

목표와 시나리오 기반의 통합적 요구 사항 분석 방안

(An Integrated Requirements Analysis Method based on Goal and Scenario)

김진태[†] 김동선^{**} 박수용^{***}
 (Jintae Kim) (Dongsun Kim) (Sooyong Park)

요약 요구 공학에서 초기에 요구 사항을 추출하고 분석하는 것은 성공적인 소프트웨어 시스템을 개발하기 위한 중요한 작업이다. 그러나 시나리오 기반의 분석, 목표기반의 분석, 시나리오를 이용한 목표기반의 분석등과 같은 다양한 분석 기법이 있음에도 불구하고, 각 기법들은 장, 단점을 가지고 있기 때문에, 하나의 기법만을 가지고 요구 사항을 분석한다면, 개발할 시스템 전체의 요구 사항을 정확하게 이해하기가 쉽지 않다. 또한 각각의 기법들은 요구 사항 추출 단계를 지원하지 않으므로 분석을 위한 가공되지 않은 데이터(raw data)를 다루기 어렵다. 본 논문에서는 요구 사항 분석을 위한 데이터 획득과 여러 기법들을 효율적으로 통합하는 통합적 요구 사항 분석 방안을 제안한다. 통합적 요구 사항 분석 방안은 2가지 관점에서 기술된다. 첫째는 어떤 요소들이 통합될 수 있는지를 기술하고, 두 번째는 그 요소들이 어떻게 통합되어야 하는지를 기술한다. 본 논문에서 제안된 방안이 어떻게 적용되는지를 미팅예약시스템(Meeting Reservation System)을 예로 설명한다. 또한 다른 기법들과의 비교를 위해 30여명의 요구 사항 분석 실무 경험자를 대상으로 한 실험의 결과를 통해 제안된 기법의 효과를 검증한다.

키워드 : 요구 사항, 소프트웨어 품질, 목표, 시나리오

Abstract In Requirements engineering, requirements elicitation and analysis are very important tasks to develop software systems successfully. Even though various methods have been proposed including scenario-based analysis, goal based analysis, and coupling goal with scenario, each method has its own strengths and weaknesses. If you use only one method, it is very difficult to elicit and to analyze system requirements completely. Since these methods don't support a comprehensive elicitation process, this paper proposes an integrated approach. This integrated approach is focused on two viewpoints. One is what factors are comprised and the other is how they are integrated. This paper also shows how the proposed approach can be applied to Meeting Reservation System (MRS) development. An experiment has been conducted using this approach to demonstrate how complete requirements are elicited and analyzed and to show relative time savings during elicitation in comparison to each method.

Key words : Requirements, Software Quality, Goal, Scenario

1. 서론

요구 공학이란 요구 사항을 정의하고 문서화하는데 필요한 요구 사항 추출, 분석, 기술, 검증, 유지 보수 및

관리를 포함한 제반 공정에 대한 체계적 접근 분야로서 활발한 연구 분야로 떠오르는 소프트웨어 공학 분야이다[1,2]. [3][20]에 의하면 소프트웨어를 개발할 때 필요한 각 단계별 여러 수정 비용이 요구 사항 추출, 분석하는 단계에서 '1-2' 비용이 드는 반면, 뒷단계로 갈수록 더 많은 비용이 필요로 함을 보여준다. 이것은 요구 공학 중에서도 요구 사항을 추출하고 분석하는 단계가 소프트웨어 개발에 있어서 상당히 중요한 부분을 차지하고 있음을 보여준다[4].

요구 사항을 추출하고 분석하는 기법은 여러 가지가

† 학생회원 : 서강대학교 컴퓨터학과
 canon@selab.sogang.ac.kr
 ** 학생회원 : 서강대학교 컴퓨터학과
 darksw@selab.sogang.ac.kr
 *** 정회원 : 서강대학교 컴퓨터학과 교수
 sypark@ccs.sogang.ac.kr
 논문접수 : 2003년 4월 7일
 심사완료 : 2004년 3월 5일

있다. 그중 대표적인 기법은 시나리오 기반의 분석, 목표기반의 분석, 시나리오를 이용한 목표기반의 분석이 있다. 이런 요구 사항 분석 기법들은 각기 장, 단점을 가지고 있다. 따라서 한 가지 분석 기법으로는 정확하게, 빠짐없는 요구 사항을 추출하고 분석하기가 어렵다. 만일 하나의 분석 기술만을 사용하여 요구 사항을 분석한다면, 완벽한 요구 사항 분석을 이루지 못해 사용자의 정확한 요구에 맞는 소프트웨어 개발이 어렵게 된다. 본 논문에서는 하나의 분석 기술만을 사용할 때 발생할 수 있는 문제점들을 해결하기 위해 통합적 요구 사항 분석 방안을 제시한다. 통합적 요구 사항 분석 방안은 2가지 측면에서 기술된다. 첫째는, 어떤 요소들이 통합되어야 하는가에 대한 것, 둘째는 그 요소들이 어떻게 통합되어야 하는가에 대한 것이다. 본 논문에서는 제안된 기법의 효과를 알기 위해 30명의 실무 경험자들을 대상으로 실험한 결과를 통해 제안하는 기법의 효과를 소개한다.

본 논문의 구성은 다음과 같다. 1장에서는 본 연구의 필요성과 방향을 제시하였고, 2장에서는 기존에 존재하는 요구 사항 분석 기술들을 소개하고 장단점을 비교한다. 3장에서는 제안하는 기법의 접근 방안을 설명하고, 4장에서는 접근 방안을 실제로 실현시키는 전체 공정을 설명한다. 5장에서는 사례 연구를 통해 제안된 기법의 적용 방법을 설명하고, 6장에서는 실무경험자들에게 실험한 결과를 설명한다. 마지막으로 7장에서는 결론과 향후 연구에 대해 설명한다.

2. 관련연구

기존 요구사항 분석 기술은 시나리오 기반 분석(Scenario based requirement analysis)[5,6], 목표 기반의 분석(Goal based requirement analysis)[7,8], 그리고 시나리오를 이용한 목표 기반의 분석(Coupling Goal with Scenario)[9] 등이 있다.

2.1 시나리오 기반의 분석

시나리오 기반의 분석은 일반 사용자가 접근하기 용이한 예나 실세계의 경험을 바탕으로 추출한 시나리오를 사용하여, 사용자가 자신의 시스템 요구사항을 기술한다[5,6].

시나리오 기반의 분석은 시나리오 모델링과 시나리오의 검증으로 이뤄진다. 시나리오 모델링은 다시 시나리오 구조모델과 시나리오 스크립트 모델로 구성된다. 시나리오 구조모델은 개발할 시스템의 영역을 설정하고 8개의 개체를 통하여 표현한다. 시나리오 스크립트모델은 개체들 간의 관계를 통하여 시나리오를 생성하게 한다. 시나리오 검증은 생성된 시나리오를 관련된 이해당사자들, 사회규범, 사용자의 요구 등을 바탕으로 검증하는

단계이다.

시나리오 기반의 분석은 시나리오가 사용자에게 친숙한 일련의 작업 순서들을 기반으로 이루어진다는 점에서 다른 분석보다 사용자의 입장을 고려하는 장점이 있다. 그러나 개체들 간의 상호작용으로 시나리오가 생성되므로 시나리오들 간의 관계가 나타나지 않는다. 시나리오는 기능을 나열한 것이므로 비 기능적인 요구 사항을 획득하기 어렵다.

2.2 목표 기반의 분석

목표 기반의 분석은 목표를 통하여 요구사항을 추출과 분석한다[7]. 목표는 이해 관계자가 개발될 시스템에서 성취하고자 하는 것이며, 추상화된 최상위 수준의 요구사항이다[10]. 목표 기반의 분석을 위한 활동들은 목표분석(goal analysis)과 목표 진화(goal evolution)로 나눌 수 있다. 목표 분석은 개발될 시스템의 후보 목표(candidate goal)를 찾아내고, 시스템의 영향을 주는 요소들을 정의하는 활동이다. 목표 분석은 세 단계의 활동 - 목표 식별(Identifying goals), 목표 분류(classifying goals), 에이전트와 이해관계자 식별(identify agents and stakeholders) - 으로 나누어진다. 목표 진화(goal evolution)는 분석된 목표들을 이루기 위한 시나리오를 생성하고 제약 상황 등을 기술한다. 우선 식별된 목표들의 집합을 핵심적인 목표들만으로 최적화하고 핵심 목표들을 명확하게 기술한다. 그리고 제약사항을 기술한 후, 목표를 정제한다. 장애가 되는 목표를 식별하고 시나리오를 추출해 낸다. 최종적으로 목표를 운용 가능케 한다.

시스템의 목표는 추상화된 요구사항이다. 따라서 목표 기반으로 요구사항을 추출, 분석하면 시스템 전체에 대한 요구사항을 파악하기 쉬우며 기능적, 비 기능적 요구사항을 균형적으로 추출할 수 있다. 하지만 초기목표를 식별하기가 어렵기 때문에 요구사항 추출, 분석의 시작단계에서 많은 시간이 필요하다. 또한 설정된 목표를 진화시키는 방법에 대한 체계적인 방법이 존재하지 않는다.

2.3 시나리오를 이용한 목표 기반 분석

시나리오를 이용한 목표 기반 분석은 목표를 성취하기 위한 시나리오 생성 방법을 제공하고 목표들 간의 관계 설정을 위한 방법을 제공한다[9]. 목표는 시나리오의 발견을 돕고(scenario authoring), 시나리오는 목표를 발견하는 것을 돕는다(goal discovery). 그림 1은 목표와 시나리오 결합 방법의 전체 공정을 나타낸다.

그림 1에서 볼 수 있듯이 목표(G)와 시나리오(Sc)의 결합으로 이뤄진 $[G0, Sc0]$ 단위가 존재하고, Sc0에 기술된 시나리오들은 G0을 달성하기 위한 하위 목표(G1, G2, ..., Gn)로 정제된다. 이와 같은 방식으로 시나리오

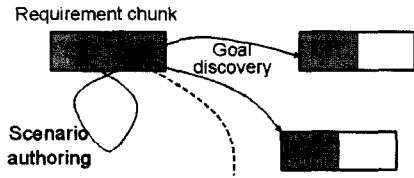


그림 1 목표와 시나리오 결합

생성을 통해 목표를 발견(Goal discovery)하고, 발견된 목표는 다시 시나리오를 생성(Scenario authoring)시킨다.

시나리오를 이용한 목표 기반 분석은 시나리오를 이용하여 목표를 체계적으로 정제하는 방법을 제시하며 목표들 간의 관계를 표현하는 방법도 제시하는 장점을 갖는다. 그러나 초기 목표의 설정이 어렵고 어느 단계(level)에서 목표 정제가 중단되어야 하는지에 대한 기준이 존재하지 않기 때문에 요구사항 추출, 분석이 효율적이지 못할 수 있다.

2.4 유스케이스 지향 분석

유스케이스 지향 분석은 유스케이스와 액터간의 상호작용이 중심이 되어 사용자의 요구사항을 분석한다. 유스케이스를 추출하고 분류할 때 시스템이 개발될 상황을 고려하기 때문에 요구사항 분석이 사용자 입장에서 개발자의 입장에 가깝게 분석된다. 따라서 요구사항의 초기 단계인 추출공정을 제공하지 않는다.

유스케이스는 각각 독립적이기 때문에 유스케이스 단위로 병렬적으로 개발할 수 있다. 하지만 각 유스케이스의 단순한 나열만으로는 시스템의 전체적인 요구사항을 묘사하기 어렵다. 또한 유스케이스가 액터와 시스템의 상호작용을 일련의 작업 순서로 표현하기 때문에 기능적인 요구사항을 표현하기에는 알맞으나, 비기능적인 요구사항을 표현하기에는 부적합한 특징을 가진다.

2.5 각 방법들의 비교, 분석, 평가

2.1~2.3에서 언급한 방법들을 요구 사항 분석 기법이 가져야 하는 유효성검사(Check validation)[11], 정형화(formalization)[11], 추상화(abstraction level)[12], 도메인 지식 표현(domain knowledge), 다양한 관점(various view)[13], 요구사항 완전성(completeness)[13]의

표 1 기법들 간의 비교 평가

Items	Scenario	Goal	Coupling	Use case
Abstraction level	none	weak	weak	weak
Formalization	none	strong	strong	weak
Domain knowledge	weak	none	none	none
Various view	none	weak	weak	weak
Completeness	none	strong	strong	weak
Check validation	weak	weak	weak	weak

항목으로 비교, 분석해 보았으며 결과는 표 1에서 볼 수 있다.

시나리오 기반 분석은 추상화 단계가 나타나지 않고, 정형화된 표현 방식이 없으며, 시스템을 이해하기 위한 다양한 뷰(view)를 제공하지 못한다. 그러므로 전체 시스템에 대한 요구 사항을 식별하기 어렵다. 또한 시나리오 생성 시에 요구 사항이 누수 될 수 있는 경우가 발생하고 시나리오를 검증하는 방법이 분석가와 사용자의 결정에 의해 이뤄진다.

목표 기반 분석은 시스템 목표를 통하여 전체 요구사항을 기술하기 쉽지만 목표를 식별하는 방법이 애매모호하다. 목표들 간의 관계 설정이 어려워 추상화 단계가 존재하지 않고, 도메인에 대한 지식 표현이 없다.

시나리오를 이용한 목표 기반 분석은 목표들 간의 관계를 설정하기 위해 시나리오를 사용하여 진행한다. 그러나 목표 지향 분석과 마찬가지로 초기 목표 식별이 어렵고, 도메인에 대한 지식을 제공해주지 못한다. 또한 추출된 요구 사항을 검증할 수 있는 방안이 없다.

유스케이스 지향 분석은 유스케이스와 액터간의 상호작용을 기초로 요구사항을 추출한다. 하지만 이러한 추출이 개발자를 위주로 지원하는 단점을 지닌다. 따라서 본 논문에서는 사용자 입장에서 쉽게 요구 사항을 도출하고 분석하기 위해 목표와 시나리오를 이용한 통합적 요구 사항 분석방안을 제안한다.

3. 접근 방안

통합적인 요구 사항 추출, 분석 기법을 위한 접근 방안은 2가지의 질문에 의해 시작된다. 첫째는 각 기법들의 어떤 요소들이 통합되어야 하는가와 둘째는 그 요소들이 어떤 방식으로 통합되어야 하는가이다.

3.1 통합되어야 할 요소들

통합되어야 할 요소들은 각 기법들이 가지고 있는 단점들을 살펴봄으로써 찾아 낼 수 있다. 그림 2는 각 기법들의 단점과 이를 해결하기 위한 요소들을 제안하고 있다.

단점 (1), (5), (6)은 추상화(Abstraction)에 의해서 해결된다. 이는 추상화를 통하여 전체 요구 사항을 나타낼 수 있고, 세부적인 요구 사항에서 추상적인 요구 사항으로 추상화되면서 목표식별이 쉽게 이뤄진다[14]. 따라서 초기 목표 식별의 어려움을 극복할 수가 있다. 단점 (2), (4)는 분류(classification)를 통하여 해결된다[15]. 의미적으로 유사한 요구 사항들은 서로 간에 유사한 관계를 갖는다. 반면에 의미적으로 다른 특성을 지닌 요구 사항들은 특수한 관계를 가진다. 단점 (3)은 생성된 시나리오에 품질속성을 매핑(Applying to quality attributes)하므로 해결가능하다. 초기단계에 품질속성을

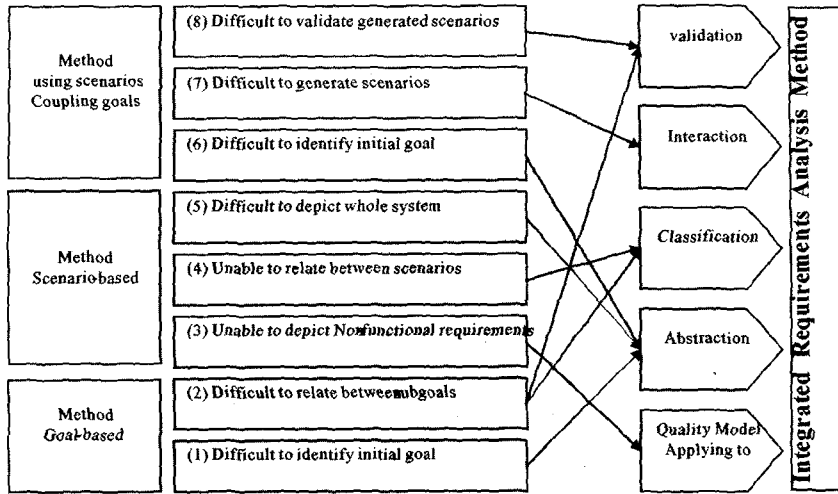


그림 2 통합되어야 할 요소들

매핑하는 것은 쉬운 일이 아니다. 하지만, 초기 단계에 조금이라도 요구 사항에 대한 품질속성을 찾아 낼 수 있다면 설계나 아키텍처 생성에 가이드라인을 제공하게 된다. 단점 (7)은 상호작용(Interaction)에 의해서 해결된다[5]. 구체적인 개체들을 정하고 그 개체들 간의 관계를 통하여 해결한다. 단점 (8)은 상향식과 하향식 방법으로 생성된 시나리오들을 통합하여 검증(validation)으로써 서로의 약점을 보완하게 된다[15]. 따라서 그림 2에서 알 수 있듯이 각 기법들의 단점들을 해결할 수 있는 요소들-추상화(Abstraction), 분류(Classification), 품질속성 매핑(Applying to quality model), 상호작용(Interaction), 검증(Validation)-이 통합되어야 한다.

3.2 통합하는 절차

통합하는 절차는 다양한 뷰(관점)를 표현하기 위해 상향식 영역(bottom up dimension), 하향식 영역(top down dimension)[15]으로 나누고, 요구 사항이 기능적인 요구 사항과 비 기능적인 요구 사항으로 나뉘지므로 기능적인 요구 사항을 보여주는 행위영역(behavior dimension)과 비 기능적인 요구 사항을 보여주는 품질영역(quality dimension)으로 이뤄진다[11]. 행위영역은 기능적인 요구 사항을 식별하게 되고, 품질영역은 비 기능적인 요구 사항을 식별하게 된다. 상향식 영역은 행위영역에서 생성된 시나리오들을 기반으로 그들 간의 관계 생성과 추상화 단계를 거쳐 시스템의 목표를 생성한다. 하향식 영역에서는 시스템 목표로부터 검증 시나리오를 생성한다. 3.1에서 언급된 요소 중에서 행위 영역에 해당하는 요소는 상호작용이며, 상향식 영역에 해당하는 요소는 분류와 추상화이다. 하향식 영역에 해당하는 요소는 검증이며, 품질 영역에 해당하는 요소는 품질

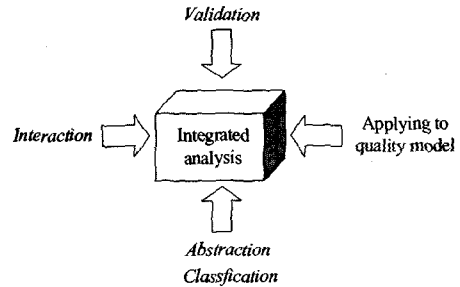


그림 3 통합 영역과 절차

속성 매핑이 된다. 그림 3은 통합하는 절차에 해당하는 4가지 영역을 보여주고 있다.

4. 통합적 요구사항 분석 방안

3장에서 언급한 통합되어야 할 요소와 통합하는 방법을 기반으로 구체적인 통합공정을 통해 통합적 요구 사항 추출, 분석 기법을 제안한다.

그림 4를 보면, 초기에 요구 사항을 추출하고 분석하기 전에 개발할 시스템의 영역을 설정해야 한다[13]. 시스템의 영역을 설정하기 위해 비즈니스를 기술(Business Description)하고 이를 바탕으로 개체들 간의 상호작용(Interaction)을 통해 시나리오를 생성하여 요구 사항을 추출한다. 생성된 시나리오는 분류(Classification), 추상화(Abstraction)를 거쳐 시스템 목표(Intial goal)를 식별하게 되고 이를 기반으로 검증 시나리오(Validation scenario)가 시나리오 작성(scenario authoring)에 의해 만들어진다. 상호작용에서 생성된 시나리오와 시나리오 작성에서 생성된 시나리오를 서로 비교

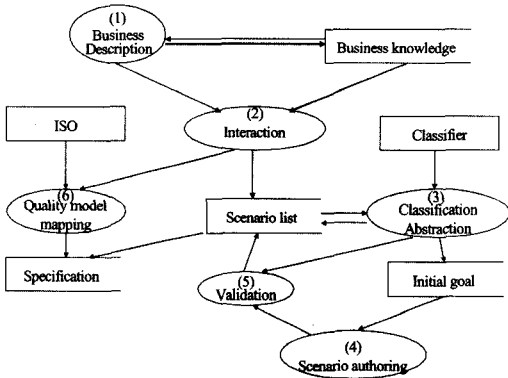


그림 4 통합적 요구 사항 분석 공정

하여 합집합을 구한 후 이를 기반으로 품질속성을 매핑한다.

4.1 비즈니스 기술

비즈니스 기술을 통하여 개발할 시스템의 영역을 결정하고, 시스템에 대한 이해를 돕는다. 비즈니스 기술은 비즈니스 내에 존재하는 에이전트, 목표, 행위, 이벤트, 객체 등의 개체를 사용하여 표현된다[5,18]. 비즈니스 내에 존재할 수 있는 에이전트, 목표, 행위, 등을 뽑아내고 개체들 간의 정적인 관계를 나타낸다. 그림 5는 비즈니스 기술 모델을 나타낸다.

비즈니스 기술 모델은 총 6개의 개체-액티비티 (activities: 목표를 성취하기 위한 단계), 객체(object: 처리의 주체이고 처리 후에는 변경), 에이전트(agents: 액티비티를 성취하는 주체), 이벤트(event: 액티비티가 일어나게 하는 행위), 후보목표(candidate goals: 요구 사항을 표현하는 것으로 에이전트에 의해서 수행), 구조 객체(structure object: 장소 정보와 물리적 환경)-들이 존재한다. 각 개체들은 '액티비티' 개체를 중심으로 관계가 설정된다.

'액티비티'는 '에이전트'가 발생시키는 '이벤트'로 활성화된다. '액티비티'내에서 '에이전트'는 '객체'를 생성 및 변경하는 일련의 작업들을 수행한다. 이것이 곧 시나리오 생성이다. '에이전트'는 각 시나리오를 '후보목표'를 기준으로 수행한다. 다시 말해, '에이전트'가 수행하는 모든 시나리오에는 '후보목표'를 달성하기 위한 일련의 작업들이다.

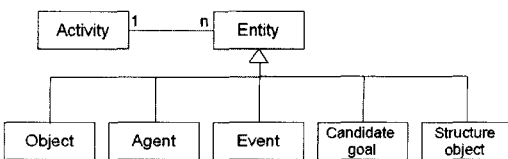
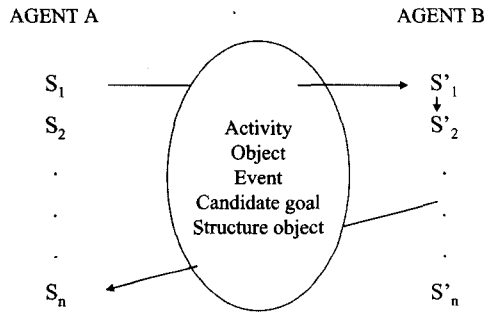


그림 5 비즈니스 기술 모델

4.2 시나리오 생성

비즈니스 기술에서 나타난 에이전트들 간의 상호작용을 통해 시나리오를 생성한다. 에이전트들 간의 상호작용은 비즈니스 기술 모델에서 나타난 여러 개체들을 참조하여 시나리오로 기술된다. 따라서 4.1과 4.2 단계를 통해 체계적인 요구 사항 추출 방법을 보여준다. 그림 6은 개체들을 참조하여 에이전트들 간의 상호작용을 통해 시나리오가 생성되는 과정을 보여준다.

생성되는 시나리오는 정상적인 시나리오와 예외적인 시나리오로 나타나게 되고 이를 바탕으로 시나리오 목록(Scenario list)을 만든다.



$$S = \{s/s \text{ is al scenario generated by Interactions between system and other agents}\}$$

그림 6 개체들을 이용한 상호작용에 의한 시나리오 생성

4.3 시나리오 분류

4.2를 통해 생성된 시나리오를 중심으로 의미적으로 유사한 시나리오들을 분류한다. 시나리오 분류 기준은 요구 사항의 의미적 유사성에 근거에 의해서 그림 7과 같이 나타난다[15].

그림 7에서 제안하고 있는 분류 의미들을 위주로 시나리오를 재작성하고 분류한다.

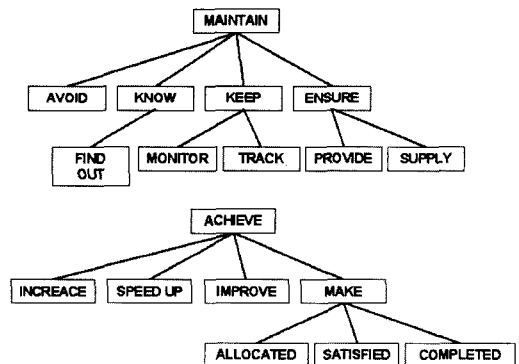


그림 7 의미적 유사성에 의한 요구 사항 분류 체계

4.4 목표 생성을 위한 시나리오 추상화

시나리오를 분류한 후 각 시나리오들 간의 관계를 규정하면서 공통된 프로세스나 역할을 중심으로 상위 목표를 생성해 나간다. 그림 8은 시나리오 추상화를 보여 준다.

시나리오 추상화에서는 유사한 부류의 목표들은 상위로 올라 갈수록 통합된 목표를 지니게 되고 하위 목표와 시나리오의 의미를 포괄하는 언어로 기술한다.

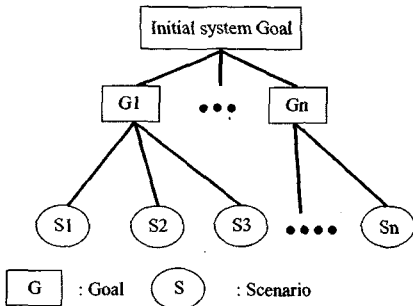


그림 8 시나리오 추상화

4.5 검증용 위한 시나리오 작성

본 단계는 4.2~4.4까지 이뤄진 시나리오 생성에 대한 검증 단계로 이뤄진다. 검증 시나리오 생성 시에 먼저 최상의 목표를 잡고 이를 정제하여 하위 목표로 나뉘는 단계를 거친다. 이때 2.3에서 소개한 기법을 이용하여 하향식으로 요구 사항을 세분화 하여 시나리오를 생성한다. 4.5에서 생성되는 시나리오를 다음과 같이 표현할 수 있다.

$$S' = \{s | s \text{ is a scenario generated in scenario authoring}\}$$

4.6 시나리오 검증 및 통합

4.2에서 생성된 시나리오와 4.5에서 작성된 시나리오를 비교 분석하는 단계이다. 4.2에서 생성된 시나리오는 개체들 간의 상호작용에 의해 생성된 것이므로 행위를 기반으로 생성되었다. 따라서 시나리오들 간의 관계에서 나타나는 요구 사항과 비 기능적인 요구 사항이 누락된다. 4.5에서 생성된 시나리오는 시나리오들 간의 관계에 의한 요구사항은 나타나지만, 시스템의 목표로부터 하향식으로 생성된 것이므로 목표를 달성하기 위한 시나리오가 완벽하지 않을 수 있다. 이를 보완하기 위해 4.2와 4.4에서 4.5에서 생성된 시나리오들의 합집합을 구한다. 그림 9는 시나리오 통합과 검증에 관한 내용을 보여준다.

시나리오 생성에서 나타난 시나리오를 S(s)라고 하고 시나리오작성(Scenario authoring)에서 생성된 시나리오

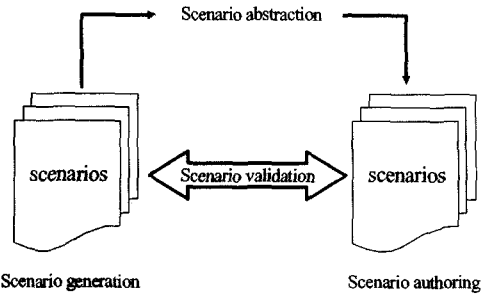


그림 9 시나리오 검증 및 통합

를 S'라고 하면 시나리오 검증 및 통합 단계를 통해 생성되는 시스템에 대한 최종 요구 사항은 SUS'로 나타난다.

4.7 품질모델 매핑

요구사항을 추출하고 분석하는 단계에서 전체 시스템에서 차지하는 요구사항 항목들을 정하고 분류하는 것은 향후 설계 및 아키텍처 생성, 구현 단계에서 개발자와 사용자 모두에게 폭 넓은 개발 방안을 제공하게 된다. 따라서 추출 분석 단계에서 품질을 고려한다면 시스템 개발 시 비용 감축을 가져올 수 있다. 이를 위해 본 논문에서는 획득된 요구 사항을 시나리오로 표현했고 시나리오들이 표현되었을 때에 나타날 수 있는 소프트웨어 품질 속성을 식별한다[16]. 식별된 품질 속성들을 해당 시나리오와 관련되어 있는지를 불리언 값(boolean value)으로 나타낸다[17]. 이를 통해 개발할 시스템에서 차지하고 있는 품질속성들의 비중을 파악하여 시스템 개발에 주의를 두어야 할 품질 속성을 제공한다. 그리고 각 시나리오 별로 품질 속성에 관련 정도를 계산하여 시나리오 기반의 요구사항들을 품질 측면에서 상대적인 비율을 제공한다.

5. 사례연구

단원 4에서 제안한 기법을 적용하기 위해 본 논문에서는 MRS(Meeting Reservation System)을 예로 든다. MRS는 방과 장비 등의 자원을 할당하기 위한 시스템이다. 자원은 조직에 의해서 바뀐다. 미팅 예약자가 MRS를 사용하여 예약을 하면, MRS는 미팅 참가자에게 이를 이메일(email)을 통해 알려주고 미팅 참가자는 다시 참석 여부를 MRS에 알려준다. 미팅참가자에게 참석여부를 받은 MRS는 해당 예약을 확정하여 미팅 예약자에게 보낸다. MRS에서 자원 할당은 서버 에서 동기적으로 일어난다.

5.1 MRS를 위한 비즈니스 기술

비즈니스 기술에서는 개발할 시스템의 영역을 설정한다. 이를 위해 시스템의 영역(비즈니스)을 몇 개의 개체

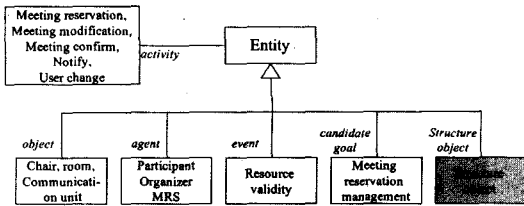


그림 10 MRS에 적용된 비즈니스 기술 모델

로 표현한다. 그림 10은 MRS에서 나타나는 시스템의 영역을 비즈니스 기술 모델로 표현한 것이다.

MRS에서는 후보목표는 미팅 예약 관리(Meeting reservatin management)으로, 액티비티는 미팅 예약(Meeting Reservation), 미팅 변경(Meeting modification), 미팅확정(Meeting confirm), 공지(Notify), 사용자 변경(User change)으로, 에이전트는 참가자(Participant), 예약자(Organizer), MRS으로, 객체는 의자(chair), 방(room), 통신수단(communication unit), 이벤트는 리소스 유효(Resource validity)으로 나타난다. 구조객체는 나타나지 않는다.

5.2 MRS를 위한 시나리오 생성

비즈니스 기술에서 나타난 개체들을 바탕으로 시나리오가 생성된다. 비즈니스 기술 모델에는 에이전트가 참석자(participant), 예약자(organizer), MRS가 나타났다. 개발할 시스템은 MRS이므로 MRS를 중심으로 각 에이전트들 간의 상호작용을 기술한다. 이때 비즈니스 기술 모델에 나타났던 액티비티, 후보목표, 이벤트 등을 참조하여 기술하게 된다. 그림 11은 에이전트들 간의 시나리오가 생성되는 것을 보여주고 있다.

시스템의 요구사항은 MRS에서 발생하는 시나리오들

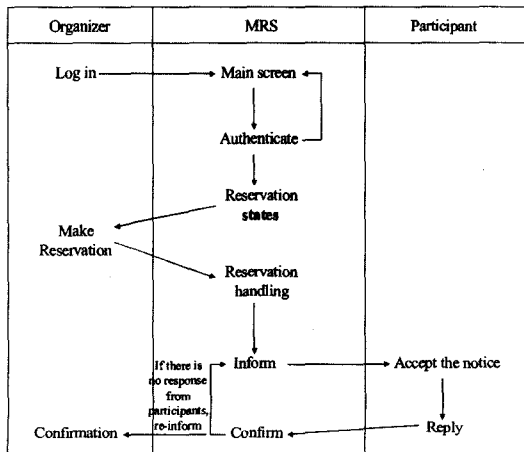


그림 11 예약자, 참석자, MRS간의 상호작용에 의한 시나리오 생성

을 나열하는 것이기 때문에 MRS 위주의 관찰이 필요하다. 미팅예약자는 로그인을 하고, 이때 MRS의 상태는 “1. Main screen”이 된다. MRS는 사용자의 로그인을 “2. authenticate”를 하고 인증이 거부되면 “2.1 Authentication error handling”가 되어 1번 상태로 돌아가게 된다. 인증을 한 후에 “3. Reservation states”를 보여주고, 사용자가 예약 활동을 하면 MRS는 “4. Reservation handling”을 하고 미팅참가자들에게 미팅을 “5. Inform”한다. 미팅참가자들은 공지를 수신하고 MRS에게 응답을 한다. 응답을 받은 MRS는 미팅을 “6. Confirm”하고 미팅 예약자에게 완료됨을 알린다. 만일 미팅 참가자에서부터 응답이 오지 않으면 “6.1 re-inform”를 한다. 단원 4에서 제안한 방법을 적용하다 보면 비즈니스 기술 모델에 나타나지 않았던 개체들이 획득되는 경우가 발생한다. MRS의 경우에 해당 비즈니스의 정책을 반영하는 작업이 필요하다. 예를 들면, 자원이 변경되어 MRS에 반영을 해야 하는 경우가 있다. 이를 위해 시스템 관리자라는 에이전트가 필요하다. 따라서 비즈니스 기술 모델에서 에이전트에 관리자가 추가된다. 추가된 에이전트와 MRS간의 상호작용을 기반으로 시나리오를 재생성 한다. 그림 12는 시스템 관리자와 MRS간의 상호작용을 나열하여 시나리오를 생성하는 것이다.

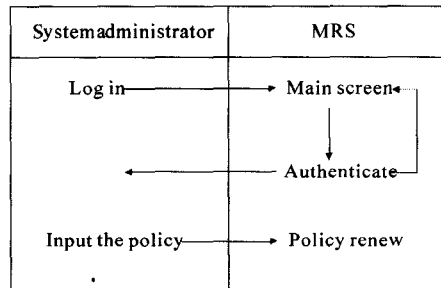


그림 12 관리자과 MRS간의 상호작용에 의한 시나리오 생성

시스템 관리자는 로그인을 하고, 이때 MRS의 상태는 “1. Main screen”이 된다. MRS는 시스템 관리자의 로그인을 “2. Authenticate”한다. 시스템 관리자는 정책을 입력하고 MRS는 “7. Policy renew”을 한다. 이러한 과정을 통해 표 2와 같은 정상 시나리오 목록과 표 3과 같은 예외적인 시나리오 목록을 만들 수 있다.

이렇게 7개의 정상 시나리오와 2개의 예외 시나리오가 생성되어, 총 9개의 MRS에 관한 시나리오가 생성되었다.

표 2 정상 시나리오

#	scenario
1	Main Screen
2	Authenticate
3	Show the Reservation condition to user
4	Reservation condition handling
5	Inform
6	Confirm
7	Policy renew

표 3 예외 시나리오

#	scenario
2.1	Authentication error handling
6.1	re-inform

5.3 MRS를 위한 시나리오 분류

5.2에서 이뤄진 요구사항을 추상화하기 위해 먼저 시나리오를 그림 7을 기준으로 분류한다. 표 4는 MRS에 적용된 시나리오 분류이다.

표 4 MRS에 적용된 시나리오 분류

Completed	
Transaction	Scenario 2.1: Authentication error handling MRS rejects and return to initial state (scenario 1) if user ID and password doesn't match
	Scenario 4: Reservation condition handling MRS handles organizer's reservation.
	Scenario 6: Confirm MRS inform to user that a reservation user make is complete
	Scenario 7: Policy renews MRS renews business policy
Monitor	
Interaction with user	Scenario 1: Main screen MRS is waiting for user input and displaying front page
Find out	
Validation check	Scenario 2: Authenticate MRS authenticates user
Provide	
Provide object	Scenario 3: Show the Reservation condition to user - MRS provides reservation information
	Scenario 5: Inform - MRS inform to organizer about new or modified reservation
	Scenario 6.1: re-inform - If there is no response from participants, MRS re-informs participants.

그림 7에서 제시한 요구사항 분류체계를 기반으로 앞 단계에서 추출된 시나리오를 공통된 속성으로 묶어 'transaction', 'interaction with user', 'validation check', 'provide objects'로 추상화되어 상위 목표가 된다.

5.4 MRS를 위한 목표 생성을 위한 시나리오 추상화

시나리오 추상화 단계를 통해 생성된 시나리오들 간의 관계를 설정하여, 최상위 목표를 도출하게 된다. 각 시나리오가 추상화 될 때에는 분류된 결과를 바탕으로 추상화가 이뤄지며, 추상화는 전체 요구 사항을 포함하는 목표가 생성될 때까지 이뤄진다. 그림 13은 MRS에 적용된 목표생성을 위한 시나리오 추상화의 결과를 보여주고 있다.

이러한 시나리오 추상화를 통해 "Provide meeting reservation management service with user"라는 최상위 단계의 목표를 도출해낸다.

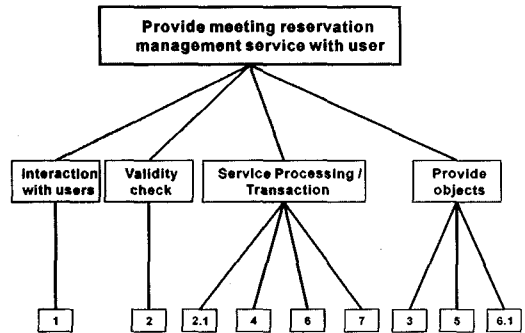


그림 13 MRS에 적용된 시나리오 추상화

5.5 MRS를 위한 검증 시나리오 생성 및 통합

앞 단계에서 이뤄진 모든 작업을 검증하기 위한 검증 시나리오를 생성한다. 그림 14에서 볼 수 있듯이, 검증 시나리오는 시스템의 목표(G1)인 "Provide meeting reservation management service with user"으로부터 시작된다. 최상위 목표(G1)인 "Provide meeting reservation management service with user"를 달성하기 위해 "MRS authenticates the user"(Sc1.1), "MRS manages policies"(Sc1.2), "MRS manages reservations"(Sc1.3)등의 시나리오가 필요하고 이러한 시나리오들은 하위 단계에서의 하위 목표(G2.1, G2.2, G2.3)가 된다. 이러한 방법으로 각 단계에서의 목표를 성취하기 위한 시나리오를 기술하고 이 시나리오들을 다음 단계의 목표가 되게 하여 목표를 추출한다. 또한 본 단계에서 시나리오들 간의 관계가 더욱 명확하게 나타난다. 예를 들어 "Waiting for user input"(G3.1)은 "User authentication"(G3.2)보다 항상 먼저 성취되어야 하는 목표가 된다. 따라서 G3.1은 G3.2에 대하여 "before" 관계를 가지게 되고 반면에 G3.2는 G3.1에 대하여 "after" 관계를 가지게 된다. 그림 14는 시스템 목표(G1)로부터 시나리오가 생성(작성)되는 것을 보여주고, 표 5는 검증을 위한 시나리오 목록을 나타낸다.

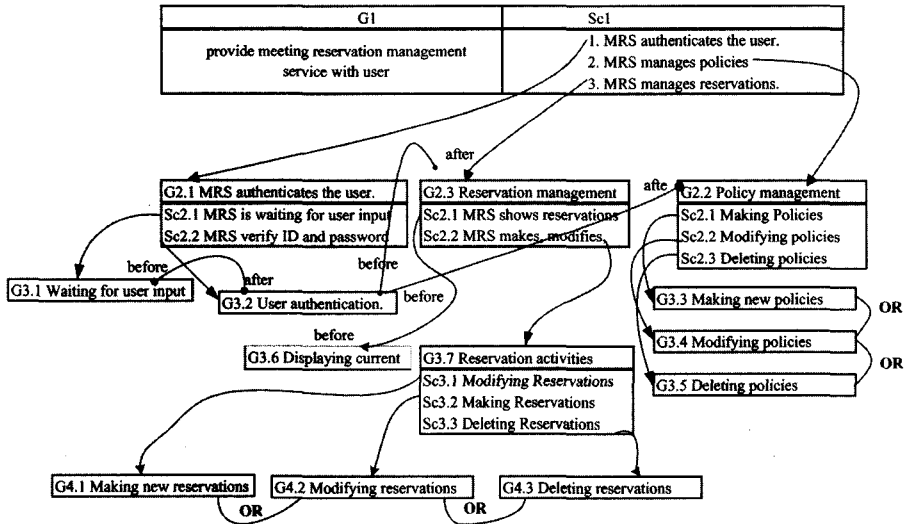


그림 14 목표로부터 검증 시나리오 생성

표 5 검증 시나리오 목록

Sc ID	Scenarios for validation
A	Waiting for user input
B	User authentication
C	Displaying current reservation status
D	Making new policies
E	Modifying policies
F	Deleting policies
G	Modifying reservation
H	Making new reservation
I	Deleting reservation

표 2, 표 3에서 나타난 시나리오와 표 5에서 나타난 시나리오를 비교 검토를 하여(표 6), 각 표에 나타나는 시나리오를 통합한다(표 7).

표 6 시나리오 집합 비교

Scenarios authored in top down approach	Scenarios generated
A. Waiting for user input	1. MRS is waiting for user input and
B. User authentication	2. MRS authenticates user.
C. Displaying current reservation status	2.1 MRS rejects and return to initial
D. Making new policies	3. MRS provides reservation information.
E. Modifying policies	4. MRS handles organizer's reservation.
F. Deleting policies	5. MRS inform to organizer about new or
G. Modifying reservations	6. MRS inform to user that a reservation
H. Making new reservation	6.1 If there is no response from
I. Deleting reservation	7. MRS renews business policy.

표 6에서 볼 수 있듯이 시나리오 생성(단원 5.2)에서 획득된 시나리오(표 6에서 우측)와 검증 시나리오 작성(단원 5.5)에서 획득된 시나리오(표 6에서 좌측)를 보면, Sc7의 경우에 시나리오 검증 단계에서 ScD, ScE, ScF로 세분화되었음을 알 수가 있다. 그리고 Sc2.1, Sc5,

표 7 정제된 시나리오 목록

Sc ID	Scenarios
1	MRS is waiting for user (organizers or participants) input and displaying main screen
2	MRS authenticates the user
2.1	MRS rejects and returns to initial state (scenario 1) if user's ID and password do not match
3	MRS displays reservation information
4-1	The organizer modifies reservation
4-2	The organizer make new reservation
4-3	The organizer deletes reservation
5	MRS serves a notice to participants about new or modified reservation
6	MRS serves a notice to the organizer that a reservation user make is completed
6-1	MRS serves a notice to participants with another media again if participants doesn't reply
7-1	Making new policies
7-2	Modifying policies
7-3	Deleting policies

Sc6, Sc6.1은 시나리오 검증 단계에서는 나타나지 않은 시나리오임을 알 수가 있다. 따라서 시나리오 생성 단계에서 나타난 시나리오 목록과 시나리오 검증 단계에서 나타난 시나리오의 합을 이루면 MRS에 대한 요구 사항이 정제된 시나리오 집합으로 나타난다.

Sc1, Sc2, Sc3, Sc5, Sc6은 정상 시나리오이며, Sc4-1, Sc4-2, Sc4-3, Sc6-1, Sc7-1, Sc7-2, Sc7-3은 Sc4, Sc6, Sc7에서 세분화 된 시나리오이고, Sc2.1은 Sc2에 대한 예외 시나리오가 된다.

5.6 MRS를 위한 품질 모델 매핑

소프트웨어 품질을 관리하기 위해 요구사항 분석 단계에서 미리 해당 요구 사항의 품질이 어떤지를 고려한다면(semi-captured), 향후 시스템을 구현할 때 어떤 요구사항이 반드시 구현되어야 하는지에 대해 기준을 제공하게 된다. 본 논문에서는 이를 위해 생성된 시나리오를 소프트웨어 품질 모델[16]에 비추어 각 시나리오 별로 어떤 품질이 나타나고 있는지를 매트릭스로 표시하였고-시나리오가 잠재적으로 갖고 있는 품질 모델 속성에 대해 개발자, 사용자가 모여서 정한다-각 속성별로 산출되는 상대적 값을 계산하는 식을 제공한다. MRS의 최종 시나리오를 품질 모델에 매핑 시킨 것은 그림 15와 같다.

#	functionality	reliability	usability	efficiency	maintainability	portability	relative cost
1	1		1				5.88
2	1						5.54
2.1	1		1				5.88
3			1				0.35
4-1	1	1					6.92
4-2	1	1					6.92
4-3	1	1					6.92
5	1					1	5.69
6	1						5.54
6.1	1					1	5.69
7-1	1	1			1		7.27
7-2	1	1			1		7.27
7-3	1	1			1		7.27
Count	12	6	3	0	3	2	26
ratio	46%	23%	12%	0%	12%	8%	100%

그림 15 품질속성 매핑 매트릭스

그림 15에서 좌측 열은 시나리오 번호(Sc ID)를 기술하고, 상단 행은 ISO 9126 [16]에서 제안하는 소프트웨어 품질 속성을 나열한다. 각 시나리오는, 분석가와 이해당사자들의 합의로, 소프트웨어 품질속성의 정의를 바탕으로 관련된 품질속성에 대해 불리언 값(boolean value)을 기입한다. 예를 들어 Sc1-“MRS is waiting for user (organizers or participants) input and displaying main screen”-의 경우에, Sc1은 기능성(functionality)과 사용 편의성(usability) 속성과 관련이 있고 따라서 Sc1은 기능성과 사용 편의성의 빈칸에 불리언 값 '1'을 기입한다. 이와 같은 방식으로 Sc2부터 Sc7-3까지 작업을 한다. 각 시나리오에 대한 품질속성 값 기입을 마치면, 품질속성 별로 시스템에서 나타나는 개수와 비율을 계산한다. 예를 들어 기능성은 Sc1, Sc2, Sc4-1, Sc4-2, Sc4-3, Sc5, Sc6, Sc6.1, Sc7-1, Sc7-2, Sc7-3에서 나타나므로 총 12개가 나타난다. 이와 같은 방식으로 신뢰성(reliability)은 6개, 사용 편의성은 3개, 효율성(efficiency)은 0개, 유지 용이성(maintainability)은 3개, 호환성(portability)은 2개가 나타난다. 따라서 MRS에서 나타나는 모든 시나리오에 나타나는 품질속성의 개수는 총 26개가 된다. 그러므로 기능성은 46%(=12/26), 신뢰성은 23%(=6/26), 사용 편의성은 12%(=

3/26), 효율성은 0%(=0/26), 유지 용이성은 12%(=3/26), 호환성은 8%(=2/26)로 MRS에서 차지하는 비율이 나타난다. 이제 각 시나리오 별로 품질 면에서 MRS에서 차지하는 상대적인 값을 계산해본다. 시나리오 Sc_i에 대해 Sc_i의 상대값(RC: relative cost)는 다음과 같다.

i : 시나리오의 개수

j : 품질 속성 종류

Sc: 시나리오

RC: 상대 중요도

$$RC(Sc_i) = \sum_j ((\text{the number of quality } j \text{ in } Sc_i) \times (\text{the rate of quality } j \text{ in } Sc_i))$$

그러므로 시나리오 1의 상대값, RC(Sc1) = 12*46% + 3*12% = 5.88을 갖게 된다. 시나리오 1의 상대적인 값이 5.88이라는 것은 RC(Sc2)=5.54를 갖는 시나리오 2에 비해 시스템에서 차지하는 품질 비중에 높다는 것을 의미한다. 따라서 소프트웨어 시스템을 개발할 때 높은 값을 갖는 시나리오에 대해 주의 깊게 개발할 필요가 있음을 알려준다.

5.7 MRS를 위한 최종 요구 사항 기술

MRS 예제에 사용된 프로세스 결과를 나타내는 최종 요구 사항을 표 8 형식으로 기술할 수가 있다. 예를 들어 시나리오 3(Sc3)에 대해, 표 8에서 번호에는 해당 시나리오의 번호인 3을 기입하고 이름에는 시나리오의 내용인 “MRS displays reservation information”을 작성한다. 종류에는 시나리오 3이 시나리오 분류체계에서 속하는 요소인 ‘Provide/Ensure/Maintain’을 작성한다. 관련 목표에는 시나리오를 추상화 시켜 생성한 ‘Provide objects’를 기입한다. 최종 목표는 해당 시나리오가 최종적으로 성취하고자 하는 시스템 목표인 “provide meeting reservation management service with user”를 작성한다. 중요도는 품질 모델 매핑의 결과로 나온 시나리오의 중요도인 0.35와 선택된 품질 모델인 ‘usability’를 기입한다. 마지막으로 관계에는 3번 시나리오가 4-1번 시나리오 앞에 나타나야 한다는 시나리오간의 관계를 기입한다.

표 8 요구사항기술

Items	Contents
Number	3
Name	MRS displays reservation information
Class	Provide/Ensure/Maintain
Related goal(s)	Provide objects
Initial goal	Providing meeting reservation management service to user
RC	0.35(usability)
Relationship	Before(4-1)

6. 실험

본 논문에서 제안한 통합적 요구 사항 분석 기법의 효율성과 용이성을 검증하기 위해, 30명의 시스템 개발 관련에 종사하는, 5~10년 된 실무 경력자들을 대상으로, 2개의 서로 다른 예제에 각각 시나리오 기반 분석 방법, 목표 기반 분석 방법, 통합적 분석 방법을 적용하였다. 적용된 예제는 본 논문에서 소개한 MRS와 전자입찰시스템(EBS: Electronic Bid System)을 적용했다. 전자입찰시스템은 회사의 사무용품을 웹 환경에서 입찰을 받아 낙찰 프로세스를 거쳐 자동으로 회사의 정책을 반영한 입찰자를 선정하는 시스템이다. 실험을 위해 30명을 5명씩 6개의 조로 나눴고, 6개의 조는 다시 A파트와 B파트로 나눴다. A파트의 경우에는 MRS예제에 시나리오 기반의 분석, 목표 기반의 분석, 본 논문에서 제안하는 분석을 적용했고, B파트의 경우에는 EBS예제에 A파트와 동일 방식으로 적용했다. 실험 결과는 크게 2가지로 얼마나 정확하게 요구 사항을 획득하는 가에 대한 완전성(completeness)과 얼마의 시간이 소요되는 지에 대한 시간비용(time cost)으로 나눴다. 완전성은 이미 요구 사항이 분석된 문서를 기준으로 하여 제안된 기법을 통해 나온 분석물이 얼마나 완전하게 분석되는지를 비교 검토하였다. MRS에 대해 기준이 되는 분석서는 OMG(Object Management Group)에서 제공하는 문서 [19]로 삼았고 EBS에 대해서는 Rational의 자료를 기준으로 삼아서 비교하였다. 요구 사항 분석의 완전성은 두 예제의 평균값을 구해 기술하였다. 시간 비용의 측정 은 5명씩 이뤄진 조별로 요구 사항 분석을 시작할 때와 마쳤을 때를 기준으로 시간을 명시하도록 하여 가장 오래 걸린 통합적 분석 기법을 100으로 놓고 다른 기법들을 사용할 때에 걸린 시간을 상대적인 값으로 기술하였다. 그림 16, 17은 각 기법들을 적용한 결과로 요구 사항 완전성과 시간비용을 보여준다.

완전성 측면에서는 예제에 상관없이 시나리오 기반의 분석이 30%, 목표 기반의 분석이 40%, 통합적인 분석

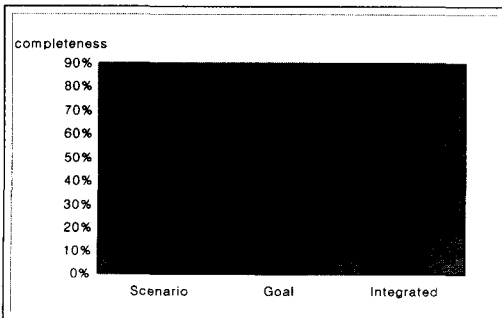


그림 16 완전성

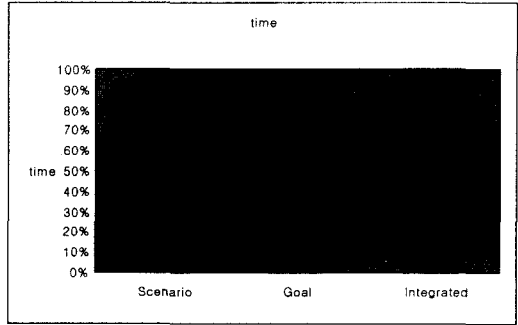


그림 17 시간 비용

이 86%의 완전성을 보여줬다. 반면에 시간 비용에서는 통합적 분석을 100으로 볼 때에 시나리오 기반의 분석이 38, 목표 기반의 분석이 50을 나타냈다. 따라서 요구 사항을 얼마나 정확하게 획득하고 분석하는 가에서는 제안된 통합적 요구 사항 분석 방안이 우수한 것으로 판명되었으나, 기법을 적용하는데 있어서는 상대적으로 많은 시간이 필요함을 알 수 있었다.

7. 결론

소프트웨어 개발에서 요구 사항을 추출하고 분석하는 것에 대한 중요성이 날로 높아지면서 이를 지원하는 요구 사항 분석 기술이 필요하게 되었다. 대표적인 기법들은 시나리오 기반의 분석, 목표 기반의 분석, 시나리오를 이용한 목표 기반의 분석 등이 있다. 각 기법은 장, 단점을 가지고 있어, 만일 하나의 기법만을 사용하게 되면, 정확한 요구 사항의 획득이 쉽지 않다. 따라서 소프트웨어 개발 시에 누수 되는 요구사항이 발생하게 되고 이는 후에 많은 비용을 유발한다. 따라서 본 논문에서는 각 기법들의 장, 단점을 보완하는 통합적인 요구 사항 분석 방안을 제안했다. 통합적인 요구 사항 분석 방안을 설명하기 위해 어떤 요소들이 통합되어야 하는지와 그 요소들이 어떻게 통합될 수 있는지에 대해 제안했다. 또한 요구 사항 획득, 분석 단계에서 간략하게(semi-captured) 소프트웨어 품질 속성을 고려한 방안을 제시했다. 제안된 방안에 대한 검증 방법으로 소프트웨어 개발을 위한 분석 경험이 5~10년이 되는 30명의 경험자에게 본 기법을 교육 후에 2개의 예제에 대해 적용해보았다. 그 결과로 본 논문에서 제안한 기법이 요구 사항을 정확하게 획득하는데 효과가 나타났으나, 적용하는데 다른 기법에 비해 많은 시간이 필요했다.

향후 연구 과제로는 본 논문에서 나타나는 검증을 위한 시나리오 작성 시에 시나리오를 이용한 목표 기반 분석 기법을 이용하므로 목표들의 정제 완료가 어디서 일어나야하는 지에 대한 명확한 방안을 제시해야 하며,

품질 속성으로 요구 사항의 중요도를 결정하기 위한 가중치 계산 방법이 추가 되어야 할 필요가 있다. 또한, 본 논문에서 제안된 기법을 적용하는데 있어서 시간을 절약하기 위한 기법의 간소화에 대한 연구가 필요하다. 제안된 방안의 활성화를 위해 유스케이스 분석 방안의 연계성에 관해서도 연구가 진행 중이다.

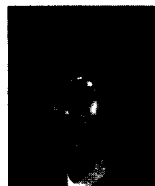
참 고 문 헌

- [1] Richard H.Thayer, Merlin Dorfman, Software Requirements Engineering, 2nd Edition IEEE Computer Society Press, 1997.
- [2] Daniel Gross, Eric Yu, From Non-Functional Requirements to Design through Patterns, Requirements Engineering, Springer-Verlag, 2001, pp. 18-36.
- [3] Steve McConnell, Code Complete: A Practical Handbook of Software Construction, Microsoft Press, 1993.
- [4] Ivar Jacobson, "Four Macro Trend in Software Development," Korea Conference of Software Engineering, 2002, pp. 1-14.
- [5] Sutcliffe, A, "Scenario based requirement analysis," Requirements Engineering Journal, 1998, pp. 48-65.
- [6] V.Plihon, J.Ralyte, A.Benjamin, N.A.M. Maiden, A. Sutcliffe, E.Dubois, and P.Heymans, "A Reuse-Oriented Approach for the Construction of Scenario Based methods," Proc. Int'l Software process Assoc. Fifth Int'l Conf. Software Process(ICSP '98), Chicago, June 1998, pp. 14-17.
- [7] Annie I. Anton, "Goal-Based Requirements Analysis," Proceedings of ICRE 96, pp. 136-144.
- [8] Anne Dardenne, Axel van Lamsweerde, Stephen Fickas, "Goal-Directed Requirements Acquisition," Science of Computer Programming 20(1-2), 1993, pp. 3-50.
- [9] C. Rolland, C. Souveyet, C. Ben Achour, "Guiding Goal Modelling using Scenarios," IEEE Transactions on Software Engineering, Special Issue on Scenario Management, Vol. 24, No. 12, Dec. 1998. pp 1055-1071.
- [10] E. Yu and J. Mylopoulos, "Why Goal-Oriented Requirements Engineering," 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.
- [11] Ian Sommerville, Software Engineering 6th Edition, Addison-Wesley, 2001, pp 137-139, pp194-211.
- [12] Colette Rolland, Georges Grosz, Regis Kla, "Experience With Goal-Scenario Coupling In Requirements Engineering," Requirements Engineering, 1999. Proceedings. IEEE International Symposium on , 1999 Page(s): 74-81.
- [13] Gerald Kotonya, Ian Sommerville, Requirements Engineering processes and techniques, Wiley, 1997, pp. 139-14.
- [14] Alan M. Davis, Software Requirements-Analysis and specification, Prentice Hall, 1990, pp. 46-51.
- [15] Annie I. Antn, Goal Identification and Refinement in the Specification of Software-Based Information Systems, Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, USA, June 1997.
- [16] International Organization for Standards, ISO/IEC 9126-1 Information Technology; Software Product Quality, International Organization for Standards, 2002.
- [17] Paul Clements, Rick Kazman, Mrak Klein, Evaluating Software Architectures : Methods and Case Studies, Addison-Wesley, 2002, pp 211-239.
- [18] Maiden N.A.M., Sutcliffe A.G., Requirements critiquing using domain abstractions. In: Proceedings First International Conference on Requirements Engineering, J. Siddiqi (ed.) IEEE Computer Society Press, 1994, 184-193.
- [19] OMG, A UML Profile for Enterprise Distributed Object Computing Specification, 2002-08, pp B-1~B-3.
- [20] B.WBoehm, "Software Engineering Economics," Prentice Hall, 1981.



김진태

1998년 서강대학교 컴퓨터학 공학사. 2000년 서강대학교 컴퓨터학 공학석사. 2001년 테크놀로지. 2002년~현재 서강대학교 컴퓨터학과 박사과정. 관심분야는 Software Variability, Ubiquitous Computing, Product Line



김동선

2003년 서강대학교 컴퓨터학 공학사. 2003년~현재 서강대학교 컴퓨터학과 석사과정. 관심분야는 Formal Methods, Embedded Software, Patterns

박수용

정보과학회 논문지 : 소프트웨어 및 응용 제 31 권 제 3 호 참조