

내장형 소프트웨어의 비기능적 요구사항 추적

최정아⁰, 이현정, 최윤석, 정기원
 숭실대학교 대학원 컴퓨터학과

todo3@lycos.co.kr⁰, kiwili@empal.com, secooling@hanafos.com, chong@comp.ssu.ac.kr

A Tracing Technique for Non Functional Requirements of Embedded Softwares

Junga Choi⁰, Hyunjeong Lee, Yunseok Choi, Kiwon Chong
 Dept. of Computing, Graduate School, Soongsil Univ.

요 약

비기능적 요구사항은 대상 시스템이 지원해야 할 기능적 요구사항의 속성 또는 품질로써 소프트웨어 품질 속성으로 반영된다. 또한 비기능적 요구사항은 설계를 결정하기 위한 중요한 기준으로 사용되므로 소프트웨어 개발 프로세스 전반에 걸쳐 꾸준히 고려하고, 관리해야 한다. 하지만 비기능적 요구사항은 가시성이 없고, 명확히 명세하기가 어려운 특성으로 인해 간과하기 쉽다. 따라서 비기능적 요구사항을 실제 모델링에 반영하고, 이를 추적하여 계속적으로 관리하기 위한 노력이 필요하다. 이에 본 논문에서는 내장형 소프트웨어의 비기능적 요구사항을 모델링 하는 절차를 제안하며, 내장형 소프트웨어의 특성을 가진 지능형 조명 제어 시스템의 사례를 통해 비기능적 요구사항을 반영한 모델링을 수행하고, 다이어그램들 사이의 비기능적 요구사항에 대한 추적 과정을 살펴본다.

1. 서 론

비기능적 요구사항(non functional requirement : NFR)은 대상 시스템이 지원해야 할 기능적 요구사항(functional requirement : FR)의 속성 또는 품질이다[1]. 이러한 NFR은 품질 속성으로 반영된다[1]. 따라서 소프트웨어 개발 프로세스 전반에 걸쳐 NFR을 꾸준히 고려하고, 관리하는 노력이 필요하다[2, 3]. 특히 높은 신뢰성, 효율성, 성능 등을 요구하는 내장형 소프트웨어의 경우에는 NFR의 중요성이 더욱 부각된다. 하지만 NFR은 가시성(visibility)이 없고, 명확하게 명세하기 어려운 특성상 그 중요성에도 불구하고 관리에 많은 비용이 소모되므로 쉽게 간과된다[3]. 따라서 NFR을 소프트웨어 개발 프로세스를 따라 계속적으로 고려하고 관리하기 위해 NFR의 추적에 관한 연구가 필요하다. 이에 본 논문은 내장형 소프트웨어의 NFR 중에서 성능 요구사항을 지능형 조명 제어 시스템의 사례를 통하여 도출하고, NFR을 반영한 모델링을 수행하고, 다이어그램 사이의 비기능적 요구사항 추적 과정을 보인다.

2. 관련연구

2.1 비기능적 요구사항 프레임워크[4]

NFR은 시스템 개발에 있어서 설계를 결정하기 위한 기준(criteria)으로서 중요한 역할을 한다[4]. 하지만 NFR의 모호한 특성상 명세의 어려움뿐만 아니라, 구체적인 NFR 항목의 정의 자체도 어려운 실정이다[4]. 이에 NFR을 표현하고 사용하기 위한 포괄적인 프레임워크를 제안했다. 다섯 가지 컴포넌트(Goal, Link types, Methods, Correlation rules, Labeling procedure)를 정의하고, 각각을 수식을 사용해 명세했으며, NFR graph[4]를 작성해 가시성이 없는 NFR에 가시성을 부여했다. 다음의 그림 1은 속

성 employee의 정확성(accuracy)을 표현하기 위한 NFR graph를 나타낸 것이다.

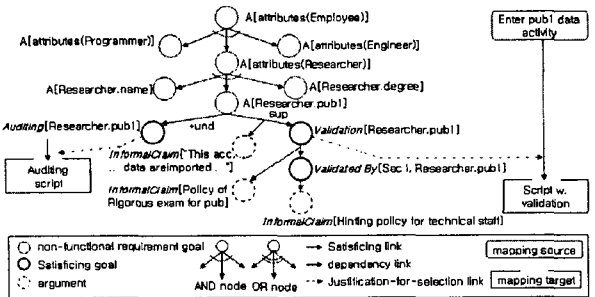


그림 1 속성 employee의 정확성을 표현한 NFR graph[4]

속성 employee의 정확성 NFR을 위한 목표(goal)를 분석 및 분해(decomposition)해, 각 목표들 간의 관계를 표현했다. 이러한 NFR graph를 통해 NFR을 분석할 수 있다.

2.2 UML을 사용한 요구사항 추적 프레임워크[5]

요구사항 추적 기능은 크게 변경 추적(change tracking)과 영향력 분석(influence analysis)으로 나눌 수 있다. 모델과 모델 사이의 그리고 모델 내부의 요소 사이의 요구사항이 일관성 있게 연결되어 있는지를 확인하기 위한 요구사항 추적 모델을 UML 다이어그램을 사용하여 작성했다. 변경 추적에서는 모델간의 추적과 요소간의 추적을 확인하고, 영향력 분석에서는 모델의 요소들이 서로 어떠한 영향을 끼치는지를 확인했다. UML의 유스케이스, 클래스, 시퀀스 다이어그램을 통해서 업무모델, 분석모델, 설계모델 간의 요소들의 추적을 보였다.

3. 내장형 소프트웨어의 비기능적 요구사항 추적

3장에서는 내장형 소프트웨어의 NFR을 추적하기 위한 기본이 되는 NFR 모델링 절차를 설명하고, 실제 제안하는 절차에 따라 지능형 조명 제어 시스템에서 특히 중요시되는 성능 요구사항의 추적 과정을 살펴본다.

3.1 비기능적 요구사항 모델링 절차

NFR 추적을 위해 우선적으로 NFR을 도출하여 NFR을 반영한 다이어그램을 작성하는 비기능적 요구사항 모델링 절차가 필요하다. 비기능적 요구사항 모델링 절차는 크게 비기능적 요구사항 명세와 비기능적 요구사항 모델링의 두 단계로 구성된다. 다음 그림 2는 총 2개의 단계와 6개의 활동으로 구성된 비기능적 요구사항 모델링 절차를 나타낸 것이다.

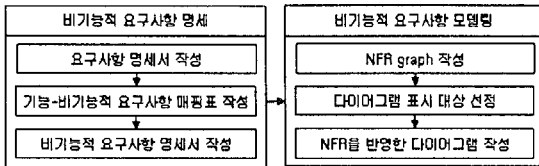


그림 2 비기능적 요구사항 모델링 절차

비기능적 요구사항 명세 단계에서는 작성된 기능 위주의 요구사항 명세서를 바탕으로 기능적 요구사항 명세로부터 NFR을 도출하기 위한 기능-비기능적 요구사항 매핑표를 작성하여, 이때 도출된 NFR을 정제하여 비기능적 요구사항 명세서를 작성한다. NFR은 기능적 요구사항의 품질 속성으로 구현되므로, FR로부터 도출한다. 비기능적 요구사항 명세서에는 NFR ID, NFR 이름, NFR 설명, 우선순위, 속성, 추적성 수준 등을 명세한다.

비기능적 요구사항 모델링 단계에서는 명세된 비기능적 요구사항 명세서를 기반으로 NFR graph를 작성하고, NFR graph를 통해 분석된 내용을 다이어그램에 표현하기 위한 표시 대상을 선정한다. NFR graph는 기능-비기능적 요구사항 매핑표에서 정의된 하나의 품질 속성에 해당하는 NFR 별로 그린다. NFR graph는 최상위의 NFR goal로부터 NFR을 분해하여 표현한 그래프이다. 우선 NFR goal이 만족하고자 하는 내용에 해당하는 NFR goal topic을 선정하고, 각 topic에 맞게 NFR goal을 분해하며, NFR 제약사항을 정의한다. NFR graph의 표기법은 비기능적 요구사항 프레임워크[4]의 표기법을 기본으로 하여 본 논문에서 조정된 표기법을 사용한다. NFR graph를 작성하고, 다이어그램에 표시 대상을 선정하면, 실제 NFR을 반영한 다이어그램을 작성한다. NFR graph에서 분석된 NFR 제약사항을 다이어그램의 속성이나 오퍼레이션의 제약사항으로 표시한다. 지금까지 설명한 두 단계를 거쳐 NFR이 어떻게 도출되며, NFR이 모델링에 어떻게 반영되는지를 알 수 있으며, 이로써 NFR의 추적이 가능하다.

3.2 내장형 소프트웨어의 성능 요구사항 추적

내장형 소프트웨어는 실시간 처리로 시간 제약사항에 매우 민감하며, 저전력을 요하는 특성으로 인해 시간효율성(time behavior)과 자원효율성(resource utilization)에 있어서의 NFR을 만족해야 한다. 따라서 내장형 소프트웨어의 주요 NFR로 ISO/IEC 9126 [6]의 소프트웨어 품질 특성 중 효율성(eficiency)을 고려하기 위한 성능 요구사항을 본 논문에서 제안하는 절차에 따라 추적한

다. 사례연구로는 거주자에게 쾌적하고 생산성 있는 거주 및 업무 환경을 제공해 주기 위해 실내 조도를 원하는 정도로 일정하게 유지하는 지능형 조명 제어 시스템의 NFR을 도출하여 추적한다. 본 시스템의 유스케이스 다이어그램은 다음의 그림 3과 같다.

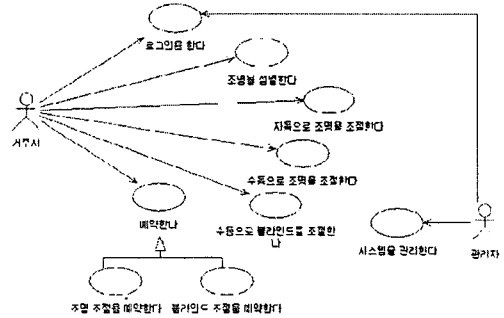


그림 3 지능형 조명 제어 시스템의 유스케이스 다이어그램

요구사항 명세서를 작성하는 활동에서 유스케이스 다이어그램이 완성되며, 이 다이어그램으로부터 NFR을 추적한다.

3.2.1 내장형 소프트웨어의 비기능적 요구사항 명세 단계

작성된 FR 위주의 요구사항 명세서를 바탕으로 NFR을 도출하기 위해 기능-비기능적 요구사항 매핑표를 작성한다. 작성된 매핑표는 그림 4와 같다.

FR ID	FR 이름	NFR ID	NFR 설명	품질 속성	NFR graph ID
FR 1	자동조명 조절	NFR 1.1	조명은 5초 간격으로 밝기를 조절한다.	Time behavior	NFRG 1.1-1
		NFR 1.2	밝기 센서는 1/10초 간격으로 빛을 읽어 들여 계속적으로 실내 밝기를 감지한다.	Time behavior	NFRG 1.2-1
		NFR 1.3	센서는 5초 동안 감지된 실내 밝기의 평균값을 계산한다.	Time behavior	NFRG 1.3-1

그림 4 기능-비기능적 요구사항 매핑표

기능-비기능적 요구사항 매핑표에서 FR은 유스케이스 단위로 나누며, 하나의 FR에 대해 여러 개의 NFR을 도출하여 이에 앞붙은 ISO/IEC 9126의 부특성(subcharacteristic)들을 정의하고, NFR의 부특성 별로 NFR graphID를 부여한다. FR 1 자동 조명 조절 기능으로부터 NFR 1.1, 1.2, 1.3을 도출하였다. 이는 모두 성능 요구사항을 반영하는 ISO/IEC 9126의 주특성 효율성에 해당하는 NFR이다. 이렇게 도출된 NFR들은 더 자세하게 비기능적 요구사항 명세서로 작성된다. 비기능적 요구사항 명세서에는 추적 가능성 여부를 표현한 추적성 수준에 대한 명세도 포함한다.

3.2.2 내장형 소프트웨어의 비기능적 요구사항 모델링 단계

기능-비기능적 요구사항 매핑표에서 각각의 NFR ID별로 NFR graph를 그린다. FR 1 자동 조명 조절 기능에서 '조명은 5초 간격으로 밝기를 조절한다.'라는 NFR 1.1에 대한 시간효율성에 관련된 NFR을 표현한 NFR graph는 그림 5와 같다.

그림 5의 NFR graph에서 goal sort는 Efficiency이고, sub sort는 Time behavior이다. 자동으로 조명의 밝기를 조절하기 위한 관련 속성을 도출하고, ChangeIntensity() 오퍼레이션이 5초 간

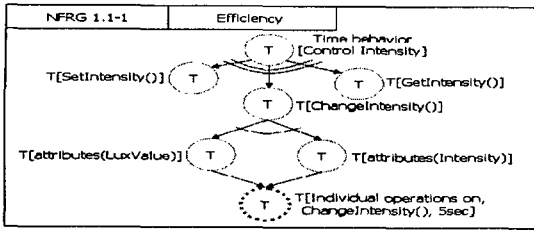


그림 5 NFR graph (NFRG 1.1-1)

적으로 수행되어야 한다는 제약사항을 표현하였다.

본 논문에서 조정한 NFR graph의 작성 가이드는 다음과 같다.

- NFR graph의 goal sort는 ISO/IEC 9126의 소프트웨어 품질 주특성을 기본으로 구분한다.
- NFR graph의 sub sort는 ISO/IEC 9126의 소프트웨어 품질 부특성을 기본으로 구분한다.
- NFR graph 원 안의 글자는 sub sort의 앞글자로 표현한다.
- 분해되는 그래프 노드들 간의 관계를 AND 또는 OR로 표현한다.
- NFR goal은 NFR을 충족시키기 위한 목표의 성격에 따라 NFR goal, *satisficing goal*, *argument goal*로 구분한다.
- sub sort의 파라미터는 NFR과 관련되는 요소를 우선적으로 선정하여 나타내고, 필요하면 자연어로 설명한다.

NFR graph가 작성되고, 다이어그램에 표시할 대상이 선정되면, 이를 기반으로 NFR을 실제 모델링에 반영하는 활동을 수행한다. 그림 6은 지능형 조명 제어 시스템의 조명 클래스를 작성한 것이다. NFR graph 분석을 통해 도출된 제약사항들 중에서 다이어그램에 표시할 대상을 선정하여 표시한다.

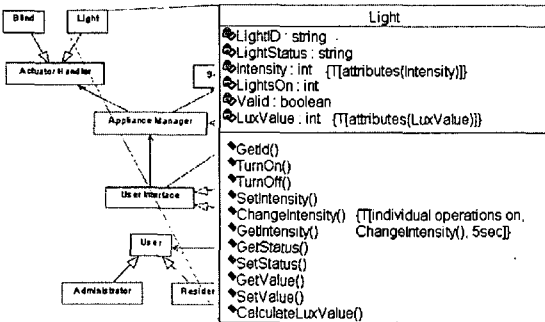


그림 6 NFR을 반영한 조명 클래스

ChangeIntensity() 오퍼레이션에 NFR graph에서 분석한 NFR 제약사항을 표시함으로써 모델링 단계에서 NFR을 반영한 모델링이 가능하다.

NFR을 반영한 다이어그램이 작성되면, 다이어그램간의 NFR의 추적이 가능하다. 다음의 그림 7은 유스케이스 다이어그램에서부터 기능-비기능 요구사항 매핑표, NFR graph, 클래스 다이어그램에 이르기까지 NFR의 추적을 나타낸 것이다.

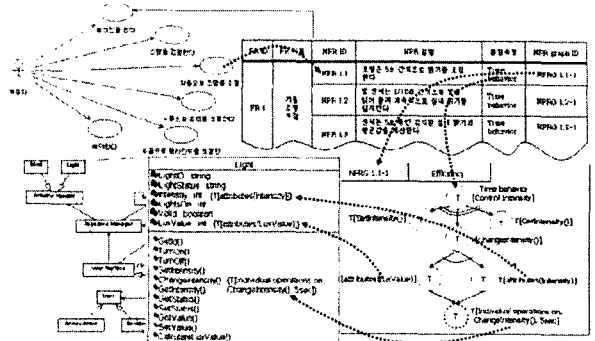


그림 7 NFR을 반영한 다이어그램간의 추적

NFR을 반영한 다이어그램을 작성하면, 이들 간에 NFR이 어떠한 흐름을 갖고, 어떠한 형태로 구현되어 가는지의 추적을 살펴볼 수 있다. 비기능적 요구사항 명세서에 표현된 NFR은 모두 모델링에 반영되어야 하며, 특정 NFR의 제약사항이 다이어그램에 표시 대상으로 선정하기 어려운 경우에는 모델링 시에 노트를 사용하여 명시한다.

4. 결론 및 향후연구

본 논문에서는 내장형 소프트웨어의 NFR을 추적하기 위한 기반으로 비기능적 요구사항 모델링 절차를 제안한다. FR을 바탕으로 NFR을 도출하여, 비기능적 요구사항 명세서를 작성하고, 이를 NFR graph에 표현하며, UML 다이어그램에 NFR을 반영하는 과정을 진행한다. 지능형 조명 제어 시스템의 사례로 NFR을 반영한 다이어그램을 작성하고, 다이어그램 사이의 추적을 보였다. 이와 같이 내장형 소프트웨어의 성능에 큰 영향을 끼치는 효율성 품질 속성에 대한 NFR을 모델링 단계에서부터 고려함으로써 보다 효율적으로 NFR들을 분석하고, NFR의 비가시성으로 인해 쉽게 간과됨을 피할 수 있다. 또한 NFR을 추적함으로써 요구사항 관리의 측면에서도 효율적이다.

향후연구로는 NFR graph에서 각각의 NFR의 품질 속성별로 충돌을 일으킬 수 있는 부분들에 대한 연구가 필요하며, 품질과 관련된 NFR 이외의 비용, 조직, 문화적 비기능성들의 확장이 필요하다.

5. 참고문헌

- [1] David C. Hay, Requirements Analysis : From Business Views to Architecture, Pearson Education, 2003.
- [2] L. Chung, B. Nixon, "Dealing with Non functional Requirements : Three Experimental Studies of a Process Oriented Approach," Proc. 17th Int'l Conference Software Engineering, pp. 24-28, April 1995.
- [3] L. Chung, B. Nixon, E. Yu, and J. Mylopoulos, "Non-Functional Requirements in Software Engineering," Kluwer Academic, 2000.
- [4] J. Mylopoulos, L. Chung, E. Yu, and B. Nixon, "Representing and Using Non-Functional Requirements : A Process-Oriented Approach," IEEE Transactions on Software Engineering, vol. 18, no. 6, pp. 483-497, June 1992.
- [5] Toshihiko, Tsumaki, Yoshitomi Morisawa, "A Framework of Requirements Tracing using UML," Software Engineering Conference, APSEC 2000. Proceedings. Seventh Asia Pacific, pp. 206-213, 5-8 December 2000.
- [6] ISO, ISO/IEC 9126 : Information Technology-Software Quality Characteristics and Metrics, 1998.