내장형 소프트웨어의 비기능적 요구사항 성능 중심 추적

(Performance-based Tracing Non-Functional Requirements of Embedded Software)

최 정 아 † 정 기 원 ††
(Junga Choi) (Kiwon Chong)

요 약 비기능적 요구사항은 대상 시스템이 지원해야 할 기능적 요구사항의 속성 또는 품질 요구사항을 말하며, 소프트웨어의 품질 속성으로 반영된다. 이러한 비기능적 요구사항은 특히 성능과 관계된 부분의 설계를 결정하기 위한 중요한 기준으로 사용된다. 따라서 비기능적 요구사항은 소프트웨어 개발 생명주기 전반에 걸쳐 계속적으로 고려되고, 관리되어야 한다. 하지만 비기능적 요구사항의 모호성과 비가시적 특성으로 인해 도출 및 명세뿐만 아니라, 관리에도 어려움이 많다. 이에 본 논문에서는 NFR 그래프를 이용해 비기능적 요구사항에 가시성을 부여하고 비기능적 요구사항을 설계에 반영하여 비기능적 요구사항 관리의 효율성을 높이기 위한 비기능적 요구사항의 추적 기법을 제안한다. 비기능적 요구사항 개발 절차에 따라 지능형 조명 제어 시스템의 성능 요구사항 추적 방법을 사례연구로 수행하였으며, 이를 통해 종합적으로 비기능적 요구사항 관리의 효율을 높이고자 한다.

키워드 : 비기능적 요구사항, 요구사항 추적, NFR 그래프, 성능 요구사항

Abstract A non-functional requirement is a property or quality that the proposed systems have to support the functional requirements. A non-functional requirement is reflected by quality attribute. These non-functional requirements play a crucial role during system development, serving as selection criteria for choosing among decisions. It should be continuously considered through the software development process. In spite of the importance of the non-functional requirements, it received little attention because of ambiguousness and invisibility of non-functional requirements. Therefore non-functional model which is a process to analyze the non-functional requirement is proposed for improving the management efficiency of non-functional requirements. Also, this paper presents the trace among the UML diagrams to the conceptual model. According to the non-functional requirement development process, this paper achieved performance-based case study. After then, non-functional requirement should be traced using the UML diagrams.

Key words : Non-functional Requirement, Requirement Tracing, NFR graph, Performance Requirement

1. 서 론

정확한 요구사항의 도출 및 분석은 소프트웨어 개발에 있어 중요한 성공 요인이다[1]. 특히나 내장형 소프트웨어는 실시간 처리로 시간 제약사항에 매우 민감하며, 저전력을 요하는 특성으로 인해 시간효율성(time behavior)과 자원효율성(resource utilization)에 있어서 높은 요구사항을 만족해야 한다. 이와 같이 내장형 소프트웨어의 특성상 기능적 요구사항(functional requirement) 외에 비기능적 요구사항(non-functional requirement)이 보다 강조된다[2]. 따라서 효율적이며 신뢰성 높은 내장형 소프트웨어를 위해서는 반드시 비기능적 요구사항을 정확히 도출하고 관리해야 한다[3]. 하지만 이러한 비기능적 요구사항은 비가시적인(invisible) 특성으로 인하여 쉽게 간과되며 설계에 반영되지 않는다[2]. 따라서 비기능적 요구사항을 도출하여 설계에 반영하며, 또한 이를 관리하기 위해 비기능적 요구사항의 추적에 관한 연구

· 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음
† 정 회 원 : 숭실대학교 컴퓨터학과
todo3@hanmail.net
†† 종신회원 : 숭실대학교 컴퓨터학부 교수
chong@ssu.ac.kr
논문접수 : 2005년 1월 26일
심사완료 : 2006년 5월 9일

가 필요하다. 본 논문에서는 소프트웨어 개발 생명주기(Software Life Cycle) 전체적인 관점에서의 비기능적 요구사항을 바라보며, 이를 설계에 반영하고, 추적하기 위한 연구를 진행하였다.

2. 관련연구

비기능적 요구사항은 설계를 결정하기 위한 기준(criteria)으로써 중요한 역할을 한다. 하지만 비기능적 요구사항에 대한 많은 연구가 진행되지 않았으며, 구체적인 비기능적 요구사항 항목의 정의 자체도 쉽지 않다. 이에 본 논문과의 비교를 위해 비기능적 요구사항과 요구사항 추적에 관한 두 개의 관련연구를 살펴본다.

2.1 비기능적 요구사항 프레임워크

요구사항은 일반적으로 개발될 시스템에 요구되고 시스템이 달성하기 위한 것으로, 목표(goal)를 통해 구현된다[4]. 특히 비기능적 요구사항은 소프트웨어 제품의 품질에 직접적으로 영향을 미치기 때문에 효율적으로 분석하고 적용되어야 한다. 이에 비기능적 요구사항과 관련된 목표를 분석하고, 표현(representation)을 제공한 것이 비기능적 요구사항 프레임워크이다[5]. 비기능적 요구사항을 표현하고 사용하기 위한 포괄적인(comprehensive) 프레임워크를 제안했으며 다섯 가지 컴포넌트(Goals, Link types, Methods, Correlation rules, Labelling procedure)를 정의하고 각각을 수식을 사용해 명세했다. 그러나 추가적으로 비기능적 요구사항들 간의 관계나 추적성에 관한 명세가 요구된다.

2.2 UML을 사용한 요구사항 추적 프레임워크

소프트웨어의 요구사항은 비즈니스나 기술적 환경의 진화(evolution)에 의해 계속적으로 변화하며, 한번 변화되면 시스템 전체적으로 모델과 요소들을 찾고, 위험요소를 분석하고, 비용을 예측하고, 각각의 항목들의 절충점을 찾아야 한다. 따라서 이러한 변화와 영향력 추적을 위해 본 프레임워크가 제안되었다. 요구사항 추적 기능을 변경 추적(change tracking)과 영향력 분석(influence analysis)으로 나누어, 모델과 모델 사이 그리고 모델 내부의 요소 사이의 요구사항이 일관성 있게 연결되어 있는지 확인한다. 본 관련연구에서는 모델과 모델 사이의 추적 관계를 표현하기 위한 요구사항 추적 모델을 UML 다이어그램을 사용하여 작성했으며, 모델간의 추적과 요소간의 추적을 확인하고, 영향력 분석에서는 모델의 요소들이 서로 어떤 영향을 끼치는지 확인하였다. 하지만 비기능적 요구사항에 대한 고려가 많지 않으며 요구사항의 가시성에 대해 추가적인 연구가 필요하다.

3. 비기능적 소프트웨어 요구사항 분석

본 논문에서는 두 단계의 비기능적 요구사항 모델 개발 절차를 제안하며, 각 단계의 활동을 그림과 같은 흐름으로 설명한다. 그림 1은 본 논문의 비기능적 소프트웨어 요구사항 분석 방법을 설명한 그림이다.

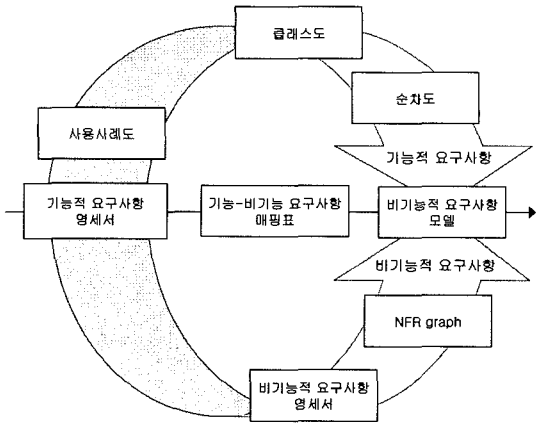

그림 1 비기능적 소프트웨어 요구사항 분석 방법

작성된 기능적 요구사항 명세서를 바탕으로 비기능적 요구사항을 도출하며, 도출된 비기능적 요구사항을 다시 기능적 요구사항의 개념 모델(conceptual model)에 반영하여 기능적 요구사항과 비기능적 요구사항을 통합한 체계적인 요구사항 분석을 수행한다.

3.1 비기능적 소프트웨어 요구사항 분석 원칙

소프트웨어 시스템에서 기능적으로 구현되어야 하는 부분 외에, 신뢰성, 보안성, 정확성, 안정성, 성능 또는 조직적 요구사항과 같은 비기능적 관점들은 비기능적 요구사항으로 처리되어야 한다. 비기능적 요구사항들을 소프트웨어 개발 초기부터 개발 프로세스를 진행해 나가는 동안에 일관적으로 고려하며, 다양한 방법으로 명세하여 모델링에 반영하고, 간과하지 않고 개발에 반영해야 한다[5]. 그러나 비기능적 요구사항은 가시성(visibility)이 없어 처리하기가 어려울 뿐만 아니라, 관리하는데 비용도 많이 든다[6,7]. 따라서 비기능적 요구사항의 분석을 위한 몇 가지 원칙을 정의한다.

① 소프트웨어의 비기능적 요구사항은 기능적 요구사항의 속성 및 품질 요구사항이다[2].
② 소프트웨어의 비기능적 요구사항에 가시성을 부여한다.
③ 소프트웨어의 비기능적 요구사항의 충돌을 처리해야 한다.
④ 소프트웨어의 비기능적 요구사항은 소프트웨어 생명주기 전체적인 흐름으로서의 추적이 필요하다.
⑤ 소프트웨어의 비기능적 요구사항의 분석을 위해 각각의 비기능적 요구사항에 ID를 부여하여 관리한다.

설명한 다섯 가지 기본 원칙들을 바탕으로 본 논문의

비기능적 요구사항을 분석한다. 제시된 다섯 가지 분석 원칙들을 이후의 비기능적 요구사항 분석에 모두 적용한다.

3.2 비기능적 소프트웨어 요구사항 모델

비기능적 소프트웨어 요구사항을 추적하기 위해 우선 비기능적 요구사항 모델을 개발한다. 비기능적 요구사항 모델은 도출된 비기능적 요구사항들을 모델링에 반영한 개념 모델(conceptual model)을 의미하며, 이를 통해 비기능적 요구사항들이 구현에 반영된다. 제안하는 비기능적 요구사항 모델 개발의 단계는 비기능적 요구사항 명세와 비기능적 요구사항 모델링의 두 단계로 이루어진다. 비기능적 요구사항 모델 개발 절차는 그림 2와 같다.

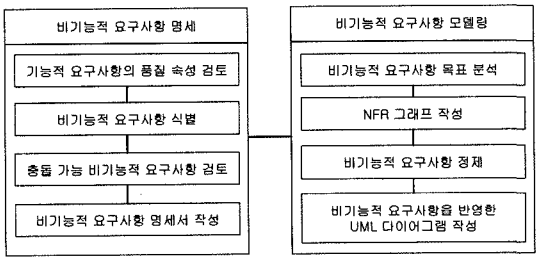

그림 2 비기능적 요구사항 모델 개발 절차

먼저 비기능적 요구사항 명세 단계에서는 작성된 기능적 요구사항 명세서를 바탕으로 기능적 요구사항의 품질 속성을 검토하여 비기능적 요구사항을 도출해 낸다. 다음 비기능적 요구사항 모델링 단계에서는 작성된 비기능적 요구사항 명세서를 바탕으로 NFR 그래프를 작성하며, 비기능적 요구사항을 반영한 UML 다이어그램을 그린다. 비기능적 요구사항 모델 개발 단계의 각 활동에서는 하나 이상의 산출물을 생성한다. 산출물들은 대부분 본 논문에서 제시하는 비기능적 요구사항 모델 개발을 위한 분석을 수행하며 만들어지는 표와 UML 다이어그램이다. 또한 각각의 활동들은 비기능적 요구사항 모델 개발을 위해 유기적으로 연관된다. 표 1은 비기능적 요구사항 모델 개발을 위한 산출물 목록을 나타낸 것이다.

각각의 활동별 산출물들에는 비기능적 요구사항 ID를 빠지지 않게 기록하며, 최종적으로 비기능적 요구사항을 반영한 UML 다이어그램인 비기능적 요구사항 모델의 작성을 위해 사용한다. 각 단계의 구체적인 설명은 다음에서 설명한다.

3.3 비기능적 요구사항 명세

본 단계에서는 비기능적 요구사항 명세를 위한 네 가지 활동을 수행한다. 비기능적 요구사항 명세 단계의 최종 산출물로는 비기능적 요구사항 명세서가 작성된다. 각각의 세부 활동 별로 수행하는 작업을 설명한다.

3.3.1 기능적 요구사항의 품질 속성 식별

작성된 기능적 요구사항들의 품질 속성(quality attribute)을 식별한다. 품질 속성은 실제 시스템의 성공 또는 실패 여부를 결정하는데 매우 중요한 역할을 하며 본 논문에서는 ISO/IEC 9126[8]의 소프트웨어 품질 속성을 기본으로 식별한다. 기능적 요구사항 명세서는 비기능적 요구사항 명세 단계 시작 전에 작성되어 있는 것을 전제로 한다. 기능적 요구사항 명세서의 내용을 바탕으로 각각의 기능적 요구사항들이 어떤 품질 속성에 연관되어 있는지를 식별하여 기능적 요구사항 품질 속성 검토표를 작성한다. 이때 개발자(developer) 관점과 사용자(user) 관점에서의 품질 속성을 고려하여 검토표를 작성한다.

3.3.2 비기능적 요구사항 식별

검토된 기능적 요구사항의 품질 속성을 기반으로 비기능적 요구사항을 식별한다. 비기능적 요구사항 식별을 위해 각각의 요구사항에 ID를 부여하는 것을 원칙으로 한다. 기능적 요구사항으로부터 도출된 비기능적 요구사항의 종속성을 표시하기 위해 기능적 요구사항 ID와 함께 비기능적 요구사항 ID를 정의하며, 각각의 비기능적 요구사항은 고유번호를 갖는다. 비기능적 요구사항의 ID를 만드는 규칙은 표 2와 같다.

표 1 비기능적 요구사항 모델 개발 산출물 목록

단계	활동	산출물
비기능적 요구사항 명세	기능적 요구사항의 품질 속성 식별	기능적 요구사항 품질 속성 검토표
	비기능적 요구사항 식별	기능-비기능적 요구사항 매핑표
	충돌 가능 비기능적 요구사항 검토	공통 비기능적 요구사항 정의서
		충돌 비기능적 요구사항 정의서
	비기능적 요구사항 명세서 작성	비기능적 요구사항 명세서
비기능적 요구사항 모델링	비기능적 요구사항 목표 분석	비기능적 요구사항 목표 분석서
	NFR 그래프 작성	NFR 그래프
	비기능적 요구사항 정제	비기능적 요구사항 명세서(정제)
	비기능적 요구사항을 반영한 UML 다이어그램 작성	비기능적 요구사항 모델

표 2 비기능적 요구사항 ID 작성 규칙

(영문자는 모두 대문자)

기능적 요구사항 식별자	구분자	고유 번호	비기능적 요구사항 식별자	구분자	고유 번호
FR (functional requirement)	- (under bar)	XX (고유번호 2자)	NFR (non-functional requirement)	- (under bar)	XX (고유번호 2자)

기능적 요구사항으로부터 비기능적 요구사항들을 도출하기 위해 매핑표를 작성한다. 하나의 사용사례 단위의 기능적 요구사항에서 품질 속성들을 고려한 후 식별된 비기능적 요구사항들을 나열한다. 기능-비기능적 요구사항 매핑표는 표 3과 같다.

표 3 기능-비기능적 요구사항 매핑표

FR ID	FR 이름	NFR ID	NFR 설명	품질 속성
FR_01	...	FR_01_NFR_01	...	...
		FR_01_NFR_02	...	...
...	...	...	...	...

3.3.3 충돌 가능 비기능적 요구사항 검토

소프트웨어 개발은 제약된 환경 하에서 이루어지기 때문에 품질 속성의 상쇄가 불가결하게 발생한다. 이런 경우의 비기능적 요구사항의 품질 속성 상쇄를 비기능적 요구사항의 충돌(conflict)로 정의하고 충돌 가능 비기능적 요구사항을 검토한다. 그림 3은 두 가지 비기능적 요구사항 사이의 충돌 정도를 나타낸 그림이다.

각각의 비기능적 요구사항 'Efficiency'와 'Functionality'에 대한 요구사항 충돌 정도를 보인다. 두 개의 요구사항이 강하게 충돌을 일으키는 경우부터 Strong conflict, weak conflict, very weak conflict, no conflict로 종류를 구분하여 명세 한다. 이러한 비기능적 요구사항의 충돌 정도 또한 충돌 요구사항 정의서에 명세 한다.

3.3.4 비기능적 요구사항 명세서 작성

비기능적 요구사항 명세의 기능적 요구사항의 품질 속성 검토, 비기능적 요구사항 식별, 충돌 가능 비기능적 요구사항 검토의 세 활동을 통해 비기능적 요구사항이 도출되고, 충돌 가능한 비기능적 요구사항들을 검토하여 비기능적 요구사항 명세서를 작성한다. 비기능적 요구사항은 ID 순서에 따라 명세 한다. 비기능적 요구사항 명세서에는 NFR ID, NFR 이름, 설명, 우선순위, 속성, 충돌 비기능적 요구사항, 공통 비기능적 요구사항, 추적성 수준 등을 기입한다.

3.4 비기능적 요구사항 모델링

비기능적 요구사항 모델링 단계는 비기능적 요구사항 명세 단계를 통해 작성된 비기능적 요구사항 명세서를 기반으로 비기능적 요구사항 목표를 분석하고 NFR 그래프를 작성하여 비기능적 요구사항을 반영한 UML 다이어그램을 작성하는 단계이다. 비기능적 요구사항 모델링 단계에서 비기능적 요구사항 명세서가 정제되며, 본 단계의 최종 산출물로 비기능적 요구사항 모델이 작성된다. 다음에 각 활동 별로 하나씩 설명한다.

3.4.1 비기능적 요구사항 목표 분석

본 활동은 비기능적 요구사항 명세 단계에서 작성된 비기능적 요구사항 명세서를 기반으로 각각의 비기능적 요구사항의 목표(goal)를 분석하는 활동이다. 비기능적 요구사항을 목표의 범주로 구분하고, 하위 목표(sub goal)로 분해하여 하위 목표들의 관계를 분석한다. 기본적으로 NFR 프레임워크[5]에서 제안하는 목표 분석의 지침을 따른다.

3.4.2 NFR 그래프 작성

비기능적 요구사항의 목표 분석이 이루어지면, NFR

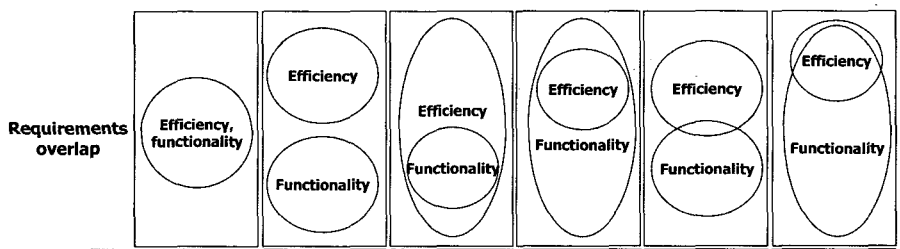

그림 3 비기능적 요구사항의 충돌 정도

표 4 NFR 그래프 ID 작성 규칙

FR ID	FR Name	NFR ID	NFR Description	Quality Attribute	NFRG ID
FR_01	...	FR_01_NFR_01	...	Security	FR_01_NFR_01_NFRG_01
				Accuracy	FR_01_NFR_01_NFRG_02
				Learnability	FR_01_NFR_01_NFRG_03
		FR_01_NFR_02	...	...	...
					...
...	...	...	...	...	...

그래프를 작성한다. NFR 그래프는 Chung의 NFR 프레임워크[5]에서 비기능적 요구사항을 표현하기 위해 제안된 그래프로, 가시성이 없는 비기능적 요구사항에 가시성을 부여한다. 본 논문에서는 NFR 프레임워크의 NFR 그래프 표기법을 따르되 수식을 사용하지 않고 작성하며, 약간 조정된 표기법(notation)으로 표현한다. NFR 그래프는 비기능적 요구사항과 마찬가지로 고유 번호로 구분하여 관리하기 위해 NFR 그래프 ID를 부여한다. NFR 그래프 ID는 비기능적 요구사항 ID에 NFR 그래프의 고유번호를 덧붙여 작성한다. 표 4에서 NFR 그래프 ID 작성 규칙을 설명한다.

도출된 비기능적 요구사항 각각의 품질 속성 별로 NFR 그래프를 작성하며 이에 따라 ID를 부여한다.

3.4.3 비기능적 요구사항 정제

비기능적 요구사항 명세서 작성 후, 비기능적 요구사항 목표 분석을 통해 NFR 그래프를 작성하며 비기능적 요구사항 명세서를 정제한다. 비기능적 요구사항 정제 활동을 통해 비기능적 요구사항 모델을 위한 비기능적 요구사항 명세서가 완성된다.

3.4.4 비기능적 요구사항을 반영한 UML 다이어그램 작성

비기능적 요구사항을 반영한 UML 다이어그램 작성 활동은 비기능적 요구사항 모델링의 마지막 활동이다. 작성된 NFR 그래프의 목표들을 UML 다이어그램에 표현한다. 본 논문의 비기능적 요구사항은 기능적 요구사항으로부터 도출하기 때문에 비기능적 요구사항은 모두 기능적 요구사항에 여러 가지 방법과 형태로 속하게 된다. 따라서 이러한 비기능적 요구사항을 기능적 요구사항 위주의 개념 모델(conceptual model)에 표현함으로써 비기능적 요구사항의 가시성을 보장하고, 기능적 요구사항과 비기능적 요구사항의 통합으로 인해 보다 완성된 요구사항을 설계에 반영하고자 한다. 이 활동을 통해 비기능적 요구사항 모델이 완성되며, 이를 통해 비기능적 요구사항을 추적한다.

4. 사례연구

본 장에서는 사례 연구로써 내장형 소프트웨어의 성능 요구사항을 추적한다. 본 논문에서 정의하는 추적은 비기능적 요구사항 도출에서부터 분석 과정에 걸쳐 어떻게 각각의 요구사항이 산출물에 반영되는가, 또한 비기능적 요구사항이 명세 된 내용대로 소프트웨어 생명주기 내에 반영되는가를 추적하는 것을 의미한다. 내장형 소프트웨어는 일반 소프트웨어에 비해 실시간성이 강조되며, 높은 신뢰성을 요구하며, 환경에 강한 내구성을 가지는 특징을 갖는다[9]. 이에 내장형 소프트웨어의 특징을 반영한 추적 원칙을 제시하며, 이에 따라 성능 요구사항을 추적한다.

① 비기능적 요구사항의 추적을 위해서 비기능적 요구사항 모델을 개발하여 추적한다.
② 각각의 비기능적 요구사항 산출물별로 ID를 부여하여 ID에 따라 추적한다.
③ 내장형 소프트웨어의 추적을 위한 추적 요인(factor)을 분석한다.
④ 비기능적 요구사항이 구현되어 가는 과정을 추적하며, 품질 속성 별로 구현되는 기술 범주를 구분한다.
⑤ 성능 중심의 비기능적 요구사항으로 ISO/IEC 9126의 효율성 측면을 본 논문에서 고려하여 이를 추적한다.
⑥ 비기능적 요구사항 ID와 추적 매트릭스(metrics)는 추적을 위해 반드시 필요한 문서는 아니며, 필요시에 작성하여 추적한다.

내장형 소프트웨어의 성능 요구사항 추적을 위한 사례연구 시스템은 '지능형 조명 제어 시스템'으로 거주자에게 쾌적하고 생산성 있는 거주 및 업무 환경을 제공해 주기 위해 실내 조도를 원하는 정도로 유지하고 조절하기 위한 시스템이다. 시스템의 사용사례도는 그림 4와 같다.

3장에서 설명한 프로세스에 따라 비기능적 요구사항 명세 단계에서의 추적과, 비기능적 요구사항 모델링 단계에서의 추적을 나누어 설명한다.

4.1 비기능적 요구사항 명세 단계에서의 추적

지능형 조명 제어 시스템의 비기능적 요구사항 모델 작성을 위해 비기능적 요구사항 명세 단계를 수행한다.

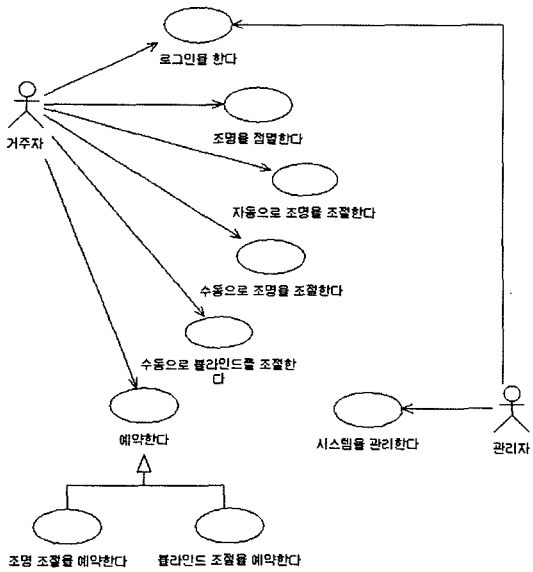

그림 4 지능형 조명 제어 시스템의 사용사례도

비기능적 요구사항 명세 단계에서는 첫 번째로 기능적 요구사항의 품질 속성을 검토하여 기능적 요구사항에서 고려 가능한 품질 속성을 도출한다. 이렇게 품질 속성의 검토를 통해 도출된 기능적 요구사항으로부터 비기능적 요구사항을 식별해 낸다.

기능-비기능적 요구사항 매핑표에서 기능적 요구사항은 사용사례 단위로 나누며, 하나의 기능적 요구사항에 대해 도출 가능한 여러 개의 비기능적 요구사항을 도출하여 이에 알맞은 ISO/IEC 9126[8]의 부특성(sub characteristic)을 정의했다. 표 5에서 FR_03 자동 조명 조절 기능으로부터 FR_03_NFR_01, FR_03_NFR_02, FR_03_ NFR_03, FR_03_NFR_04, FR_03_ NFR_05를 도출했다. 도출된 다섯 개의 비기능적 요구사항은 모두 ISO/IEC 9126의 부특성 시간효율성(time behavior)에 속한다.

다음 활동은 도출된 비기능적 요구사항들 사이에 충돌 가능한 비기능적 요구사항을 검토하는 활동이다. 특정 품질 속성들은 특성들이 결합되면 상쇄 효과가 발생하게 된다[5]. 따라서 사용자와 개발자는 어떤 특성이 보다 중요한지를 판단해야 하며, 의사결정시에 이들 간의 우선순위를 결정해야 한다[10]. 이러한 비기능적 요구사항 분석 과정을 거쳐 비기능적 요구사항 명세서를 작성한다.

4.2 비기능적 요구사항 모델링 단계에서의 추적

본 단계에서는 명세 된 각각의 비기능적 요구사항의 목표를 분석하여 NFR 그래프를 그리고, 이를 UML 다이어그램에 표현한다. 기능-비기능적 요구사항 매핑표에서 각각의 비기능적 요구사항의 품질 속성 별로 NFR 그래프를 그린다. 자동 조명 조절 기능에서 '조명은 5초 간격으로 계속적으로 밝기를 조절한다.'에 대한 시간효율성(time behavior) 비기능적 요구사항을 표현한 NFR 그래프는 그림 5와 같다.

아래의 NFR 그래프에서 Goal sort는 Efficiency이고, Sub sort는 Time behavior이다. 조명의 밝기를 조정하기 위해, 관련된 속성을 도출하고, ChangeIntensity() 오퍼레이션이 5초 간격으로 수행되어야 한다는 제약사항(constraint)을 표현하였다.

그림 6은 지능형 조명 제어 시스템의 조명 클래스를 작성한 것이다. NFR 그래프 분석을 통해 도출된 비기능적 요구사항의 하위 목표 요소들을 클래스에 표현한

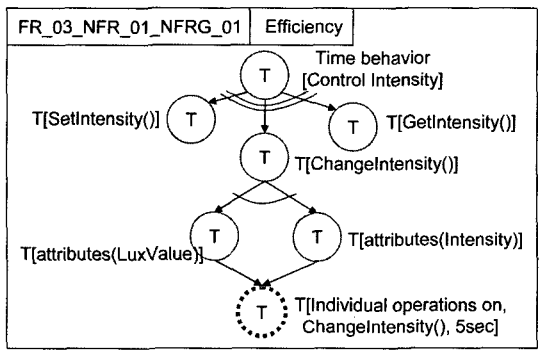

그림 5 NFR 그래프의 예(FR_03_NFR_01_NFRG_01)

표 5 기능-비기능적 요구사항 매핑표의 예

FR ID	FR 이름	NFR ID	NFR 설명	품질 속성
FR_03	자동조명 조절	FR_03_NFR_01	조명은 5초 간격으로 계속적으로 밝기를 조절한다.	Time behavior
		FR_03_NFR_02	빛 센서는 1/10초 간격으로 빛을 읽어 들여 계속적으로 실내 밝기를 감지한다.	Time behavior
		FR_03_NFR_03	센서는 5초 동안 감지된 실내 밝기의 평균값을 계산한다.	Time behavior
		FR_03_NFR_04	조명 제어 시스템은 자동 조절 기능에 의해 1초 이내로 시스템이 반응을 시작해야 한다.	Time behavior
		FR_03_NFR_05	실내의 조도를 감지하는 빛 센서는 일정주기 2~3초 간격으로 실내 조도를 감지하여 반응해야 한다.	Time behavior
...	...	...	...	...

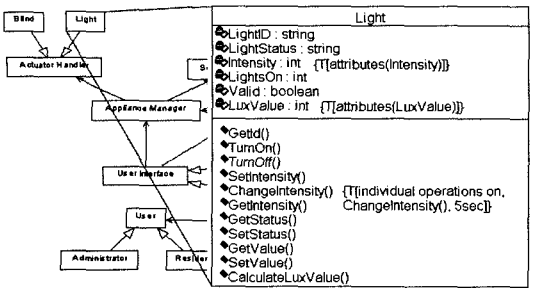

그림 6 비기능적 요구사항을 반영한 조명 클래스

다. ChangeIntensity() 오퍼레이션에 비기능적 요구사항을 제약사항으로 표시함으로써 모델링 단계에서 비기능적 요구사항을 반영한 모델링이 가능하다. 클래스에서는 속성과, 오퍼레이션에 각기 제약사항을 표시한다. 속성과 오퍼레이션으로써 표현되지 않는 비기능적 요구사항은 노트(note)를 사용하도록 한다.

이렇게 다이어그램이 작성되면, 비기능적 요구사항 다이어그램간의 추적이 가능하다. 그림 7은 사용사례도에서부터 기능-비기능적 요구사항 매핑표, NFR 그래프, 클래스도에 이르기까지 비기능적 요구사항 모델 산출물의 추적을 나타낸 것이다.

각각의 비기능적 요구사항은 ID를 따라 도출되어 분석된 후, 비기능적 요구사항이 기능적 요구사항에 어떤 요소로 속하며 구현을 위해 어떻게 처리되는지의 확인이 가능하다. 품질 속성은 기능 이외의 요구사항이기는 하지만, 파생된 기능 요구사항, 설계 가이드라인 또는 바람직한 품질 속성을 만들어 내는 다른 유형의 기술 정보로 연결 될 수 있다. 보통 아키텍처나 설계 제약사항과 같은 기능 범주에 속해 실제 구현에 반영된다.

5. 기존 연구와의 비교 평가

본 논문에서는 기존의 비기능적 요구사항 분석 방법들을 분석하고, 비기능적 요구사항을 관리하기 위해 내장형 소프트웨어의 성능 중심의 비기능적 요구사항을 추적하였다. 기존 관련연구에서 분석한 논문들과 본 논문에서 제안하는 방법을 여러 항목들로 비교한다. 비교 항목은 비기능적 요구사항의 특성을 고려하여 반드시 고려되어야 하는 비기능적 요구사항의 속성들로 도출되었으며, 가시성 확보와 추적성 부분에서 주요 비교 항목들을 선정하였다. 비교 평가 내용은 표 6과 같다.

본 논문의 비기능적 요구사항 분석 방법은 기존 연구들에 비해 비기능적 요구사항의 가시성 확보를 염두에

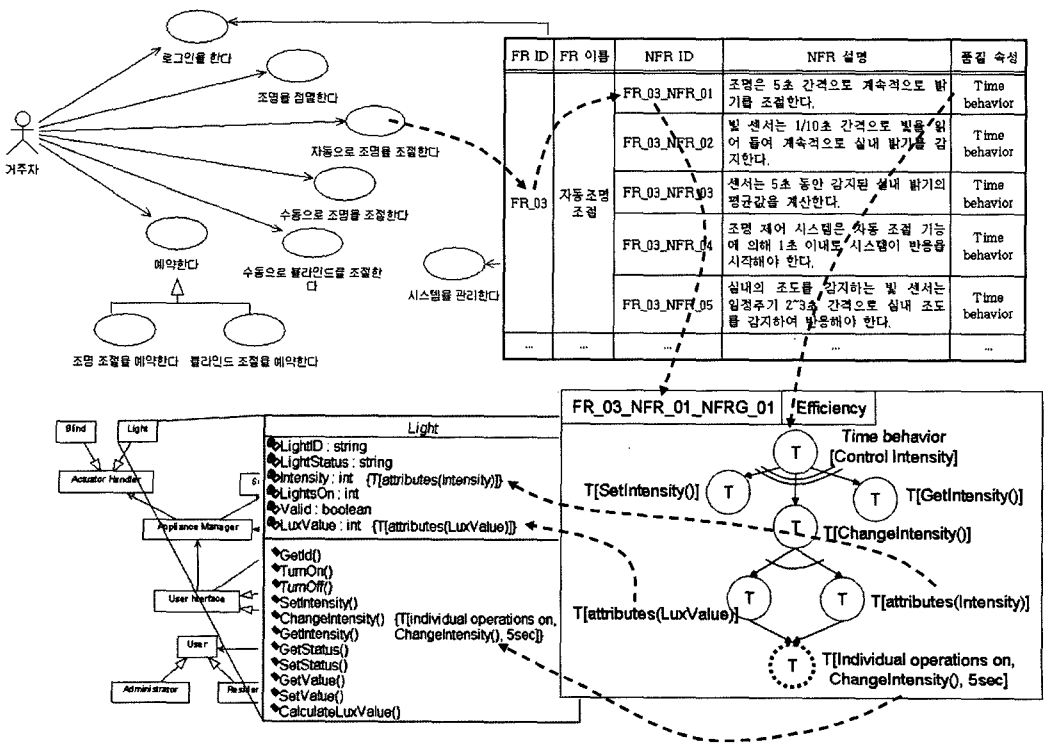

그림 7 비기능적 요구사항을 반영한 산출물간의 추적

표 6 기존 연구와의 비교 평가

분석 방법 / 비교 항목	본 논문의 방법	비기능적 요구사항 프레임워크[5]	요구사항 추적 프레임워크[1]
비기능적 요구사항 도출	△	○	X
Process-oriented 접근	△	△	△
비기능적 요구사항의 가시성 확보	○	○	△
비기능적 요구사항간의 관계 명세	△	X	X
비기능적 요구사항 추적성 확보	○	X	○
UML 사용	○	○	○
NFR 그래프 사용	○	○	X

두었고, 비기능적 요구사항간의 관계 명시나 비기능적 요구사항 추적성 확보 부분을 보완하였다. 비기능적 요구사항 프레임워크에 비해 비기능적 요구사항 도출에 대한 구체적인 가이드라인을 제공하지 않았으나, 소프트웨어 개발 생명주기 전체적인 프로세스 기반으로 비기능적 요구사항을 추적하여 Process-oriented 접근을 이루었다. 하지만 개발 생명주기 전 과정에 걸친 비기능적 요구사항의 검증 및 확인에 대한 부분은 미흡하다.

6. 결론 및 향후 연구

본 논문에서는 내장형 소프트웨어의 비기능적 요구사항을 추적하기 위한 비기능적 요구사항 모델 개발 절차 및 추적 기법을 제안하였다. 비기능적 요구사항 모델 개발 절차를 통해 비기능적 요구사항을 도출하여, 비기능적 요구사항 명세서를 작성하고, 이를 NFR 그래프에 표현하며, UML 다이어그램에 비기능적 요구사항을 표현하여 반영하였다. 또한 사례연구로 내장형 소프트웨어의 성능 요구사항을 반영한 UML 다이어그램간의 추적을 보였다. 이를 통해 내장형 소프트웨어의 성능에 큰 영향을 끼치는 효율성에 대한 비기능적 요구사항을 모델링 단계에서부터 고려함으로서 보다 빠르게 비기능적 요구사항을 분석하고 설계에 반영하였으며, 또한 기능적 요구사항과 비기능적 요구사항을 통합하여 체계적인 요구사항 분석을 수행하며, 비기능적 요구사항을 추적함으로써 요구사항 관리의 측면에서도 효율성을 높였다.

향후연구로는 NFR 그래프에서 각각의 비기능적 요구사항의 품질 속성별로 충돌을 일으킬 수 있는 부분들에 대한 구체적인 명세가 필요하며, 본 논문에서 제시된 다이어그램 외에도 내장형 소프트웨어를 설계하기 위한 실시간적 측면을 반영한 다양한 다이어그램에 정형화된 방법으로 비기능적 요구사항을 표현할 수 있는 명세 측면에서의 연구가 필요하다. 또한 비기능적 요구사항 명세서에서 비기능적 요구사항 추적 요소를 도출하여 소프트웨어가 개발된 이후에 수락 테스트에 반영하여 품질을 검증할 수 있는 점검 항목(checklist)으로 활용하기 위한 연구도 필요하다.

참 고 문 헌

[1] Tsumaki, T., and Morisawa, Y., "A Framework of Requirements Tracing using UML," Proceedings of Seventh Asia Pacific Software Engineering Conference, pp. 206-213, December 2000.

[2] Cysneiros, L. M., do Prado Leite, J. C. S., and Sabat Neto, J. D. M., "A framework for Integrating Non-Functional Requirements into Conceptual Models," Requirements Engineering, pp. 97-115, 2001.

[3] Alan, C., Microprocessor Systems Design 68000 Hardware, Software and Interfacing, 3rd Edition, PWS Publisher, 1998.

[4] Pasternak, T., "Using trade-off analysis to uncover links between functional and non-functional requirements in use-case analysis," Proceedings of IEEE International Conference on Software.

[5] Mylopoulos, J., Chung, L., Yu, E., and Nixon, B., "Representing and Using Non-Functional Requirements: A Process-Oriented Approach," IEEE Transactions on Software Engineering, Vol. 18, No. 6, pp. 483-497, June 1992.

[6] Cysneiros, L. M., and do Prado Leite, J. C. S., "Integrating Non-Functional Requirements into Data Model," Proceeding of Fourth International Symposium on Requirements Engineering, June 1999.

[7] Davis, A., Software Requirements: Objects Functions and States, Prentice Hall, 1993.

[8] ISO, ISO/IEC 9126: Information Technology-Software Quality Characteristics and Metrics, 1998.

[9] 정기석, 김태환, "내장형 시스템 설계: 개론", 정보과학회지 제20권 제7호 통권 제158호, pp. 5-13, 2002.

[10] Karl, E. W., Software Requirements, 2nd Edition, Microsoft Press, 2003.

최 정 아

2005년 숭실대학교 대학원 컴퓨터학과 졸업(공학석사). 현재는 NHN㈜ QA&프로세스개발팀. 관심분야는 비기능적 요구사항, 소프트웨어 품질, 소프트웨어 재사용

정 기 원

정보과학회논문지 : 소프트웨어 및 응용
제 33 권 제 4 호 참조