

시나리오 기반 요구사항 추출 도구의 설계

김치수^{1*}, 김영태¹, 공헌택¹, 임재현¹
¹공주대학교 컴퓨터공학부

Design of Scenario-based Requirements Extraction Tool

Chi-Su Kim^{1*}, Tae-Young Kim¹, Heon-Tag Kong¹ and Jae-Hyun Lim¹

¹Dept. of Computer Engineering, Kongju National University

요 약 사용자 요구사항 공학에서 가장 어려운 문제 중의 하나는 다양한 말단 사용자, 책임자와 소프트웨어 기술자 사이에 존재하는 상호이해의 불일치이다. 시나리오는 다양한 책임자에게 일부 추상 모델 대신 그들의 언어로 기술하고 검토할 수 있도록 해준다. 본 논문에서는 도출, 분석과 검증을 포함한 다양한 요구사항 공학 단계에 시나리오를 사용하는 진보적이고 반복적으로 삽입되는 프로세스를 제안한다. 프로세스 모델은 본 논문의 TRES 시스템에 적용되었다. 제안하는 시스템에서는 기존 도구들의 단점을 없애기 위해 시제품, 시나리오와 사용 사례를 하나의 포괄적인 프레임워크로 조합한다.

본 시스템은 시나리오 기반 요구사항 추출을 위한 XML 기반 시스템이다. TRES 시스템 내에서 시나리오는 사용의 문맥을 포함하여 사용자와 작업에 관한 정보를 얻을 수 있다. 본 TRES 시스템에서 시나리오는 XML 기반의 데이터 베이스에 보관되고 XML 표기법을 사용하여 기술된다.

Abstract One of the most difficult problems in user requirement engineering is the communication gap that exists between different end-users, stakeholders, and software engineers. Since scenarios allow different stakeholders to describe and review the problem in their own language instead of some abstract model, they are also a solution to the problem. We propose a progressive, iterative and interleaved process of using scenario at different requirement engineering stages including elicitation, analysis and validation. This process model has been applied to our TRES System. In our proposed system, we combine prototype, scenarios and use cases in a single and comprehensive framework to avoid most of the shortcomings in other tools.

Our system is an XML-based system for scenario-driven requirement engineering. Within the TRES system, scenarios are stories capture information about users and their tasks, including the context of use. In our TRES system, scenarios are stored in XML-based database and described using XML notation.

Key Words : Requirements Extraction, Scenario, Use Case

1. 서론

Moltke와 Clausewitz에 따르면, 시나리오의 개념은 이미 2000년 전부터 존재해왔다[1]. 초기의 시나리오는 주로 군인의 전략적 계획으로 구현되었다. 1970년대 초기 시나리오 개념은 경제 분야에서 사용되어왔다. 즉 군인의 전략적 계획의 모델이 비즈니스 환경에 적용된 것이다.

현재, 시나리오는 소프트웨어 공학, 인간과 컴퓨터의 상호작용, 디지털 예술, 조직 운영과 같은 다양한 영역과 학문에 적용되고 있다.

소프트웨어 공학에서 시나리오는 요구사항을 도출하고, 분석하고, 모호성을 발견하고, 정의된 특징들 사이의 불일치를 막고, 요구사항을 확인하고 검증하기 위해 사용된다[2].

본 연구는 공주대학교 자체기술연구 과제로 수행되었음

*교신저자 : 김치수(cskim@kongju.ac.kr)

접수일 09년 02월 02일

수정일 (1차 09년 04월 26일, 2차 09년 06월 29일)

계재확정일 09년 7월 22일

Sutcliffe는 시나리오를 다음과 같이 정의했다. 이벤트의 한 단계는 사용 사례의 하나의 경로이다. 많은 시나리오가 하나의 사용 사례를 위해 정의될 것이고, 각각의 시나리오에는 일어날 수 있는 이벤트의 예를 표현한다. 각 시나리오에는 정상적이거나 비정상적인 동작을 기술할 것이다[3].

요구 사항 공학을 위한 많은 도구들이 존재하지만 이들 중 일부는 DOORS, CaliberRM, Rational RequisitePro와 같은 정보 관리와 추적 도구이고, 일부는 소프트웨어 시스템의 설계 모델링을 위한 AxiomDsn과 용어집, 요구 사항, 참조와 같은 문서화 관리를 위한 EasyRM requirement manager와 같은 CASE 도구이다. 이러한 도구들은 시나리오와 직접 관계가 없다.

요구사항 공학을 위한 기존 상용 도구는 DOORS처럼 시나리오 분석을 지원하지 않거나 Rational ROSE처럼 사용 사례 편집 기능만을 제공한다[4].

본 논문에서는 사용자 및 책임자와 소프트웨어 기술자 사이에 존재하는 불일치를 줄이기 위하여 시나리오로부터 사용자 요구사항을 도출하고, 검증하고, 공유하는 도구인 TRES(Tool for Requirements Extraction based on Scenario) 시스템을 제시함으로써 기존의 시나리오를 다루는 요구사항 공학 도구의 단점을 수용하고자 한다.

이 시스템은 시나리오, 사용자, 사용 목표, 프로토타입, 사용사례의 5개의 구성 요소를 갖으며, 시나리오 구성 요소는 시나리오 편집과 시나리오 분석과 같은 여러 가지의 특징을 제공한다.

2. 관련 연구

요구사항 공학에서 시나리오를 다루는 도구는 몇 되지 않는다. 다음은 시나리오를 다루는 요구사항 공학을 위한 몇 개의 도구를 소개하며 각 도구의 단점도 소개한다. 본 논문에서는 이러한 도구들의 단점을 수용하기 위한 요구사항 공학 도구를 제안하고자 한다.

2.1 NDRASS

옥스퍼드 브룩스 대학교 Hong Zhi와 Lingzi Jin은 요구사항 분석 지원 시스템인 NDRASS에서 시나리오 기반 요구사항 분석을 위한 자동화 도구를 소개했다. 그들은 시스템의 사용에 합리적으로 발생할지도 모르는 일반적인 특징들의 일련의 상황으로 시나리오의 개념을 정의했고, 시나리오의 특성을 사용자 에이전트, 사용 목적, 동작 조건 측면에서 정의하였다[5]. NDRASS 시스템은 요구사

항 정의 언어 NDRDL-2를 갖는다. NDRDL-2 언어를 위한 파서는 언어를 자동화 도구에 의해 처리될 수 있는 내부 형식으로 변환하기 위해 개발되었다.

이 도구는 시나리오가 상이한 관점의 DFD, ERD, STD를 사용하여 표현되기 때문에 시나리오 사이의 불일치와 모호성, 시나리오의 복잡도가 증가한다는 단점이 있다.

2.2 PRIME-CREWS

PRIME-CREWS는 시나리오를 사용하여 협력적인 요구사항 공학을 하기 위한 통합 모델링 환경 프로세스이다. PRIME-CREWS 환경은 목표 모델의 구조와 모델 검증에 시나리오 사용을 지원한다. 이 환경에서 시나리오는 현재와 향후의 시스템 사용의 구체적 예를 표현한다[6]. PRIME-CREWS는 peter Haumer가 제안한 접근 방법을 사용한다.

그러나 이 접근 방법은 모든 프로젝트에 동등하게 적용되지 않는다는 단점이 있다. 기능이 잘 관찰될 수 있는 오래된 시스템, 그리고 새로운 시스템에서 제공하는 큰 기능, 혹은 새로운 시스템의 새로운 목표를 끌어내기 위해 시스템의 단점을 관찰하는 프로젝트에 적합하다.

2.3 CREWS-SAVRE

CREWS-SAVRE 도구는 Sutcliffe에 의해 제안된 4단계 방법을 지원하기 위해 실행된다. 이 방법은 객체지향 개발을 위한 사용사례 접근방법과 통합 시나리오 기반 요구사항 공학을 위한 방법이다. CREWS-SAVRE는 사용 사례로부터 시나리오의 자동 생성 등의 중요한 기능을 지원한다[3].

CREWS-SAVRE는 시스템 요구사항을 확인하는 두 가지 방법을 제공한다. 첫 번째로 동작, 이벤트, 시스템 요구사항의 패턴을 명시한 검증 구조를 적용한다. 두 번째로 시스템 요구사항의 검토와 검증을 위해 요구사항 문서와 함께 각 시나리오를 사용자에게 제출한다.

그러나 설계자들은 보통 자신들이 이해하기 쉽고, 설계 단계를 용이하게 하는 모델링 언어와 같은 정밀한 언어로 요구 사항 문서를 기술하기 때문에 사용자가 요구사항 문서를 이해하는 것은 더 어렵다.

2.4 시나리오 Plus

Ian alexander에 의해 개발된 시나리오 Plus는 목표를 생성하고, 모델링 한다. 또한 배경 시나리오를 추출하고, 확인하고, 실행한다[7].

자신의 요구 사항을 소프트웨어 개발자에게 기술하고 싶어 하는 책임자를 위해 설계된 시나리오 Plus는 시나리

오 기반 요구사항 도출 및 분석에 사용되는 DOORS(Dynamic Object Oriented Requirements System)을 사용 가능하게 하는 사용 사례 툴킷, 다이어그램 툴킷, 확장 툴킷의 애드온 도구로 구성된다[7].

DOORS는 Telelogic사에 의해 개발된 정보 관리와 추적 가능 도구로 Telelogic사의 웹 사이트에 기술된 바와 같이 요구 사항은 DOORS 내에서 이산 객체로 처리된다 [8].

3. 시나리오 기반 요구사항 추출 프로세스

본 논문에서는 책임자가 사용하는 자연어와 개발자가 사용하는 모델링 언어 사이의 불일치를 줄이기 위해 점진적이고 반복적인 요구사항 추출 프로세스 모델을 제안한다. [그림 1]은 TRES 시스템의 실행순서를 나타낸다. TRES는 요구 사항 도출을 지원하는 도구로서 다양한 유형의 사용자를 포함하며 다음과 같은 작업을 포함한다.

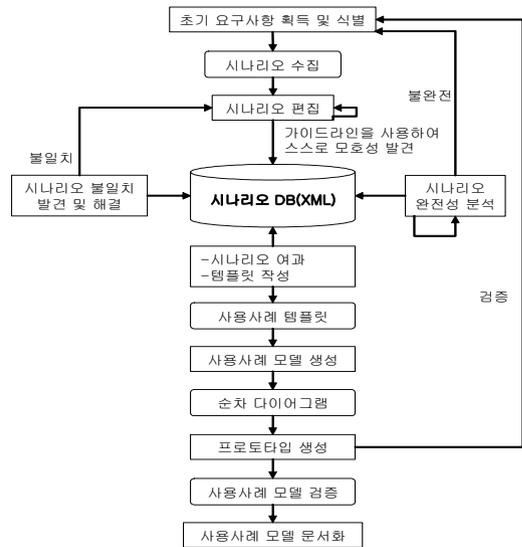
3.1 초기 요구사항의 획득 및 식별

최종사용자와 책임자로부터의 기본적인 요구 사항을 획득하고, 시나리오의 자연어 설명으로 요구 사항을 제공한다. 그 다음 환경, 에이전트, 목표를 분석하여 일련의 시나리오를 식별한다. 시나리오 식별 방법은 Jacobson[9]과 Maiden [10] 등의 방법을 사용한다.

요구 사항 내용은 소프트웨어의 목적, 범위와 일반적인 제약사항, 시스템의 총체적인 기능의 개요, 기능적인 요구사항, 비 기능적인 요구 사항, 도메인 지식을 포함 한다.

일반적으로 최종사용자와 책임자는 소프트웨어 개발 지식이 없기 때문에, 일부 정보를 무시할 수도 있지만 이것은 후반의 요구 사항 개발에 사용될 수도 있다. 게다가, 제공되는 정보가 항상 적절하게 구성되는 것은 아닐 것이다. 그러므로 TRES에서 최종사용자와 책임자는 자주 사용성 전문가의 도움을 받아야 한다.

사용성 전문가는 인터뷰와 설문 조사와 같은 방법을 사용하여 가이드라인에 따라 시나리오를 생성한다. 그때부터 시나리오는 유용하고 분석에서 서 보다 용이하게 사용된다.



[그림 1] TRES 시스템의 시나리오 구성 흐름도

3.2 시나리오 편집

요구 사항을 획득하고, 식별하고 난 후에 얻어진 시나리오는 XML 표기법으로 기술될 것이고, 시나리오 데이터베이스에 저장된다.

시나리오 내용에 추가적으로, 시나리오의 ID, 작성자, 작성일, 마지막 수정일과 같은 부가적인 정보 역시 데이터베이스에 저장된다. 데이터베이스에 저장되기 전에, 시나리오는 어떤 모호성이 있는지 발견하고 해결하기 위해 스타일과 내용 가이드라인을 사용하여 확인되어야 한다.

또한, 시나리오의 각각의 절은 Rolland의 텍스트 구조로 기술된다[11]. 이것은 후후 시나리오 분석과 사용 사례 모델 생성에 유용하다. 그러나 시나리오의 이러한 편집은 시나리오가 정제되고 난 후에 이루어져야 한다.

시나리오가 정제될 때 절에서 명사, 동사, 형용사, 부사, 전치사, 대명사, 관사를 무시하거나 제거할 수도 있다. 명사는 요구 사항 수집과 식별 단계에서 대명사로 대체되었어야 한다. 계속하여, 주어, 본동사, 목적어와 구조 해석을 위해 필요한 보어는 명사, 동사, 관사로부터 추출 된다.

3.3 시나리오 정제

프로세스 모델의 다음 단계는 편집된 시나리오를 정제하는 것이다. 첫 번째로 불일치를 발견하고 해결한다. 두 번째로 시나리오의 완전성을 분석한다. 본 시스템에서 사용성 전문가는 정제 작업에 대한 책임이 있다. 이 단계를 마치게 되면 명사, 동사, 형용사, 부사, 전치사, 대명사,

관사의 모음을 얻을 것이다.

1) 불일치 발견

대부분 복잡한 소프트웨어 시스템을 위해, 많은 시나리오가 있을 것이라고 인식한다. 시나리오는 사용 사례의 특별한 경로이다. 몇 개의 시나리오는 하나의 사용 사례에 대한 설명일 수도 있다. 그리고 중복되는 것들이 있을 지도 모른다. 따라서 그것은 시나리오 사이의 불일치 가능성을 증가시킨다.

또한, 여러 책임자가 같은 기능에 대해 모순되는 요구사항을 제공할지도 모른다. 게다가 최종사용자와 책임자는 그들의 용어로 요구사항을 표현하고, 그들의 이해력에 따라 용어의 모호성을 초래할 수도 있다.

불일치를 발견하고, 해결하기 위해서 채택할 수 있는 몇 가지 접근방법이 존재한다. Keller가 제안한 접근방법은 시나리오 사이의 중복을 해결하기 위해 같은 사용 사례에 주석을 추가한 모든 XML 형식의 시나리오를 병합한다[12].

사용성 전문가는 하위어, 상위어, 동의어를 발견하기 위해 편집된 시나리오를 분석하고, 언어의 불일치를 줄이기 위해 용어집으로부터 일관되고 대체 가능한 것으로 바꾼다.

2) 시나리오 완전성 분석

어쩌면 최초의 실행에서 잠재적인 소프트웨어를 위해 충분한 시나리오를 구성하지 않았을 수도 있다. 그 결과 빠져 있는 요구 사항을 포함시키기 위해 시나리오 수집과 식별 단계로 되돌아가게 된다. 게다가, 시나리오 절과 문장에 불완전한 의미가 있을지도 모른다.

Rolland는 불완전한 의미를 발견하고, 해결하기 위해 적용할 규칙을 제안했다[11]. 예를 들어, 완료 규칙은 예시된 절의 언어 패턴이 빠진 요소가 있으면 작성자는 빠진 요소를 제공할 것을 요청하는 것을 말한다.

[그림 2]는 완료 규칙 CO1의 예를 보여준다. 이것은 모든 동작 V에 대해 에이전트가 객체에 대해 이 동작 V를 수행한다면, 객체 O가 있음을 뜻한다. 그러나 에이전트 요소가 빠져 있다면, 저자는 절을 완전하게 하기 위해 빠진 요소의 제공을 요청할 것이다[11].

CO1 : √V , ∃O: Action (V) [Agent:? ; Object:O] ASK(<<Complete : V by...(agent of the action)>>)

[그림 2] 완료 규칙의 예

3.4 시나리오 여과 및 템플릿 작성

잠재적인 소프트웨어의 액터(에이전트)와 작업(동작), 그것들 사이의 관계, 제약사항을 이해하기 위해 편집된 시나리오를 분석한다. 바꾸어 말하면, 비 형식적인 표현인 시나리오와 형식적인 사용 사례 모델 사이의 차이를 줄이기 위해 텍스트의 의미를 이해하는 것은 필수적이다.

이 단계는 시나리오 내에서 텍스트의 의미를 추출하기 위해 의미 패턴의 사용을 제안한 Rolland의 접근방법에 의거한다[11]. 사례 문법은 에이전트, 객체, 수신자, 송신자와 같은 일련의 의미의 사례를 사용하여 의미를 표현한다.

의미 패턴은 텍스트의 많은 의미를 표현한다. 예를 들어 "고객이 주문을 전송한다."에 대한 의미 패턴은 Action (전송) [Agent:'고객'; Object:'주문']이다. 텍스트의 의미는 다른 표현(예를 들면 "고객이 주문을 전송한다.", "주문이 고객에 의해 전송된다.", "고객에 의한 주문의 전송" 등)에 의해 표현된다.

필터링 과정 동안, 의미 패턴은 의미 사례(에이전트, 객체, 수신자, 발신자)를 절과 문장 각각의 요소에 연관시키기 위한 템플릿으로 작용한다. 의미 패턴은 동작, 에이전트, 객체, 객체의 초기상태와 최종상태, 두 객체 사이의 관계, 제약사항, 조건, 상태, 동작들 사이의 관계와 같은 사용 사례 모델에서 필요한 요소의 여러 의미를 정의한다.

사용 사례 요소의 의미는 의미 패턴에 의해 차례로 표현된다. "주문이 고객에 의해 전송된다."는 전달 행동으로서의 이 시나리오 절은 의미적으로 완전하지 않다. 완료 규칙을 적용하면 새로운 시나리오 "주문은 고객에 의해 X 회사로 전송된다."가 될 것이다. 텍스트 구조는 다음과 같이 기술된다.

[['order'](**Subject**)Object['is sent'](**Main Verb**) Communication['by customer'] (**Complement**) Agent-Source['to company X'] (**Complement**) Destination](**VG passive**)Communication

텍스트 구조에 분석 규칙을 적용하여 생성된 의미 패턴의 예는 다음과 같다.

Communication('send') [Agent:'customer'; Object:'order'; Source:'customer'; Destination:' company X']

3.5 사용 사례 모델 생성

시나리오를 여과하고 난 후에, 템플릿을 사용하여 시

나리오에 관한 정보를 기록한다. 우리는 시나리오를 표현하기 위해 의미 패턴을 사용하며, 의미 패턴은 동작, 에이전트, 객체, 객체의 초기상태와 최종상태, 두 객체 사이의 관계, 제약사항, 조건, 상태, 동작들 사이의 관계(순차, 병렬, 반복)와 같은 사용 사례 모델에서 필요한 요소의 여러 의미를 정의한다. 그러므로 사용 사례 모델에서 필요한 요소의 의미는 템플릿에 기록된다. 기록 정보는 사용 사례 모델을 생성하기 위해 사용될 것이다.

같은 지식을 나타내기 위해 많은 다른 방법을 제공하는 자연어와 비교하면, 의미 패턴은 상당한 양의 텍스트의 의미를 지원하기 위해 유일한 난해한 수준의 표현을 제공한다[11]. 따라서 우리가 시나리오를 제공하기 위해서 의미 패턴을 사용할 때, 이 속성은 불일치 가능성 축소와 모호성 해결을 돕는다.

3.6 사용 사례 모델의 검증 및 문서화

사용 사례 모델이 생성되고 난 후에, 책임자는 이 모델을 확인하고, 피드백을 제공하는 것에 대해 책임이 있을 것이다. 피드백은 사용성 전문가와 요구 사항 엔지니어가 사용 사례모델을 정제할 뿐만 아니라, 시나리오의 불일치와 불완전을 다루는 것을 돕는다.

프로토타입은 책임자와 소프트웨어 기술자 사이의 불일치를 줄이는 해결책이다. 그러므로 책임자가 검증을 더 용이하게 하기 위해 신속한 프로토타입을 생성하고, 검증과 평가를 위해 그것을 최종사용자 또는 책임자에게 제공한다.

Mohammed와 Kell은 시나리오로부터 순서도를 이끌어내고, 그 다음 순서도로부터 프로토타입을 생성하는 접근 방법을 제안했다[12]. 사용 사례 모델의 생성에서 사용 사례와 관련된 시나리오가 있기 때문에, 시나리오로부터 순서도를 만들고, 검증을 위해 순서도로부터 프로토타입을 생성하는데 Kell의 방법을 적용한다.

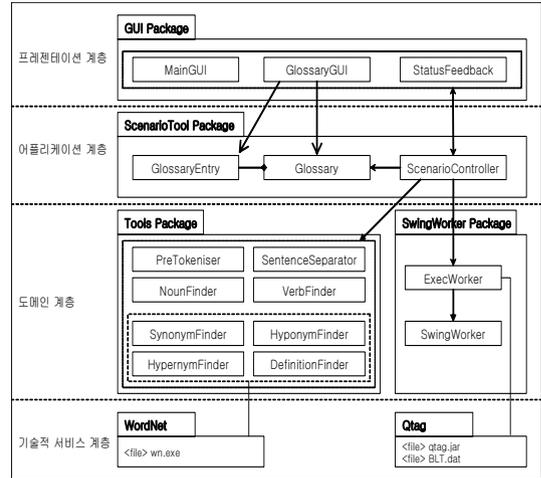
책임자의 조언과 제안에 의해 기존 시나리오가 변경되거나 새로운 시나리오가 추가될 수 있다. 어느 쪽이든, 프로세스 모델이 반복 실행된다. 반복은 요구사항 공학 기술자와 사용자가 동의할 때까지 계속된다.

4. TRES 프로토타입의 구현

[그림 3]은 TRES 시스템의 여러 패키지를 보여준다. 패키지는 여러 계층으로 구성되며 최상위 계층은 유저 인터페이스다. 최상위 계층 아래의 각 계층은 유저 인터페이스로부터 점점 더 추상화를 표현하고, 점차 더 기술

적인 클래스들로 구성된다.

GUI 패키지는 TRES가 디스플레이 되는 것을 관리하는 뷰이다. 도메인 계층과 기술적 서비스 계층의 패키지는 도구가 사용하는 데이터 또는 값을 관리하는 기능을 한다.



[그림 3] TRES 시스템의 구조

ScenarioTool 패키지는 사용자가 TRES와 상호 작용할 때 무엇이 일어나는지 결정하는 제어기이다. ScenarioController는 시스템의 중심으로 모든 GUI 클래스와 메인 클래스 사이의 메시지를 전달한다. ScenarioController는 메인 클래스에 대한 사용자의 입력을 받는다. 그 다음 메인 클래스는 ScenarioController에 정보를 리턴하고, ScenarioController는 GUI 클래스에 이 정보를 보낸다. GUI 클래스는 순서대로 사용자에게 정보를 디스플레이 한다. 따라서 TRES의 전체 제어가 ScenarioController에 달려있다.



[그림 4] 새로운 시나리오 추가 화면

TRES 시스템의 메인 인터페이스는 두 영역으로 나뉜다. 상단 영역은 TRES 시스템에서 지원하는 5가지 주요 기능들을 나타낸다. 그 중 Scenario 요소는 시나리오 편집에 관련된 요소이다. 하단 영역은 인덱스를 이용한 시나리오 검색, 추가, 편집 등의 작업 영역이다. [그림 4]는 새로운 시나리오를 추가하기 위한 화면으로 사용자가 UI의 필드를 채움으로써 시나리오에 관한 정보를 제공하면 시나리오는 시나리오 데이터베이스에 추가된다. 추가된 시나리오는 인덱스를 이용한 검색, 편집 등이 가능해진다.

다음은 TRES의 시나리오 데이터베이스에서 시나리오를 제공하기 위해 사용되는 XML 파일의 예이다.

```
<?xml version="1.0"?>
<!DOCTYPE ScenarioCollection SYSTEM
"ScenarioCollection.dtd">
<ScenarioCollection>
<scenarios id="1">
<ScenarioName>ABC</ScenarioName>
<CreatedDate>2007-07-07</CreatedDate>
<ModifiedDate>2007-07-18</ModifiedDate>
<Author>
<name>zerot</name>
<address>공주대학교 천안공대 컴퓨터공학과
</address>
<email>zerot@kongju.ac.kr</email>
<url>brain.kongju.ac.kr</url>
</Author>
<Notes>
<NotePos ln="20" col="50"></NotePos>
<NoteText>This is a memo</NoteText>
</Notes>
<HighLighter>
<startpos ln="10" col="15"></startpos>
<endpos ln="18" col="25"></endpos>
</HighLighter>
<BodyText>
사용자가 시나리오를 추가하고자 한다. 사용자는 시나리오명과 시나리오의 내용, 작성자의 이름, 작성자의 email, 작성자의 주소, 작성자의 URL을 입력한다. 모든 필드를 입력한 후 입력 버튼을 누르면 시나리오는 데이터베이스에 추가된다.
</BodyText>
</scenarios>
</ScenarioCollection>
```

5. 결론 및 향후 연구

본 논문에서는 책임자에게 쉽고, 책임자와 요구사항 공학자 사이의 불일치를 줄이고, 기존 도구의 단점을 극복하고자 TRES 시스템을 제안했다.

TRES 시스템은 사용 사례와 시나리오를 조합하고, 시나리오로부터 사용 사례를 추출해 낸다. 바꾸어 말하면 책임자가 필요한 요구 사항을 이야기하는 것처럼 표현한다. 그 다음, 수집된 시나리오를 분석하고, 사용 사례를 생성한다.

TRES 시스템은 사용자를 직접 포함하고, 보다 정확한 요구 사항을 제공한다. 이렇게 모아진 요구 사항은 보다 더 일관되고 완전성이 높게 된다.

본 논문에서는 데이터 유연성과 이식성을 늘리기 위해 시나리오를 저장하는 시나리오 데이터베이스에 현재 이용 가능한 가장 탁월한 메타 데이터인 XML을 채택했다.

TRES 프로토타입은 계속적으로 연구 중인 프로젝트의 일부이다. 이것을 통한 효과는 소프트웨어 기술자가 시나리오로부터 일련의 요구 사항을 반자동식으로 도출할 수 있다.

향후 해결해야 할 문제는 다음과 같다. 첫째 시나리오로부터 집단화나 연관과 같은 사용 사례 사이의 관계를 추출하는 방법. 둘째 일관성의 분석과 시나리오의 완전성 향상. 셋째 시나리오로부터 사용사례를 생성하기 위해 사용되는 템플릿의 향상 방법을 연구하는 것이다.

참고 문헌

- [1] Ute von Reibnitz, Scenario Techniques. Hamburg, New York, McGraw-Hill Book Company GmbH, 1998.
- [2] John M. Carroll, Making Use Scenario-Based Design of Human-Computer Interactions. London, England, The MIT Press Cambridge, Massachusetts, 2000.
- [3] Alistair G. Sutcliffe, Neil A. M. Maiden, Shailey Minocha, and Darrel Manuel, Supporting Scenario-based Requirements Engineering. IEEE Transactions on Software Engineering, val. 24, issue 12, (December), pp. 1072-1088, 1998.
- [4] Hong Zhu and Lingzi Jin, Scenario Analysis in an Automated Tool for Requirements Engineering, Journal of Systems and Software vol 61,no.2,(March), pp. 145-169, 2002.
- [5] Hong Zhu and Lingzi Jin, Scenario Analysis in an Automated Tool for Requirements Engineering,

Journal of Systems and Software vol 61,no.2,(March), pp. 145-169, 2002.

- [6] Peter Haumer, Klaus Pohl, and Klaus Weidenhaupt, Requirements Elicitation and Validation with Real World Scenes. IEEE Transaction on Software Engineering, vol. 24, no. 12(December), pp. 1036-1054, 1998.
- [7] Ian Alexander, In Scenario Plus. [Internet] May 10, 2002.
- [8] DOORS In Telelogic Company, 2001.
- [9] Ivar Jacobson, Object-Oriented Software Engineering: A Use-case Driven Approach, Addison-Wesley, 1992.
- [10] Neil Maiden, Shailey Minocha, Keith Manning, and Michele Ryan, A Software Tool for Scenario Generation and Use. In Proceedings of the Third International Workshop on Requirements Engineering: Foundation for software Quality. Barcelona, Spain, 1997.
- [11] Colette Rooland and Camille Ben Achour, Guiding the Construction of Textual Use Case Specifications. Data and Knowledge Engineering, vol. 25, no.3 (April), pp. 125-160, 1998.
- [12] Mohammed Elkoutbi and Rudolf K.Keller, User Interface Prototyping based on UML Scenarios and High-level Petri Nets, 2000.

김 치 수(Chi-Su Kim)

[정회원]



- 1984년 2월 : 중앙대학교 전자계산학과 졸업(학사)
- 1986년 8월 : 중앙대학교 대학원 전자계산학과 졸업(석사)
- 1990년 8월 : 중앙대학교 대학원 전자계산학과 졸업(박사)
- 1990년 9월 ~ 1992년 8월 : 공주교육대학교 전임강사
- 1992년 9월 ~ 현재 : 공주대학교 컴퓨터공학부 교수

<관심분야>

소프트웨어 개발 방법론, 온톨로지

김 영 태(Young-Tae Kim)

[정회원]



- 2000년 2월 : 공주대학교 전자계산학과 졸업(학사)
- 2002년 2월 : 공주대학교 전자계산학과 졸업(석사)
- 2004년 2월 : 공주대학교 컴퓨터공학과 박사과정 수료

<관심분야>

데이터통합, UML, 온톨로지

공 헌 택(Heon-Tag Kong)

[정회원]



- 1984년 5월 : Northeast Missouri State Univ. 전산학과(학사)
- 1987년 12월 : Utah State Univ. 전산학과(석사)
- 1998년 2월 : 단국대학교 전산통계학과(박사)
- 1988년 1월 ~ 1990년 3월 : 한국국방연구원 전산체계연구부 근무
- 1990년 4월 ~ 현재 : 공주대학교 컴퓨터공학부 교수

<관심분야>

병렬 알고리즘, 병렬처리 컴퓨터, 데이터베이스

임 재 현(Jae-Hyun Lim)

[정회원]



- 1986년 2월 : 중앙대학교 전자계산학과 졸업(학사)
- 1988년 8월 : 중앙대학교 대학원 전자계산학과 졸업(석사)
- 1998년 8월 : 중앙대학교 대학원 컴퓨터공학과 졸업(박사)
- 1998년 9월 ~ 현재 : 공주대학교 컴퓨터공학부 부교수

<관심분야>

상황인식, RFID/USN, 온톨로지, 인터넷 기술