

# 유비쿼터스 서비스 시나리오 생성 및 시스템 분석 시스템 개발

이연남

경희대학교 유비쿼터스 서비스&비즈니스 연구센터  
(yonnim@khu.ac.kr)

정동영

경희대학교 유비쿼터스 서비스&비즈니스 연구센터  
(dyjung@khu.ac.kr)

권오병

경희대학교 국제경영학부  
(obkwon@khu.ac.kr)

.....

실제 물리적 공간에 유비쿼터스 서비스를 개발하여 실현하는 작업이 유비쿼터스 지능 공간 구축을 위해 진행 중에 있다. 또한 이와 같은 유비쿼터스 지능 공간 개발은 전 세계적으로 다양한 분야에서 이루어지고 있다. 이러한 유비쿼터스 지능 공간을 개발하기 위한 방법론 중에서 먼저 시나리오를 제작 분석하고 이에 따라 요구분석과 설계를 하기도 하며 역으로 개발에 고려하고 있는 유비쿼터스 기술로 먼저 설계나 요구분석을 실시하고 이후에 시나리오를 통하여 실현 가능성을 보이기도 한다. 그러나 시나리오 제작과 요구분석이 상호 유사한 내용을 공유하고 있음에도 불구하고 별도로 제작하는 과정에서 불일치와 많은 노력이 소모된다. 따라서 본 논문의 목적은 시나리오 제작과 요구분석 문서를 생성하는 시스템을 제안하는 것이다.

.....

논문접수일 : 2008년 10월    논문수정일 : 2008년 11월    게재확정일 : 2008년 12월    교신저자 : 권오병

## 1. 서론

유비쿼터스 서비스 개념의 확장으로 최근 서비스 제공 공간인 유비쿼터스 지능 공간(USS : Ubiquitous Smart Space)에 대한 관심이 높아지고 있다. 여기서 USS란 기존의 지능 공간에 상황인식, 증강현실, 유비쿼터스 센서 네트워크 등 유비쿼터스 컴퓨팅 기술이 추가되어 수요자 중심의 사용자 인터페이스가 보장되는 증강된 공간으로 이해할 수 있다. Abowd(1998)는 이를 ‘매일의 경험 획득, 정보에의 접근, 의사소통과 협업 지원, 자연스러운 인터페이스, 훈련 혹은 학습 등을 시스템적으로 제

공 혹은 도움을 주는 공간’으로 정의하기도 했다. 지금까지 개발된 많은 USS는 다양한 유비쿼터스 컴퓨팅 관련 요소, 원천 기술 및 제품들이 등장하는 실현 가능한 서비스 시나리오들을 바탕으로 제안되고 있다(김성후 등, 2006). 개발 초기 단계에서 프로젝트의 비전 및 내용 혹은 미래의 생활상을 보여주는 서비스 시나리오를 작성하고, 이를 통해 개발 내용에 대한 요구분석을 수행한 후 USS를 설계하고 구현하고 있다. 이렇게 시나리오 방법을 자주 활용하는 이유는 USS가 기존의 물리공간과는 완전히 다른 새로운 환경으로 그 서비스의 양상은 기존 서비스의 내용과는 상당히 다르게 전

\* 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스컴퓨팅 및 네트워크 원천기반기술개발사업의 08B3-S1-10M 과제와 경희대학교 대학원의 2008학년도 1차 우수연구논문 장학금으로 지원받아 연구된 것임.

개될 것이기 때문에, 이러한 새로운 USS에 대한 체험을 개념화 하는 미래 서비스를 개발함에 있어 미래연구 방법으로 가장 널리 쓰이고 있는 시나리오 분석 방법론이 효과적인 도구가 되기 때문이다. 실제 USS의 서비스 아이디어 제안을 위한 방법이자 도구로서 시나리오는 매우 유망한 방법으로 증명되고 있다(장종인, 2007). 또한 시나리오는 새로운 아이디어를 다른 개발자에게 설명하거나 시스템이 구현되기 위해 요구되는 사항을 포착하는 것과 다양한 목적에 적용될 수 있어, 실제 유비쿼터스 컴퓨팅 환경 하의 서비스 개발을 위한 방법으로써 시나리오의 개발이 활발하게 사용되기도 한다(Annie et al., 1998).

하지만 지금까지는 USS 시나리오를 작성하는 것에 대한 표준화된 방법론을 따르는 시스템이 존재하지 않았다. 이로 인해 현재 대부분의 USS 시나리오는 요구분석과 관련이 적은 사람들에 의해 직접 작성되고 있는 실정이며, 실제 지극히 개인적 혹은 작성 주체가 속한 조직의 성향에 의해 그 내용이 결정되고 있다. 또한 시나리오가 구체적인 시스템 개발을 위한 요구분석과 명료하게 연결되지 않고 있다. 이러한 주된 이유는 요구분석은 시스템 분석 및 개발팀이 시나리오 작성은 시스템 기획팀이나 경영팀이 주로 사용해 온 기법이기 때문에 그 동안 별개로 진행되어 왔기 때문이다. 그러나 유비쿼터스 지능공간 프로젝트와 같은 미래적이고 수요와 개발자가 함께 구축해 나가야 하는 경우에는 이 두 부분이 통합되어야 하는데 아직은 통합 방법론에 소개되지 못했다. 이는 시나리오 작성주체와 요구분석 주체 사이의 커뮤니케이션 문제를 발생시킬 뿐만 아니라, 실제 요구분석 작업을 시나리오 작성 작업에 들였던 노력과 관계없이 수동적으로 완전히 처음부터 다시 진행하게끔 하고 있다. 결국 최종적으로 요구분석에 대한 고려 없이

일방적이며 주관적으로 작성된 시나리오로 인해 시나리오 및 요구분석서 간 이중 작성, 내용 불일치, 시간 및 비용 낭비 등의 문제가 발생되고 있는 것이다.

더욱이 이와 같은 시나리오는 시나리오에 대한 객관적인 분석 및 평가, 요구분석의 시나리오 반영도 검증 등의 작업 또한 쉽지 않게 한다. 주관적인 시나리오를 평가할 객관적인 기준을 마련하는 것도 쉽지 않을뿐더러 시나리오와 별개의 작업으로 작성된 요구분석서에 대해 시나리오의 반영도를 검증하는 것은 의미가 없기 때문이다.

이에 본 논문에서는 USS 개발 효율성 증대를 위해 시나리오 제작과 요구분석에 이르는 과정을 구조적으로 통합하는 방법론을 제시한다. 시나리오 작성 및 요구분석을 수행하기에 앞서 시나리오의 구성요소 및 그들의 연관관계를 데이터의 흐름으로 연결한 망구조의 모델링 도구인 시나리오 기술도를 먼저 작성하고 이를 기반으로 두 작업을 동시에 진행하는 것이다. 이는 시나리오 제작과 요구분석 두 작업 내 일관성을 보장하고 불필요한 중복적 노력을 감소시키는데 기여할 것이다. 또한 제안 방법론을 적용하여 개발한 케이스 도구는 S/W 생산성 및 효율성을 높일 수 있을 것이다. 특별히 본 연구는 시나리오 기반의 S/W공학 분야 내 시나리오 기반 요구분석 활동 중 발생하는 이슈에 초점을 두고 있으며 본 논문에서 연구하는 시나리오는 단기적 관점에서 개발될 기술의 실용성 증명을 위한 시나리오인 기술 주도 방식(Technical-push approach)의 시나리오에 한정하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 기존 문헌과 자료들을 통하여 기존의 시나리오 작성 방법론 및 시나리오 기반 요구분석 방법론을 정리한다. 제 3장에서는 USS 개발을 위한 시나리오 구축 및 요구분석을 위한 구조적 통합 방법론에 대

해 구체적으로 기술한다. 제 4장에서는 현재까지 진행된 제안된 방법론의 프로토타입 시스템을 보이도록 하며, 마지막으로 제 5장에서는 연구의 결론 및 공헌, 그리고 향후 연구방향 등을 논의한다.

## 2. 문헌 연구

### 2.1 시나리오 작성 방법론

최근 논의되고 있는 요구분석을 위한 시나리오 작성 방법론에는 자연어 접근법(Natural language approach), 상태도 표현기법(State chart representation), 계층적 도면 표시법(Hierarchical graphical notation), 유즈케이스 맵(Use case maps), 시개념 오토마타(Timed automata) 등이 있다.

첫째로, 자연어 접근법은 기능적 요구사항을 도출하고 분석하기 위해 UML 표준의 한 요소인 유즈케이스를 사용한다(Amyot et al., 2000; Jacobson et al., 1992). 유즈케이스는 시나리오로 분해되며 자연어를 사용하여 기술되며 초기 요구사항 도출 단계에서 유효한 것으로 알려져 있다. 또한 자연어 사용 및 최종 사용자와의 상호작용으로 도출 및 검증 작업을 간단하게 하며, 그 내용을 읽고 이해하기가 쉽다는 장점이 있다. 하지만 이 방법은 적시성, 사용 리소스, 분석가 및 디자이너의 상태 중심 행동과 같은 시스템의 상세한 내용은 제공하지 않는다. 게다가 Glintz(2000)는 지속적인 상태에 대한 모델링 없이 유즈케이스만으로 기능적 요구사항을 제공하는 것은 불충분하다고 주장하고 있다. 이와 같이 자연어 기반 프로세스는 비즈니스 프로세스를 정의하기에는 좋으나 시스템의 요구 분석을 위해서는 미흡한 점이 있다.

둘째로, 상태도 표현방법은 Glintz가 제안한 상태도를 사용하여 시나리오를 표현하는 접근법이

다. 이 접근법은 요약된 단일 시나리오뿐만 아니라 혼합된 시나리오를 표현할 수 있는 수단을 제공한다. 단일 시나리오는 구조화된 텍스트와 상태도로 그려질 수 있다. 이때 구조화된 텍스트는 순환, 반복, 사용자 입력과 시스템 반응 사이의 구별 등을 제공한다. 이는 시스템 속성을 분석할 수 있는 구조에서 시나리오에 대한 계층적이며 상세한 수준의 관점을 준다.

셋째로, 계층적 도면 표시법은 Regnell 등이 제시한 방법으로 그래픽 표기법을 통해 유즈케이스 모델을 여러 다른 수준의 시나리오로 확장한다(Regnell et al., 1996). 이때 수준이란 환경적 수준, 구조적 수준, 이벤트적 수준을 의미한다. 환경적 수준에서는 유즈케이스와 액터 및 그들의 목표를 달성하기 위한 기능적 요구사항 간의 관계를 표현한다. 구조적 수준에서는 시퀀스, 예외, 방해, 반복 등을 포함한 각각의 에피소드들의 시퀀스로 완성된 유즈케이스를 표현하며(Grabowski et al., 1993), 이벤트적 수준에서는 메시지 시퀀스 차트를 통해 각각의 에피소드 내에서 일어나는 이벤트를 기술한다.

유즈케이스 맵(Buhr et al., 1996)은 소프트웨어 시스템의 행동적 시간적 요구사항을 보여준다. 하나 혹은 그 이상의 유즈케이스들의 수행 책임들을 원인과 결과의 관점에서 연결하는 가지적인 표기법을 제공한다. 이러한 연결 관계에는 시스템의 구조를 표현하는 소프트웨어나 하드웨어 구성요소를 덧붙인다. 이 표기법은 유즈케이스에 맞춰 시나리오의 진행을 보여주며 요구분석가에게 기능적 구성요소 및 요소 각각의 행동, 시스템의 구조적 조직 등을 설명해 준다. 이처럼 맵을 통한 경로는 해당 시스템의 분석을 용이하게 한다.

시개념 오토마타는 일반적으로 시스템의 요구사항의 모델링 및 공식화를 통해 소프트웨어 요구

분석서의 질을 향상시키고자 하는 방법이다(Hsia et al., 1994; Leveson et al., 1994). 공식화 방법은 수학적 기반의 기술로 요구사항, 요구분석서, 개발 내 디자인 단계에서의 오류를 제거해 나감으로써 신뢰도를 증가시키는 결함 회피 기술이다. 이 접근법을 사용하는 또 다른 주요 이점은 시나리오 집합을 구조화 시킬 수 있다는 점이다. 특히 Some 이 제시한 시개념 오토마타에서는 보다 정확하게 사용자 요구사항을 이해하기 위해 시간적 제약 사항에 대해 기술하는 것으로 특히 실시간 시스템 요구분석에 유용하다. 하지만 이 방법은 요구분석서를 읽고 이해하기가 어려우며 실제 상황은 모든 면을 모델링 할 수 없다는 단점을 가진다. <표 1>은 이상의 방법론들을 정리한 것이다(Saiedian et al., 2005).

이처럼 시나리오 작성을 위한 다양한 방법론들이 제안되었지만 그들 중에서 시나리오 작성과 요구분석 작업간의 연결성을 찾아보기는 매우 힘들다. 대부분의 방법론이 요구분석에 대한 실질적인

고려없이 단순히 시나리오 작성 및 그 표현에만 집중하고 있다. 결국 이러한 방법론적 한계가 시나리오와 요구분석서의 이중 작성은 물론 시스템 개발의 소요시간 및 비용을 낭비하는 결과를 초래하고 있다.

## 2.2 시나리오 기반 요구분석법

시나리오 기반의 분석은 일반 사용자가 접근하기 용이한 예나 실세계의 경험을 바탕으로 추출한 시나리오를 사용하여 사용자가 자신의 시스템 요구사항을 기술한다(Potts, 1999; Sutcliffe et al., 1998). 시나리오 기반의 요구분석 방법론은 크게 ScenIC과 SCRAM로 나눌 수 있다(Sutcliffe et al., 1998).

먼저 ScenIC 방법론은 목표, 목적, 과제, 장애물, 액터로 구성된 시나리오와 관련된 지식 스키마를 제안한다(Potts, 1999). 이때 시나리오는 에피소드, 액터(사람 혹은 기계)에 의해 수행되는 액션으로

<표 1> 시나리오 작성 방법론

구 분	자연어	상태도 표현법	계층화된 도면 표기	유즈케이스 맵	시개념 오토마타 접근법
기본 개념	유즈케이스, 액터, 사전/사후 조건, 시나리오 등	이벤트와 트랜지션, 시나리오 표현	유즈케이스, 액터, 시퀀스, 예외 처리, 반복, 대안, 시나리오 등	유즈케이스, 맵, 사전/사후 조건, 트리거, 시나리오	시개념 오토마타, 제약식, 오퍼레이션 시맨틱스
표기법	자연어, 화살표와 타원형으로 관계 표현	계층과 시나리오 표현 위한 상태도	도식화된 표기로 시나리오 표현, 이벤트 수준 등	유즈케이스의 비전 표기 링크, 시나리오 등	기술언어로서의 시개념 오토마타로 시나리오 표현, 시나리오를 오퍼레이션의 순서로 인식
주요 역할	요구 사항 기술, 객체 탐색	시나리오의 구조화	이벤트 수준에서의 시나리오 표현	행태적 관계 표현	시나리오의 오퍼레이션 뷰 표현, 고 수준의 정확도 제공

구성된다. 목표는 특정 상태를 성취하거나 유지하거나 피하는 것이다. 반면 장애물은 목표의 성취를 방해하거나 성공적인 과제 완수를 막는다. 본 방법론에서 시나리오는 반구조적인 형식으로 표현된다. 시나리오를 분석 및 점검을 포함한 모든 사이클을 통해 요구분석 및 명세서 작성이 이루어진다. 가이드라인 집합은 시나리오의 이야기를 형식화하고, 목표, 액션, 장애물을 밝힌다. 에피소드는 목표가 장애물의 방해에도 불구하고 시스템 및 액터의 과제 수행에 의해 성취되었는지에 의해 평가될 수 있다. 이처럼 시스템의 요구사항을 충족시키기 위해 목표, 과제, 리소스, 액터들 간의 종속 관계성을 확인한다. 목표가 성취될 수 있음을 확인하기 위해 종속성 분석 및 수단-목적 분석에 기반 하여 과제 및 액터의 능력을 시험한다.

다음으로 SCRAM 방법론(Sutcliffe et al., 1998)은 명시적인 모델링 및 명세서를 제공하지 않으며, 디자이너에 의해 선택된 소프트웨어 공학 방법론에 따라 병렬적으로 진행된다. Alistair에 의하면 이 방법론은 첫째, 도메인을 분석하고 초기 요구사항을 확인하며, 둘째 스토리보드를 작성하고 디자인 방향을 설정하고 셋째, 요구사항을 발굴하고, 마지막으로 프로토타이핑 및 요구사항을 검증하는 4가지 단계로 구성된다. 이때 도메인 분석 및 초기 요구사항 확인 단계에서는 첫 번째 컨셉 데모를 개발하기 위한 충분한 정보를 얻기 위해 전통적인 인터뷰 방식 및 팩트-파인딩 기술을 활용하고, 소수의 고객을 방문한다. 스토리보드 작성 및 디자인 방향 설정 단계에서는 요구된 시스템의 초기 비전을 정의하고 스토리보드를 통해 사용자에게 설명함으로써 실행 가능성에 대한 피드백을 획득한다. 요구사항 발굴 단계에서는 사용자에게 시나리오 기반의 반상호적인 컨셉 데모 및 초기 프로토타입을 사용하여 보다 상세한 디자인을 설

명하고 요구사항을 발굴한다. 마지막으로 프로토타이핑 및 요구사항 검증 단계에서는 모든 기능을 포함한 프로토타입을 개발하고 모든 사용자에게 받아들여질 때까지 요구사항을 계속적으로 다듬어 나가게 된다.

한편 시나리오 기반의 요구분석을 위한 지금까지의 상용화된 도구로서는 DOORS, Rational Rose, RETH, CREWS-SAVRE 등이 있다. DOORS 및 Rational Rose는 시나리오를 형식화하고 그 일관성을 확인한다(Do Prado Leite et al., 2000). RETH 하이퍼텍스트 도구(Kaindl, 1995)는 시나리오 검증을 지원하기 위한 목적으로 시나리오, 목표, 기능적 요구사항을 연결한다. CREWS-SAVRE 도구(Maiden et al., 1998)는 가능한 사건 시퀀스를 확장함으로써 초기 시나리오의 변경을 돕는다. 여기서 사건 시퀀스는 유즈케이스와 같은 행동 모델로부터 추출될 수 있다.

한편 시나리오 생성에 대한 방법론은 주로 수리적 모형의 생성에 주력해 왔으며 주로 시뮬레이션 분야의 관심이었다(Guastaroba et al., 2009). 그러나 시스템 개발을 목적으로 한 시나리오 문서의 생성에 대한 연구는 아직 이루어지지 않고 있는 실정이다.

앞서 설명한 현재까지의 방법론들은 모두 시나리오 기반 요구분석 방법론이라고 되어 있지만 실제로 그들의 요구분석 과정은 시나리오 작성 과정과 완전히 분리되어 있다. 그들은 오직 시나리오 작성이 완전히 완료된 후에 시스템 작성을 시작한다.

### 3. 구조적 통합 방법론

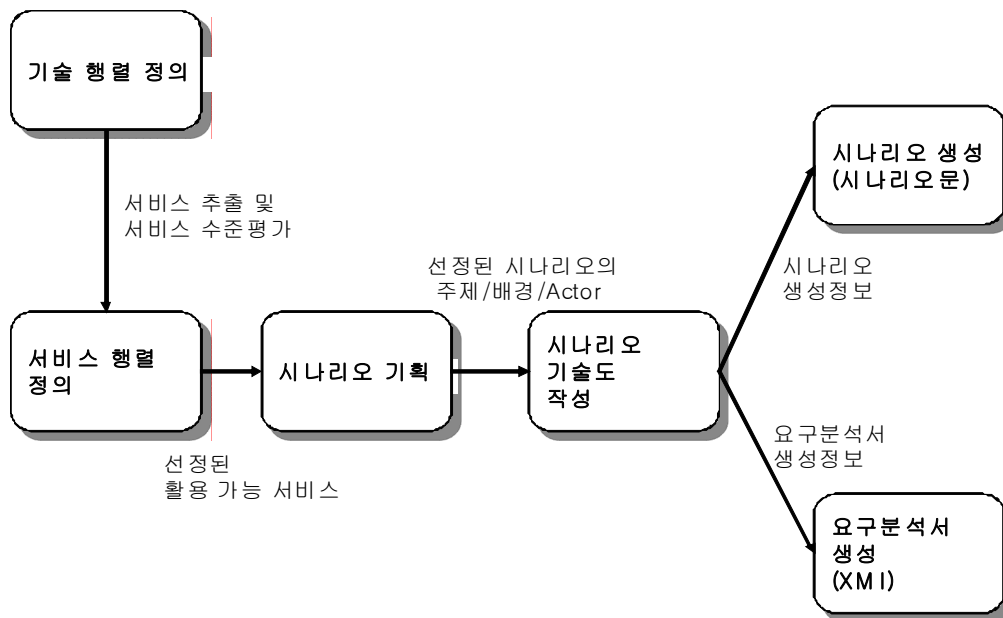
#### 3.1 전체 프레임워크

시나리오를 작성하는 방법에는 크게 두 가지 방

법이 있다. 하나는 현상에 대한 분석이 먼저 이루어진 후에 이를 기반으로 시나리오를 작성하는 것(guided approach)이고, 다른 하나는 이러한 현상 분석 틀에서 벗어나 자유롭게 미래에 일어날 일들에 대해 시나리오를 작성하는 것(open-ended approach)이다(Ranta et al., 2003). 본 연구는 앞서 밝혔듯이 Technical-push 방식의 시나리오 작성 즉 단기적 관점에서 개발될 기술의 실용성 증명을 위한 시나리오 작성에 중점을 두고 있으므로 시나리오 작성에 어느 정도의 틀을 제공하며 이후의 요구분석 및 구현 단계가 실제적으로 이루어질 수 있도록 하는 현상분석 기반의 시나리오 작성 방법(guided approach)을 도입했다. 본 논문에서는 서비스 시나리오를 작성하기 전 현재 활용이 가능한 기술 및 장치, 서비스를 분석하는 것을 현상분석 단계로 하고 이를 기반으로 시나리오를 작성하고자 한다.

이와 같은 접근 방법에 따라 USS 개발의 초기 단계에서 고려하게 되는 시나리오 구축 방법 및 요구분석 방법의 구조적 통합을 위해 본 연구에서는 <그림 1>과 같은 전체적인 구조적 통합 방법론을 제안한다.

먼저 시나리오 작성자는 시나리오에서 현재 활용이 가능한 주요 기술 및 장치를 수집 및 정리하여 기술 행렬을 정의한다. 이때 기술행렬이란 작성할 시나리오에서 활용 가능한 주요 기술 및 장치에 대한 정의 및 개략적 설명을 정리한 표를 의미한다. 정의된 기술 행렬 내 기술간 조합을 통해 시나리오에서 활용 가능한 서비스를 추출하고 그에 대한 정의를 표의 형태를 가진 서비스 행렬로 정의한 뒤 서비스 수준 평가를 실시하여 시나리오에서 수용 가능한 서비스 수준을 가진 서비스들만을 정리한다. 이렇게 서비스들이 정리되고 나면 이를 기반으로 시나리오를 기획하고 기획된 시나리오



<그림 1> 시나리오 구축 및 요구분석 위한 구조적 통합 방법론의 전체 개념도

를 근간으로 시나리오 기술도를 그린다. 여기서 시나리오 기술도는 약속된 기호 및 절차를 통해 시나리오를 표현하는 정형화된 방법을 의미하며, 상황도(배경도), 레벨-0 다이어그램, 레벨-1 다이어그램으로 구성된다. 완성된 시나리오 기술도를 공동기반으로 하여 시나리오를 생성하고 요구분석서를 작성한다.

### 3.2 기술 행렬 정의

기술행렬이란 작성할 시나리오에서 활용 가능한 주요 기술 및 장치에 대한 정보를 수집하고 각 기술별로 개략적인 설명서를 작성하는 것을 의미한다. 기술행렬은 시나리오에 활용할 수 있는 가용 유비쿼터스 컴퓨팅 기술 및 장치들을 간단한 설명과 함께 제안하는 것이다.

### 3.3 서비스 행렬 정의

앞서 정의된 기술 행렬을 기반으로 개별 기술 혹은 특정 기술 간의 조합을 통해 시나리오에서 사용 가능한 서비스를 추출한다. 여기서 추출된 서비스는 서비스 행렬을 통해 그 내용이 구체화되고 서비스 수준 측정 과정에서 시나리오 반영 여부를 확정한다. 또한 기술 및 장치를 반영한 서비스 명이 다수 인식되었으면 각 서비스 별로 <표 4>와 같은 서비스 행렬 정의 양식에 따라 내용을 보다 상세히 기술한다. 이때 특별히 형용사나 부사형으로 표현되는 기능성 측면에 대해서는 어떤 기술로 인해 그 같은 기능의 구현이 가능해지는지 최대한 정확히 설명한다.

이렇게 정의된 각각의 서비스에 대해 서비스 수준 측정을 위한 체크리스트를 활용하여 서비스 수준을 측정하고 그 결과를 서비스 요청자와 함께 검토하여 해당 서비스의 채택여부를 결정한다. 채

택되지 못한 서비스는 서비스 행렬 목록에서 제거하며 만약 추가적인 서비스 발굴이 필요한 경우에는 서비스 추출 단계부터 다시 재작업 한다. 본 단계는 서비스 수준 측정을 통해 각 서비스의 시나리오 반영 여부를 결정하는데 그 목적이 있다.

## 3.4 시나리오 기획

### 3.4.1 시나리오 주제 및 배경 설정

개발할 서비스가 결정되면 시나리오의 주제 및 배경을 설정한다. 이때 시나리오의 주제 및 배경은 다음과 같은 방법으로 설정한다.

- 시나리오에 등장하는 서비스들을 통해 궁극적으로 사용자에게 전달하고자 하는 가치 정의한다.
- 해당 시나리오와 수익관계를 가지는 이해관계자를 식별한다.
- 주인공이 하고자 하는 일 또는 목표를 설정하고 필요한 경우 주인공의 목표에 대한 갈등 및 장애물을 설정한다.
- 시나리오의 개략적인 내용을 기술한다.

### 3.4.2 시나리오 행위자(Actor) 및 역할 설정

시나리오 안에서 실질적으로 서비스를 수행하는 주체가 되는 행위자를 정의하고 각 행위자의 성격(장비, 에이전트, 온톨로지 등)을 구분하여 역할을 설정한다. <표 2>은 시나리오의 행위자 및 역할 정의 양식이다.

## 3.5 시나리오기술도 작성

기획된 시나리오를 근간으로 시나리오 기술도를 그린다. 여기서 시나리오기술도란 체계적인 시나리오 작성을 위해 약속된 기호와 절차를 적용하

여 시나리오를 표현하는 정형화된 방법을 의미하며, 배경도, 레벨-0 다이어그램, 레벨-1 다이어그램으로 구성된다. 다음의 <표 2>은 본 연구에서 제안하는 것으로 시나리오기술도의 구성요소를 정의한 것이다.

### 3.5.1 시나리오 배경도 작성

먼저 시나리오 배경도에서는 시나리오의 수익 모델 표현한다. 이 다이어그램에서는 해당 시나리오와 수익관계를 가지는 이해관계자들 간의 관계가 설명되며 시나리오에 등장하는 서비스의 대상 범위가 결정된다. 시나리오 배경도 작성순서는 다음과 같다.

- ① 시나리오와 수익관계를 가지는 이해관계자

를 식별한다.

- ② 시나리오명을 Sequence로 표시한다.
- ③ 식별된 이해관계자들을 관련자로 표시한다.
- ④ 시나리오와 각각의 이해관계자들 간의 상세한 수익관계를 데이터흐름을 통해 표현한다. 시나리오 배경도가 작성된 예는 <그림 2>와 같다.

### 3.5.2 시나리오 레벨-0 다이어그램 작성

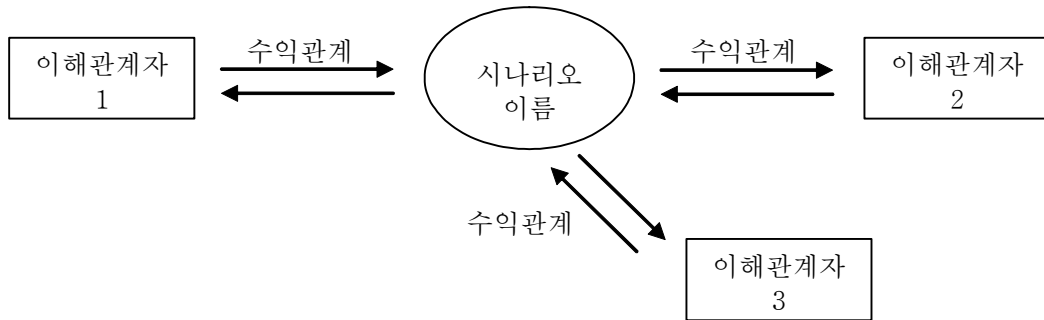
시나리오에서 수행되는 모든 서비스 및 그 입출력 데이터를 표현한다. 시나리오 레벨-0 다이어그램 작성순서는 다음과 같다.

- ① 해당 시나리오에서 수행되어야 할 세부 서비스 요소를 식별한다.
- ② 식별된 세부 서비스 요소들을 Sequence로

<표 2> 시나리오기술도의 구성요소

구성요소	기 호	설 명
시퀀스		시나리오에서 제공하는 서비스를 표현하며 하나의 시퀀스는 한 개 이상의 장면(Scene)을 가진다.
장면		데이터를 활용한 실질적인 서비스 제공 활동을 표현하며 서비스 제공 주체인 행위자 및 서비스 내용을 가진다.
관련자		시나리오에서 제공되는 서비스와 수익관계를 가지는 이해관계자를 표현한다.
데이터 흐름		서비스간 혹은 실질적인 행위자활 동간의 일반적인 데이터 이동을 표현한다
		서비스간 혹은 실질적인 행위자활 동간의 선택적 데이터 이동을 표현한다
		서비스간 혹은 실질적인 행위자활 동간의 Loop의 데이터 이동을 표현한다
데이터저장소		데이터가 저장되어 있는 곳을 표현하며 다수의 상이한 물리적인 형태를 취할 수 있다.
정보 흐름		장면과 데이터 저장소간 정보 이동을 표현하며 장면은 특정정보를 요청하고 데이터 저장소는 이를 제공한다.





<그림 2> 시나리오 배경도

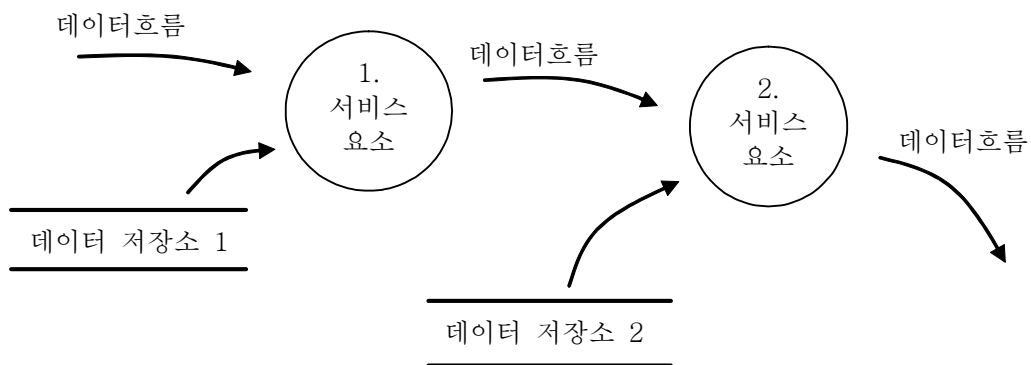
표시한다. 이때 순서가 있는 경우 순서에 맞춰 고유번호 부여하고 그 순서에 맞춰 배열한다.

- ③ 모든 서비스와 관련된 입출력 데이터 및 데이터 저장소를 식별한다.
- ④ 식별된 입출력 데이터 및 데이터 저장소를 Sequence로 표기된 해당 서비스에 데이터 흐름 및 데이터 저장소로 연결하여 표현한다.
- ⑤ 만일 서비스간 연결이 제대로 이루어지지 않는 경우 서비스 요소 혹은 입출력 데이터의 식별이 완전하지 않은 것이므로 이 같은 사항들을 다시 점검한 후 재연결 한다.

레벨-0 다이어그램의 작성 예는 다음 <그림 3>과 같다.

### 3.5.3 시나리오 레벨-1 다이어그램 작성

각각의 서비스에 등장하는 행위자 및 그 활동을 구체적으로 표현한다. 이때 모든 행위자는 Event, Action, Response을 가지며 Event는 행위자의 Action을 유발하는 입력적 행동, Action은 행위자가 직접 수행하는 활동, Response는 행위자의 Action으로 발생하는 반응적 행동을 각각 의미한다. 이러한 행위자와 그 활동들은 Action 행렬 형태로 정리할 수 있다. 레벨-1 다이어그램 작성순



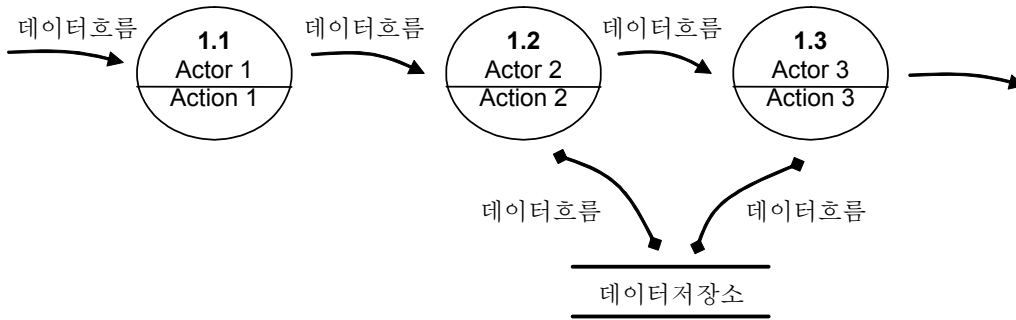
<그림 3> 시나리오 레벨-0 다이어그램

서는 다음과 같다.

- ① 각 서비스에 등장하는 행위자 및 각 행위자별 Event, Action, Response를 식별한 뒤 표 7의 양식으로 Action 행렬을 작성한다.
- ② 식별된 각 행위자의 Action을 장면으로 표시한다. 이때 순서가 있는 경우 순서에 맞춰 고유번호 부여한다.
- ③ 식별된 Event 및 Response를 기반으로 각

Action과 관련된 입출력 데이터 및 데이터 저장소를 정의한다.

- ④ 식별된 입출력 데이터 및 데이터 저장소를 해당 장면에 데이터 흐름 및 데이터 저장소로 연결하여 표현한다. 이때 장면은 순서대로 배열한다.
- ⑤ 레벨-0 다이어그램에서 정의한 Sequence의 입출력 데이터와 첫 장면의 입력 데이터 및



<그림 4> 시나리오 레벨-1 다이어그램

```

for(int i=0; i<uss_mdb.count; i++){
    if(uss_mdb.sce_DS[i].equals("")){
        //시나리오 생성 Rule 1
        jTextArea1.append(uss_mdb.sce_Actor[i]+"(은,는,이,가) "+uss_mdb.sce_Action[i]+"한다"+newline);
        if(i==(uss_mdb.count-1)){
            break;
        }
        else if(i!=(uss_mdb.count-1)&&uss_mdb.sce_Actor[i+1]!=""){
            // 시나리오 생성 Rule 3
            jTextArea1.append(uss_mdb.sce_Actor[i]+"(은,는,이,가) "+uss_mdb.sce_Actor[i+1]+"에게 "+uss_mdb.sce_Response[i]+"한다"+newline);
        }
    }
    else{
        // 시나리오 생성 Rule 2
        jTextArea1.append(uss_mdb.sce_Actor[i]+"(은,는,이,가) "+uss_mdb.sce_DS[i]+"에 "+uss_mdb.sce_DSSend[i]+"(을,를)"+" 전송하고 "+uss_mdb.sce_DS[i]+"(으로부터,로 부터) "+uss_mdb.sce_DSReceived[i]+"(을,를)"+" 가져온다"+newline);
        // 시나리오 생성 Rule 1
        jTextArea1.append(uss_mdb.sce_Actor[i]+"(은,는,이,가) "+uss_mdb.sce_Action[i]+"한다"+newline);
        if(i==(uss_mdb.count-1)){
            break;
        }
        else if(i!=(uss_mdb.count-1)&&uss_mdb.sce_Actor[i+1]!=""){
            // 시나리오 생성 Rule 3
            jTextArea1.append(uss_mdb.sce_Actor[i]+"(은,는,이,가) "+uss_mdb.sce_Actor[i+1]+"에게 "+uss_mdb.sce_Response[i]+"한다"+newline);
        }
    }
}
    
```

<그림 5> 시나리오 생성 규칙

마지막 장면의 출력데이터 흐름이 일치하는지 확인한다.

레벨-1 다이어그램의 작성 예는 다음 <그림 4>와 같다.

### 3.6 시나리오 생성

시나리오는 시나리오 레벨-0 다이어그램의 Sequence 단위로 시나리오 레벨-1 다이어그램의 내용을 기반으로 시나리오 자동생성 규칙을 적용하여 자동 생성한다. 이때 이 규칙은 Scene의 순서에 따라 적용되며 생성규칙은 <그림 5>와 같다.

### 3.7 요구분석서 생성

요구분석서는 시나리오 기술도를 기반으로 XMI를 생성함으로써 UML 도구에서 제공하는 역공학 기법으로 작성되는 방법을 택하였다. 본 연구에서는 요구분석을 위해 소프트웨어 시스템을 시각화하고, 기술하고, 구축하며 또한 산출물을 문서화하는데 널리 사용되고 있는 모델링 언어인 UML을

사용하였다. 특히 OMG의 공식 표준으로 자리잡은 UML의 4개 모델과 8개 다이어그램 중 시퀀스 다이어그램을 채택하였다. 생성 규칙은 Star UML의 XMI 작성 방법에 근거하였다.

## 4. 시스템 구현

제안된 방법론의 가능성을 보이기 위하여 프로토타입 시스템을 구현하였다. 시나리오기술도 작성도구는 Java 1.6.x 버전의 애플리케이션으로 요구분석서는 XMI 형태로 표현되도록 하였다.

또한 구현된 시나리오 구축 및 요구분석 위한 구조적 통합 방법론 지원 도구의 용이성 및 실효성을 확인하기 위해 우리는 해당 방법론을 ‘유비쿼터스 쇼핑 안내 시나리오’에 적용하여 사례 연구를 진행하였다. 유비쿼터스 쇼핑 안내 시나리오란 자동 고객인식 및 상품 제안, 증강현실을 통해 가상으로 옷 착용모습 전시 및 원격 전송, 자동 상품결제 및 배송 등의 서비스를 기반으로 사용자의 쇼핑을 보다 실제적으로 돕기 위해 제안된 시나리오

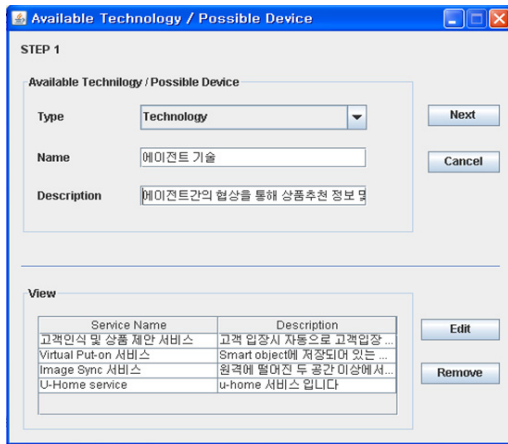
<표 3> 기술행렬 정의

구 분	기술 명	내 용
기 술	에이전트 기술	에이전트간의 협상을 통해 상품추천 정보 및 결제 정보 제공
	의미망	온톨로지 언어(Ontology Language : OWL)로 작성된 데이터를 통해 Semantic Information 제공
	uPAN Scale-free 무선전송 기술	매장의 네트워크 AP와 u-PDA 사이의 데이터 전송
	UMO	정해진 성능을 유지하는 기기가 아닌 상황에 따른 재구성이 가능하고 스마트 오브젝트와의 computing resource를 공유하며 궁극적으로 Invisible community computing platform을 지향하는 사람이 소지하고 다니는 Mobile Object 혹은 Device
장 치	UMO	고객이 소지한 스마트 오브젝트
	Smart Wall 또는 AR Table	매장이나 고객의 집에 설치된 스마트 디스플레이 장치 디스플레이에 관한 중요한 역할을 함
	RFID 센서	장치 간의 인지 역할을 함

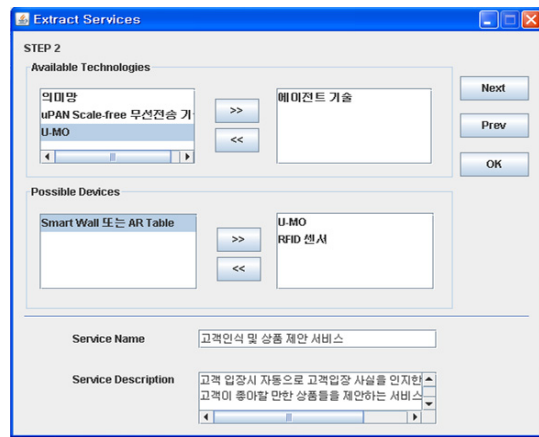
이다.

유비쿼터스 쇼핑 안내는 실제적으로 한국의 유비쿼터스 개발 사업단에서 고려하고 있는 서비스로 에이전트 기술, 의미망 기술, uPAN기술, 소형의 유비쿼터스 모바일 단말기인 UMO기술, 스마트 월 또는 증강현실 테이블 그리고 RFID 센서 기술 등을 활용하여 시스템을 개발하려고 하고 있었다. 이에 기술행렬 정의는 다음 <표 3>과 같이 이루어졌다.

이에 따라 시나리오 기술도는 사전 정의된 기술 행렬과 서비스 행렬을 기반으로 작성된다. <그림 6(a)>는 기술 행렬을 정의할 수 있도록 입력하는 화면이다. 한편 서비스 수준 평가는 서비스 수준 측정용 통해 서비스 요청자가 각 서비스의 시나리오 반영 여부를 결정하는 단계로 본 사례연구에서는 서비스 요청자가 모든 서비스의 반영을 결정하는 것으로 가정하였다. 도구상에서 이러한 서비스 행렬을 정의하는 화면은 <그림 6(b)>와 같다.

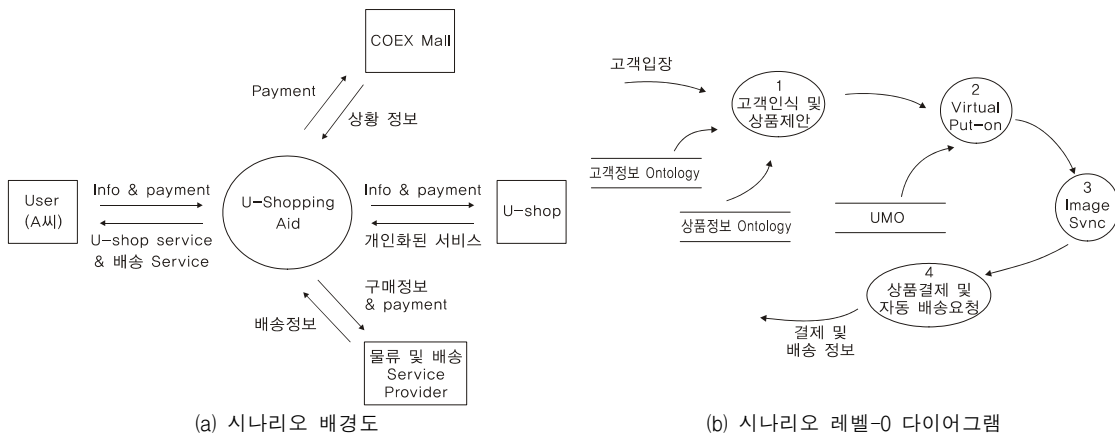


(a)



(b)

<그림 6> 기술행렬 및 서비스행렬 정의 화면



(a) 시나리오 배경도

(b) 시나리오 레벨-0 다이어그램

<그림 7> 시나리오 다이어그램

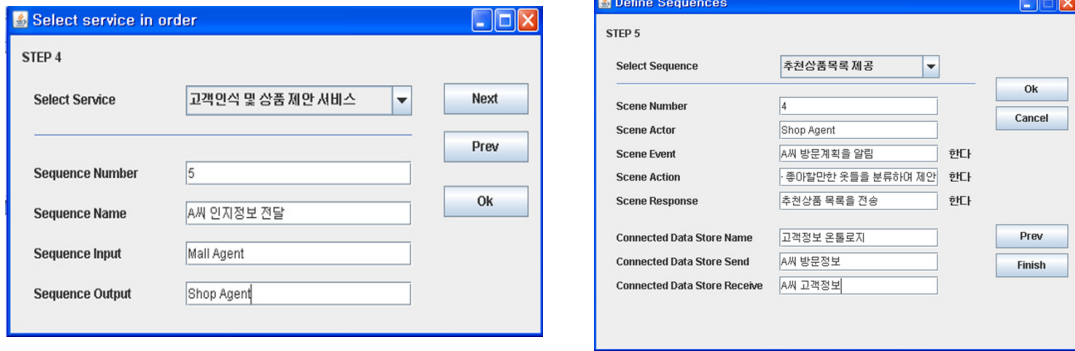
다음으로 시나리오 기획을 위해 시나리오 행위자 및 역할이 설정되면 앞에서 인식된 기술, 서비스 및 시나리오를 통하여 다음 <그림 7(a)> 및 <그림 7(b)>와 같이 시나리오 배경도와 시나리오 레벨-0 다이어그램이 도출되었다. 레벨-1 다이어그램은 같은 요령이므로 생략하였다.

시나리오 레벨-0 다이어그램과 시나리오 레벨

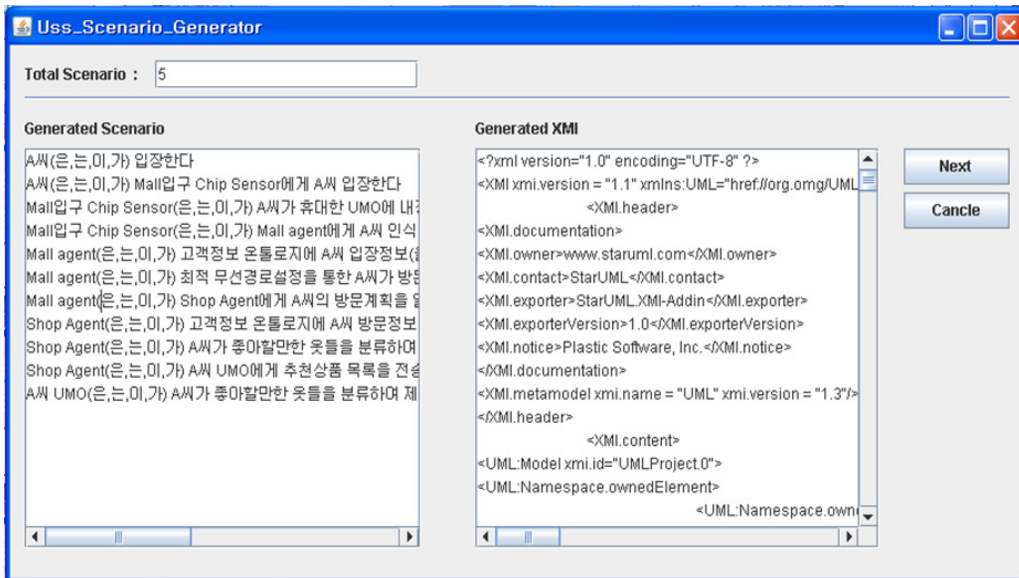
-1 다이어그램은 각각 <그림 8(a)>, <그림 8(b)>과 같은 입력화면을 통해 작성된다.

이렇게 시나리오기술도가 완성되고 나면 앞의 <그림 5>와 같은 시나리오 생성규칙을 통해 <그림 9>의 왼쪽 창과 같은 모양으로 시나리오가 자동 생성된다.

마지막으로 요구분석서에 대한 명세서는 <그림



(a) (b)  
<그림 8> 시나리오 레벨-0 다이어그램 및 레벨-1 다이어그램 작성 화면



<그림 9> 시나리오 자동 생성 화면

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <XMI xmi.version="1.1" xmlns:UML="href://org.omg/UML/1.3" timestamp="Mon Mar 24 15:27:6 2008">
3   <XMI.header>
4     <XMI.documentation>
5       <XMI.owned>www.stacuml.com</XMI.owned>
6     <XMI.contact>StarUML</XMI.contact>
7     <XMI.exporter>StarUML</XMI.exporter>
8     <XMI.exporterVersion>1.0</XMI.exporterVersion>
9     <XMI.notice>Plastic Software, Inc.</XMI.notice>
10    </XMI.documentation>
11    <XMI.metamodel xmi.name="UML" xmi.version="1.3"/>
12  </XMI.header>
13  <XMI.content>
14    <UML:Model xmi.id="UMLProject.0">
15      <UML:Namespace.ownedElement>
16        <UML:Package xmi.id="UMLPackage.0" name="SamplePackage" visibility="public" isSpecification="false" namespace="UMLProject.0" isRoot="false" isLeaf="false">
17          <UML:Class xmi.id="UMLClass.2" name="" visibility="public" isSpecification="false" namespace="UMLPackage.0" clientDependency="UMLRealization.200" isRoot="false" isLeaf="false">
18            <UML:Classifier.feature>
19              <UML:Attribute xmi.id="UMLAttribute.3" name="" visibility="private" isSpecification="false" ownerScope="instance" changeability="changeable" targetScope="instance">
20                <UML:BehavioralFeature.parameter>
21                  <UML:Parameter xmi.id="UMLParameter.4" name="return_parameter" visibility="public" isSpecification="false" kind="return" behavioralFeature="UMLOperation.0" type="UMLClass.2">
22                    <UML:BehavioralFeature.parameter>
23                      <UML:Parameter xmi.id="UMLParameter.5" name="in_parameter" visibility="public" isSpecification="false" kind="in" behavioralFeature="UMLOperation.0" type="UMLClass.2">
24                        <UML:Operation>
25                          <UML:BehavioralFeature.parameter>
26                            <UML:Parameter xmi.id="UMLParameter.6" name="return_parameter" visibility="public" isSpecification="false" kind="return" behavioralFeature="UMLOperation.0" type="UMLClass.2">
27                              <UML:Parameter xmi.id="UMLParameter.7" name="in_parameter" visibility="public" isSpecification="false" kind="in" behavioralFeature="UMLOperation.0" type="UMLClass.2">
28                                <UML:BehavioralFeature.parameter>
29                                  <UML:Operation>
30                                    <UML:BehavioralFeature.parameter>
31                                      <UML:Parameter xmi.id="UMLParameter.8" name="return_parameter" visibility="public" isSpecification="false" kind="return" behavioralFeature="UMLOperation.0" type="UMLClass.2">
32                                        <UML:Parameter xmi.id="UMLParameter.9" name="in_parameter" visibility="public" isSpecification="false" kind="in" behavioralFeature="UMLOperation.0" type="UMLClass.2">
33                                          <UML:BehavioralFeature.parameter>
34                                            <UML:Operation>
35                                              <UML:BehavioralFeature.parameter>
36                                                <UML:Parameter xmi.id="UMLParameter.10" name="return_parameter" visibility="public" isSpecification="false" kind="return" behavioralFeature="UMLOperation.0" type="UMLClass.2">
37                                                  <UML:Parameter xmi.id="UMLParameter.11" name="in_parameter" visibility="public" isSpecification="false" kind="in" behavioralFeature="UMLOperation.0" type="UMLClass.2">
38                                                    <UML:BehavioralFeature.parameter>
39                                                      <UML:Operation>
40                                                        <UML:BehavioralFeature.parameter>
41                                                          <UML:Parameter xmi.id="UMLParameter.12" name="return_parameter" visibility="public" isSpecification="false" kind="return" behavioralFeature="UMLOperation.0" type="UMLClass.2">
42                                                            <UML:Parameter xmi.id="UMLParameter.13" name="in_parameter" visibility="public" isSpecification="false" kind="in" behavioralFeature="UMLOperation.0" type="UMLClass.2">
43                                                              <UML:BehavioralFeature.parameter>
44                                                                <UML:Operation>
45                                                                  <UML:BehavioralFeature.parameter>
46                                                                    <UML:Parameter xmi.id="UMLParameter.14" name="return_parameter" visibility="public" isSpecification="false" kind="return" behavioralFeature="UMLOperation.0" type="UMLClass.2">
47                                                                      <UML:Parameter xmi.id="UMLParameter.15" name="in_parameter" visibility="public" isSpecification="false" kind="in" behavioralFeature="UMLOperation.0" type="UMLClass.2">
48

```

<그림 10> 작성된 XMI 형태의 요구분석서 예

9>의 오른쪽 창에 나타나게 되며, 형태는 다음 <그림 10>과 같은 XMI 문서 형태로 작성된다. 이 명세서는 파일로 저장되게 생성된 XMI 문서를 UML 도구에서 읽어 들이면 UML 내에서의 여러 가지 모델들이 작성된다. 본 연구에서는 Star UML을 사용하였으며 시퀀스 다이어그램의 여러 다이어그램들이 생성된다.

이러한 본 연구에서 제안한 방법론은 기존의 방법론에 비하여 다음 <표 4>과 같은 차이가 있다.

## 5. 결 론

USS 개발을 성공과 효율성 증진을 위해 먼저 시나리오를 구축한 후에 요구사항을 추출하고 분석하는 것에 대한 중요성이 날로 높아지면서 이를 지원하는 시나리오 구축 및 요구분석 통합 방법론이 필요하게 되었다. 하지만 지금까지 사용되어오

고 있는 기존 USS 시나리오 작성 방법론들은 USS 시나리오에 대한 정형화된 작성 방법론이 존재하지 않고 전체적인 USS 개발 단계에 대한 고려도 없이 시나리오 작성을 위한 가이드라인만 제시하고 있는 실정이라서 그 효용성이 떨어진다. 이러한 이유로 실제로 구축되는 시나리오는 작성자의 주관적 성향에 의해 그 내용이 결정되고 있으며, USS 개발을 위한 구체적인 요구분석과 연결되지 않고 있다. 이에 시나리오 및 요구분석서간 이중 작성, 내용 불일치, 시간 및 비용 낭비 등의 문제가 발생되고 있으며 시나리오에 대한 객관적인 분석 및 평가, 요구분석의 시나리오 반영도 검증 등의 작업 또한 어렵게 하고 있다.

본 연구에서 제안한 시나리오 구축 및 요구분석을 위한 구조적 통합 방법론을 통하여 보다 정형화되고 구조화된 모습의 시나리오 기술도를 구축하고 이를 기반으로 시나리오 작성 및 요구분석

<표 4> 기존 방법론과 구조적 통합 방법론의 비교

기존 방법론	기존 방법론의 한계점	구조적 통합 방법론의 개선점
자연어	자연어의 사용으로 시나리오 작성자와 시스템 개발자간의 커뮤니케이션은 용이하나 시스템의 동작과정이 모두 글의 형태(textual format)로 표현되어 있어 실제 시스템 개발시 필요한 데이터 요소 및 프로세스 등을 요구분석 하는데 중복적인 시간이 소요됨	기술적인 시나리오 형태로만 작성되던 시나리오를 시스템 구성요소 및 프로세스를 고려한 구조화되고 그래픽한 표기법을 개발하고 이를 통해 시나리오를 작성함
상태도 표현법	구조화된 텍스트와 상태도를 통해 이벤트와 이행(transition) 등을 표현하여 비교적 작성과 해석이 쉬우나 시스템 요구분석시 필요한 데이터베이스 등의 시스템 구성요소들을 구별하여 표현하지 못함	구조적 통합 방법론에서의 표기법은 데이터베이스, 리소스 등의 시스템 구성요소를 사전에 고려하여 개발되어 시나리오에 등장하는 해당 요소들의 표현이 모두 가능함
계층화된 도면 표기	시나리오를 이벤트 레벨에서 그래픽한 표기법을 사용해 표현하여 시나리오와 운영 환경 속성간의 종속성 등을 잘 보여주나 구체적인 데이터의 이동 흐름 등을 알 수 없어 시스템 요구분석서의 재작성이 불가피함	구조적 통합 방법론에서 시나리오 기술도는 구체적인 이동 데이터, 이동의 흐름 등을 모두 표현하여 요구분석서로 작성시 바로 활용이 가능함
유즈케이스 맵	비주요한 표기법을 통해 기능 컴포넌트들 간의 관계를 명쾌하게 표현하나 시나리오에서 등장하는 리소스 및 일시적 프로퍼티 등이 표현되지 않아 시나리오와 요구분석서 간의 차이가 발생함	데이터베이스, 리소스 등의 시스템 구성요소들이 모두 표현된 시나리오 기술도를 동일한 기반으로 하여 기술식 시나리오와 요구분석서를 작성함으로써 두 문서 간 일치성이 높음
시개념 오토마타 접근법	시스템의 요구사항의 모델링 및 공식화하여 표현한 수학적 기반의 기술로 매우 정교하나 작성작업이 매우 어렵고 복잡함	사전 정의된 표기법을 활용하여 쉽게 작성할 수 있음

작업을 수행할 수 있다. 이는 시나리오 제작과 요구분석 두 작업 내 일관성을 유지할 수 있게 하고 불필요한 중복작업을 제거하여 USS개발 효율성을 높이는데 기여할 것으로 보인다. 또한 본 논문에서 제안한 시나리오 구축 및 요구분석 위한 구조적 통합 방법론을 프로토타입으로 구현하였는데 이렇게 개발된 도구 소프트웨어 생산성 및 효율성을 높이는 도구로 활용될 수 있다.

현재 연구는 서비스 정의와 기술행렬에서 시작하여 XMI 문서 생성까지 수행한 것이다. 따라서 향후 연구방향은 현재 방법론을 적용한 프로토타입 시스템에 실제 그래픽 버전의 UML 생성 기능, 시나리오 업데이트 기능, UML을 통한 시나리오 작성 기능 등을 추가하는 것이다. 또한 본 연구의 결과물에 대한 실제적인 유용성을 사용자 테스트 형식으로 증명할 예정이다.

## 참고문헌

- 김성후, 박창호, 김청택, “유비쿼터스 서비스 평가를 위한 u-서비스 체험성의 분석”, *조사연구*, 7권 1호(2006), 1~28.
- 장종인, “미래연구 방법론 : 사례를 통해 살펴본 시나리오 방법론”, *정보통신정책*, 19권 9호(2007), 5~16.
- Amyot D. and G. Mussbacher, “On the extension of UML with use case maps concepts”, *Proceedings of UML 2000, Lecture notes in Computer Science*, No.1939(2000), 16~31.
- Annie I. and Colin P., Anton, “A Representational Framework for Scenarios of System Use”, *Requirements Engineering*, Vol.3, No.3/4(1998), 219~241.
- Buhr R. and R. Casselman “Use case maps for

- object-oriented systems”, Prentice Hall, 1996.
- Do Prado Leite, Julio Cesar Sampaio, Hadad, D. S. Graciela, Doorn, Jorge Horacio, and Kaplan, Gladys “A Scenario Construction Process”, *Requirements Engineering*, Vol.5, No.1(2000), 38~61.
- Glinz M, “Improving the quality of requirements with scenarios”, Proceedings of the Second World Congress for Software Quality(2WCSQ), Yokohama, (2000), 55~60.
- Glinz M. “A lightweight approach to consistency of scenarios and class models”, Proceedings 4<sup>th</sup> IEEE International Conference on Requirements Engineering, Schaumburg, IL, (2000), 49~58.
- Hsia P., J. Samuel, J. Gao, and D. Kung, “Formal approach to scenario analysis”, *IEEE Software*, Vol.11 No.2(1994), 33~41.
- Jacobson I., M. Christerson, P. Jonsson, and G. Overgaard, “Object-oriented software engineering—a use case driven approach”, Addison-Wesley, Reading, MA, 1992.
- Kaindl, H. “An Integration of Scenarios with Their Purposes in Task Modeling”, DIS 95 Conference Proceedings, (1995), 227-235, New York : ACM Press.
- Leveson, N. G., Heimdahl, M. P. E., Hildreth, H., Reese, J.D., “Requirements specifications for process-control systems”, *IEEE transactions on software engineering*, Vol.20, No.9, 1994, pp.684-707.
- Guastaroba, G., R. M. Mansini, and G. Speranza, “On the effectiveness of scenario generation techniques in single-period portfolio optimization,” *European Journal of Operational Research*, Vol.192, No.2( 2009), 500~511.
- Grabowski, J., P. Graubmann, and E. Rudolph, “The standardization of message sequence charts”, *Software Engineering Standards Symposium*, (1993), 48~63.
- Maiden, N. A. M., S. Minocha, K. Manning, and M. Ryan, “CREWS-SAVRE : systematic scenario generation and use”, *Requirements Engineering*, Third International Conference on, (1998), 148~155.
- Potts, C., “ScenIC : A Strategy for Inquiry-Driven Requirements Determination”, 4<sup>th</sup> IEEE International Symposium on Requirements Engineering, Los Alamitos CA : IEEE Computer Society Press, (1999), 58-65.
- Ranta, M., and H. Asplund, “Utilizing scenarios in service idea generation—a case of distant participation for a seminar”, *Good, the bad and the irrelevant : the use and the future of informAcommunication technologies, a transdisciplinary, proactive and collaborative knowledge-building conference*, (2003), 379-386.
- Regnell, B., M. Andersson, and J. Bergstrand, “A hierarchical use case model with graphical representation”, *Engineering of Computer-Based Systems, IEEE Symposium and Workshop on*, (1996), 270~277.
- Saiedian, H., P. Kumarakulasingam, and M. Anan, “Scenario-based requirements analysis techniques for real-time software systems : a comparative evaluation”, *Requirements Engineering*, Vol.10, No.1(2005), 22~33.
- Sutcliffe, A. G. and S. Minocha, “Scenario-based Analysis of Non-Functional Requirements”, *International workshop on requirements engineering : foundation for software quality*, (1998), 219~234.
- Sutcliffe, A. G., and M. Ryan, “Experience with SCRAM, a Scenario Requirements Analysis Method” *Requirements Engineering*, Third International Conference on, (1998), 164~171.



Abstract

## A System Development of Generating Ubiquitous Service Scenarios and System Analysis

Yon Nim Lee\* · Dong Young Jung\* · Oh Byung Kwon\*\*

Development and realization of ubiquitous services in physical space is now in progress to develop ubiquitous smart space (USS). As the development methodologies for USS, scenario development is performed before requirement analysis and design, or vice versa. However, even though lots of redundant elements could be found between scenarios and requirement analysis results, a sort of structural approaches to join them together has been still very rare. Hence, the aim of this paper is to propose a system which generates ubiquitous service scenarios and system analysis specifications according to a novel integration methodology. To do so, scenario and requirement analysis are integrated in a structured manner.

**Key Words** : Ubiquitous Smart Space, Scenario Analysis, Requirement Analysis, Structured Method

---

\* Research Center for Ubiquitous Business and Services, Kyung Hee University

\*\* School of International Management, Kyung Hee University

## 저자 소개



**이연님**

현재 경희대학교 국제경영학과 박사과정에 재학 중이며 동시에 유비쿼터스 비즈니스&서비스 연구센터(RCUBS; Research Center for Ubiquitous Business and Services)의 연구원으로 활동하고 있다. 2000년에 한동대학교 경영경제학부에서 학사, 2003년 이화여자대학교에서 석사학위를 취득하였다. 석사학위 취득 후 4년간은 금융권과 IT 컨설팅 분야에서 IT 컨설턴트로 활동하였으며, 현재는 지식경제부 UCN 사업 및 과학기술부 특정기초연구지원사업 과제에 참여하고 있다. 관심 연구분야는 상황인식 시스템, 상황인식 컴퓨팅내 프라이버시 이슈, 에너지관리 시스템 등이다.



**정동영**

한양대학교 컴퓨터공학과를 졸업하고, 현재 경희대학교 국제경영학과 MIS 전공으로 석사과정 및 경희대학교 유비쿼터스비즈니스&서비스연구센터 연구원으로 활동하고 있다. 관심분야는 유비쿼터스 컴퓨팅 서비스, 증강현실, U-Commerce 등에 있다.



**권오병**

2004년부터 경희대학교 국제경영학부 교수로 재직 중이다. 1995년에 한국과학기술원에서 박사학위를 취득하였으며 1996년부터 2003년까지 한동대학교 경영경제학부 교수로 재직한 바 있다. 2002년에는 미국 카네기멜론대학 전산학부 ISRI의 MyCampus 프로젝트에서 시맨틱 웹, 다중 에이전트, 웹서비스 및 추론시스템을 공동으로 구축한 바 있다. 현재는 경희대학교 내 유비쿼터스비즈니스&서비스연구센터(RCUBS)의 센터장으로서 유비쿼터스 서비스 관련 프로젝트를 수행하고 있다. 관심 연구분야는 다중 에이전트 시스템, 상황인식 기술, 의사결정지원시스템 등이다.