들을 작성하기 위한 형식을 명세표로 나타내면 <표 6>과 같다.

〈표 3〉 개별적인 의미 p 작성 명세표

$p_i$ $(0 \leq i \leq n)$	명제	
	대상	
	대상의 행위나 기능	
	행위나 기능의 특성	

〈표 4〉 개별적인 의미 p 작성 명세표의 작성 예

$P_1$	명제	학생은 수강등록정보를 확인한다.
	대상	학생
	대상의 행위나 기능	수강등록정보를 확인한다.
	행위나 기능의 특성	수강신청이 완료된 후

〈표 3〉에서 "명제" 란은 요구사항 문장 S에서 도출한 대상과 대상의 행위나 기능을 명제 형식으로 작성한다. "대상" 란은 개별적인 의미에서 시스템에서 행위나 기능의 주체로 선정된 대상을 작성한다. "대상의 행위나 기능" 란은 대상이 시스템에서 수행하는 행위나 기능을 작성한다. 마지막으로 "행위나 기능의 특성" 란은 선정된 대상과 행위나 기능에 대한 추가적인 정보들을 작성한다. 〈표 4〉는 작성 예를 나타낸다.

요구사항 의미명제 도출 과정 3단계는 요구사항 문장에서 도출해낸 개별적인 의미들의 의미적 관계를 분석하여 의미명제를 도출한다. 도출한 개별적인 의미들의 의미적 관계를 분석하여 의미명제를 도출하는 이유는 도출된 모든 개별적인 의미들이 요구사항 문장 S를 나타내지 않기 때문이다. 요구사항 문장 S의 특정한 개별적인 의미가 다른 개별적인 의미들과의 의미적 관계를 통해 요구사항 문장을 구성할 수 있지만, 구성하는데 꼭 필요하지 않은 개별적인 의미가 될 수 있기 때문이다. 그래서 요구사항 문장에서 의미명제를 도출하기 위해서는 요구사항 문장에서 도출할 수 있는 모든 개별적인 의미들을 의미적 관계 분석을 수행해야 한다.

$$S_k = \{ p_0, p_1, \dots, p_i, 0 \leq i \leq S_{total} \}$$

$p_i$ 들 중에서 의미명제 P가 도출되면

$$S_k = \{ P_0, P_1, \dots, P_1, 0 \leq 1 \leq S_{total} \} \quad (3)$$

의미적 관계 분석을 위해 선택된 선행되는 개별적인 의미를  $p_i$ (where  $0 \leq i \leq S_{total}$ )라고 정의하고, 선택된 개별적인 의미  $p_i$ 와 분석을 위해 선택되는 후행되는 개별적인 의미를  $p_j$ (where  $0 \leq j \leq S_{total}, i \neq j$ )라고 정의한다. 개별적인 의미  $p_i$ 와  $p_j$ 의 의미적 관계를 분석하여 도출된 의미명제는  $P_1$ (where  $0 \leq j \leq S_{total}$ )이라고 정의하고,  $S_{Total}$ 은 요구사항 문장에서 도출한 개별적인 의미들의 총 개수이다.

개별적인 의미  $p_i$ 와  $p_j$ 가 의미적 관계 분석의 네 가지 조건에 의해 분석되면 개별적인 의미  $p_i$ 와  $p_j$ 는 요구사항 문장  $S_k$ 를 구성하는데 의미적 관계를 형성하고 있기 때문에  $p_i$ 는 의미명제  $P_1$ 로 도출된다.

개별적인 의미  $p_i$ 와  $p_j$ 가 의미적 관계 분석의 네 가지 조건에 의해 분석되지 않으면  $p_i$ 와  $p_j$ (where  $0 \leq j+1 \leq S_{total}$ )는

요구사항 문장 S를 구성하는데 의미적 관계를 형성하고 있지 않기 때문에  $p_i$ 에 대한 다음 개별적인 의미  $p_{i+1}$ 과의 의미적 관계 분석을 수행한다. 모든 개별적인 명제들 간의 의미적 관계 분석을 수행한다. 모든 개별적인 의미들 간의 의미적 관계로 분석하였을 때 그 어떤 개별적인 의미들과도 의미적 관계에 있지 않은 개별적인 의미들은 생략된다.

의미명제  $P_1$ 로 선정된 개별적인 의미  $p_i$ 에 대한 정보들과 의미적으로 관계를 갖는 개별적인 의미  $p_i$ 에 대한 의미적 관계 정보들을 추가하여 의미명제  $P_1$ 를 구성할 수 있다. 의미명제의 구성 명세는 〈표 5〉와 같은 형식으로 작성된다.

〈표 5〉에서 "명제", "대상", "대상의 행위나 기능", "행위나 기능의 특성" 란은 의미명제  $P_1$ 로 도출된 개별적인 의미  $p_i$ 의 "명제", "대상", "대상의 행위나 기능", "행위나 기능의 특성"을 작성한다. "의미적 관계 분석"은 의미명제  $P_1$ 로 도출된 개별적인 의미  $p_i$ 와 의미적 관계를 맺고 있는 의미명제와 의미적 관계 분석의 조건을 작성한다. 〈표 6〉은 작성 예를 나타낸다.

〈표 5〉 의미명제 작성 명세표 P

$P_k$ $(0 \leq k \leq n)$	명제	
	대상	
	대상의 행위나 기능	
	행위나 기능의 특성	
	의미적 관계 분석	

〈표 6〉 의미명제 작성 명세표 P의 작성 예

$P_1$	명제	학생은 수강등록정보를 확인한다.
	대상	학생
	대상의 행위나 기능	수강등록정보를 확인한다.
	행위나 기능의 특성	수강신청이 완료된 후
	의미적 관계 분석	$P_1$ 은 $P_0$ 의 결과이다.

지금까지의 살펴본 요구사항 의미명제 도출과정에 대한 전체적인 구성 명세표를 제안하면 〈표 7〉과 같다. 〈표 8〉은 전체 분석 자료의 작성 예이다.

〈표 7〉 요구사항 분석 명세표

요구사항 텍스트 R		
...요구사항 텍스트 내용...		
요구사항 문장 $S_0$ : ...요구사항 문장 내용...		
$P_0$	명제	
	대상	
	대상의 행위나 기능	
	행위나 기능의 특성	
	의미적 관계 분석	
$P_1$	명제	
	.....	
.....		
$P_j$ $(0 \leq j \leq S_{Total})$	명제	
	.....	

요구사항 문장  $S_1$  : ...요구사항 문장 내용...

$P_0$	명제	
	.....	

$P_1$	명제	
	.....	

.....

$P_j$ ( $0 \leq j \leq S_{Total}$ )	명제	
	.....	

.....

요구사항 문장  $S_i$  ( $0 \leq i \leq R_{Total}$ ): ...요구사항 문장 내용...

.....

$P_j$ ( $0 \leq j \leq S_{Total}$ )	명제	
	.....	

<표 8> 요구사항 분석 명세표의 작성 예

요구사항 텍스트 R :  
학생은 수강등록을 한다. 수강등록이 완료된 후, 자신의 수강등록 정보를 확인하기 위해서 시스템을 사용한다. 교수는 해당 학기에 가르칠 강좌를 개설하고, 수강등록이 완료된 후, 자신의 개설강좌에 등록된 학생들의 정보를 확인하기 위해서 시스템을 사용한다.

요구사항 문장  $S_0$  :  
학생은 수강등록을 한다. 수강등록이 완료된 후, 자신의 수강등록정보를 확인하기 위해서 시스템을 사용한다.

$P_0$	명제	학생은 수강등록을 한다.
	대상	학생
	대상의 행위나 기능	수강등록을 한다.
	행위나 기능의 특성	없음
의미적 관계 분석		$P_0$ 은 $P_1$ 의 원인이다.

$P_1$	명제	학생은 수강등록정보를 확인한다.
	대상	학생
	대상의 행위나 기능	수강등록정보를 확인한다.
	행위나 기능의 특성	수강신청이 완료된 후
의미적 관계 분석		$P_1$ 은 $P_0$ 의 결과이다.

요구사항 문장  $S_1$  :  
교수는 해당 학기에 가르칠 강좌를 개설하고, 수강등록이 완료된 후, 자신의 개설강좌에 등록된 학생들의 정보를 확인하기 위해서 시스템을 사용한다.

$P_0$	명제	교수는 강좌를 개설한다.
	대상	교수
	대상의 행위나 기능	강좌를 개설한다.
	행위나 기능의 특성	없음
의미적 관계 분석		$P_0$ 은 $P_1$ 의 원인이다.

$P_1$	명제	교수는 강좌에 등록된 학생들의 정보를 확인한다.
	대상	교수
	대상의 행위나 기능	등록한 학생들의 정보를 확인한다.
	행위나 기능의 특성	수강신청이 완료된 후
의미적 관계 분석		$P_1$ 은 $P_0$ 의 결과이다.

3.2 거시명제를 이용한 핵심적인 요구사항 정의

요구사항 텍스트를 거시구조로 분석하여 의미명제들을 도출하였다. 의미명제들은 요구사항 텍스트를 나타내는 명제 단위의 구성요소들이다. 요구사항 텍스트가 가진 의미명제들에 거시규칙을 적용하면 의미명제들의 전체적인 의미 즉, 거시명제를 정의할 수 있다. 거시규칙은 요구사항 텍스트의 주제를 명확히 파악하는 척도가 되기 때문에 거시규칙의 적용으로 정의된 거시명제는 핵심적인 요구사항이 된다. 2.1.3 절에서 제시한 거시규칙 네 가지를 반 다이크(van Dijk)의 텍스트 학의 거시규칙[8]을 본 논문에서 성격에 맞도록 정리하면 다음과 같다.

첫째, 생략 규칙은 여러 의미명제들 중에서 핵심적이지 않은 부가적인 의미명제들은 생략한다. 각 요구사항 문장의 의미명제들이 모두 핵심적이지 않기 때문이다.

둘째, 선택규칙은 의미적 관계에 의해서 도출된 의미명제들 중에서 대표적인 의미명제만을 선택한다. 조건, 전제, 결과로 표현되는 의미명제는 생략되고 나머지 의미명제가 선택된다.

셋째로 일반화 규칙은 의미명제들을 상위 개념으로 표현하는 의미명제로 일반화되어 표현된다. 모든 거시 규칙은 일반화 속성을 가지고 거시명제를 도출한다.

넷째로 구성화 규칙은 여러 의미명제들을 어떤 상태, 사건, 과정, 보통조건, 상황, 요소, 결과와 같은 총괄적이고 통합된 표현으로 대체할 수 있다. 의미명제는 새로운 정보에 의해 대체되지만 제거되거나 선택되지는 않는다.

거시규칙들은 서로 연관성을 가지고 의미적 관계에 대하여 영향을 받는다. 선택 규칙은 의미명제들에서 전체적인 주제의 의미명제를 선택할 수 있고, 그 밖에 의미명제들은 생략규칙을 이용하여 생략할 수 있다. 구성화 규칙은 선택 규칙과 같은 조건에 의해서 명제로 대체되지만, 차이점은 구성화 규칙은 여러 의미명제들을 통합하여 거시명제를 나타낸다는 점이다. 일반화 규칙은 선택 규칙과 구성화 규칙을 이용하여 거시명제를 도출하는 의미명제들은 모두 일반화 규칙이 적용된다.

지금까지 살펴본 거시규칙을 의미명제들에 적용하여 거시명제를 정의할 수 있고, 정의한 거시명제는 요구사항 텍스트에서 전체적인 주제를 나타내는 핵심적인 요구사항이 된다.

의미명제들에 거시규칙을 적용하여 거시명제를 도출하기 위해서 <표 9>의 거시규칙 적용 명세표를 이용하여 도출하도록 한다.

<표 9> 시규칙 적용 명세표

요구사항 문장	의미명제	거시규칙	거시명제
$S_0$	$P_0, P_1, \dots$ $P_j(0 \leq j \leq S_{Total})$	적용규칙	$M_{0-0}, M_{0-1}, \dots$ $M_{i-k}(0 \leq k \leq S_{Total})$
$S_1$	$P_0, P_1, \dots$ $P_j(0 \leq j \leq S_{Total})$	적용규칙	$M_{1-0}, M_{1-1}, \dots$ $M_{i-k}(0 \leq k \leq S_{Total})$
.....			
$S_i$ ( $0 \leq i \leq R_{total}$ )	$P_0, P_1, \dots$ $P_j(0 \leq j \leq S_{Total})$	적용규칙	$M_{i-0}, M_{i-1}, \dots$ $M_{i-k}(0 \leq k \leq S_{Total})$

거시명제는  $M_{i-k}$  (where  $0 \leq i \leq R_{total}$ ,  $0 \leq k \leq S_{total}$ ) 이라고 정의하고,  $R_{total}$ 은 요구사항 텍스트를 구성하는 일반적인 문장들의 총 개수이고,  $S_{total}$ 은 요구사항 문장에서 도출한 개별적인 의미들의 총 개수이다.

<표 9>에서 "요구사항 문장"은 거시규칙을 적용할 요구사항 문장으로 거시명제는 요구사항 문장 단위로 도출된다. 각 요구사항 문장들은 독립적인 의미를 나타내는 같은 의미들로 구성된 문장들의 집합이기 때문이다. "의미명제"는 요구사항 문장을 구성하는 의미명제들을 말한다. "거시규칙 적용"은 의미명제들이 갖는 의미적 관계 조건에 따라 적용할 거시규칙의 종류를 작성한다. "거시명제"는 거시규칙을 통해 도출된 대상과 행위나 기능으로 구성된 명제를 작성한다. <표 10>은 작성 예를 나타낸다.

<표 10> 거시규칙 적용 명세표의 작성 예

문장	의미명제	거시규칙	거시명제
$S_0$	$P_0$ : 학생은 수강신청을 한다. $P_1$ : 학생은 수강 등록정보를 확인한다.	선택 생략 일반화	$M_{0-0}$ : 학생은 수강신청을 한다.
$S_1$	$P_0$ : 교수는 강좌를 개설한다. $P_1$ : 교수는 강좌에 등록된 학생들의 정보를 확인한다.	선택 생략 일반화	$M_{1-0}$ : 교수는 가르칠 강좌를 개설한다.

의미명제들에 거시규칙을 적용하여 정의된 거시명제들을 <표 11>의 거시구조 구성 명세표를 이용하여 작성하면 세부적인 정보와 함께 표현할 수 있다.

<표 11> 거시구조 구성 명세표

$M_{i-k}$ ( $0 \leq i \leq n$ ) ( $0 \leq k \leq n$ )	거시명제	
	요구사항 문장	
	대상	
	대상의 행위나 기능	
	하위 개념의 의미명제	
	행위나 기능의 특성	

<표 12> 거시구조 구성 명세표의 작성 예

$M_{0-0}$	거시명제	학생은 수강신청을 한다.
	요구사항 문장	$S_0$
	대상	학생
	대상의 행위나 기능	수강신청을 한다.
	하위 개념의 의미명제	
	$P_0$ : 학생은 수강신청을 한다. $P_1$ : 학생은 수강 등록정보를 확인한다.	
	행위나 기능의 특성	
	$P_1$ : 수강신청이 완료된 후	

<표 11>에서 "거시명제"란은 의미명제들에 거시규칙을 적용하여 도출된 거시명제를 대상과 행위나 기능 형태로 작성한다. "요구사항 문장"란은 거시명제를 도출하는데 사용된 요구사항 문장을 작성한다. "대상"란은 거시명제에서 행위나 기능의 주체를 선정된 대상을 작성한다. "대상의 행위나 기능"란은 대상이 시스템에서 수행하는 행위나 기능을 작성한다. "거시규칙"란은 거시명제를 정의하기 위하여 의미명제들에 적용한 거시규칙의 종류를 작성한다. "하위 개념의 의미명제"란은 거시명제가 구성하고 있는 의미명제들의 정보를 작성한다. 마지막으로 "행위나 기능의 특성"란은 거시명제가 구성하고 있는 의미명제들의 행위나 기능의 특성을 작성한다. <표 12>는 작성 예를 나타낸다.

거시규칙을 통해 도출된 거시명제들은 명확하게 분석된 고객의 요구를 의미하기 때문에 분석자에 의해 분석된 거시명제는 고객과의 명확하고 원활한 의사소통 도구로 사용될 수 있다.

### 3.3 거시명제를 적용한 Use-case Model의 작성

요구사항 텍스트에서 분석하여 도출한 의미명제들과 거시명제들을 Use-case Model을 이용하여 작성할 수 있다.

Use-case Model에서 Actor는 시스템으로부터 어떠한 서비스를 제공받기 위해 존재하는 대상을 말한다. 즉, 자신의 목적을 달성하기 위해서, 시스템과 상호작용하는 어떤 사람 또는 어떤 것의 역할들이 모두 Actor가 된다. Use-case는 시스템에 의해 제공되는 기능들을 나타낸다. 일반적으로 하나로 완료되는 중요 기능을 나타내는 시스템에 의해 수행되는 일련의 트랜잭션이다.

의미명제와 거시명제는 요구사항 텍스트를 분석하여 대상에 대하여 시스템에 의해 수행되는 일련의 행위나 기능을 표현하고 있다. 그러므로 의미명제와 거시명제의 대상은 Actor로 표현할 수 있고, 대상에 대한 행위나 기능은 Use-case로 표현할 수 있다.

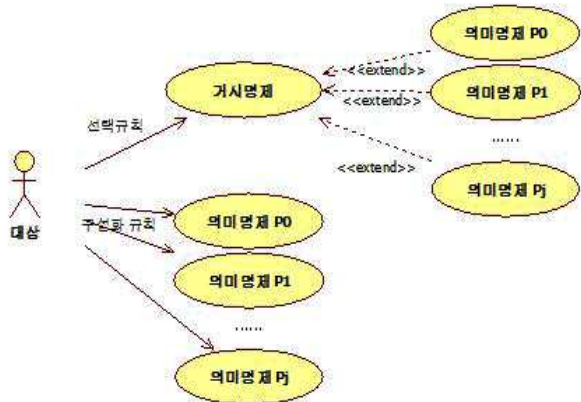
거시명제는 거시규칙을 이용하여 요구사항 문장의 전체적인 의미에서 핵심적인 의미만을 추출하여 표현한다. 거시명제의 구성요소로 포함되어 표현되지 못하는 의미명제들은 거시규칙의 종류에 따라 Use-case의 관계표현으로 세부적인 정보를 표현할 수 있다.

선택 규칙의 적용으로 생략된 의미명제들은 확장 관계(extend)를 이용하여 부수적이고 선택적인 동작들로 묘사될 수 있다.

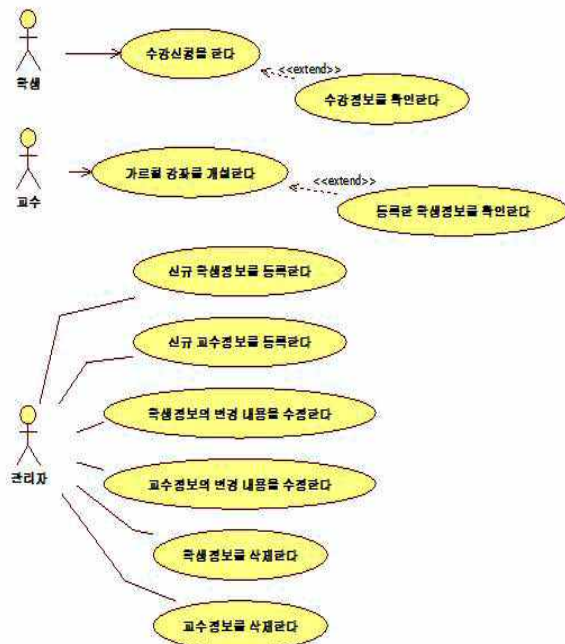
구성화 규칙의 적용으로 도출된 거시명제는 포함하고 있는 의미명제들의 대표가 되는 상위개념으로 시스템에서 제공하고 있는 하나로 완료되는 중요 기능을 나타내지 못하기 때문에 Use-case로 표현될 수 없다. 그래서 구성화 규칙에 의해 도출된 거시명제는 포함하고 있는 의미명제들을 Use-case로 표현한다.

(그림 2)는 Use-case Model를 이용하여 거시명제와 의미명제를 작성한 그림이다. (그림 3)는 작성 예를 나타낸다.





(그림 2) Use-case Model을 이용한 거시명제와 의미명제의 작성



(그림 3) Use-case Model 작성 예

거시구조 분석에 대한 거시명제와 의미명제의 정보, Use-case Model과 함께 본 논문에서는 제시되지 않았지만 Use-case Model에서 제시하고 있는 각 Use-case에 부가적인 정보들을 표현할 수 있는 명세서를 이용하면 고객과 사용자 그리고 개발자들 간의 시스템의 기능과 행위에 대해서 효율적인 의사소통 도구로 사용될 수 있다.

#### 4. 거시구조 개념을 이용한 요구사항 분석방법의 이점

요구공학에서 대표적으로 사용되는 기법 중에서 목표기반의 분석과 비교하여 본 논문에서 제시한 분석방법의 이점을 제시하도록 한다.

대표적인 기법들 중에서 목표 기반의 분석은 목표를 통하여 요구사항을 추출하고 분석한다[13][14].

목표 기반의 분석의 대표적인 장점은 요구사항을 추출, 분석하면 시스템 전체에 대한 요구사항을 파악하기 쉬워지며, 기능적, 비 기능적 요구사항을 균형적으로 추출할 수 있고, 추출된 요구사항에 대한 생성 근거를 제시할 수 있다 [11].

목표기반의 분석의 대표적인 단점은 첫째로 초기목표를 식별하기가 어렵기 때문에 요구사항 추출, 분석의 시작단계에서 많은 시간이 필요하다. 둘째로 목표들의 분화로 요구사항을 분석하는 기법이기 때문에 어느 단계에서 목표의 분화를 멈춰야 할지 모호하다. 이런 단점을 해결하기 위하여 목표기반의 분석에서는 4개의 단계로 비즈니스 단계, 서비스 단계, 상호작용 단계, 기능 단계로 나누고, 요구사항 추상화 계층에 따라 목표를 분화(decomposition)하고, 정제(refinement)하여 구체화(concretion)한다. 셋째로 설정된 목표를 진화시키는 방법에 대한 체계적인 방법이 존재하지 않는다[11][12].

목표 기반의 분석 방법과 비교하여 본 논문에서 제시한 요구사항 분석방법의 이점을 정리하면 다음과 같다.

첫째는 목표기반의 분석에서는 최상위 목표를 통하여 기능목표를 추출하기 위한 4개의 요구사항 추상화 단계에서 분석 진행은 각 단계마다 분석되고, 제공되어야 하는 목표를 정의하고 만족 시켜야한다. 초기목표를 식별하기가 어렵기 때문에 요구사항 추출, 분석의 시작단계에서 많은 시간이 필요하다는 단점이 발생한다. 본 논문에서 제시된 분석 방법은 요구사항 텍스트에서 거시명제를 도출하는 과정은 4단계로 단계마다 분할된 요구사항 정보는 계층적으로 구성하면서 분석하기 때문에 각 단계의 분화를 멈춰야 하는지와 각 단계마다의 입출력되는 정보의 모호성이 없고, 각 단계마다 발생하는 목표를 정의하고 만족 시키는 것이 어렵지 않다.

둘째는 요구사항을 명확하게 정의하기 위한 기술적인 측면에서 다양한 분석 및 검증기법에 적용하여 요구사항을 명확하게 도출 및 분석할 수 있다. 요구사항이 텍스트 형태로 제공된다면 거시명제를 핵심적인 요구사항으로 도출하여 다양한 분석 및 검증기법에 적용할 수 있게 다양하게 재구성하여 사용할 수 있다. 목표 기반의 방법에서는 분화의 모호성의 해결책으로 제시한 4개의 단계에 각 단계마다 목표로 정의되는 정보들은 텍스트로 구성되면, 거시구조 분석은 분화와 정제의 역할을 하고 거시규칙은 구체화 역할을 수행할 수 있다.

마지막으로 목표 기반의 방법에서는 목표를 통하여 요구사항을 추출하고 분석한다. 그러나 요구사항을 추출하고 분석하기 위한 초기목표를 식별하기가 어렵기 때문에 요구사항 추출, 분석의 시작단계에서 많은 시간이 필요하다. 게다가 여러 조건에 의해서 초기의 목표 처리과정은 실제 상황에 반영되지 않고 이상적인 부분에 많은 반영으로 인해 애매한 목표의 개념은 비효율적인 요구사항들이 초래하게 된다. 따라서 목표 기반의 분석에서는 의미적인 수행을 할 수 있도록 목표가 제공되어야한다[16]. 본 논문의 분석방법에



서 핵심적인 의미와 목표 기반의 분석에서 말하고 있는 의미적인 수행은 추출하고 싶은 요구사항의 의미라는 점에서 동일하다. 주어진 요구사항 텍스트를 거시구조로 분석하고 거시규칙을 적용하므로 핵심적인 의미 즉, 목표기반의 분석에서 도출하고자 하는 목표를 쉽게 도출할 수 있다.

본 논문에서 목표 기반의 분석에서 비교하고자 하는 부분은 개발될 시스템에서 왜 필요한지에 대한 성취하고자 하는 목표를 추출하여 요구사항을 추출하고 분석하는 방법에 대한 것이다. 제시된 장점에서 살펴본 것과 같이 목표라는 핵심적인 의미를 도출하는데 있어서 목표기반의 분석의 목표 도출을 이용한 요구사항의 추출 및 분석보다는 일관성 있고 명확하게 목표를 도출하고, 다양한 형태 및 다른 분석방법에 적용하여 효율적으로 사용할 수 있다.

### 5. 결 론

본 논문에서는 고객이 원하는 핵심적인 요구사항을 명확하게 도출하기 위해, 텍스트언어학(textlinguistics)의 거시구조(Macrostruktur)를 이용하여 요구사항을 분석하여, 의미명제를 도출하고, 도출된 의미명제들에 거시규칙을 적용하여 핵심적인 요구사항을 정의한다.

고객이 요청하는 시스템을 개발하기 위한 의사소통 방법이 문 또는 텍스트의 형태로 이루어지기 때문에 문 또는 텍스트를 단순히 구성형태에서 필요한 정보만을 추출하여 요구사항을 이해, 분석, 문서화하고, 변경을 관리 통제하는 다양한 기술과 방법은 요구사항을 빠짐없이 명확하게 분석하기에는 많은 어려움이 있고, 유연성이 많이 필요하다.

본 논문에서 제시된 요구사항 분석 방법은 사람에게서 발생한 언어발화를 기초로 한 순수 말이나 문이 뜻하는 의미를 분석하여 핵심적인 의미를 도출한다. 핵심적인 의미는 최소한의 정보를 이용하여 명제단위로 표현하므로 애매모호한 표현들을 줄이고 명확하게 표현한다. 명확하게 표현된 핵심적인 의미는 요구사항을 빠짐없이 명확하게 분석하여 도출한 정보로서 고객과 분석자간의 의사소통의 효율을 높일 수 있다.

본 논문에서는 텍스트언어학(textlinuistics)의 거시구조를 이용하여 Use-case Model의 방법만을 적용하였으나 다른 분석방법과의 적용과 이론에 대한 확장을 통하여 더욱 효율적으로 요구사항이 내포하고 있는 핵심적인 의미를 도출할 수 있을 것이다. 따라서 현재 기술적인 측면에서 이론적으로 많이 사용되고 있는 다양한 분석 및 검증기법과 비교 및 상호보완에 대한 연구를 진행하여 요구사항 도출뿐만이 아니라 소프트웨어 설계에 적용할 수 있는 방법을 연구를 진행할 것이다.

### 참 고 문 헌

[1] 소프트웨어 요구사항 관리 사례 연구, 최정은, 최순규, 이선아, 한국정보과학회 봄 학술발표논문집 2002.

[2] The Standish Group, Standish Group Report, 2005.  
 [3] 소프트웨어사업 요구사항 명세화 표준지침 개발, 이병걸 외 5명, 정보과학회논문지: 소프트웨어 및 응용 제 36권 제 7호(2009. 7)  
 [4] UML을 활용한 GLORY의 기능적 요구사항 분석 및 검증, 궁상환, 이재기, 남궁한, 한국콘텐츠학회논문지 08, Vol.8 No.5  
 [5] 텍스트언어학의 이해(Understanding Textlinuistics), 강창우 외 16명, 2004  
 [6] 텍스트 연구회 편 : 문 주제 중심의 텍스트 요약과 거시규칙, 김재봉, 한국텍스트언어학회 텍스트언어학, 1995  
 [7] 요약 능력과 요약 규칙, 서혁, 국어교육학회 국어교육학연구, 1994년  
 [8] Text학, van Dijk, 정시호 역, 1995  
 [9] 텍스트 요약 전략에 대한 국어교육학적 연구, 김재봉, 1999  
 [10] 상태 기반의 요구사항 명세 및 코드 생성방안, 2010 한국 소프트웨어공학 학술대회 논문집 제 12권 제 1호 , 박수진, 황현진, 박수용  
 [11] 목표와 시나리오 기반의 통합적 요구사항 분석방안, 김진태, 김동선, 박수용, 정보과학회논문지 : 소프트웨어 및 응용 제 31권 제 5호 (2004. 5)  
 [12] 아키텍처 모델링을 위한 요구사항 정량화 기법, 김진태, 양원석, 정창해, 박수용, 정보과학회논문지 : 소프트웨어 및 응용 제 33권 제 1호 (2006.1)  
 [13] "Goal-Based Requirements Analysis", Annie I. Anton, Proceedings of ICRE 96  
 [14] "Why Goal-Oriented Requirements Engineering.," E. Yu and J. Mylopoulos, Proceedings of the 4th International Workshop on Requirements Engineering: Foundations of Software Quality(8-9 June 1998, Pisa, Italy). E. Dubois, A.L. Opdahl, K. Pohl, eds. Presses Universitaires de Namur, 1998. pp.15-22.  
 [15] UML기반 시스템 분석설계 CHAPTER 5 요구사항 정의, 2008, 장연세  
 [16] "Guiding Goal Modeling Using Scenarios", Colette Rolland, Carine Souveyet, and Camille Ben Achour, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, Vol.24, No.12, DECEMBER, 1998.  
 [17] 텍스트는 체계인가? - 반 다이크의 분석모델을 중심으로., 이성만 (1993), 독일문학 50, 303-332.  
 [18] Einführung in die Textlinguistik. Tübingen: Niemeyer. , de Beaugrande, Robert-Alain/Dressler, Wolfgang Ulrich (1981), (김 태호/이현호 (역) (1995): 담화.텍스트언어학입문. 한신문화사)  
 [19] Eine Einführung in Grundbegriffe und Methoden., Brinker, Klaus (1992), Linguistische Textanalyse. 3., durchges. und erweit. Aufl. Berlin: Schmidt.  
 [20] Textlinguistik., Coseriu, Eugenio (1980), Eine Einführung. Tübingen: Narr. (신 익성 (역) (1995): 텍스트언어학. 사회문화연구소)  
 [21] Textlinguistik. Eine Einführung., Heinemann, Wolfgang/Viehweger, Dieter (1991), Tübingen: Niemeyer.  
 [22] van Dijk, Teun A. (1997), The Study of Discourse. In: Teun

A. van Dijk (ed.): Discourse Studies: A Multidisciplinary Introduction. Bd. I: Discourse as Structure and Process London: Sage, 1-34.

- [23] 프레임, 스크립트 이론과 텍스트 정보처리과정. 박정준 (1994), 텍스트언어학 2, 61-107.
- [24] 텍스트 이해의 관점들, 정동현(1995), 독일어문학 6, 501-520
- [25] Strategies of discourse comprehension. Dijk, Teun A. van/Kintsch, Walter (1983), Paris et al. Academic Press.



### 조 용 균

e-mail : toyongkyun@gmail.com  
 2009년 단국대학교 컴퓨터과학과(학사)  
 2011년 단국대학교 전자계산학과(석사)  
 관심분야: 소프트웨어 공학, 요구공학



### 박 용 익

e-mail : dialog62@gmail.com  
 1987년 단국대학교 독어독문학과(학사)  
 1993년 독일 뮌스터대학교 일반언어어학과 (Ph.D)  
 2001년~2004년 연세대학교 연구교수  
 2005년~2011년 서울대학교 인문학연구원  
 전임연구원

관심분야: 대화분석, 이야기분석, 의료커뮤니케이션 교육



### 박 용 범

e-mail : ybpark@dankook.ac.kr  
 1985년 서강대학교 전자계산학과(학사)  
 1987년 Ph.D N.Y.Polytechnic Univ(석사)  
 1991년 Ph.D N.Y.Polytechnic Univ(박사)  
 현 재 단국대학교 컴퓨터과학과 교수  
 관심분야: 패턴인식, 정보아키텍처, 멀티 미디어 보안