

소프트웨어 프로덕트 라인을 위한 목표, 시나리오, 휘처 기반의 도메인 분석 방안

(A Domain Analysis Method for Software Product Lines Based on Goals, Scenarios, and Features)

김민성[†] 박수용^{**}
(Minseong Kim) (Sooyong Park)

요약 소프트웨어 재사용을 위한 새로운 패러다임으로서 대두되고 있는 소프트웨어 프로덕트 라인에 서 가장 중요하며, 기본이 되는 것은 바로 재사용 가능한 소프트웨어 자산을 개발하기 위한 프로덕트 라인의 공통성과 가변성 식별이라고 하겠다. 현재 이를 위해 휘처 중심의 도메인 분석 방법이 많이 사용되고 있으나, 이 방법은 휘처를 식별하고, 식별된 휘처의 근거를 제시하기 위한 체계적인 방법을 제공하지 못하고 있다. 또한 프로덕트의 공통성과 가변성 분석 결과가 프로덕트 라인 개발 조직의 최상위 수준 목표(goals)를 만족시키고, 그 근거를 보여줄 수 있어야 하지만 현재 이러한 부분에 대한 연구가 부족한 실정이다. 따라서 본 논문에서는 기존의 휘처 중심의 도메인 분석 방법에서의 문제점들을 해결하고, 보안하기 위해 프로덕트 라인을 위한 목표, 시나리오, 휘처 기반의 도메인 분석 방안을 제안하였다. 이것은 목표와 시나리오 그리고 휘처의 관계를 통하여 프로덕트 라인을 위한 도메인 요구사항 모델(DRM: domain requirements model)을 제시하고, 그러한 모델을 바탕으로 도메인 요구사항 모델링 방법(domain requirements modeling method)을 제안한다. 마지막으로 제안된 방안을 지원하는 도구 (IDEAS)를 설명하고, 이를 통해 주택 통합 시스템(HIS)에 적용함으로써 제안된 방법을 검증하였다. 제안된 방법은 체계적으로 휘처를 식별하고, 그에 대한 근거 및 공통성과 가변성에 대한 근거를 제공할 수 있을 것이다.

키워드 : 소프트웨어 재사용, 소프트웨어 프로덕트 라인(Software product lines), 요구공학, 도메인 분석, 휘처 중심의 도메인 분석 방법(Feature-oriented approach)

Abstract Software product lines (SPL) are recently an emerging software reuse paradigm, which helps organizations develop their products from reusable core assets rather than from scratch. For developing these assets, understanding commonality and variability (C&V) is essential. A feature-oriented approach has been used extensively for C&V analysis in the SPL. However, this contains no proposal to systematically identify features and provide the rationale for the features. Further, the approach does not directly show how the results of C&V analysis will satisfy an organization's high-level business goals and provide the rationale for the C&V. Therefore, this paper presents a domain analysis method for the SPL based on goals, scenarios, and features in order to overcome some of the deficiencies and limitations of the feature-oriented approach. In particular, the paper proposes a domain requirements model (DRM) and a domain requirements modeling method based on the DRM. This method has been applied to the home integration system (HIS) domain to demonstrate its feasibility with a supporting tool, namely IDEAS. Our approach makes it possible to systematically identify the features and provide the rationale for both the features and the C&V.

Key words : software reuse, software product lines, requirements engineering, domain analysis, feature-oriented approach

· 본 연구는 한국과학재단 특장기초연구(R01-2006-000-10536-0)지원으로 수행되었음

† 학생회원 : 서강대학교 컴퓨터학과
minskim@sogang.ac.kr

** 정회원 : 서강대학교 컴퓨터학과 교수
sypark@sogang.ac.kr

논문접수 : 2004년 8월 4일

심사완료 : 2006년 5월 16일

1. 서론

소프트웨어 재사용은 소프트웨어 공학의 가장 큰 목표 중의 하나이며, 소프트웨어 공학의 태동과 함께 그 시각을 같이하고 있는 가장 중요하고, 오래된 소프트웨어 공학의 개발 접근 방법이다. 현재 이러한 소프트웨어

재사용을 위한 새로운 패러다임으로서 소프트웨어 프로덕트 라인(Software product lines)이 대두되고 있다. 소프트웨어 프로덕트 라인은 체계적이고, 계획된 보다 큰 범위의 재사용을 가능하게 하는 소프트웨어 개발 프레임워크로서 단일 시스템을 처음부터 새롭게 개발하는 것이 아니라 하나 이상의 여러 제품들을 재사용 가능한 소프트웨어 자산(software asset)을 통해 개발하는 것을 의미한다[1-3]. 이러한 소프트웨어 프로덕트 라인에서 가장 중요하며, 기본이 되는 것은 바로 특정 도메인(domain) 즉, 프로덕트 라인에서 여러 프로덕트들 간의 공통성(commonality)과 가변성(variability)을 어떻게 정확하게 식별하고, 표현할 것인가 하는 점이다[4-7]. 이는 프로덕트 라인에 포함된 각각의 프로덕트들을 위해 사용될 재사용 가능한 자산을 개발할 때, 이러한 공통성과 가변성을 이해하는 것이 가장 본질적이기 때문이다.

현재 프로덕트 간의 공통성과 가변성을 식별하기 위한 방법으로 휘처 중심의 도메인 분석 방법(feature-oriented approach)이 광범위하게 사용되고 있다[3,4,5,8]. 그러나 이 방법은 프로덕트 라인 개발 시에 프로덕트 라인에서 휘처가 식별되고, 정의되어야 하는 경우, 휘처 식별을 위한 체계적인 방법을 제공하지 못하고 있다. 특히, 어떤 조직에서 성숙되지 않은 도메인이나 새로운 시장에 진입하기 위해 지금까지는 존재하지 않았던 프로덕트를 목표로 할 경우에 휘처 식별에 어려움이 있으며, 휘처 중심의 방법을 적용하는 것이 적절하지 못할 수도 있다[9-11]. 또한 프로덕트 라인을 위한 도메인 분석의 결과인 도메인 모델 즉, 공통성과 가변성 분석의 결과가 프로덕트 라인 개발 조직의 최상위 수준 목표(high-level goals)들을 만족시키도록 식별되고, 그 근거를 보여줄 수 있어야 하지만 기존의 휘처 중심의 방법에서는 이러한 부분에 대한 연구가 부족한 실정이다.

기존의 요구공학(requirements engineering) 분야에서는 개발 조직의 목표(goals)와 사용자의 요구를 만족시키기 위한 요구사항을 추출하고 식별하기 위한 효과적인 방법으로서 목표와 시나리오 기반의 방법(goal-driven and scenario-based requirements analysis)이 많이 사용되고 있다[12-15]. 그러나 이러한 요구사항 분석 방법은 단지 단일 시스템을 개발하기 위한 요구사항 분석 방법으로서 프로덕트 라인을 고려하진 않고 있다.

따라서 기존의 휘처 중심의 도메인 분석 방법에서의 문제점들을 해결하고, 보완하기 위해서 본 논문에서는 소프트웨어 프로덕트 라인을 위한 목표, 시나리오, 휘처 기반의 도메인 분석 방법을 제안하고자 한다. 이는 기존의 도메인 분석 방법에서의 휘처에 기존의 요구공학에서 사용되고 있는 목표와 시나리오 개념을 통합하여 프

로덕트 라인을 위한 도메인 요구사항 모델(domain requirements model)을 제안하고, 그러한 모델을 바탕으로 도메인 요구사항 모델링 방법(domain requirements modeling method)을 제시한다.

아울러, 본 논문에서는 제안된 방법을 지원하기 위한 도구 즉, **IDEAS** (**I**ntegrated **D**omain **r**equirements **A**nalysis for **S**oftware product lines)를 개발하였다. 따라서 제시된 도메인 요구사항 모델과 모델링 방법은 이를 지원하는 도구와 함께 휘처 식별을 위한 체계적인 방법이 될 수 있을 뿐 만 아니라 식별된 프로덕트 휘처와 프로덕트 라인의 공통성과 가변성 분석에 대한 근거를 제공하는 체계적이고, 구체적인 도메인 분석 방법이 될 것이다. 또한 이를 통해 사용자, 고객, 개발자와 같은 이해 관계자(stakeholder)들 사이에서의 효과적인 의사소통이 가능해질 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에 대한 관련 연구 및 문제점에 대해 살펴보고, 3장에서는 소프트웨어 프로덕트 라인을 위한 도메인 요구사항 모델을 제안하며, 4장에서는 이러한 모델을 바탕으로 도메인 요구사항 모델링 방법을 제시한다. 5장에서는 3장, 4장에서 제안된 도메인 요구사항 모델과 모델링 방법을 지원도구를 통해 주택 통합 시스템(HIS)에 적용하여 평가해 보고, 6장에서는 결론과 향후 연구 방안을 논의하고자 한다.

2. 관련 연구 및 문제점

2.1 프로덕트 라인에서의 도메인 분석 및 휘처 중심의 방법

소프트웨어 프로덕트 라인에서 공통성과 가변성을 분석하기 위해 사용되는 도메인 분석 방법들로는 휘처 중심의 방법[5,6,8,16,17], 객체 지향 방법인 Reuse-Driven Software Engineering Business(RSEB)[18], 그리고 FODacom[19], FeaturSEB[20], PLUS[21]와 Product Line Analysis(PLA)[11]가 있다. 이 외에도 많은 도메인 분석 방법들이 존재하나 현재 휘처 중심의 도메인 분석 방법이 1990년 Software Engineering Institute(SEI)에 의해서 FODA(Feature Oriented Domain Analysis)[8]가 소개된 이후 산업계와 학계에서 광범위하게 사용되고 있다[3,4,5,7]. 휘처 중심의 도메인 분석 방법의 핵심은 도메인 즉, 프로덕트 라인에서 관련된 제품군의 공통성과 가변성을 휘처의 관점에서 분석하고, 이러한 휘처를 이용하여 재사용 가능한 소프트웨어 자산을 개발하는데 있다. 여기서 프로덕트 휘처는 고객과 개발자가 모두 이해하는 프로덕트의 본질적인 추상화로서 일반적으로 고객과 개발자는 프로덕트가 가지고 있거나 제공해야 하는 휘처의 관점에서 프로덕트의 특징

을 말하게 된다.

이러한 휘처 중심의 방법에서는 프로덕트 라인 개발 시 프로덕트 개발 조직 내에 관련된 프로덕트들이 존재하지 않거나 그것들과 관련된 휘처 집합이 존재하지 않을 경우, 휘처 모델링을 하기 위해서 프로덕트 라인의 휘처들이 우선 식별되고, 정의되어야 한다[1]. 이 때, 휘처 중심 방법에서는 휘처 식별 프레임워크(feature identification framework)으로써 휘처 범주(feature categories)를 제공하는 것에 의해 휘처들을 식별하고 있다. 즉, 성숙되고, 안정된 도메인의 경우, 도메인 용어를 분석하고 휘처 범주에 의해 휘처들을 식별하고 있다[4,9]. 또한 [5]에서는, 마케팅 & 프로덕트 계획(marketing and product plan, MPP)의 수립 및 정련을 통하여 휘처의 관점에서 프로덕트의 특징을 식별하고, 초기 휘처 모델을 만들게 된다. 그러나 성숙되지 않거나, 새로운 도메인의 경우에 휘처 중심 방법에서는 도메인 전문가들 사이에 존재하는 도메인 용어 사용에 대한 모호성 때문에, 휘처 식별 이전에 도메인 용어에 대한 표준화와 도메인 영역 결정(domain scoping)이 선행되어야 한다고 말하고 있다[4,9]. 하지만 그러한 상황에서는 휘처 식별의 바탕이 되는 도메인 지식과 책, 사용자 매뉴얼, 설계 문서 그리고 프로그램 코드 등과 같은 관련 자료들이 부족하여 휘처 식별이 쉽지 않다. 특히, 어떤 조직에서 새로운 시장에 진입하기 위해 지금까지는 존재하지 않았던 프로덕트를 목표로 할 경우에는 그 조직이 목표로 하는 프로덕트에 관련된 프로덕트들이 많이 존재하지 않을 뿐만 아니라, 존재한다 하더라도 그 조직에서 목표로 하는 프로덕트와는 많이 다른 경우가 대부분일 것이므로 기존의 방법으로는 많은 어려움이 있다[10]. 또한 조직 내에서 항상 도메인 지식 추출을 위한 도메인 전문가가 유용하지 않거나, 잘 이해되지 않은 시스템의 특징에 대해서 분석을 할 때 휘처 중심의 방법을 적용하는 것은 적절하지 못하게 된다[11].

FODACom[19], FeatuRSEB[20], 그리고 PLUS[21]와 같은 방법에서는 유스케이스(Use Case) 모델과 휘처 모델을 함께 사용하여 도메인 분석을 하고 있으며, PLA[11]에서는 프로덕트 라인 요구사항 분석으로서 휘처 모델과 함께 유스케이스 모델(Use Case model)과 객체 모델(object model)을 사용하고 있다. 그러나 이러한 방법들 역시 휘처 식별을 위한 체계적인 방법과 식별된 휘처에 대한 근거를 보여주기 위한 체계적이고, 구체적인 방법을 제시하지는 못하고 있다.

2.2 목표와 시나리오 기반의 요구사항 분석

기존의 요구공학 분야에서는 개발 조직의 목표와 사용자들의 요구를 만족시키기 위한 요구사항을 추출하고, 그 근거를 제공하기 위한 효과적인 방법으로서 목표와

시나리오 기반의 방법이 많이 사용되고 있다[12-15].

특히, 시스템들이 비즈니스와 조직의 문제에 대한 해결책을 제공하기 위해 중요하게 사용됨에 따라 시스템들과 그 시스템이 수행될 환경 사이에서의 관계가 목표를 기반으로 표현되고, 분석되고 있다[22]. 이는 목표 중심의 요구사항 분석(goal-driven requirements analysis)이 소프트웨어 개발 프로젝트의 근거가 되는 조직적인 목적 즉, 시스템이 왜 개발 되어야 하는지에 초점을 두고 있기 때문이며, 이러한 조직적인 목적과 실제 소프트웨어 요구사항 사이의 상호 의존성을 식별하는 것에 의해 요구사항의 근거를 제공할 수 있기 때문이다[14,23-26]. 또한 목표 모델링(goal modeling)은 상위 수준의 목표를 성취하기 위한 하위 수준의 요구사항들을 식별하는 것에 의해 요구사항 추출의 효과적인 방법이 된다[15,24-27]. 여기서 목표 식별을 위한 목표 정제 과정(refinement)은 상위 수준의 전략적인 목표로부터 하위 수준의 기술적인 요구사항까지의 연결(traceability link)을 제공하게 된다[24]. 반면에, 시나리오 기반의 요구사항 분석 방법(scenario-based requirements analysis)[12,13,28]은 시스템의 요구사항을 이야기 예제 중심의 형식을 가지는 시나리오를 통해 사용자 입장에서는 이해하기 쉽고, 분석가 입장에서는 보다 구체적인 요구사항 명세가 가능하도록 한다[12].

따라서 최근에는 이러한 목표와 시나리오를 결합한 분석 방법[12,14,15,27]이 제안되고 있으며, 이러한 분석 방법은 목표와 시나리오 간의 양방향 관계(unidirectional relationship)을 바탕으로 요구사항을 추출하고, 식별하는데 사용되고 있다[15,29].

그러나 기존의 목표와 시나리오 기반의 요구사항 분석 방법은 단지 단일시스템 만을 위한 요구사항 분석 방법이다. 따라서 프로덕트 라인에서 분석되어야 하는 여러 프로덕트들 즉, 관련된 제품군에 대한 고려하고, 있다. 이는 프로덕트 라인에 포함된 모든 프로덕트에 공통적인 요구사항(common requirements)과 각각의 프로덕트 마다 다른 요구사항(variation requirements)을 모두 포함하는 도메인 요구사항에 대해서는 고려하지 않고 있음을 의미한다.

따라서 본 논문에서는 기존의 휘처 중심의 도메인 분석 방법에서의 문제점들을 해결하고, 보안하기 위해 소프트웨어 프로덕트 라인을 위한 목표, 시나리오, 휘처 기반의 도메인 분석 방법을 제안한다.

3. 프로덕트 라인을 위한 도메인 요구사항 모델

본 논문에서 제안하는 도메인 분석 방법에서는 프로덕트 라인을 위한 도메인 요구사항 모델(DRM, domain requirements model)을 제시한다. 이 모델은 목표와 시

나리오 그리고 휘처에 의해 구성된 도메인 요구사항을 바탕으로 하고 있으며, 그것들 사이의 관계를 통하여 정의된다. DRM에 존재하는 관계들로는 구성(composition), 대체(alternative), 선택(optional)과 정제(refinement) 그리고 첨부(attachment)가 있다. 그림 1은 이러한 DRM의 개괄적인 모습을 보여주고 있다.

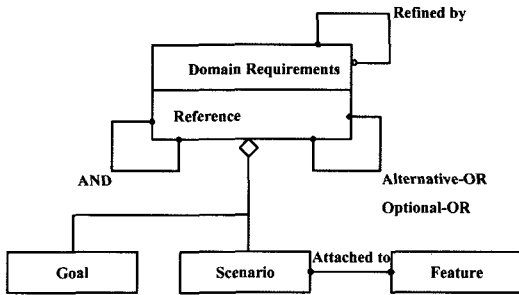


그림 1 개괄적인 도메인 요구사항 모델

이 모델은 [15,29]에서 목표와 시나리오에 의해 구성된 요구사항 단위 모델(requirements chunk model)을 바탕으로 하고 있으나, 단일 시스템의 요구사항이 아닌 프로덕트 라인을 위한 도메인 요구사항을 분석하기 위해서 휘처 중심 방법의 휘처를 통해 확장되었다.

3.1 목표, 시나리오 그리고 휘처의 개념

도메인 요구사항 모델(DRM)은 요구공학 분야에서 요구사항 분석을 위해 사용되고 있는 목표와 시나리오, 그리고 휘처 중심의 방법에서 사용되고 있는 휘처에 의해 구성된다. 따라서 본 절에서는 각각의 개념에 대해 살펴보고, 각 개념 정의를 바탕으로 각각이 서로 어떻게 관련되는지에 대해 알아보려 한다.

본 논문에서는 기존의 목표에 대한 정의들[14,15,24]을 바탕으로 프로덕트 라인 관점에서 다음과 같이 목표를 정의한다.

- 목표란 “프로덕트 라인에서 비즈니스, 조직 혹은 프로덕트의 최상위 수준의 목적(objective)이며, 프로덕트 라인의 이해 관계자들이 성취하고자 하는 것”이다.

시나리오는 기존 논문[12,15,28]에서 정의하고 있는 개념을 바탕으로 프로덕트 라인 관점에서 다음과 같이 정의한다.

- 시나리오는 “프로덕트 라인에서 어떤 목표를 성취하기 위한 의도적인 행위”이다.

2.2절에서 설명했듯이 최근에는 이러한 목표와 시나리오를 결합한 분석 방법이 제안되고 있다. 이러한 방법에서는 시나리오가 실제 상황 또는 특정한 어떤 행위를 묘사하기 때문에, 실제의 구체적인 요구사항들을 추출하게 되며, 그러한 시나리오의 수행에 의해 목표가 성취되

는 관계를 가지게 된다.

다음으로 프로덕트 휘처는 고객과 개발자가 모두 이해하는 프로덕트의 본질적인 추상화이며, 고객과 개발자는 일반적으로 프로덕트가 가지고 있거나, 제공해야 하는 휘처의 관점에서 프로덕트의 특징을 말하게 된다[5]. 따라서 본 논문에서는 프로덕트 휘처를 다음과 같이 정의한다.

- 프로덕트 휘처는 “프로덕트 라인에서 이해 관계자들에게 보여지는 프로덕트 들의 특징이며, 프로덕트 기능의 추상화”이다.

또한 이 논문에서는 프로덕트 휘처를 어떤 행위(behavior)에 의해 성취되는 결과 또는 영향으로 볼 수 있다고 제안한다. 여기서 행위라고 하는 것은 프로덕트 라인에서 이해 관계자들에게 보여지는 프로덕트의 외부적인 행위(external behavior) 혹은 내부적인 행위(internal behavior)를 의미하게 된다. 예를 들어, [8]에서 휘처의 예로서 제시된 자동차의 자동 변속기 휘처의 경우를 생각해 보면, 자동 변속기 휘처라고 하는 것은 기체가 자동으로 클러치와 기어변속을 하는 행위에 의한 결과로서 볼 수 있으며, 이 휘처는 클러치와 기어변속을 하는 행위에 의해 자동차에서 출력축의 회전 속도 및 회전력을 바꾸게 된다. 그러므로 휘처들은 실 세계의 상황 또는 행위를 묘사하는 시나리오들과 관련될 수 있으며, 결과적으로 목표와 휘처는 시나리오를 통해 간접적으로 관련을 맺게 된다.

3.2 목표, 시나리오 그리고 휘처의 관계

고객과 개발자들은 일반적으로 프로덕트가 가지고 있거나 제공해야 하는 휘처의 관점에서 프로덕트의 특징을 말하게 된다[5]. 그러한 휘처들은 다양한 이해 관계자들에 의해 요구되고, 프로덕트 라인에 포함되는 경우에, 도메인 요구사항 즉, 프로덕트 라인의 공통적인 요구사항과 각각의 프로덕트 마다 다른 요구사항이 될 수 있다. 따라서 앞 절의 시나리오와 휘처의 정의에 의해 프로덕트 라인에서 프로덕트 휘처들은 시나리오로서 표현되는 프로덕트의 행위들을 수행 가능하게 한다(그림 2에서 “enables”). 즉, 각각의 프로덕트 휘처들은 프로덕트의 어떤 행위를 나타내며[30], 그 행위가 가능하도록 만드는 것이다. 아울러, 그러한 프로덕트의 행위들에 의해 성취되는 영향 또는 결과는 휘처에 의해 표현될 수 있게 된다. 이는 그러한 프로덕트의 행위들이 프로덕트의 휘처를 만족시키게 됨을 의미한다(그림 2에서 “fulfills”). 따라서 본 논문에서는 이러한 휘처와 시나리오의 관계를 휘처가 시나리오에 포함된 행위에 첨부되는 것으로 정의한다(그림 1). 결국, 프로덕트 라인에서 휘처와 목표는 이 논문에서 제안하는 도메인 요구사항 모델에서 시나리오에 포함된 행위에 의해 간접적으로

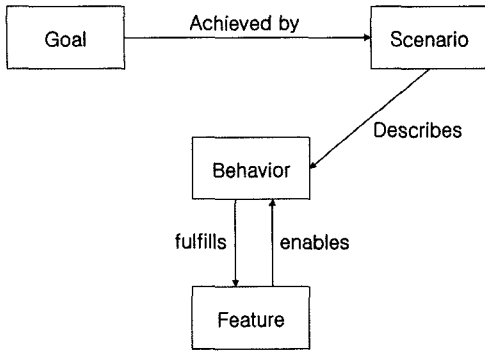


그림 2 목표, 시나리오, 휘처의 관계

관련을 맺게 된다. 그림 2는 이러한 DRM(그림 1)의 바탕이 되는 목표, 시나리오 그리고 휘처의 관계를 보여준다.

목표와 시나리오 그리고 시나리오와 휘처의 관계는 다음과 같이 좀 더 일반적으로 정의된다.

정의 1. G 는 목표들의 집합이고, S 는 시나리오 집합이다. 이때 목표와 시나리오의 관계 $\langle \text{Goal}, \text{Scenario} \rangle \subseteq G \times S$ 는 다음과 같이 정의 된다.

$\langle \text{Goal}, \text{Scenario} \rangle = \{ (G, S) \mid G \in G \wedge S \in S \wedge G \text{는 } S \text{의 수행에 의해 성취된다} \}$

정의 2. A 는 행위(action)들의 집합이고, S 는 시나리오들의 집합이다. 여기서 시나리오는 순차적인 여러 행위들을 포함한다. 그리고 F 는 시나리오의 행위들에 의해 만족되는 휘처들의 집합이다. 이 때 시나리오와 그러한 시나리오의 행위들에 첨부(attach)되는 휘처들의 관계 $\langle \text{Scenario}, \text{Feature} \rangle \subseteq S \times F$ 는 다음과 같이 정의된다.

$\langle \text{Scenario}, \text{Feature} \rangle = \{ (S, F) \mid S \in S \wedge F \in F \wedge S \text{는 } A_i \in A \text{인 } (A_1, \dots, A_n) \text{을 포함한다} \wedge A_i \text{에 첨부되어지는 } F \text{를 가진 } i \text{가 존재한다} \}$

이러한 정의들을 바탕으로, 목표와 시나리오 그리고 휘처에 의해 구성되는 도메인 요구사항 모델(그림 1)은 다음과 같이 정의될 수 있게 된다.

정의 3. 도메인 요구사항 모델은 $(G, S, A, F, \langle \text{Goal}, \text{Scenario} \rangle, \langle \text{Scenario}, \text{Feature} \rangle)$ 이며, 각각의 요소는 다음과 같다.

- 1) G 는 목표들의 집합이다.
- 2) S 는 시나리오들의 집합이다.
- 3) A 는 시나리오에 포함되는 행위(action)들의 집합이다.
- 4) F 는 휘처들의 집합이다
- 5) $\langle \text{Goal}, \text{Scenario} \rangle \subseteq G \times S$ 는 목표 G 와 시나리

오 S 의 관계이다. (정의 1에 의해서)

6) $\langle \text{Scenario}, \text{Feature} \rangle \subseteq S \times F$ 는 시나리오 S 와 휘처 F 의 관계이다. (정의 2에 의해서)

위와 같은 목표, 시나리오, 휘처의 관계는 [31,32,33]을 바탕으로 하고 있다. 이 논문들에서는 목표와 시나리오 그리고 기능(function)을 바탕으로 단일 시스템 설계를 위한 모델을 제안하고 있다. 그러나 본 논문에서는 위의 논문들에서 제시된 기능을 휘처로서 추상화 시키는 것에 의해 기존 모델을 확장하여 프로젝트 라인을 위한 도메인 요구사항 모델을 정의하였다. 이것은 프로젝트 휘처가 프로젝트 라인에서 이해 관계자들에게 보여지는 프로젝트의 특징이며, 프로젝트 기능의 추상화로써 정의되는 것을 통해 이루어진다.

3.3 도메인 요구사항 모델의 계층(The Hierarchy of Domain Requirements Model)

도메인 요구사항 모델에서는 5가지 종류의 관계가 존재하며, 이러한 관계는 도메인 요구사항들의 계층적인 구조를 형성하게 된다.

3.3.1 구성 관계(Composition Relationship)

구성 관계는 도메인 요구사항 사이의 수평적인 AND 구조를 나타낸다. 이것은 기존의 목표 분석 방법 [14,15,34]에서 사용된 목표간 AND 관계를 바탕으로 하며, 상위 수준의 목표를 만족시키기 위해 함께 요구되는 목표들을 나타낸다. 그러나 DRM에서 정의된 도메인 요구사항의 AND 관계는 프로젝트 라인을 위한 것으로서 여러 관련 프로젝트들을 고려하고, 그것들 사이의 공통성과 가변성 분석을 바탕으로 이루어지게 된다.

3.3.2 대체와 선택 관계(Alternative and Optional Relationship)

OR 관계는 도메인 요구사항 사이의 수평적인 OR 구조를 나타낸다. 특히, DRM에서는 기존의 OR 관계를 프로젝트 라인을 위해 대체(Alternative OR)와 선택(Optional OR) 관계로 나누고 있다. 여기서 대체 관계는 프로젝트 라인에서 어떤 목표를 성취하기 위한 대안적인 방법을 의미하며, 선택 관계는 어떤 목표를 성취하기 위해 선택이 가능한 것들을 표현하고 있다.

3.3.3 정제 관계(Refinement Relationship)

추상화(abstraction)는 어떤 것의 본질적인 측면에 중점을 두기 위해 너무 구체적이거나 자세한 사항은 숨기는 방법이다[15]. 따라서 본 논문에서는 정제를 각기 다른 추상화 수준에서 도메인 요구사항을 묘사하고, 식별하기 위해 사용한다. 특히, 기존의 정제 관계 개념을 프로젝트 간의 공통성과 가변성을 고려하기 위하여 확장시켰다.

정제는 DRM의 시나리오 부분과 관련되며, 레벨 i 에서 시나리오 Sc 의 모든 행위들은 레벨 $i+1$ 에서 성취되

기 위한 또 다른 목표로서 생각될 수 있게 된다. 따라서 정제 관계는 도메인 요구사항들 사이의 수직적인 관계를 나타내게 된다.

3.3.4 첨부 관계(Attachment Relationship)

첨부 관계는 단일 시스템이 아닌 프로덕트 라인을 위해 기존의 목표 분석 방법과 휘처를 연결시켜주게 된다. 따라서 이것은 도메인 요구사항들 사이에서 시나리오와 그러한 시나리오의 행위들에 첨부되는 휘처들의 관계를 표현한다. 첨부 관계는 그림 1에서 보여지는 것과 같이 휘처와 시나리오 부분에 관련되며, 시나리오 Sc에 포함된 각각의 행위들은 프로덕트 휘처를 만족시키게 되고, 휘처는 그러한 행위를 가능하도록 만들게 된다. 이 관계는 그림 1에서 'attached to'에 의해 표현된다.

4. 도메인 요구사항 모델링 방법

본 장에서는 3장에서 정의된 도메인 요구사항 모델(DRM)을 바탕으로 도메인 요구사항 모델링 방법을 제안한다. 따라서 도메인 요구사항을 3가지 유형으로 나누고, 이를 바탕으로 도메인 요구사항 식별 공정과 프로덕트 라인을 위한 도메인 요구사항 모델링 공정을 제안한다.

4.1 도메인 요구사항의 3가지 추상화 수준

본 절에서는 기존에 존재하는 요구사항의 추상화 수준[15,35,36]을 바탕으로 도메인 요구사항 식별을 위해 다음과 같은 세 가지 도메인 요구사항 추상화 수준을 정의한다. 따라서 이것은 도메인 요구사항 식별의 관심 분리(separation of concerns)를 가능하게 함으로써 요구사항의 추출 및 식별에 도움을 주게 된다[15,29].

• 정책 수준(Policy level)

조직이나 고객들의 관점 즉, 이해 관계자들의 관점에서 개발될 시스템이 성취하여야 하는 최상위 수준 목표 또는 전략적인 목적을 식별하고, 이해한다. 그리고 이것은 개발될 시스템이 조직에 제공해야 하는 서비스(service)들과 그것들에 대한 근거 식별을 포함하게 된다. 따라서 이 수준의 도메인 요구사항은 비즈니스 목표와 서비스 목표를 포함하는 도메인 요구사항으로 표현된다.

• 상호작용 수준(Interaction level)

정책 수준에서의 비즈니스 도메인 요구사항을 성취하기 위하여 무엇이 제공되어야 하는가에 대한 정보를 포함한다. 이것은 정책 수준에서 요구된 서비스들을 성취하기 위한 시스템과 사용자 사이에서의 상호작용을 고려하게 되며, 결국 프로덕트 라인 안에서 프로덕트들과 사용자 또는 시스템 사이의 상호 작용에 초점을 맞추게 된다.

• 내부 수준(Internal level)

상위 수준의 도메인 요구사항들을 만족시키기 위해

시스템이 제공하여야 하는 기능들을 정의한다. 특히, 상위 수준에서 분석된 상호 작용을 만족시키기 위해서 시스템이 무엇을 해야 하는지를 다루게 된다. 그러나 같은 서비스들 혹은 기능들이라고 해도 각각의 도메인 내에서 각기 다른 프로덕트들 사이에서는 도메인마다 다른 요구사항들이 존재하게 된다. 따라서 내부 수준에서는 시스템 도메인 요구사항과 특정 도메인마다 다른 도메인 종속적인 요구사항들도 다루게 된다.

4.2 도메인 요구사항 식별 공정

도메인 요구사항 식별 공정은 4.1절에서 정의된 도메인 요구사항의 3가지 추상화 수준과 4.3절에서 설명될 공통성과 가변성 분석의 결과를 바탕으로 목표 식별(goal discovery), 시나리오 기술(scenario authoring) 그리고 휘처 첨부(feature attachment)로 이루어진다. 따라서 DRM의 목표, 시나리오 그리고 휘처의 관계를 바탕으로 목표가 식별 된 후, 목표에 의해 시나리오 기술이 이루어지고, 다시 시나리오에 포함된 각각의 행위에 휘처가 첨부된다. 여기서 휘처 첨부는 곧 목표 식별과 시나리오 기술에 의한 휘처의 식별을 의미하게 된다. 그러므로 도메인 요구사항 식별 공정은 목표 식별과 시나리오 기술 그리고 휘처 첨부의 과정이 반복되면서 서로 상호보완적인 역할을 수행하는 것에 의해 이루어지게 된다. 다음 그림 3은 전체적인 도메인 요구사항 식별 공정을 보여준다.

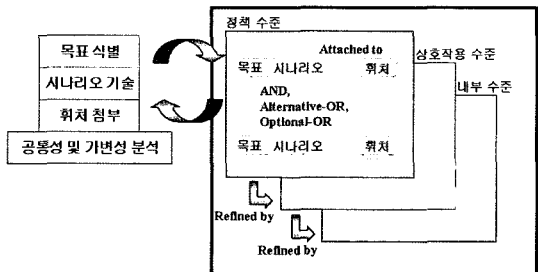


그림 3 도메인 요구사항 식별 공정

여기서 식별된 목표, 시나리오 그리고 휘처는 3.3절에서 정의된 DRM의 5 가지 관계 즉, 'AND', 'Alternative-OR', 'Optional-OR', 'Refined by' 그리고 'Attached to'에 의해 표현된다.

요구사항을 추출하기 위해서 기존의 논문들[12,15,34]에서는 전략들을 바탕으로 한 공정을 제안하고 있다. 본 논문에서는 도메인 요구사항을 식별하기 위해서 5가지 전략 즉, 정제(refinement), 구성(composition), 대체(alternative), 선택(option) 그리고 첨부(attachment) 전략을 사용하였다. 이러한 식별 전략들은 앞에서 정의된 DRM의 5가지 관계를 바탕으로 하고 있으며, 4.3 절에

서 설정된 프로덕트 라인을 위한 공통성과 가변성 분석 결과를 통해 이루어진다. 그러므로 3장에서 도메인 요구사항 모델에 대한 정의 3: (G, S, A, F, <Goal, Scenario>, <Scenario, Feature>) 에서

- 정책 전략은 G 보다 하위 추상화 수준에 있는 목표 Gi 를 찾는다.
- 구성 전략은 G와 'AND' 관계에 있는 목표 Gj 를 찾는다.
- 대체 전략은 G와 'Alternative-OR' 관계에 있는 목표 Gk 를 찾는다.
- 선택 전략은 G와 'Optional-OR' 관계에 있는 목표 Gl 을 찾는다.
- 첨부 전략은 G를 성취하기 위한 Sc 에 첨부되는 휘처 Fi 를 찾는다.

4.3 도메인 요구사항 모델링 공정

그림 4와 같이 도메인 요구사항 모델링 공정은 프로덕트 라인 영역 결정(product line scoping)과 프로덕트 라인을 위한 도메인 요구사항 분석(domain requirements analysis) 그리고 공통성과 가변성 분석(commonality and variability analysis)의 세 가지 활동으로 이루어진다. 이러한 활동들은 순차적으로 진행되는 것이 아니라 반복적으로 서로에게 영향을 주면서 진행된다. 프로덕트 라인 영역결정은 도메인 요구사항의 정책 수준에서의 활동이 되며, 도메인 요구사항 분석은 상호작용 및 내부 수준에 해당되는 활동이 된다. 그리고 공통성과 가변성 분석은 이러한 두 활동을 연결시켜 주며, 도메인 요구사항의 공통성과 가변성 분석을 한다.

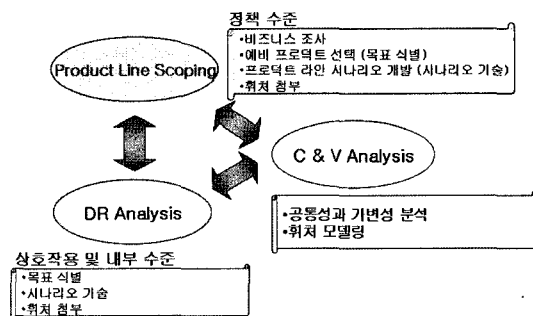


그림 4 도메인 요구사항 모델링 공정

4.3.1 프로덕트 라인 영역 결정

영역 결정은 소프트웨어 프로덕트 라인 개발에 있어서 프로덕트 라인의 지속적인 실용 가능성(long-term viability)을 결정하는 중요하고, 기본적인 활동으로서 프로덕트 라인 안에 무엇이 포함되고, 무엇이 포함되지 않을 것인지에 대해 결정하게 된다. 일반적으로 프로덕트 라인 영역 결정의 입력 값으로는 프로덕트 개발 조

직 내에서의 관련 프로덕트의 기술서, 마케팅 분석, 도메인 지식과 비즈니스 비전 그리고 마케팅 전략등이 있을 수 있다[10].

따라서 본 논문의 프로덕트 라인 영역 결정에서는 위와 같은 입력 값들을 바탕으로 다음과 같은 활동을 포함하게 되며, 이것은 결국 프로덕트 라인 영역결정이 4.1절에서 정의된 도메인 요구사항 추상화 수준에서의 정책 수준에 해당되는 활동임을 의미한다.

- 비즈니스 조사(business investigation)
프로덕트 라인에서 성취되어야 하는 최상위 수준의 비즈니스 목표를 식별하고, 이해할 뿐 아니라 프로덕트 라인 개발에 있어서의 제약사항들(constraints)도 고려한다. 따라서 프로덕트 영역 결정의 입력 값들을 바탕으로 관련 정보를 수집하고, 4.1절에서 정의된 정책 수준의 비즈니스 도메인 요구사항을 식별하고, 이해한다.
- 예비 프로덕트 선택(candidate product selection)

프로덕트 라인에 포함되어야 하는 프로덕트들이 무엇인지를 결정한다. 만약 프로덕트 개발 조직 내에 기존의 관련 프로덕트 집합이 존재하여 그것을 바탕으로 프로덕트 라인을 개발할 경우에는, 그러한 관련 프로덕트를 바탕으로 예비 프로덕트를 결정하게 된다. 그러나 새로운 도메인에서 처음부터 프로덕트 라인을 개발 경우에는 예비 프로덕트가 그 조직을 위한 비즈니스 전략이나 경쟁사에 의해 개발된 다른 프로덕트 들을 바탕으로 결정되어야 한다. 따라서 이 활동은 4.2절에서 제시된 도메인 요구사항 식별 공정의 목표 식별 단계로서, 프로덕트 라인을 위한 초기 목표 식별을 하게 된다. 이는 예비 프로덕트들로 선택된 것들이 결국 프로덕트 라인에서 성취되어야 하는 목표가 되기 때문이다.

- 프로덕트 라인 시나리오 개발(developing product line scenarios)

프로덕트 라인 시나리오는 프로덕트 라인의 범위를 결정하기 위한 중요한 요소가 된다[2]. 그것들은 프로덕트 라인 안에서 프로덕트들과 사용자 또는 시스템 사이의 상호 작용을 묘사하게 된다. 따라서 이 시나리오는 프로덕트 라인에서 모든 프로덕트에 공통적인 상호 작용과 각각의 프로덕트 들에만 존재하는 특별한 상호작용도 식별하게 된다. 본 논문에서는 [13]에서 제안된 시나리오 구조 모델(scenario structure models)과 시나리오 스크립트(scenario scripts)을 바탕으로 프로덕트 라인 시나리오를 개발한다. 여기서 시나리오 구조 모델은 개발될 시스템과 그 시스템 환경에 대한 사실을 포함하게 되며, 이것은 프로덕트 라인 관점에서 프로덕트 라인 영역 결정을 위한 문맥 모델(context model)이 된다. 그림 5는 8개의 개체로 구성되는 이러한 시나리오 구조 모델을 보여준다.

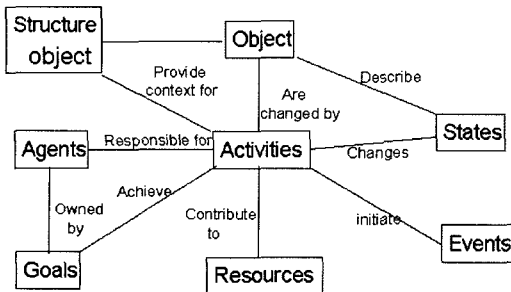


그림 5 시나리오 구조 모델

시나리오 스크립트는 시나리오 구조 모델을 바탕으로 시스템 사용 즉, 시스템 행위에 대해서 묘사한다. 이는 각 시나리오 구조 모델에서 설정된 개체와 그들 사이의 관계를 바탕으로 그것들 사이의 상호작용을 보여지게 된다. 따라서 시나리오 스크립트를 통한 프로덕트 라인 시나리오 개발은 정책 수준에서의 시나리오 기술에 해당하게 된다.

아울러, 정책 수준에서 개발된 프로덕트 라인 시나리오 오는 상호작용 수준과 내부 수준에서 이루어지는 도메인 요구사항 식별 공정을 돕게 된다. 즉, 프로덕트 라인을 위한 문맥 모델로서 역할을 하게 되는 시나리오 구조 모델과 이를 바탕으로 개발된 시나리오에 의해 상호작용 수준과 내부 수준의 목표 식별과 시나리오 기술이 이루어지게 된다.

• 휘처 첨부(Feature attachment)

프로덕트 라인 시나리오 개발 단계에서 식별된 시나리오를 바탕으로 휘처들을 각각의 시나리오에 포함된 행위들에 첨부한다. 이것은 4장에서 정의된 휘처 첨부에 의해 이루어지며, 곧 휘처 식별을 의미한다. 휘처 첨부 결과는 공통성과 가변성 분석과 도메인 요구사항 분석을 위한 입력 값이 된다.

4.3.2 프로덕트 라인을 위한 도메인 요구사항 분석

앞에서 설명하였듯이, 프로덕트 라인 영역 결정은 도메인 요구사항의 정책 수준에 해당하는 활동이다. 따라서 도메인 요구사항 분석은 상호작용과 내부 수준에서의 활동이 된다. 그러므로 프로덕트 라인 영역 결정 결과를 바탕으로 공통성과 가변성 분석이 이루어지고, 그 결과를 통해 상호작용 수준과 내부 수준에서 반복적으로 도메인 요구사항 분석이 이루어진다. 이 때, 상호작용 수준과 내부 수준 각각에서는 4.2 절에서 정의된 도메인 요구사항 식별 공정을 통해 분석이 이루어진다. 즉, 목표가 식별되고, 목표에 의해 시나리오 기술이 이루어지며, 마지막으로 그러한 시나리오에 포함된 각각의 행위에 휘처가 첨부된다.

4.3.3 공통성과 가변성 분석

공통성과 가변성 분석에서는 그림 4와 같이 프로덕트 라인 영역 결정과 도메인 요구사항 분석 활동을 서로 연결시켜 준다. 즉, 프로덕트 라인의 상위 추상화 수준에서 식별된 목표, 시나리오 그리고 휘처를 바탕으로 공통성과 가변성 분석을 함으로써 하위 추상화 수준의 도메인 요구사항을 추출하고 식별하기 위한 바탕을 제공하게 된다. 여기서 공통성과 가변성 분석은 프로덕트 라인의 여러 프로덕트들에서의 공통성을 식별하고 가변성을 다루게 된다.

아울러 도메인 요구사항의 공통성과 가변성 분석의 결과는 휘처 모델링의 바탕이 된다. 이는 식별된 휘처들에 공통성과 가변성 분석을 한결과를 통해 휘처 모델링이 이루어질 수 있음을 의미한다. 여기서 휘처 모델링은 기존의 휘처 중심의 도메인 분석[5,8,9]에서 제안된 휘처 모델링 방법에 의해 이루어 지게 된다. 따라서 제안된 방법에 의해 생성된 휘처 모델은 각각의 프로덕트 휘처 식별에 대한 근거를 제시할 수 있게 되며, 식별된 휘처들이 프로덕트 라인 목표들과 어떻게 관련 되는지를 보여줄 수 있게 된다. 이는 이해 관계자 입장에서 보여지는 프로덕트 휘처들이 프로덕트 라인의 프로덕트들에서 어떻게 프로덕트 라인 목표들을 성취하기 위해 사용될 것인가에 대해 보여지게 된다.

5. 주택 통합 시스템(HIS) 예제 적용

본 장에서는 3,4 장에서 제안한 프로덕트 라인을 위한 도메인 요구사항 모델과 모델링 방법을 이를 지원하기 위해 개발된 도구를 통해 주택 통합 시스템(HIS, Home Integration System)에 적용한다.

5.1 문제 도메인

본 논문에서 제시하는 도메인 분석 방안을 검증하기 위하여 기존에 소프트웨어 프로덕트 라인에서 예제로 사용되고 있는 주택 통합 시스템(HIS, Home Integration System)[5,11,37]를 선택하였다. HIS는 주택의 안락성(comfort), 안전성(safety) 그리고 보안성(security)을 높이기 위한 시스템으로서 주택 보유자(home owners)들이 그들의 집에서 다음과 같은 장치(equipment)들에 접근하고, 그것들을 관리하며, 통합할 수 있도록 해준다.

- 기후 관리 시스템(climate control systems) - 냉방과 난방(heating and cooling)
- 보안 시스템(security systems) - 침입(intrusion), 화재(fire) 그리고 홍수(flood)의 탐지(detection) 및 관리(control)
- 오락 시스템(entertainment systems) TV, 라디오, 음악 재생 장치들

HIS는 100억 달러의 사업 시장을 가지고 있으며,

HIS 관련 분야들 즉, 통합, 네트워크, 주택 또는 회사의 보안 시스템 등과 같은 분야에서 경험과 지식을 가지고 있는 조직이 새로운 시장으로 진입할 수 있는 기회를 제공하게 된다.

따라서 이러한 HIS 시장은 상대적으로 새롭고 아직 미성숙된 분야이며, 새로운 장치(devices)들이나 제조업자들이 계속적으로 나타남에 따라 빠르게 변화하게 된다. 현재 대부분의 주택 보유자들은 HIS에 대한 경험이 없으며, 그것의 가치나 필요성에 대해 확신을 가지지 못하고 있을 수 있다.

다음 절에서부터는 어떻게 본 논문에서 제안된 방법이 이를 지원하기 위해 개발된 도구 즉, **IDEAS** (Integrated **D**omain **r**equirements **A**nalysis for **S**oftware product lines)를 통해 적용되는지를 설명한다.

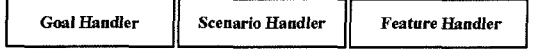
5.2 IDEAS 아키텍처

지원 도구 IDEAS는 그림 6과 같이 계층형 아키텍처로서 설계되었으며, 표현 계층 (presentation layer), 어플리케이션 계층(application layer), 그리고 데이터 계층(data layer)의 세 계층으로 구성된다. 여기서, 표현 계층은 미리 정해진 화면을 보여주며, 사용자에게 인터페이스를 제공한다. 따라서 도구의 인터페이스는 요구사항의 3가지 추상화 수준과 식별 공정에 따라 정책(그림

Presentation Layer



Application Layer



Data Layer

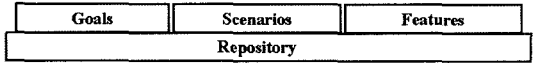


그림 6 IDEAS 아키텍처

7), 상호작용(그림 10), 그리고 내부 수준으로 구성된다. 그리고 각각의 인터페이스는 목표, 시나리오, 그리고 휘처 관점에서 도메인 요구사항들을 작성, 추가, 수정, 삭제하기위한 화면을 제공한다.

어플리케이션 계층은 표현 계층과 데이터 계층을 관리하고 연결시켜주는 역할을 하며, 목표, 시나리오, 그리고 휘처 관리자로 이루어진다. 따라서 표현 계층에서 작성된 목표와 시나리오 그리고 휘처가 각각의 관리자에 의해 적절한 형태로 변환되고, 데이터 계층에 저장된다. 그리고 사용자가 목표와 시나리오 또는 휘처를 찾고자 할 경우에는 각각의 관리자가 그러한 것들을 찾아서, 표현계층에 맞는 형식으로 변환하여 보여주게 된다.

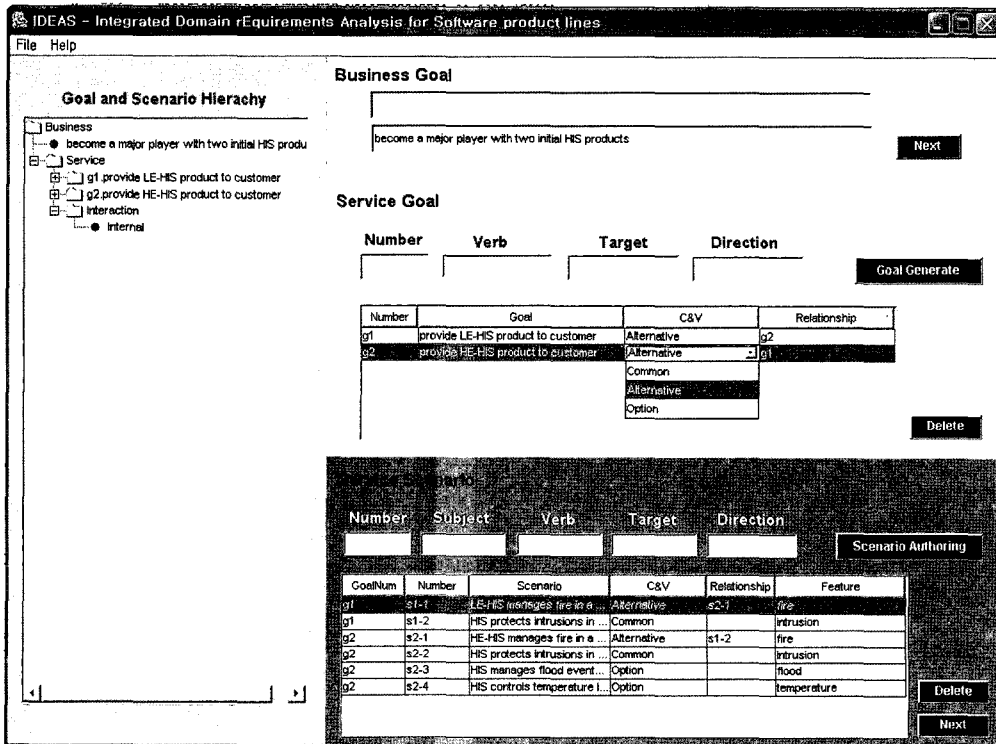


그림 7 정책 수준에서의 도메인 요구사항

마지막으로 데이터 계층은 도구에 의해서 저장, 수정, 삭제되는 여러 데이터들 즉, 목표, 시나리오, 휘처, 그리고 그것들 사이의 관계를 저장하고, 관리한다. 특히, 목표와 시나리오에 대한 정보를 저장하기 위해 각각의 목표와 시나리오의 구조는 ID, "Verb", "Target", "Direction", "C&V", "Relationship", 그리고 관련 목표 또는 관련 시나리오 ID와 같은 정보를 포함한다.

IDEAS는 Java 언어를 통해 설계되고, Java 개발 Toolkit 인 JDK 1.4.0_02를 사용하여 개발되었다. 다음 절에서부터는 앞에서 설명된 HIS 예제를 통해 지원도구를 사용하여 만들어진 화면들을 보여줌으로써 제안된 방법과 지원도구를 좀 더 자세히 설명하고자 한다.

5.3 문제 도메인에 적용

5.3.1 정책 수준에서의 프로덕트 라인 영역 결정 및 구성성과 가변성 분석

프로덕트 라인 영역 결정 단계는 프로덕트 라인에서 성취되어야 하는 최상위 수준의 비즈니스 목표를 식별하고 이해하며, 프로덕트 라인 개발에 있어서의 제약사항들도 고려한다. 또한 프로덕트 라인에 포함되어야 하는 예비 프로덕트들을 결정하고, 프로덕트 라인 시나리오를 생성한다. 이 단계는 4.1절에서 정의된 도메인 요구사항의 최상위 수준인 정책 수준에 해당하게 된다.

그림 7은 정책 수준에서 분석된 도메인 요구사항의 결과를 전체적으로 보여주며, 자세한 설명은 다음과 같다.

- HIS 프로덕트 라인의 비즈니스 목표 및 제약 사항

HIS 프로덕트 라인을 개발하려는 조직은 주택 통합 시스템을 위한 100억 달러 이상의 시장이 존재함을 인식하고, 우선 두 개의 HIS 프로덕트에 해당하는 시장에 진입하고자 한다. 따라서 이 조직은 HIS를 위한 기본적인 상호작용만을 제공하는 프로덕트(a low-end product)와 HIS에 포함 될 수 있는 여러 다른 기능 또한 제공하는 프로덕트(a high-end product) 시장을 목표로 하고 있다.

또한 이 조직의 중요한 마케팅 전략은 확장 가능한 제품을 소비자에게 제공하는 것이다. 다시 말해서, 예산에 민감한 소비자들에게 적은 휘처들을 가지고 있는 작은 시스템으로부터 시작하여, 새로운 제품을 다시 사는 것이 아니라 새로운 휘처들을 추가시킴으로써 좀 더 큰 제품을 제공하는 것이다. 그러므로 프로덕트의 확장성(scalability)은 이 프로덕트 라인의 상위 비즈니스 목표가 된다. 그림 7에서 "Business Goal" 부분은 이러한 상위 비즈니스 목표를 보여준다. 도구의 왼쪽에는 요구사항 수준 별로 생성된 목표와 이를 성취하기 위한 시나리오들이 나타난다.

- HIS 프로덕트 라인의 예비 프로덕트 선택
- 앞 부분에서의 비즈니스 목표 및 제약 사항에 대한

조사를 바탕으로 이 프로덕트 라인에서는 두 가지의 초기 HIS 프로덕트 즉, HIS를 위한 기본적인 기능 즉, 적은 휘처들을 가지고 있는 프로덕트(LE-HIS)와 여러 부가적인 휘처들 또한 제공하는 프로덕트(HE-HIS)를 포함하게 된다. 그리고 이러한 예비 프로덕트 들은 앞에서 제시된 도메인 요구사항 식별 공정에서의 초기 목표 식별의 결과로서 생각 될 수 있다. 이것은 LE-HIS와 HE-HIS가 결국 프로덕트 라인의 최상위 수준의 비즈니스 목표를 바탕으로 이 프로덕트 라인에서 성취하여야 하는 목표가 되기 때문이다. 그림 7의 "Service Goal" 부분은 도구에 의해 작성된 목표들을 보여주고 있다. 여기서 LE-HIS가 G1이 되고 HE-HIS가 G2가 됨으로써 'Alternative-OR' 관계를 가지게 되는 것은 HIS 프로덕트 라인을 위해서 이 두 개의 프로덕트가 서로 대체 가능한 관계이기 때문이다.

- HIS 프로덕트 라인을 위한 프로덕트 라인 시나리오 개발

HIS 프로덕트 라인을 위한 프로덕트 라인 시나리오 개발을 위해서 4.3절에서 제안된 시나리오 구조 모델과 시나리오 스크립트를 기술한다. 그림 8에서는 전체 HIS의 시나리오 구조 모델 중 일부분을 보여주고 있다. 이 모델은 프로덕트 라인 영역 결정을 위한 문맥 모델이 된다.

다음으로는 시나리오 구조 모델을 바탕으로 시나리오 스크립트를 생성한다. 이것은 시스템 사용 즉, 시스템 행위를 묘사함으로써 HIS 프로덕트 라인의 시나리오를 생성하게 된다. 그림 7에서 "Service Scenario" 부분은 도구의 "Scenario Authoring"에 의해 기술된 프로덕트 라인 시나리오를 보여준다. 그림에서 보이듯이 여기서 각각의 시나리오는 관련된 목표의 ID, 시나리오 ID, 가변성 종류(C&V 부분), 그리고 시나리오들 간의 관련성을 포함한다.

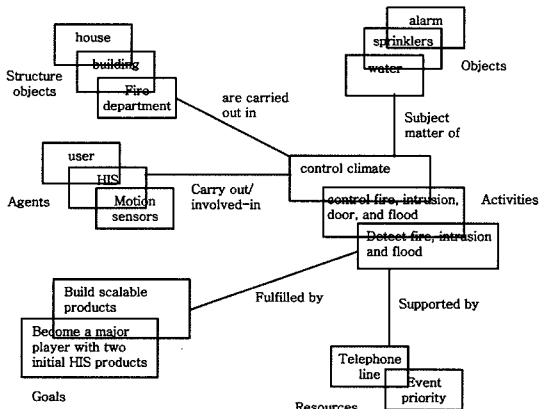


그림 8 HIS의 시나리오 구조 모델

• 휘처 첨부

프로덕트 라인 시나리오 개발 단계에서 식별된 시나리오를 바탕으로 휘처들을 각각의 시나리오에 포함된 행위들에 첨부한다. 이것은 4.2 절에서 정의된 첨부 전략에 의해 이루어진다. 그림 7의 “Service Scenario” 부분에서는 각각의 시나리오들에 화재(fire), 침입(intrusion), 홍수(flood) 그리고 기온(temperature) 휘처들이 첨부된 것을 보여주고 있다. 이것은 곧 시나리오를 바탕으로 휘처가 식별된 것을 의미한다. 그리고 이러한 휘처들은 기존의 휘처 중심의 도메인 분석[5,8,9]의 휘처 범주에서 가장 상위 수준의 휘처들이 된다.

• 공통성과 가변성 분석

정책 수준에서 식별된 도메인 요구사항을 위한 공통성과 가변성 분석을 한다. 이것은 앞 단계에서 식별된 목표 즉, 예비 프로덕트들과 생성된 시나리오, 그리고 휘처들을 바탕으로 이루어진다. 따라서 각각의 시나리오를 중심으로 공통성(common) 뿐만 아니라 가변성 즉, 대체 가능한 것(alternative) 그리고 선택 가능한 것(option)을 분석하게 된다. 그림 7에서 S1-1과 S2-1은 이 프로덕트 라인을 위해서 서로 간에 대체 가능한 것이며, S1-2와 S2-2는 공통적인 것이 된다. 그리고 S2-3, S2-4는 선택 사항들임을 나타내고 있다.

정책 수준에서의 이러한 공통성과 가변성 분석의 결과는 상호작용 수준에서의 분석에 대한 바탕이 된다. 즉, 정책 수준에서 공통적인 것은 상호작용 수준에서 ‘AND’ 관계로 분석되며, 대체 가능한 것은 ‘Alternative-OR’ 관계로, 그리고 선택 가능한 것은 ‘Optional-OR’ 관계로 분석된다. 그리고 이렇게 휘처 첨부에 의해 식별된 휘처들과 공통성 및 가변성 분석의 결과는 모두 휘

처 모델링의 바탕이 된다.

5.3.2 상호작용 수준에서의 도메인 요구사항 분석 및 공통성과 가변성 분석

상호작용 수준에서의 도메인 요구사항 분석에서는 정책 수준에서의 비즈니스 도메인 요구사항을 성취하기 위하여 무엇이 제공되어야 하는가에 대한 정보를 포함하게 된다. 따라서 이는 4.1절에서 제시된 도메인 요구사항 식별 공정인 목표 식별, 시나리오 기술 그리고 휘처 첨부에 의해 이루어진다. 그리고 이렇게 식별된 도메인 요구사항들에 대해 공통성과 가변성 분석을 함으로써 내부 수준에서의 분석을 위한 바탕을 제공하게 된다. 그림 9는 상호작용 수준에서의 분석 결과를 부분적으로 보여주고 있다.

우선, 정책 수준에서 식별된 목표 즉, 예비 프로덕트들과 시나리오, 그리고 휘처들에 대한 공통성과 가변성 분석을 바탕으로 하여 4.2 절에서 정의된 정제, 구성, 대체, 선택 전략을 통해 목표들을 식별한다. 이때, 정책 수준에서의 공통성과 가변성 분석의 결과로서 공통적인 것은 ‘AND’ 관계로 분석되며(구성 전략), 대체 가능한 것은 ‘Alternative-OR’ 관계로, 그리고 선택 가능한 것은 ‘Optional-OR’ 관계로(선택 전략) 분석된다. 그림 7과 그림 9에서 보여지는 바와 같이 정책 수준에서 공통성과 가변성 분석이 끝난 시나리오 S1-1과 S2-1을 성취하기 위해 정제 전략에 의해 G1-1 and G2-1이 식별되고, 그것들은 대체 가능한 것이기 때문에 ‘Alternative-OR’ 관계를 가지게 된다. G1-2는 S1-2와 S2-2에서 식별된 것으로서, 공통적인 것이므로 구성 전략에 의해 ‘AND’ 관계로 나타난다. 그리고 G2-3과 G2-4는 S2-3과 S2-4가 선택 가능한 것이므로 선택 전략에 의

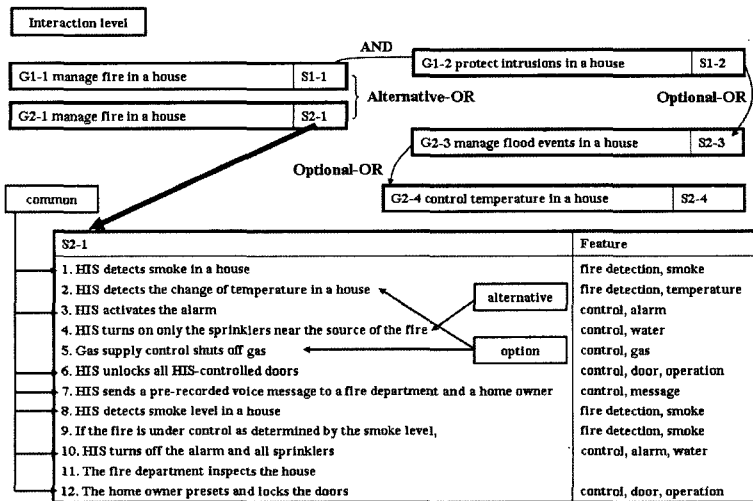


그림 9 상호작용 수준에서의 도메인 요구사항

해 'Optional-OR' 관계로서 나타나게 된다.

다시 말해서, 상호작용 수준에서의 목표들은 정책 수준의 시나리오로부터 식별되며, 따라서 상위 수준의 시나리오들이 상호작용 목표들을 위한 예비 목표들이 된다. 그리고 이중에서 도메인 전문가 혹은 분석가에 의해 실제 상호작용 목표들이 선택되고, 이는 도구에서 "Select" 버튼에 의해 수행된다(그림 10).

다음으로는 목표를 위한 시나리오들이 기술된다. 그림 9에서는 G2-1을 위한 시나리오 S2-1을 보여주고 있다. 즉, S2-1은 HE-HIS를 위해 화재를 탐지하고, 제어하기 위해 기술된 시나리오이다.

그리고 생성된 시나리오에 휘처 첨부가 이루어진다. 그림 9와 10의 하단 부분은 4.2 절에서 정의된 첨부 전략에 의해 휘처들이 첨부된것을 보여주고 있다. 여기서 첨부된 휘처들은 휘처 범주에서 "능력(capability)"이나 "운영 환경(operating environment)" 또는 "도메인 기술(domain technology)" 휘처들이 된다.

마지막으로 정책 수준에서와 같이 공통성과 가변성 분석을 통해 시나리오를 중심으로 공통적인 것, 대체 가능한 것 그리고 선택 가능한 것을 분석하게 된다. 그림 9의 하단 왼쪽, 그리고 그림 10의 "C&V" 부분은 이러한 공통성과 가변성 분석의 결과를 보여주고 있다.

5.3.3 내부 수준에서의 도메인 요구사항 분석 및 공통성과 가변성 분석

내부 수준에서의 도메인 요구사항 분석에서는 상위 수준의 도메인 요구사항들을 만족시키기 위해 시스템이 제공하여야 하는 기능들을 정의하게 된다. 특히, 상호작용 수준에서 분석된 프로덕트 라인에서의 프로덕트들과 사용자 또는 시스템 사이의 상호 작용을 만족시키기 위해 시스템이 무엇을 해야 하는지를 다루게 된다. 내부 수준에서의 분석 결과가 다음 그림 11에서 부분적으로 보이고 있다. 즉, 그림 11에서는 상호작용 수준의 분석 결과인 그림 9의 시나리오를 바탕으로 내부 수준에서 식별된 목표, 시나리오 그리고 휘처들을 보여주고 있다.

우선, 내부 수준에서는 상호작용 수준에서의 결과를 바탕으로 4.2 절에서 정의된 식별 전략들을 통해 목표들을 식별한다. 그러므로 그림 9과 그림 11에서 보이는 바와 같이 상호작용 수준에서 공통성과 가변성 분석이 끝난 시나리오 S1-1과 S2-1을 바탕으로 그것들을 성취하기 위한 정책 전략에 의해 목표 G1-1-1부터 G1-1-10이 식별되고, 그 각각은 구성, 대체, 선택 전략에 의해 관계를 가지게 된다. G1-1-3과 G2-1-3은 대체 가능한 것이기 때문에 'Alternative-OR' 관계를 가지게 되고, G1-1-9와 G1-1-10은 선택 가능한 것이므로 선택 전략

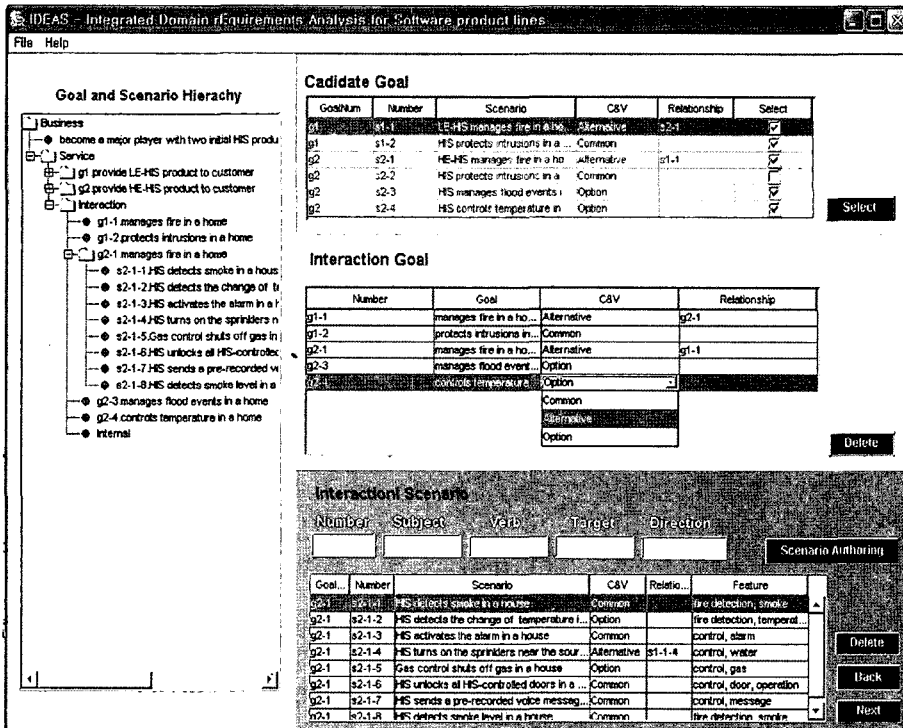


그림 10 도구 IDEAS에서 상호작용 수준

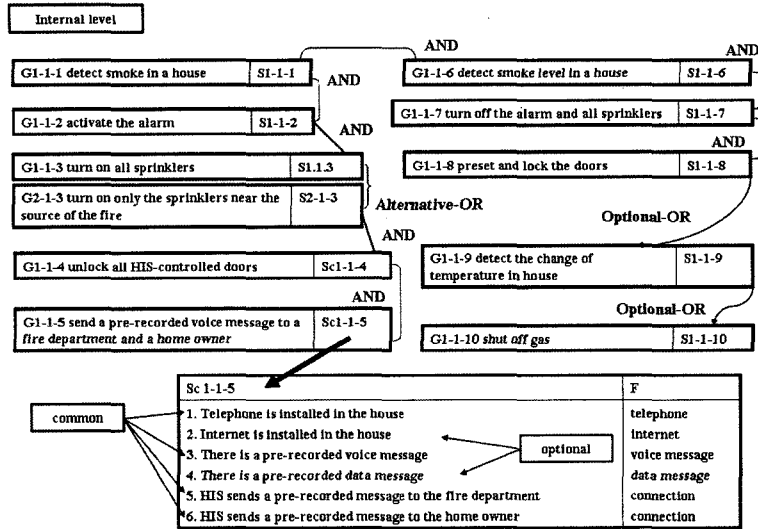


그림 11 내부 수준에서의 도메인 요구사항

에 의해 ‘Optional-OR’ 관계로서 연결되며, 나머지는 ‘AND’ 관계를 가지게 된다. 그리고 그림 11에서는 시나리오 기술에 의해 G1-1-5를 위한 시나리오 S1-1-5를 보여주고 있다. 즉, S1-1-5는 화재가 발생하였을 경우, 그것을 소방서와 주택 소유자에게 알려주는 시나리오이다. 다음으로는 시나리오 S1-1-5의 행위들에 4.2 절에서 정의된 침부 전략에 의해 휘처들이 첨부된다. 이것이 그림 11의 하단 오른쪽에 보이고 있다. 여기서 첨부된 휘처들은 휘처 범주에서 “능력”, “운용 환경”, “도메인 기술” 또는 “구현 기술(implementation technology)” 휘처들이 된다.

끝으로 앞 단계 수준에서와 같이 공통성과 가변성 분석을 통해 시나리오를 중심으로 공통적인 것, 대체 가능한 것 그리고 선택 가능한 것을 분석하게 되며, 이러한 결과를 바탕으로 휘처 모델링이 가능하게 된다. 그림 11의 하단 왼쪽은 이러한 공통성과 가변성 분석의 결과를 보여주고 있다.

5.4 비교

본 절에서는 기존의 휘처 중심의 분석 방법에서의 문제점을 바탕으로 크게 두 가지 관점 즉, 체계적인 휘처 식별, 그리고 휘처 및 공통성, 가변성의 근거 제공이라는 측면에서 제안된 방법과 다른 방법들을 비교하고자 한다.

우선, 휘처 식별 방법에 있어서 기존의 휘처 중심의 방법에서는 도메인 용어의 표준화와 도메인 영역 결정을 바탕으로 휘처 범주를 통해 휘처들을 식별하고 있으며, 최근에는 MPP[5]를 바탕으로 초기 휘처들을 식별하고 있다. 그러나 앞에서 살펴 보았듯이, 휘처 중심의 방

법에서는 휘처 식별을 위한 체계적인 방법이 제시되지 못하고 있으며, 식별된 프로덕트 휘처에 대한 근거와 도메인 분석의 결과인 공통성과 가변성 분석에 대한 근거를 제공하지 못하고 있다.

휘처와 유스케이스를 통합한 방법들의 경우에는 휘처와 유스케이스를 통합하거나(FODacom[19], FeatuRSEB[20], PLUSS[21]), 휘처와 유스케이스 모델 그리고 객체 모델을 통합하고 있다([11]). 그러나 이러한 방법들 역시 휘처 중심의 방법과 같이 도메인 분석을 위한 체계적이고, 구체적인 방법을 제공하지 못하고 있다. 아울러, 유스케이스를 바탕으로 휘처들을 분석하므로 유스케이스 분석이 가지는 단점 즉, 유스케이스가 추출되는 근거가 제시되지 못한다는 점에 의해 요구사항에 대한 정당한 근거를 제시하기 어렵다. 이것은 “왜”에 대한 질문을 통해서 요구사항의 추출이 이루어지지 못한다는 것이다.

이에 반해, 본 논문에서 제안한 방법은 3.4장에서 제시한 시나리오, 목표, 휘처를 통합한 도메인 요구사항 모델과 도메인 요구사항 식별 공정을 바탕으로 휘처 식별을 위한 체계적인 방법을 제안한다. 5.3절에서의 HIS 예에서 알 수 있듯이, 정책, 상호작용, 내부 수준에서 도메인 요구사항 식별이 목표 식별, 시나리오 기술 그리고 휘처 첨부에 의해 이루어짐으로써 체계적인 휘처 식별이 가능하게 된다. 다음 표 1은 5.3절의 HIS 예제에서 제안된 방법에 의해 화재 탐지 및 관리를 위해 각 수준별로 식별된 휘처들을 보여주고 있다.

아울러 제안된 방법은 프로덕트 휘처 식별의 근거와 프로덕트 라인의 공통성과 가변성 분석에 대한 근거를

표 1 각 수준별로 식별된 휘처들

정책 수준	상호작용 수준	내부 수준
화재 탐지 및 관리를 위해 식별된 휘처들	fire for LE-HIS, fire for HE-HIS	smoke, fire detection, smoke detection, control, alarm, water, door operation, message, gas
		smoke sensors, monitoring, detecting, sprinklers, alarm, telephone, internet, voice message, data message, connection

제공할 수 있다.

다음 그림 12는 표 1에서 화재 탐지 및 관리를 위해 각 수준별로 식별된 휘처들 중 상호작용 수준의 door operation 휘처가 제안된 방법을 통해 어떻게 식별되었는지를 보여주며, 이를 통해 그 휘처에 대한 근거를 보여주고 있다. 여기서 door operation 휘처는 목표 G1-1과 G2-1을 위한 시나리오 S1-1-6와 S2-1-6에 첨부되는 것에 의해 식별되었다. 그리고 이러한 상호작용 수준에서의 목표 G1-1과 G2-1은 정책 수준에서의 시나리오 S1과 S2의 행위 1을 만족시키기 위한 것이 된다. 그러므로 최상위 수준의 비즈니스 목표로부터, 도메인 요구사항 모델링 방법에 의해 휘처까지의 연결을 보여줌으로써 그 근거를 제공하게 된다.

그리고 상위 수준의 도메인 요구사항들을 성취하기 위해 DRM에 존재하는 구성(AND), 대체(Alternative-OR) 그리고 선택(Optional-OR) 관계를 통해 하위 수준의 도메인 요구사항들이 식별됨으로써 이를 통한 공통성과 가변성 분석의 근거를 제공할 수 있게 되었다. 즉,

구성 관계는 상위 목표를 만족시키기 위해 함께 요구되는 것들로서 공통적인 것을 나타내며, 대체 관계는 어떤 목표를 성취하기 위한 대안적인 것, 선택 관계는 목표 성취를 위해 선택 가능한 것을 나타냄으로써 상위 목표를 만족시키기 위한 프로덕트들 사이에서의 공통성(구성 관계)과 가변성(대체 및 선택 관계)을 보여주게 된다. 그리고 이는 도메인 요구사항 식별 후 공통성과 가변성 분석 활동의 바탕이 된다. 그림 12의 정책 수준에서는 대체 관계에 의해 최상위 비즈니스 목표 성취를 위한 대안적인 것들(G1과 G2)을 보여주고 있다. 그리고 이것은 시나리오 S1과 S2의 행위 1에 대한 공통성과 가변성 분석의 바탕이 된다.

6. 결론 및 향후 연구

소프트웨어 프로덕트 라인에서 재사용 가능한 소프트웨어 자산을 개발할 때, 프로덕트 라인의 공통성과 가변성 식별을 위해 휘처 중심의 도메인 분석 방법이 많이 사용되고 있다. 그러나 프로덕트 라인 개발 시에 프로덕트 라인에서 휘처가 식별되고, 정의되어야 하는 경우, 기존의 휘처 중심의 도메인 분석 방법에서는 휘처를 식별하고, 식별된 휘처의 근거를 제시하는 체계적인 방법에 대한 연구가 미흡한 실정이다. 또한 프로덕트 라인을 위한 도메인 분석 결과인 프로덕트 라인의 공통성과 가변성은 프로덕트 라인 개발 조직의 최상위 수준 목표를 만족시키고, 그 근거를 제시하여야 하지만 기존의 휘처 중심의 방법에서는 이러한 부분에 대한 연구가 부족하다.

따라서 본 논문에서는 기존의 휘처 중심의 도메인 분

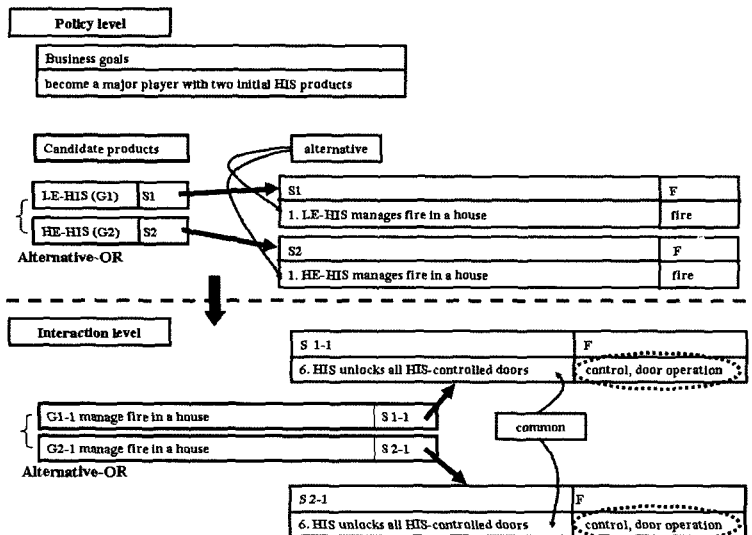


그림 12 식별된 휘처의 근거

석 방법에서의 문제점들을 해결하고, 보안하기 위해 기존의 요구사항 분석 방법에서의 목표와 시나리오 그리고 기존의 도메인 분석 방법에서의 휘처의 관계를 바탕으로 프로덕트 라인을 위한 도메인 요구사항 모델과 모델링 방법을 제시하였다. 이러한 도메인 분석 방법은 목표를 통해 프로덕트 라인의 최상위 수준 요구사항들에 대한 식별의 용이성을 제공하고, 그 근거를 보여줄 수 있었다. 또한 시나리오가 그러한 목표의 식별을 돕고, 식별된 목표를 성취할 뿐만 아니라 목표와 휘처를 연결시켰다. 휘처는 이해 관계자의 관점에서 재사용을 위한 측면, 즉 재사용을 위한 식별과 표현에 더욱 초점을 맞추게 되었다.

아울러, 본 논문에서 제안된 방법은 이를 지원하기 위해 개발된 도구인 IDEAS 와 함께 프로덕트 라인을 위한 도메인 분석에 적용되었다. 즉, 제안된 방법을 지원 도구와 함께 주택 통합 시스템(HIS)에 적용함으로써 예제를 통해 제안된 도메인 분석 방안의 공정을 구체적으로 제시하였고, HIS 프로덕트 라인을 위한 프로덕트 라인 영역 결정과 도메인 요구사항 분석, 그리고 공통성 및 가변성 분석을 통해 프로덕트 라인의 공통성과 가변성을 식별하고, 그 근거를 지원도구의 도움으로 프로덕트 라인의 목표를 바탕으로 제시할 수 있었다.

향후 연구로는 본 논문에서 제안된 도메인 분석 방법에서의 공통성과 가변성 분석에 대한 근거를 더욱 구체적으로 제공하기 위한 방안이 필요하다고 하겠다. 즉, 본 논문에서는 DRM의 구성, 대체 그리고 선택 관계를 통해 공통성과 가변성을 나타내고, 이를 바탕으로 도메인 요구사항의 공통성과 가변성을 식별하고 있으나, 최상위 수준의 비즈니스 목표를 바탕으로 그것을 더욱 만족시키기 위해 기존의 공통성과 가변성 분석의 연구 분야에서 사용되고 있는 방법들을 통합할 필요가 있다. 또한 이 연구를 바탕으로 프로덕트 라인 개발 조직의 비즈니스 목표를 만족시키는 프로덕트 개발을 위해 가장 중요한 요소 중의 하나인 프로덕트 라인의 비기능적인(non-functional) 부분 즉, 품질 요소(quality attributes)에 대한 연구가 좀 더 구체적으로 이루어져야 한다. 이는 기존 요구공학 분야에서 비기능적인 요구사항을 분석하기 위해 사용되고 있는 NFR(Non-Functional Requirements) 프레임워크와 같은 방법론의 연구를 통해 프로덕트 라인을 위한 체계적인 비기능적 요구사항 관리 방법이 제공되어야 함을 의미한다.

참고 문헌

- [1] J. Bosch, *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*, Addison-Wesley, Boston, 2000.
- [2] P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*, Boston, MA: Addison Wesley Longman, Inc., 2001.
- [3] Donohoe, P., ed., *Software Product Lines: Experience and Research Directions*. Denver, Colorado, August 28-31 2000. Boston, MA: Kluwer Academic Publishers, 2000.
- [4] K.C. Kang, K. Lee, J. Lee, S. Kim, "Feature Oriented Product Line Software Engineering: Principles and Guidelines," *Domain Oriented Systems Development: Practices and Perspectives*, Taylor & Francis, pp. 19-36, 2003.
- [5] K.C. Kang, J. Lee, P. Donohoe, "Feature-Oriented Product Line Engineering," *IEEE Software*, Vol. 9, No. 4, Jul./Aug., pp. 58-65, 2002.
- [6] D. Fey, R. Fajta, A. Boros, "Feature Modeling: A Meta-model to Enhance Usability and Usefulness," G. Chastek, ed., *Proc. 2nd Software Product Line Conf.*, Springer Lecture Notes in Computer Science, vol. 2379, Heidelberg, Germany, 2002.
- [7] W. B. Frakes and K. C. Kang "Software Reuse Research: Status and Future," *IEEE Transactions on Software Engineering(TSE)*, Vol. 31, No. 7, July, pp.529-536, 2005.
- [8] K.C. Kang, S. Cohen, J. Hess, W. Novak, A. Peterson, *Feature-Oriented Domain Analysis (FODA) Feasibility Study*, tech. report CMU/SEI-90-TR-21, Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh, PA, 1990.
- [9] K.C. Kang, S. Kim, J. Lee, K. Kim, E. Shin, M. Huh, "FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures," *Annals of Software Engineering*, Vol. 5, pp. 143-168, 1998.
- [10] S. Thiel, F. Peruzzi, "Starting a Product Line Approach for an Envisioned Market: Research and Experience in an Industrial Environment," Patrick, ed. *Proc. 1st Software Product Line Conf.(SPLCI)*, Kluwer Academic Publishers, Boston, pp.495-512, 2000.
- [11] G. Chastek, P. Donohoe, K.C. Kang, S. Thiel, *Product Line Analysis: A Practical Introduction*, tech. report CMU/SEI-2001-TR-001, Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh, 2001.
- [12] C. Potts, K. Takahashi, and A. I. Antn, "Inquiry-based requirements analysis," *IEEE Softw.*, vol. 11, no. 2, pp. 21-32, 1994.
- [13] A. Sutcliffe, "Scenario-based requirement analysis," *Requirements Engineering Journal*, pp. 48-65, 1998.
- [14] A.I. Antn, "Goal-based requirements analysis," in *Proc. 2nd Int. Conf. Requirements Engineering (ICRE'96)*, Colorado Springs, CO, April, pp. 136-144, 1996.
- [15] C. Rolland, C. Souveyet, and C. Ben Achour, "Guiding goal modeling using scenarios," *IEEE*

- Trans. Software Eng.*, vol. 24, pp. 1055-1071, 1998.
- [16] J. Lee and K. C. Kang, "Feature Binding Analysis for Product Line Component Development," Proceedings of the Fifth International Workshop on Product Family Engineering (PFE-5), Siena, Italy, November 4-6, pp.266-276, 2003.
- [17] K. Lee, K. C. Kang, M. Kim and S. Park, "Combining Feature-Oriented Analysis and Aspect-Oriented Programming for Product Line Asset Development", Proc. of 10th International Software Product Line Conference (SPLC 2006), Baltimore, Maryland, 2006 (accepted for publication).
- [18] I. Jacobson, M. Griss, P. Jonsson, *Software Reuse: Architecture, Process and Organization for Business Success*, Addison-Wesley Publishing Company, May 1997.
- [19] A. D. Vici and N. Argentieri, "FODacom: An Experience with Domain Analysis in the Italian Telecom Industry," Proceedings of the 5th International Conference on Software Reuse (ICSR), Victoria, BC, Canada, pp. 166-175, June 1998.
- [20] M. Griss, J. Favaro, M. d'Alessandro, "Integrating Feature Modeling with the RSEB," Proc. 5th International Conference on Software Reuse (ICSR'98). IEEE Computer Society Press. Victoria BC, Canada, June2-5, 1998.
- [21] M. Eriksson, J. Borstler, and K. Borg, "The PLUSS Approach - Domain Modeling with Features, Use Cases and Use Case Realizations," 9th International Software Product Line Conference (SPLC Europe), Rennes, France, September 26-29, pp.33-44, 2005.
- [22] E. Yu, "Why Goal-Oriented Requirements Engineering," *Proc. Fourth Int'l Workshop Requirements Eng.: Foundation for Software Quality (RESFQ '98)*, Pisa, Italy, June 1998.
- [23] J. Mylopoulos, L. Chung, and E. Yu, "From Object-Oriented to Goal-Oriented Requirements Analysis," Communications of the ACM, 42(1), pp. 31-37, 1999.
- [24] A. van Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour," *Proc. RE'01 5th Intl. IEEE Symp. on Requirements Engineering*, August , pp. 249-263, 2001.
- [25] E. Letier, A. van Lamsweerde, "Reasoning about partial goal satisfaction for requirements and design engineering," SIGSOFT FSE pp.53-62, 2004.
- [26] H. Van, A. van Lamsweerde, Philippe Massonet, Christophe Ponsard: Goal-Oriented Requirements Animation. Proc. of 12th IEEE Int. Req. Engineering Conf. (RE'04), pp. 218-228, 2004.
- [27] B. Nuseibeh, S. Easterbrook, "Requirements Engineering: A Roadmap," In: The Future of Software Engineering, Special Issue 22nd International Conference on Software Engineering, ACM, pp. 37-46. 2000.
- [28] P. Hsia, J. Samuel, J. Gao, D.Kung,Y. Toyoshima, and C. Chen, "Formal approach to scenario analysis," *IEEE Softw.*, vol. 11, no. 2, pp. 33-41, 1994.
- [29] C. Rolland, G. Grosz, R. Kla, "Experience With Goal-Scenario Coupling In Requirements Engineering," Fourth IEEE International Symposium on Requirements Engineering, 1999.
- [30] K. C. Kang, S. Kim, J. Lee, and K. Lee, "Feature-Oriented Engineering of PBX Software for Adaptability and Reusability," *Software- Practice and Experience*, Vol. 29, Issue 10, pp. 875-896, 1999.
- [31] B. Chandrasekaran and H. Kaindl, "Representing functional requirements and user-system interactions," in *Proc. AAAI-96 Workshop on Modeling and Reasoning about Function*, Portland, OR: AAAI, Aug.1996.
- [32] H. Kaindl, "An integration of scenarios with their purposes in task modeling," in *Proc. Symp. Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '95)*. Ann Arbor, MI: ACM, August, pp. 227-235, 1995.
- [33] H. Kaindl, "A Design Process Based on a Model Combining Scenarios with Goals and Functions," IEEE Transactions on Systems, Man, and Cybernetics, Part A, Vol. 30, No. 5, September 2000.
- [34] A. Dardenne, A. van Lamsweerde and S. Fickas, "Goal-Directed Requirements Acquisition," *Science of Computer Programming*, Vol.20, pp. 3-50, 1993.
- [35] Karl E. Wiegers, *Software Requirements*, Microsoft Press, 1999.
- [36] S. Jacobs and R. Holten, "Goal-Driven Business Modelling Supporting Decision Making within Information Systems Development," *Proc. Conf. On Organizational Computing Systems*, Milpitas, Calif. pp. 96-105, 1995.
- [37] G. Chastek, P. Donohoe, J. McGregor, *Product Line Production Planning for the Home Integration System Example*, tech. note CMU/SEI-2002-TN-029, Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh, 2002.



김민성

2001년 서강대학교 컴퓨터학과(공학사) Summa Cum Laude. 2003년 서강대학교 컴퓨터학과(공학석사). 2003년~현재, 서강대학교 컴퓨터학과 박사과정. 관심분야는 요구공학, 소프트웨어 프로토타입, 자기 적응형 소프트웨어

박수용

정보과학회논문지 : 소프트웨어 및 응용 제 33 권 제 1 호 참조