Aspect-Oriented 소프트웨어 개발을 위한 목표-시나리오 모델링 기반의 횡단관심사 식별 및 명세화 방법

(An Identification and Specification Method of Crosscutting Concerns based on Goal-Scenario Modeling for Aspect-Oriented Software Development)

김 선 화 † 김 민 성 † 박 수 용 ††
(SunHwa Kim) (MinSeong Kim) (SooYong Park)

요 약 관점지향 소프트웨어 개발방법론(Aspect-Oriented Software Development)에서 가장 중요한 고려사항중 하나는 요구사항 분석단계에서 횡단관심사(Crosscutting Concerns)를 식별하는 것이다. 이는 개발 초기단계에 횡단관심사를 식별함으로써 요구사항의 일관성(consistency)을 증진시켜 시스템의 유지보수를 쉽게 하고, 개발단계의 산출물들 사이의 추적성(traceability)을 제공하여 체계적인 변경관리를 지원할 수 있기 때문이다. 따라서 소프트웨어 개발 초기단계에 횡단관심사를 식별하고, 이를 독립적인 모듈로 구현하는 것을 지원하기 위해서는 다음과 같은 사항들을 고려해야 한다. 첫째, 복잡하게 분산되고 엉킨 요구사항의 관심사를 분리되어야 한다. 둘째, 횡단관심사가 시스템을 횡단하는 시점이 식별되어야 한다. 셋째, 횡단관심사를 구현한 모듈과 이것이 횡단하는 다른 모듈들 사이에 발생할 수 있는 요구사항 충돌을 통합 이전단계에 관리할 수 있는 방법이 지원되어야 한다. 이를 위해 본 논문에서는 목표와 시나리오 기반의 요구사항 분석 방법을 기반으로 횡단관심사를 식별하는 방법을 제안한다. 그리고 제안된 방법을 지능형 로봇 소프트웨어 개발 사례에 적용하여 그 유용성을 검증한다.

키워드 : 관점, 횡단관심사, 목표, 시나리오

Abstract Identifying crosscutting concerns during requirements engineering phase is one of the most essential parts in Aspect-Oriented Software Development. Considering crosscutting concerns in the earlier phase of the development improves consistency among requirements so that it can help maintain software systems efficiently and effectively. It also provides a systematic way to manage requirements changes by supporting traceability throughout the software lifecycle. Thus, identifying tangled and scattered concerns, and encapsulating them into separate entities must be addressed from the early phase of the development. To do so, first, functional and non-functional concerns must be clearly separated. Second, a pointcut where a main concern meets crosscutting concerns should be defined and specified precisely. Third, it is required to detect conflicts being occurred during composition of crosscutting concerns from the earlier phase. Therefore, this paper proposes a systematic approach to identifying and specifying crosscutting concerns using goal-scenario based requirements analysis. And we demonstrate the applicability of the approach by applying it into the intelligent service robot system.

Key words : Aspect, Crosscutting concerns, Goal, Scenario

· 이 논문은 2006년도 정부(교육과학기술부)의 재원으로 한국과학재단 특정기초연구사업의 지원을 받아 수행된 연구임(No. R01-2006-000-10536-0)

† 학생회원 : 서강대학교 컴퓨터학과
sunua@sogang.ac.kr
minskim@sogang.ac.kr
†† 정 회 원 : 서강대학교 컴퓨터학과 교수
sypark@sogang.ac.kr

논문접수 : 2007년 10월 8일
심사완료 : 2008년 6월 25일

정보과학회논문지: 소프트웨어 및 응용 제35권 제7호(2008.7)

1. 서 론

관심 분리(Separation of Concerns)는 문제영역을 독립적인 모듈로 분리하고 개발하기 위한 중요한 소프트웨어 개발 원칙 중 하나이다. 그러나 실제 프로젝트에서 관심을 완전히 독립적인 모듈로 분리 및 개발하는 것은 매우 어렵다. 이것은 특정 관심이 시스템의 다른 관심들에게 영향을 미치는 횡단관심사(Crosscutting Concerns)가 존재하기 때문이다. 여기서 횡단 관심사는 엉키고(tangling, 하나의 모듈에 여러 개의 관심사가 엉켜 있음) 분산된(scattering, 하나의 관심사가 여러 개의 모듈에 분산되어 있음) 관심사를 의미한다. 최근에는 이러한 횡단 관심사를 지원하기 위해 관점지향 소프트웨어 개발방법론(Aspect-Oriented Software Development)이 제안되고 있다. 이 방법론에서는 횡단관심사를 관점(Aspect)이라는 독립적인 모듈로 구현함으로써 개발의 효율성을 높이고, 유지보수 비용을 절감하고자 한다.

특히, 횡단관심사 중에서 요구사항 분석단계에서 식별된 횡단관심사는 이후 개발 단계에서 여러 개의 모듈로 설계 및 구현된다. 이와 같이 여러 개의 설계 및 구현모듈에 하나의 관심사가 분산되어 있는 것은 요구사항의 일관성(consistency)을 떨어뜨리고 체계적인 변경관리를 어렵게 하여 시스템의 유지보수 및 재사용을 힘들게 하는 원인중 하나이다. 그러므로 요구사항의 일관성을 증진시키고, 개발단계에서의 추적성(traceability)을 제공하여 체계적인 변경관리를 지원하기 위해서는 개발 초기단계에 횡단관심사를 식별하여 이것을 독립적인 모듈로 구현하도록 지원하는 방법이 필요하다.

따라서 개발 초기단계에 횡단관심사를 식별하고, 이것을 설계 및 구현단계에서 관점으로 전환되는 것을 지원하기 위해서는 다음과 같은 사항들이 고려되어야 한다. 첫째, 복잡하게 분산되고 엉킨 요구사항들을 관심 분리를 통해 시스템의 기능적 관심사 및 비기능적 관심사로 분리해야 한다. 분리된 관심사들은 관심사 간의 관계 분석을 통해 횡단관심사로 식별할 수 있다. 둘째, 횡단관심사를 독립적인 구현모듈인 관점으로 전환하는 것을 지원하기 위해서는 시스템에 삽입되어 동작하는 횡단시점을 식별할 수 있어야 한다. 셋째, 횡단관심사를 구현하는 모듈은 기존의 관심사를 구현하는 여러 모듈들에 영향을 미치기 때문에, 통합 이전단계에서 이들의 영향관계를 미리 파악하는 요구사항 충돌 검증방법이 필요하다. 이를 통해 유지보수 과정에서 횡단관심사를 구현한 모듈의 추가 및 삭제 이후에 발생할 수 있는 위험을 최소화 할 수 있다.

본 논문에서는 위 세가지 활동을 지원하기 위해 목표와 시나리오 기반의 요구사항 분석방법을 사용하여 횡단관심사를 식별하는 방법을 제안한다. 먼저 목표 그래프를 사용하여 각 관심사를 구조화 하고 목표와 시나리오의 상호작용을 통해 기능적, 비기능적 관심을 분리한다. 횡단관심사는 시스템의 여러 부분에 분산된 관심사이므로, 목표 그래프에 중복되어 나타나는 목표를 횡단관심사로서 식별한다. 또한 목표에 대응되는 시나리오를 통해 요구사항 수준에서 횡단 시점을 식별한다. 그리고 목표를 위해 기술된 시나리오를 바탕으로 횡단관심사에 대한 영향 관계 분석을 지원하고자 한다. 즉, 식별된 횡단관심사를 위한 각각의 시나리오를 모델링하고 통합함으로써, 이를 바탕으로 [1]에서 제안된 방법을 통해 개발 초기단계에서 요구사항 검증이 가능해 진다.

본 논문의 구성은 다음과 같다. 2장에서는 기존연구들을 비교하고, 3장에서는 목표와 시나리오 기반의 요구사항 분석방법에 대한 배경 지식을 살펴본다. 4장에서는 개발 초기단계에 횡단관심사를 식별하는 방법을 제안하고, 5장에서는 제안된 방법을 적용한 사례연구를 살펴본다. 마지막으로 6장에서는 결론 및 향후 연구를 제시한다.

2. 관련 연구

2.1 기존의 접근 방법

Rashid는 기존의 뷰포인트(Viewpoint)기반의 요구공학 모델인 PREview를 기반으로 횡단관심사를 식별, 명세하는 모델을 제안하였다[2]. 이 모델은 관심들을 여러 뷰포인트에서 식별함으로서 여러 곳에 영향을 미칠 수 있는 관심사들을 식별한다. 일반적으로 분석초기에 식별되는 횡단 관심들은 높은 수준의 비기능적 요구사항인 경우가 많으므로 비기능적인 관심에 대한 쉬운 접근방법을 제공하지만, 기능적인 관심에 관련된 횡단관심사를 식별하는 구체적인 방법을 언급하고 있지 않다.

V-graph 모델은 전통적인 목표기반의 접근방법인NFR-Framework(Non-Functional Requirements Framework)를 기반으로 관점을 식별하는 방법을 제안 하였다[3]. V-graph 모델은 요구사항간의 협력 관계를 표현하므로 횡단관심사를 식별하고 그들간의 영향관계를 파악하기에 용이하다. 그러나 V-graph의 기반인 NFRFramework이 주로 비기능적인 요구사항을 분석하는데 초점을 맞추고 있으며, 구체적인 횡단 시점 식별을 위한 방법을 언급하고 있지 않다.

Jacobson은 유스케이스 그 자체가 횡단관심사가 될 수 있다고 하였다[4]. 그는 횡단관심사를 확장 유스케이스(use-case extension)를 식별하면서 횡단 시점인 확장 포인트(extension point)도 함께 식별하는 방법을 제안하였다. 비록 비기능적인 요구사항을 표현하는 개념을 도입하긴 했지만, 유스케이스는 주로 사용자와의 상호작용을 기반으로 요구사항을 분석하기 때문에 비기능적인

요구사항을 식별 및 명세 하는데 있어 낮은 수준의 지원을 제공한다.

2.2 관련 연구 비교

본 장에서는 개발 초기단계에 횡단관심사를 식별하기 위해 고려해야 할 사항들을 기준으로 각 연구들을 비교 및 평가한다. 평가방법은 중간수준의 지원(**)을 기준으로, 각 접근방법이 특별히 초점을 맞추고 있는 항목에 더 높은 점수를 부여하였다. 표 1에서 나타난 것과 같이, 뷰포인트 및 목표 기반의 접근방법이 비기능적인 관심사에 초점을 맞추고 있는 반면 유스케이스기반의 방법은 기능적인 관심사에 초점을 맞추고 있다. 따라서 뷰포인트 및 목표기반의 접근방법은 비기능관심사 식별 항목에, 유스케이스기반의 접근방법은 기능관심사 식별 항목에 각각 높은 점수를 부여하였다. 그리고 유스케이스기반의 접근방법은 횡단시점을 식별하는 방법을 제공하며 목표기반의 접근방법은 관심사간의 영향관계를 식별하는 방법을 지원한다.

3. 배경 지식

본 장에서는 목표와 시나리오 기반의 요구사항 분석 방법[7]을 설명한다. 이 방법에서는 상위 목표를 성취하기 위한 구체적인 흐름을 시나리오를 통해 기술하고, 시나리오의 흐름을 바탕으로 하위 목표를 식별한다. 또한 목표와 시나리오를 4개의 추상수준으로 분리하여 하위수준으로의 분화과정을 마쳐야 하는 지점을 정의한다. 4개의 추상화 수준은 다음과 같다.

- 비즈니스 수준 : 시스템에 대한 궁극적인 목표를 정의하고 기술한다.
- 서비스 수준 : 시스템이 조직이나 사용자에게 제공해야 하는 서비스들을 식별한다.
- 상호작용 수준 : 시스템과 사용자들 간의 상호작용을 기술한다.
- 내부 수준 : 시스템 내부 객체나 외부 객체를 포함하는 시스템의 행위들을 나타낸다.

본 논문에서 목표와 시나리오 기반의 분석방법을 사용한 이유는 다음과 같다. 첫째, 관심사는 일반적으로 시스템이 성취해야 하는 목표에 대응하므로, 목표를 분석하는 활동은 관심사들을 식별하고 구조화 하는 것을 지원하기 때문이다. 둘째, 이 방법을 통해 낮은 수준의 목표를 분석함으로써 시스템 내부 객체에 분산되어 있는 횡단관심사의 식별이 가능해 진다. 마지막으로, 시나리오를 사용하여 요구사항을 분석하고 명세하는 것은 개발 초기단계에 횡단관심사와 다른 관심사간의 충돌을 식별할 수 있도록 지원하기 때문이다[1].

4. 목표 및 시나리오 기반의 횡단관심사 식별 방법

본 장에서는 목표와 시나리오 기반의 요구사항 분석 방법을 기반으로 횡단관심사를 식별하는 방법을 제안한다. 그림 1은 목표와 시나리오 기반의 요구사항 분석방법을 사용하여 횡단관심사를 식별하는 절차를 보여준다. 이 방법에서, 시스템의 전체 관심은 목표와 시나리오 기반의 요구사항 분석방법을 사용하여 분석된다. 그리고 분석된 목표의 중복을 찾아내는 활동을 통해 횡단관심사를 식별할 수 있으며, 식별된 횡단관심사는 관점 후보(Candidate Aspect)가 되어 이후 설계 및 구현단계에서 관점으로 분리되어 개발 될 수 있다. 그리고 각 목표에 대응하는 시나리오를 기반으로 횡단관심사가 어떤 핵심 기능의 어느 시점에 삽입되어 동작하는지를 식별한다. 마지막으로, 기존의 시나리오에서 횡단관심사를 분리하고 각각의 시나리오를 명세하여 독립적인 모듈로 설계 및 구현하는 것을 지원한다.

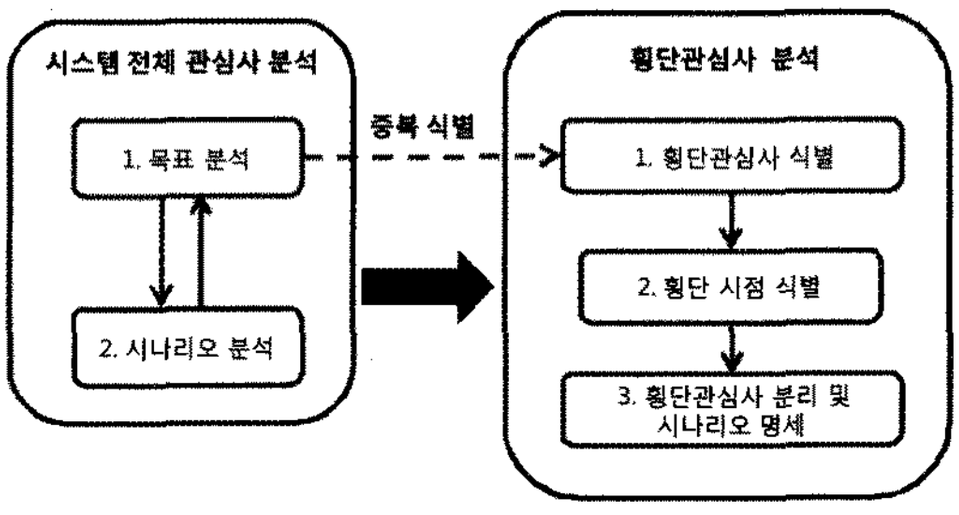


그림 1 횡단관심사 식별 절차

4.1 시스템 전체 관심사 분석

본 장에서는 시스템의 전체 관심을 식별하기 위해 목표와 시나리오 기반의 요구사항 분석 방법을 사용한다. 식별된 목표와 시나리오는 다음과 같은 명세 규칙을 갖는다[5].

표 1 관련 연구 비교 및 평가

	기능 관심사 식별	비기능 관심사 식별	횡단시점 식별	관심사간의 영향관계 식별
뷰포인트	*	***	–	–
목표	*	***	–	**
유스케이스	***	*	**	–

(–: 지원하지 않음, *: 낮은수준 지원, **: 중간수준 지원, ***: 높은수준 지원)

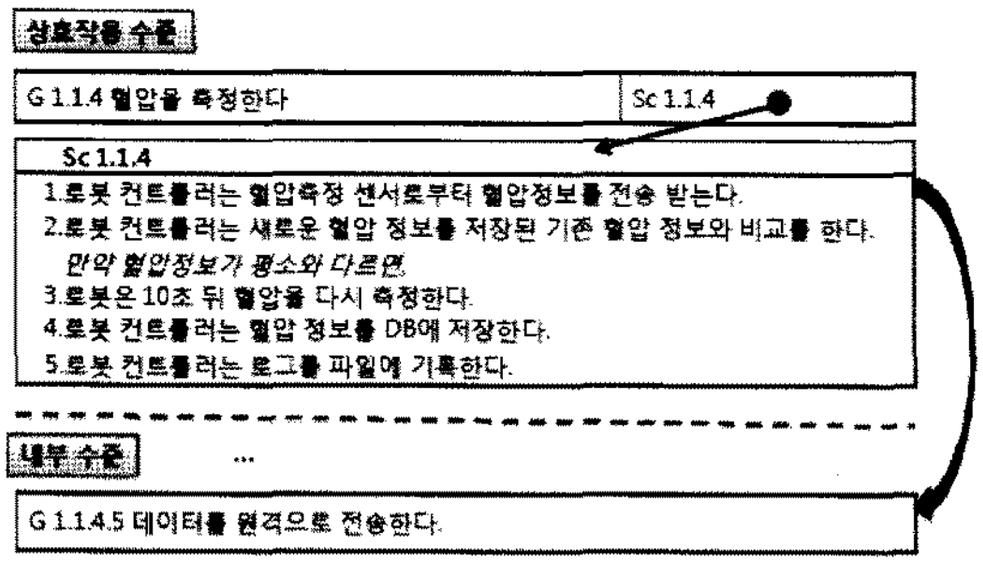


그림 2 상호작용 수준에서의 목표 그래프

- 목표는 <목적어(Target) + 방법(Direction) + 동사(Verb) > 형식으로 기술한다.
- 시나리오는 <주어(Subject) +방법(Direction) + 목적어(Target) + 동사(Verb)> 혹은 <주어(Subject) + 목적어(Target) + 방법(Direction) + 동사(Verb)>의 형식으로 기술한다.

위의 명세규칙에 따라 그림 2에 나타난 것과 같이 상위 목표 달성을 위해 기대되거나 가능한 흐름을 시나리오로서 기술한다. 그리고 명세된 시나리오 통해 하위 목표를 식별한다.

4.2 횡단 관심사 분석

4.2.1 횡단관심사 식별

본 장에서는 식별된 목표와 시나리오를 기반으로 횡단관심사를 식별한다. 횡단관심사는 시스템의 여러 관심사에 대해 횡단관계를 갖는 관심사이므로, 목표 그래프에서 중복되어 나타나는 목표는 횡단관심사로 식별될 수 있다.

횡단관심사는 동종 횡단(homogeneous crosscuts)과 이종 횡단(heterogeneous crosscuts) 두 가지로 분류될 수 있다[6]. 동종 횡단은 똑같은 관심이 시스템의 여러 곳에 분산되어 있는 것을 말하고, 이종 횡단은 큰 관심은 같지만 세부적으로는 조금씩 다른 관심이 시스템의 여러 곳에 분산되어 있는 것을 말한다. 방법(Direction)과 행위(Target, Verb)가 같은 목표가 목표 그래프에 중복되어 나타나는 경우는 동종 횡단관심사로 식별될 수 있다. 반면에 방법(Direction)은 다르나 행위(Target, Verb)가 같은 목표가 목표 그래프에 중복되어 나타나는 경우는 이종 횡단관심사로 식별될 수 있다.

그림 3은 동종 횡단관심사를 식별한 예이다. 목표 그래프에서 'G1.1.4혈압을 측정한다.'라는 상호작용 수준의 목표는 'G1.1.4.5로그를 파일에 기록한다.'라는 내부 수준의 목표를 가진다. 또한 'G1.1.5정보를 병원으로 전송한다.' 라는 상호작용 수준의 목표 역시 'G1.1.5.4로그를 파일에 기록한다'라는 내부 수준의 목표를 가진다. 이 경우 '로그를 기록한다.'라는 행위는 목표모델에서 중복

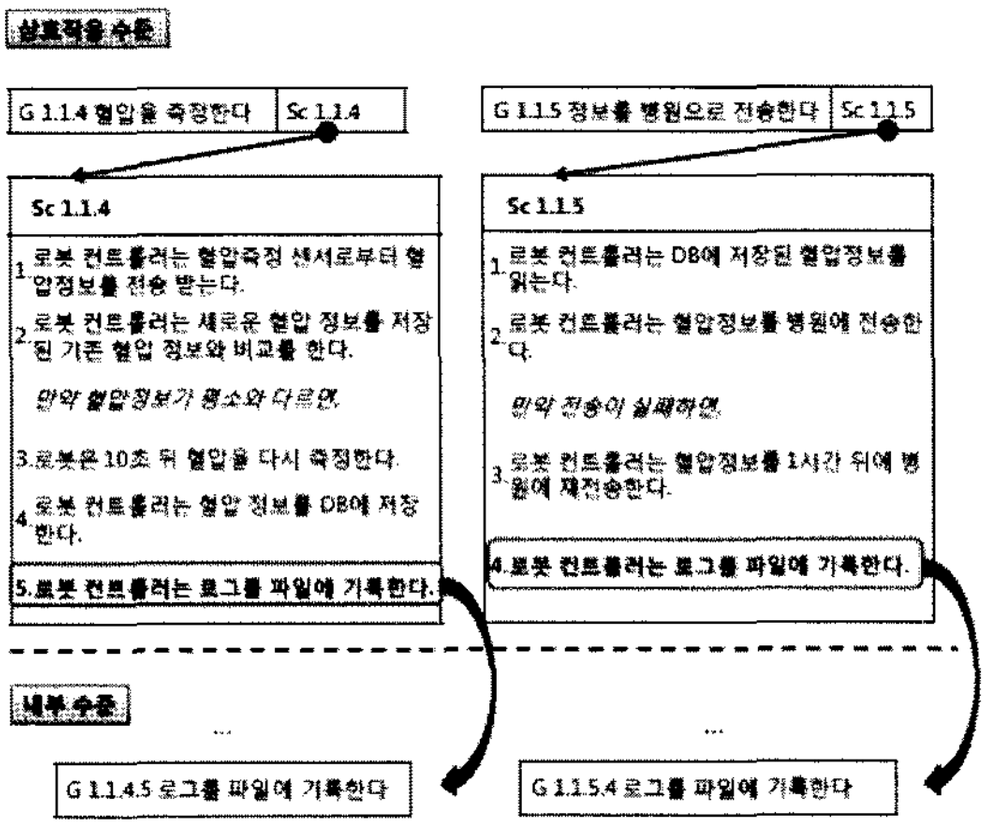


그림 3 동종 횡단관심사 식별

되어 나타나므로 횡단관심사로 식별될 수 있다. 또한 'G1.1.4.5로그를 파일에 기록한다.' 와 'G1.1.5.4로그를 파일에 기록한다.'는 방법(Direction)과 행위(Target, Verb)가 모두 똑같으므로 동종 횡단관심사이다.

그림 4는 이종 횡단관심사를 식별한 예이다. 목표 그래프에서 'G1.1.4혈압을 측정한다.'라는 상호작용 수준의 목표는 'G1.1.4.3타이머를 10초마다 동작시킨다.'라는 내부 수준의 목표를 가진다. 또한 'G1.1.5정보를 병원으로 전송한다.'라는 상호작용 수준의 목표는 'G1.1.5.3타이머를 1시간마다 동작시킨다.' 라는 내부 수준의 목표를 가진다. 이 경우 '타이머를 동작시킨다.' 라는 행위는 목표모델에서 중복되어 나타나므로 횡단관심사로 식별될 수 있다. 또한 'G1.1.4.3타이머를 10초 마다 동작시킨다.'와 'G1.1.5.3타이머를 1시간마다 동작 시킨다.'는 방법(Direction)은 다르나 행위(Target, Verb)가 같으므로 이종 횡단관심사이다.

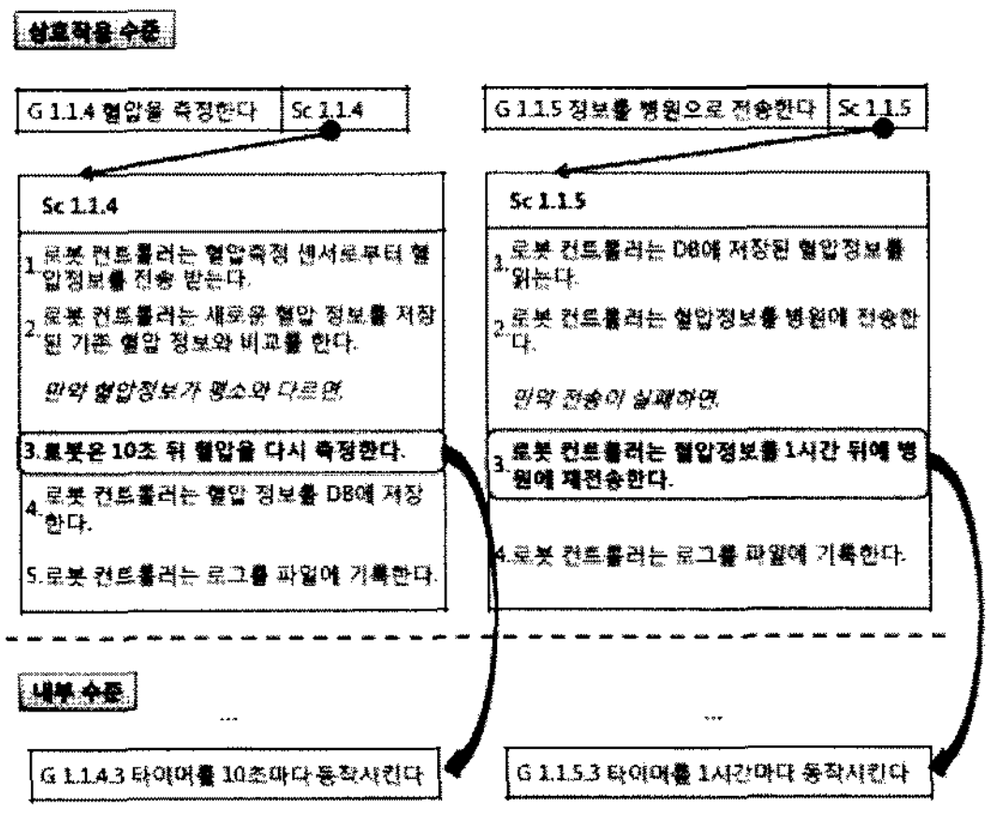


그림 4 이종 횡단관심사 식별

이렇게 식별된 횡단관심사는 관점 후보(Candidate Aspect)가 되며, 이후 설계 및 구현단계에서 관점으로 구현될 수 있다.

4.2.2 횡단 시점 식별

본 장에서는 식별된 목표에 대응하는 시나리오를 기반으로 횡단관심사가 다른 관심의 어느 시점을 횡단하는지를 식별하는 방법을 제안한다. 그림 5는 식별된 횡단관심사가 다른 관심의 어느 시점을 횡단하는지를 나타내는 예이다. 'G1.1.4.3 타이머를 10초마다 동작시킨다'라는 목표는 'G1.1.4혈압을 측정한다.'라는 목표에서 '만약 혈압정보가 평소와 다르면'이라는 에러시점을 횡단한다. 따라서 횡단 시점은 혈압정보가 평소와 다른 시점이다. 또한 'G1.1.5.3타이머를 1시간 마다 동작시킨다.'라는 목표는 'G1.1.5정보를 병원으로 전송한다.'라는 목표에서 '만약 전송이 실패하면'이라는 에러 시점을 횡단한다. 따라서 횡단 시점은 전송이 실패한 시점이다.

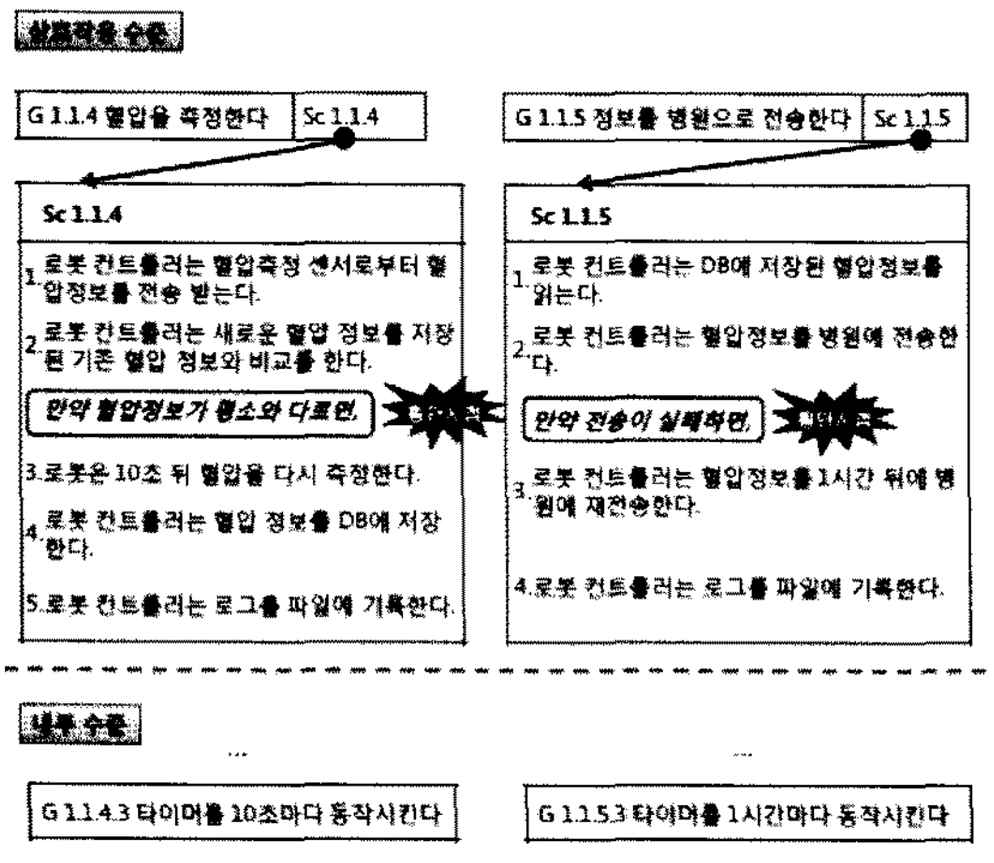


그림 5 횡단 시점 식별

4.2.3 횡단관심사 분리 및 시나리오 명세

본 장에서는 기존의 시나리오에서 횡단관심사를 분리하고 각각의 시나리오를 명세하는 방법을 설명한다. 식별된 횡단관심사에 대한 흐름은 원래 포함되어있던 기존의 관심사의 흐름에서 분리되어 명세된다. 각각의 명세된 시나리오는 설계 및 구현단계를 거쳐 기존의 기능을 구현하는 모듈과 횡단관심사를 구현하는 모듈로 생성되고, 이를 통합하는 위빙(Weaving)단계를 거쳐 함께 동작한다.

분리된 횡단관심사의 시나리오는 UML기반의 PSs (Pattern Specifications)를 사용하여 명세한다[9]. PSs는 패턴을 명세하는 도구로서 구조 및 행위가 역할(role)을 바탕으로 하기 때문에 여러 곳에서 중복되어

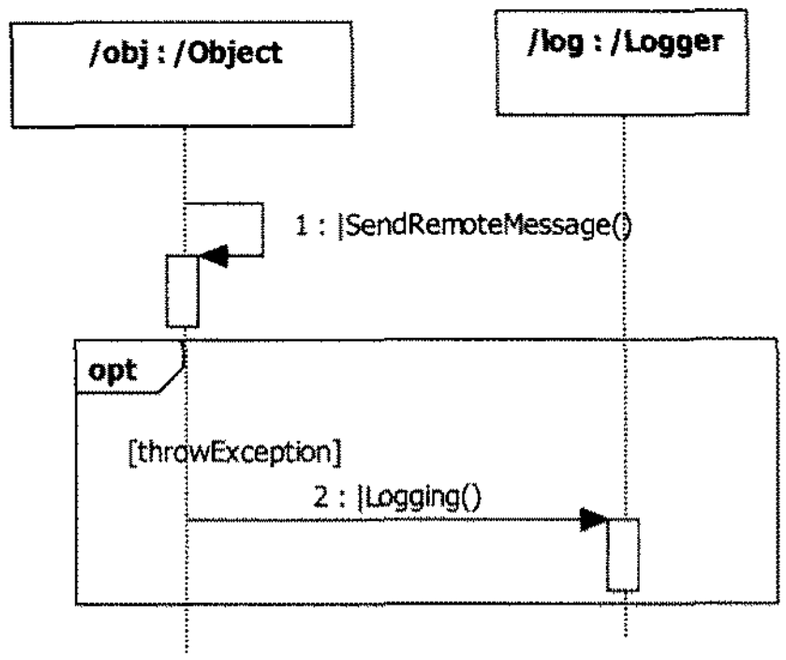


그림 6 횡단관심사 명세

발생하는 횡단관심사를 하나로 대표하여 표현하는데 적합하다. 본 논문에서는 PSs중에서도 행위를 명세하는 도구인 IPSs(Interaction Pattern Specifications)를 사용 하였다. 그림 6은 4.2.1장에서 식별된 '로그를 파일에 기록한다'라는 횡단관심사를 IPSs를 사용하여 명세한 것이다. 그리고 횡단관심사를 분리한 기존의 시나리오는 UML 시퀀스 다이어그램을 사용하여 명세한다.

5. 사례 연구

본 장에서는 지능형 서비스 로봇 시스템(T-Rot)[10]의 요구사항을 대상으로 횡단관심사를 식별한 사례를 살펴본다. T-Rot은 다양한 하드웨어와 소프트웨어로 이루어진 복잡한 시스템으로, 하나의 요구사항에는 여러 개의 관심사들이 엉켜있으며, 관심사가 여러 개의 하드웨어 동작 및 소프트웨어 기능에 분산되어 있다. 이로 인해 시스템을 독립된 모듈로 분리하여 구현하는 것이 어렵고, 변경사항이 각 개발단계의 어떤 영향을 미치는지 파악하는 것이 어렵다. 따라서 요구사항 일관성 및 추적성을 제공하기 위해서는 개발 초기 단계에 횡단관심사를 식별하는 것을 지원하여 독립적인 모듈로 구현하는 것이 필요하다.

5.1 검증 방법

본 논문에서 제안하는 방법을 검증하기 위해, T-Rot의 요구사항을 대상으로 하여 목표와 시나리오 기반의 방법 및 유스케이스 기반의 방법으로 각각 횡단관심사를 식별하고 그 갯수를 비교해 보았다. T-Rot과 같은 실시간으로 상황을 판단하고 적응하는 시스템에서는 타이머 동작 및 로깅등이 대표적인 횡단관심사 및 관점으로 식별될 수 있다. 요구사항 분석단계에서 식별된 횡단관심사는 설계 및 구현단계에서 관점으로 모듈화 되어 구현되지만, 식별된 모든 횡단관심사가 관점으로 전환되는 것은 아니다. 따라서 결과 데이타의 정확성을 보장하기 위해 구현 단계에 관점으로 식별될 수 있는 모듈들

을 기준 데이타로 정의하였다. 한편 구현단계에서 식별되는 관점들은 소스코드에 중복되어 나타나거나 다수의 요구사항 실현에 관여하는 특성이 있다. 그러므로 이 실험에서는 리팩토링을 통해 여러 부분에 중복되어있는 코드를 찾고, 함수 호출트리를 통해 다수의 테스트 케이스에 의해 호출되는 함수를 찾아 각각을 관점으로 식별하였다. 실험의 정확성을 위해 최소한 3군데 이상의 중복 코드가 발생거나, 3개 이상의 테스트케이스에 의해 호출되는 코드를 관점으로 식별하는 것으로 가정하였다. 이것은 개발경험에 근거한 기준으로, 일반적으로 전체 소스코드에 최소한 3곳 이상 중복되거나 호출되는 부분은 관점으로 모듈화 시킬 수 있다고 가정한 것이다.

5.2 검증 결과 및 분석

그림 7에서 나타난 것과 같이, 전체 31개의 기준 데이타 중에서 본 논문의 방법을 사용했을 경우에는 21개의 횡단관심사가 식별되었으며, 유스케이스 기반의 방법을 사용했을 경우에는 11개의 횡단관심사가 식별되었다. 따라서 본 논문의 방법(68%)이 유스케이스 기반의 방법(35%)에 비해 더 효과적인 횡단관심사 식별방법을 제공하는 것으로 판단할 수 있다. 그 이유 중 하나는 본 논문의 분석방법이 시스템의 비기능적인 관심사를 분석하는 가이드라인을 제공하기 때문이다. 또한 유스케이스 기반의 방법이 액터와 시스템 수준에서 요구사항을 분석하는 반면, 본 논문의 방법은 내부 수준의 목표를 식별하는 공정을 포함하기 때문에 좀 더 하위 수준의 횡단관심사를 식별할 수 있다.

또한 기준 데이타의 32%가 어떤 방법으로도 식별되지 않은 것으로 나타나는데, 이와 같은 현상의 이유는 개발 초기단계에 모든 관점후보를 식별할 수 있는 것은 아니며, 본 논문에서는 요구사항 분석단계에서 식별하는 관점후보만을 대상으로 하기 때문이다.

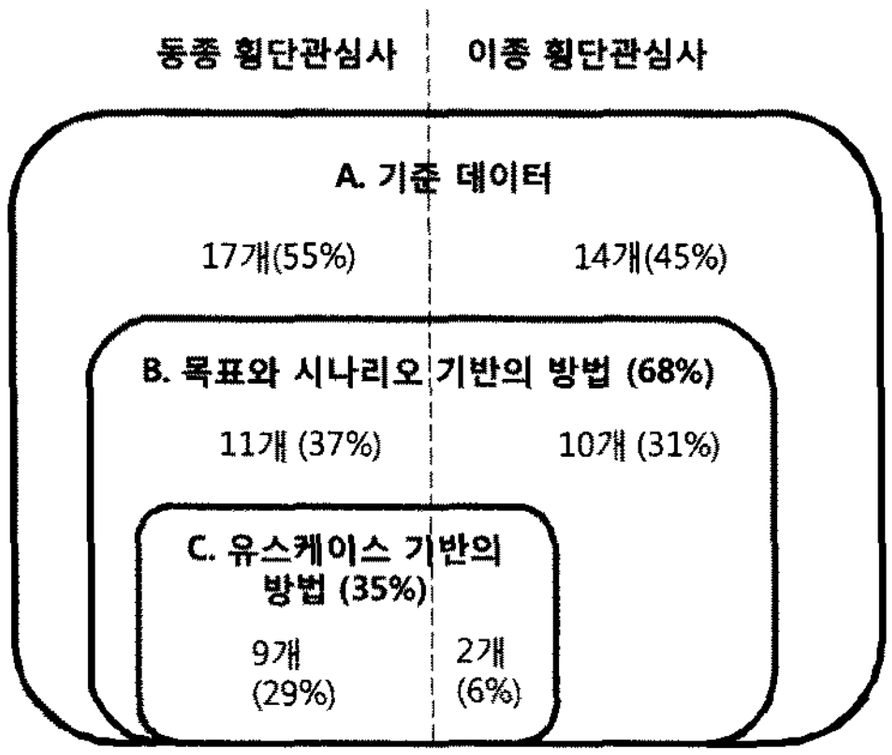


그림 7 목표와 시나리오 기반의 방법과 유스케이스 기반의 방법 비교

이와같이 본 논문에서 제안한 방법이 횡단관심사를 식별하는 효과적인 가이드라인을 제공하는 것을 알 수 있다. 이 방법은 특히, 실시간 임베디드 시스템 등 비기능적인 요구사항이 많이 발생하는 도메인에서 개발 초기단계에 횡단관심사를 분리하는 데 사용하면 유용할 것이다.

6. 결 론

횡단관심사를 소프트웨어 개발 초기단계에 식별하고 분석하는 것은 요구사항의 일관성 및 추적성을 제공하고, 체계적인 변경관리를 돕는다는 점에서 매우 중요하다. 이러한 횡단 관계를 초기에 식별하기 위해서는 몇가지 고려해야 할 점이 있다. 먼저 기능적 관심사 뿐 아니라 비기능적 관심사를 분리할 수 있어야 하며, 횡단관심사가 시스템을 횡단하는 시점 및 다른 관심사들과의 영향관계를 식별할 수 있어야 한다. 이를 위해 본 논문에서는 목표와 시나리오 기반의 요구사항 분석 방법을 기반으로 횡단관심사를 식별하는 방법을 제안했다. 제안된 방법에서는 목표 그래프를 통하여 시스템의 관심을 분리하고, 중복되어 나타나는 목표를 횡단관심사로서 식별하였다. 또한 목표에 대응하는 시나리오를 통해 횡단 시점 및 다른 관심사들과의 영향관계를 식별할 수 있었다.

본 논문에서 제시하는 방법은 목표의 행위의 중복을 근거로 횡단관심사를 식별한다. 그러나 목표와 시나리오가 자연어로 기술되기 때문에 같은 의미라도 다른 표현으로 기술될 수 있어, 사람이 직접 중복을 식별해야 한다는 문제점을 가지고 있다. 따라서 향후에는 온톨로지(ontology)를 기반으로 자동으로 행위의 중복을 찾아 횡단관심사를 식별하는 툴을 지원하고자 한다. 또한 개발 초기단계에 식별된 횡단관심사를 관점으로 구현하는 것을 지원 하기 위해, 본 논문에서 제안하는 방법을 기반으로한 관점지향 소프트웨어 개발방법론을 제안하고 이를 지원하는 도구를 개발할 예정이다.

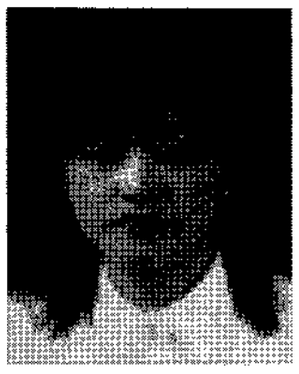
참 고 문 헌

[1] J. Whittle, J. Araujo, "Scenario Modeling with Aspects," IEEE Software, Vol. 151, Issue 4, pp. 157-171. 2004.

[2] A. Rashid, "Website: Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design," URL: Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design, 2005.

[3] Y. Yu, J. C. S. P. Leite, J. Mylopoulos, "From Goals to Aspects: Discovering Aspects from Requirements Goal Models," Proceedings of Requirements Engineering Conference, Kyoto, Japan, pp. 38-47, 2004.

[4] I. Jacobson, "Use Cases and Aspects-Working Seamlessly Together," Journal of Object Technology, Vol. 2, pp. 7-28, 2003.
[5] M. Kim, S. Park, V. Sugumaran, H. Yang, "Managing Requirements Conflicts in Software Product Lines: A Goal and Scenario Based Approach", Data and Knowledge Engineering Journal, Vol. 61, Issue 3, pp. 417-432, 2007.
[6] K. Lee, K.C. Kang, M. Kim, S. Park, "Combining feature-oriented analysis and aspect-oriented programming for product line asset development," Proceedings of the 10th International Software Product Line Conference (SPLC '06), Baltimore, MD, pp. 103-112, 2006.
[7] R. France, D. Kim, S. Ghosh, E. Song, "A UML-Based Pattern Specification Technique," IEEE Transactions on Software Engineering, Vol. 30, pp. 193-206, 2004.
[8] M. Kim, S. Kim, S. Park, M. Choi, M. Kim, H. Gomaa, "UML-Based Service Robot Software Development: A Case Study," International Conference on Software Engineering, pp. 534-543. 2006.

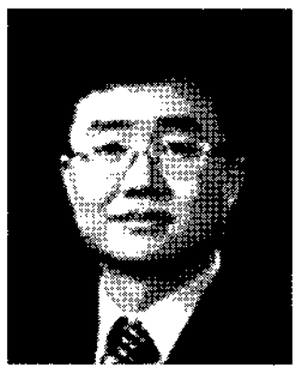


김 선 화

2006년 중앙대학교 컴퓨터학과(공학사) 2006년~현재 서강대학교 컴퓨터학과 석사과정. 관심분야는 요구공학, 소프트웨어 프로세스, 소프트웨어 아키텍쳐, 관점지향 소프트웨어

김 민 성

2001년 서강대학교 컴퓨터학과(공학사) Summa Cum Laude. 2003년 서강대학교 컴퓨터학과(공학석사). 2006년~2007년 독일 Fraunhofer Institute for Experimental Software Engineering(IESE), 방문 연구원. 2003년~현재 서강대학교 컴퓨터학과 박사과정. 관심분야는 요구공학, 소프트웨어 프로덕트 라인, 적응형 소프트웨어, 로봇 소프트웨어



박 수 용

1986년 서강대학교 전자계산학과 공학사 1988년 플로리다 주립대 전산학 석사 1995년 George Mason University 정보기술학박사, 연구 조교수. 1996년~1998년 TRW Senior Software Engineer 1998년~2002년 서강대학교 컴퓨터학과 조교수. 2002년~현재 서강대학교 컴퓨터학과 부교수. 관심분야는 요구공학, 적응형 소프트웨어, 로봇 소프트웨어