UML 기반의 품질 요구 표현 및 설계 추적성 연구☆

UML-based Quality Requirement Specifications and Design Traceability

서 동 수*
Dongsu Seo

요 약

사용자 요구 분석에 있어 품질 요구에 대한 표현과 검증은 다른 활동에 비해 체계적인 연구가 미진한 분야로 알려져 왔다. 본 논문은 UML을 기반으로 정보시스템을 개발할 경우 적용이 가능한 품질 요소 중 사용성, 성능, 이해성 등에 대한 표현과 구현 추적 방법을 제안한다. 특히 유스케이스와 시퀀스 다이어그램의 품질 표현 확장을 통해 기능 요구의 구현 추적과 더불어 관계되는 품질 요구에 대한 표현과 추적을 할 수 있도록 한다. 이를 통해 개발자는 기능요구와 비기능 요구가 올바로 구현되었음을 사전에 확인할 수 있다.

Abstract

Less efforts has been made in developing specifications and verification techniques for quality requirements. This paper suggests specification and traceability methods supporting quality issues such as usability, performance, and understandability when developing information systems with UML notation. With traceability checking methods that augment quality extension to the use cases and sequence diagrams, developers can confirm the level of implementation for both functional and non-functional requirements beforehand.

☞ keyword : requirement engineering, quality analysis, UML extension

1. 서 론

성공적인 시스템 개발을 위해 보장되어야 하는 활동 중의 하나는 사용자의 요구 사항을 정확히 파악하는 것이다. 정확하지 않은 요구 명세로 인한 영향은 많은 경우 시스템 개발의 비용 초과, 공정 지체, 품질 저하의 원인이 된다. 사용자 요구 분석의 목적은 사용자의 요구가 무엇인지를 정확히 도출하기 위해 요구사항을 수집하며, 분석하고, 기록하는 활동이라 정의된다[1].

일반적으로 사용자의 요구를 정확하게 파악해내는 것은 무척 어려운 일로 알려져 있다. 통계에 의하면 프로젝트 실패에 직접적인 영향을 미치는 요인으로 요구 분석의 오류가 13.1%, 불충분한 사용자의 참여가 12.4%, 그리고 요구사항의 변경이 8.1%로 사용자 요구와 관련된 오류 요인이 전체의 32% 가량을 차지하고 있는 것으로 보고되었다[2].

통계에 나타나듯이 요구 분석 과정에서 사용자의 의사 반영은 분석 작업의 성패를 좌우하는 중요한 요소이다. 그럼에도 불구하고 그들의 요구가 충분히 반영되고 있지 못한 이유 중 하나로 개발자는 기능 요구보다는 상대적으로 품질 요구를 정확히 파악하는데 소홀하다는 점을 들 수 있다. 이 과정에서 개발자가 품질 요구와 관련된 설계의 관련성을 파악하는데 실패한다면 품질 목표 달성에도 실패할 수밖에 없다.

* 정 회 원 : 성신여자대학교 컴퓨터정보학부 교수
dseo@sungshin.ac.kr
[2008/11/17 투고 - 2008/11/18 심사 - 2008/12/11 심사완료]
☆ 본 논문은 2006년도 성신여자대학교 학술연구조성비 지원에 의하여 연구되었음

품질에 관한 표현과 분석 기술이 상대적으로 초보 수준에 머물러 있는 이유는 첫 째, 소프트웨어 대한 품질 측정 대상이 명확하지 않고, 둘 째, 품질에 대한 표현 역시 자연어 수준의 부정확한 표기법에 의존하며, 셋 째, 설사 대상이 명확하다 하더라도 정량적인 측정이 힘들다는 점을 꼽는다[3].

ISO-9126에서는 기능성, 사용성, 유지보수성, 신뢰성, 효율성, 이식성 등의 내/외부 품질 특성을 정의하고 있다. 본 논문에서는 이들 중 기능의 분화가 상대적으로 명확한 사용성, 성능, 이해성 등의 요소에 대한 요구 분석 단계의 표현 방법과 이들이 설계 단계에서 나타날 때 추적을 위한 상호 참조 메커니즘을 제안한다.

개발자는 본 논문에서 제안하는 품질 구현에 대한 추적 메커니즘을 통해 품질과 관련된 기능 요구 사항을 적절한 수준에서 추적함으로서 품질 요구에 관련된 구현 문제를 조기에 파악할 수 있다.

2. 관련 연구

품질 요구에 관한 대표적인 연구로는 텍사스 대학의 정(Chung)과 토론토 대학의 닉슨(Nixon)[4]의 연구를 들 수 있다. 이들은 비기능 요구(non-functional requirement), 혹은 품질 요구를 표현하는 과정에서 품질 목표(goal) 개념을 도입하고, 이를 달성하기 위한 세부 목표와 품질 명세 프레임워크를 제안한다. 이들의 연구에서 주목할 것은 소프트 목표(softgoal)라는 제약에 관한 표현을 사용하여 세부 목표들 간의 관계를 명시적으로 나타내도록 하였다는 것으로서 자연스러운 품질 표현이 가능하도록 한다.

반면, 그린츠(Glintz)[5]의 연구에서는 기존의 기능 요구와 비기능 요구와 같은 이분법적인 구분을 비판하며 4개의 패싯(facet)으로 보아야 함을 주장했다. 먼저 종류(kind) 패싯에서는 기능, 성능, 제약을, 표현 패싯에서는 운용적인지, 정량적인지, 정성적인지를 구분하며, 만족성 패싯에서는 쉬운지 어려운지를, 마지막으로 역할 패싯에서는 선언적인지 가정적인지를 구분하여 명세하도록 주장한 바 있다.

두 연구를 비교할 때 정과 닉슨의 연구는 품질 요구에 대한 선도적인 연구를 수행하였지만 개발자 입장에서 일반적인 프로젝트에 적용시킬 만큼 체계적이고 단순한 절차로 정리되지 못하였다는 한계와 더불어 실제 세부적으로 분해된 품질 요소들이 설계나 구현에 어떤 영향을 미치는지에 관한 설명에 대해서는 간과하고 있다. 또한 그린츠의 연구는 기존의 이분법적인 시각이 오히려 비기능 요소의 특성 파악에 방해가 된다고 주장함으로써 기존의 표현 문제를 새로운 시각에서 보고 있다. 그러나 기존의 개발자들이 친숙해있는 기능, 비기능의 개념을 뒤집을 정도의 급진적인 견해는 개발자에게 혼란을 가중시킬 위험이 있다.

본 논문은 정과 닉슨의 표기법에서 제안된 품질 표기 방법을 기반으로 실무 개발자가 적용할 수 있는 수준으로 단순화시켰으며, 이들의 연구에서 간과되었던 품질 요구 특성이 설계에 반영되어 상호 참조할 수 있는 메커니즘 제안에 비중을 두어 연구한다.

3. 품질의 표현과 설계 관계

3.1 품질의 표현

품질 요구에 대한 원천 정보는 사용자가 갖는 품질에 대한 요구 수준과 시스템 환경이 갖는 고유한 제약 사항으로부터 나온다. 품질에 대한 요구 수준이 막연할 경우 개발자와 사용자 모두 명확하지 못한 기준에서 출발할 수밖에 없으며 이는 결과물에 대한 실망으로 연결될 위험이 있다. 예를 들어 개발자가 사용자

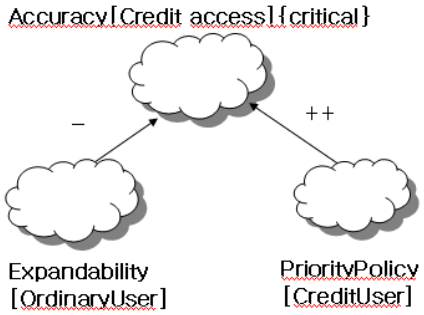
요구 명세서에 근거해 정보 검색 기능을 제공한다 하더라도 결과 값의 표현 형태나 탐색 방법이 까다롭다면 이에 대한 사용자의 만족도는 떨어질 수밖에 없다.

이러한 문제를 완화시킬 수 있는 방법으로 개발자는 사용자 만족성을 확보하기 위해 어떠한 설계요소를 고려했는지를 사전에 사용자에게 설명한다면 많은 도움이 될 것이다. 목표지향(goal-oriented) 접근 방법에서는 사용자의 품질 목표를 선언적으로 명시한다. 이 방법은 각각의 품질 요구를 목표로 설정하여 이를 충족시키기 위해 기존의 문제 분할 및 점령 방식으로 목표를 분해하는 과정을 반복한다. 목표 분할 과정에서는 다음의 활동이 수행된다.

1) 상위 품질 목표의 정의와 분할 : 시스템의 목표는 품질 요구사항을 의미하는 것으로서 설계에 고려해야 할 제약사항을 표현한다. 만일 한 목표가 너무 추상적이거나 큰 경우 다수의 세부 목표로 분할한다.
2) 목표 간의 관계 명세: 어떤 목표에 대해 예측 가능한 관계에 있는 다른 목표를 파악 후 이들의 관계를 서술한다. 대응되는 목표 간에는 보완, 갈등, 협력 관계가 있다. 각 품질 목표들 간의 관계에 있어 상승 작용이 있는 경우는 (+)를, 상쇄 관계에 있는 경우는 (-)기호로 표현하며, 이들 사이의 관계를 (++)와 같은 형식으로 결합 강도를 표현한다. 목표지향 명세에 있어 품질 목표는 다음과 같은 형식을 따른다.

품질 목표명[목표 대상]{우선순위}

이 때 품질 목표 명은 달성해야 할 품질 요소를, 목표 대상은 사용자가 직접 확인할 수 있는 엔티티를, 그리고 우선 순위는 품질요소의 상대적인 중요성을 의미한다.



(그림 1) 품질 목표의 표현

예를 들어 그림 1은 사용자 엔티티인 신용정보접근(credit access)에 대해 정확성을 중요한 수준에서 요구하고 있다. 이와 관련된 세부 품질 요소로 일반 고객 대상의 확장은 (-)의 강도의 부정적인 요인으로, 신용 사용자의 우선순위 정책은 (++) 강도의 긍정적인 요인으로 작용함을 도식으로 표현한다.

3.2 UML의 확장 표현

품질에 대한 목표지향 표현 방법은 사용자 요구 사항을 파악하는 단계에서 기능 요구 파악과 함께 사용될 수 있다. 만일 개발자가 UML을 사용하여 기능 요구 명세를 수행한다면 품질 목표 명세를 별도로 수행하는 것보다는 이를 통합하여 표현하는 것이 효과적이다.

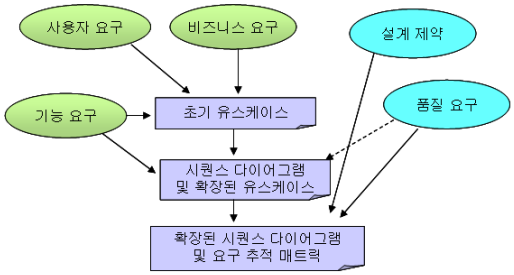
UML의 유스케이스는 시스템의 사용을 통해 얻고자 하는 서비스 기능을 명세하는 용도의 모델이므로 이 과정에서 예측할 수 있는 품질 요구 수준도 함께 파악하는 것은 자연스러운 일이다. 이를 지원하기 위해 도입되는 첫 번째 확장은 유스케이스 명세 부분에 품질 관련 서비스 요구를 명시적으로 표현하는 것이다. 이것은 사용자가 특정 활동에 대해 어떤 품질을 요구하는지에 관한 정보를 효과적으로 전달해준다. 이를 위해 유스케이스 명세는 다음과 같은 형식으로 확장되며 품질제약에서는 해당 시나

리오 부분에 대해 구체적인 품질 내용과 수준을 명세한다. 단, 기호 []는 선택을 의미한다.

유스케이스 명세::= 사전조건 + 사후조건 + 이벤트 흐름 + [대체흐름]+[예외흐름]+[품질제약]

두 번째 확장은 시나리오 명세와 관련한 시퀀스 다이어그램에 대해 스테레오타입 <<quality>> 표기를 도입하여 품질관련 이벤트와 객체를 명시적으로 표현하는 것이다. 스테레오타입 <<quality>>를 활용하는 예는 4장의 케이스 스터디에서 자세히 설명된다.

그림 2는 앞서 설명한 확장된 유스케이스와 시퀀스 다이어그램을 사용하여 품질 명세를 할 경우 관련 활동과 생성되는 문서들 간의 관계를 보여준다.

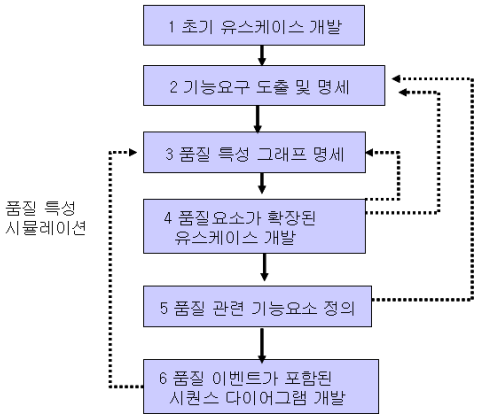


(그림 2) 유스케이스와 개발 문서의 관계

3.3 적용 절차

확장된 유스케이스와 시퀀스 다이어그램을 사용하여 품질 속성을 명세할 경우 개발자는 크게 두 단계의 명세를 수행한다. 먼저, 그림 3의 1, 2 항에 나타난 것처럼 초기 유스케이스와 기능 요구를 도출하는 활동을 수행한다. 다음 단계에서는 품질 요구를 결정짓기 위해 품질 특성 그래프와 품질 요소가 확장된 유스케이스를 개발하는 활동, 그리고 품질 이벤트를 포함하는 시퀀스 다이어그램을 확정하는 활동을 반복적으로 수행한다.

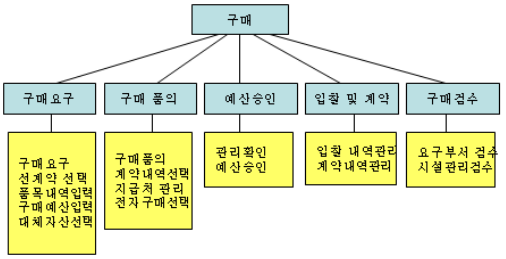
특히 품질 이벤트가 포함된 시퀀스 다이어그램 개발에서는 품질 항목에 대해 책임을 갖는 객체와 이와 관련된 이벤트를 정의하여 설계 단계에서 어떤 요소가 반영되어야 하는지 결정한다. 이 과정에서 얻어진 시퀀스 다이어그램은 개발자가 사용자에게 구현될 시스템의 서비스 수준을 확인받는 근거 자료로 활용한다.



(그림 3) 품질 명세를 포함하는 개발 절차

4. 케이스 스터디

본 절은 앞서 설명한 품질 표기 절차에 근거하여 실제 프로젝트에 적용시킨 과정을 소개한다. 케이스 스터디의 대상은 30 M/M 규모의 인력으로 개발된 구매 시스템이다. 이 시스템은 구매 처리를 위한 구매요구 처리, 확인 및 예산승인, 입찰 및 계약 관리, 구매 품의, 구매 검수 등의 업무 영역으로 구성되어 있다 (그림 4).



(그림 4) 구매 프로세스의 서비스 모듈

초기 구매 요구 분석 단계에서 파악된 총 요구 사항의 개수는 65개이며 이들은 사용자와의 협의를 통해 품질 요구와 우선순위를 포함하여 표 1과 같은 형식으로 표현된다.

(표 1) 초기 구매요구의 요구 정의서 예시

요구ID	기능 요구 내용	순위
R02001 단가내역 관리	각 모델별 부품 내역 관리가 가능하도록 보완하고 단가계약품목의 구매요구 시에는 해당 부품별 내역을 구매요구에 입력할 수 있도록 개선한다 - 해당 부품의 선택은 사용성을 최대화하여 내용을 확인하도록 한다	2
R2002 단가내역 조회	단가계약 상세내용을 일반사용자들이 검색할 수 있도록 한다 - 검색 결과는 가독성을 최대한 높이도록 표현한다 - 빠른 검색을 지원한다	3

초기 유스케이스 명세서는 초기 요구 정의서의 내용을 토대로 액터-시스템 간의 인터렉션을 서비스 단위로 서술하는 내용을 담는다. 표 3의 구매 유스케이스 UC1-01은 품질 제약 부분에서 구매 처리 검색 과정의 사용성을 높이며, 검색결과의 가독성, 그리고 검색어 처리 성능을 높일 것을 명시하고 있다.

(표 2) 구매 기능의 품질조건 정제

품질조건	품질 지원 기능
Q2001: 사용성 향상	Q2001F1: 하이퍼링크의 제공
	Q2001F2: 입력 자료의 undo 기능 제공
Q2002: 가독성 향상	Q2002F1: 중요 자료의 하일라이팅 기능 제공
	Q2002F2: 창 길이를 조절하는 스크롤바 제공
	Q2002F3: 계약별 자료 표현시 별도 창을 제공
Q2003: 속도 향상	Q2003F1: 빈번히 검색되는 자료는 캐쉬에 보관

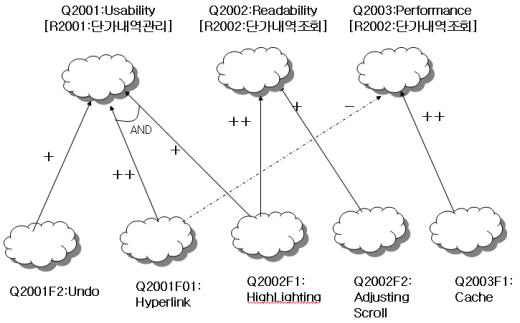
(표 3) 품질 확장 유스케이스의 명세 예

서브 시스템	구매	패키지	기초정보
유스케이스 ID	UC1-01	우선순위	중
유스케이스명	단가계약내역관리	액터명	총무팀
개요	단가계약내역을 관리한다.		
사전조건	해당없음		
사후조건	구매요구/구매품의/품목코드조회시 단가계약내역을 참조한다		
이벤트 흐름	사용자	시스템	
	[기본흐름] 1. 액터는 단가계약내역관리를 요구한다. 3. 액터는 단가계약내역을 입력, 조회, 수정, 삭제한다.	2. 시스템은 단가계약내역관리 화면을 보여준다.	
	[신규] 1. 액터는 신규입력을 요구한다 3. 액터는 품목코드, 상품명, 상세스펙, 단가 등을 입력하고 저장한다. 3. a 품목코드가 없다	2. 시스템은 단가계약내역을 입력할 수 있는 화면을 보여 준다. 4. 시스템은 입력 항목의 유효성을 검사한다. (4.e) 입력 항목이 유효하지 않다 5. 시스템은 단가계약내역을 저장하고 정상적으로 저장되었는지 확인 메시지를 보여준다.	
	[조회] 1. 액터는 품목팝업에서 품목을 선택하거나 상품명, 계약기간 등의 조회조건을 입력하고 단가계약 내역을 조회한다. 3. 액터는 조회된 단가계약 목록화면에서 단가계약 내역을 선택하여 수정, 삭제한다	2. 시스템은 조회조건에 맞는 단가계약목록을 보여준다. (2.e) 조회조건에 해당하는 자료가 없다 이하 생략..	

대체흐름	[신규] 3.a.1 품목코드관리(UC02-5101-01)에서 품목코드를 입력한다.
예외흐름	[신규]4.e.1 시스템 메시지를 보여준다.(입력 항목이 유효하지 않습니다.) [조회]2.e.1 시스템 메시지를 보여준다.(조회 항목이 존재하지 않습니다.) [수정]4.e.1 시스템 메시지를 보여준다.(수정 항목이 유효하지 않습니다.)
품질제약	Q2001: 구매 향식 처리의 사용성을 향상한다 Q2002: 결과물의 가독성 향상한다 Q2003: 질의 처리 속도를 향상시킨다

개발자는 파악된 품질 확장 유스케이스의 내용을 설계 단계에서 표현하기 위해 품질과 직접 관련되는 이벤트를 정의한다.

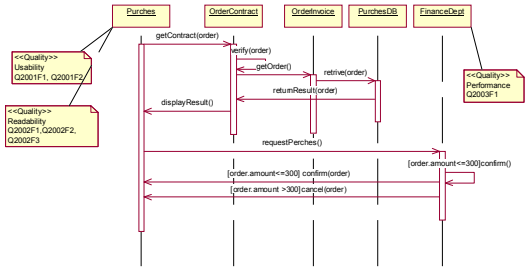
표 2는 3개 영역에 대한 6개의 품질 기능 이벤트 함수의 내용을 담고 있다. 표 2에서 정의된 품질 이벤트들 간의 관계는 그림 5와 같은 품질 특성 그래프를 통해 구체적으로 표현된다. 이 그래프에 의하면 하이퍼링크 기능은 사용성에 (++)의 강도로 긍정적인 기여하는 반면 성능 특성에는 (-) 강도로 부정적인 작용을 하고 있음을 알 수 있다. 개발자는 사용자에게 하이퍼링크 기능을 제공함으로써 얻는 득과 실을 이러한 그래프를 통해 설명함으로써 합리적인 수준의 품질 목표를 설정할 수 있다.



(그림 5) 구매요구의 품질 특성 그래프

이벤트 간의 상호작용은 UML의 시퀀스 다이어그램으로 표현된다. 시퀀스 다이어그램은 객체들끼리 주고받는 메시지의 순서를 시간의 흐름에 따라 보여주므로 시간에 따른 품질 제어 과정이 잘 이해될 수 있다.

예로서 그림 6은 앞서 파악된 유스케이스의 품질 제약 내용을 담은 초기 품질 확장 시퀀스 다이어그램으로 사용성, 가독성, 성능 요소가 고려되고 있음을 보여준다. 반면 이것이 확장된 그림 7의 경우는 사용성과 가독성 품질을 제공해야 하는 모듈로서 PurchesGUI는 품질 이벤트 Q2001F1, Q2001F2 이벤트와 더불어 사용성에 대한 책임을, CacheMgr 객체는 Q2003F1 품질 이벤트를 이용하여 성능 서비스에 대한 책임에 기여하고 있음을 보여준다.

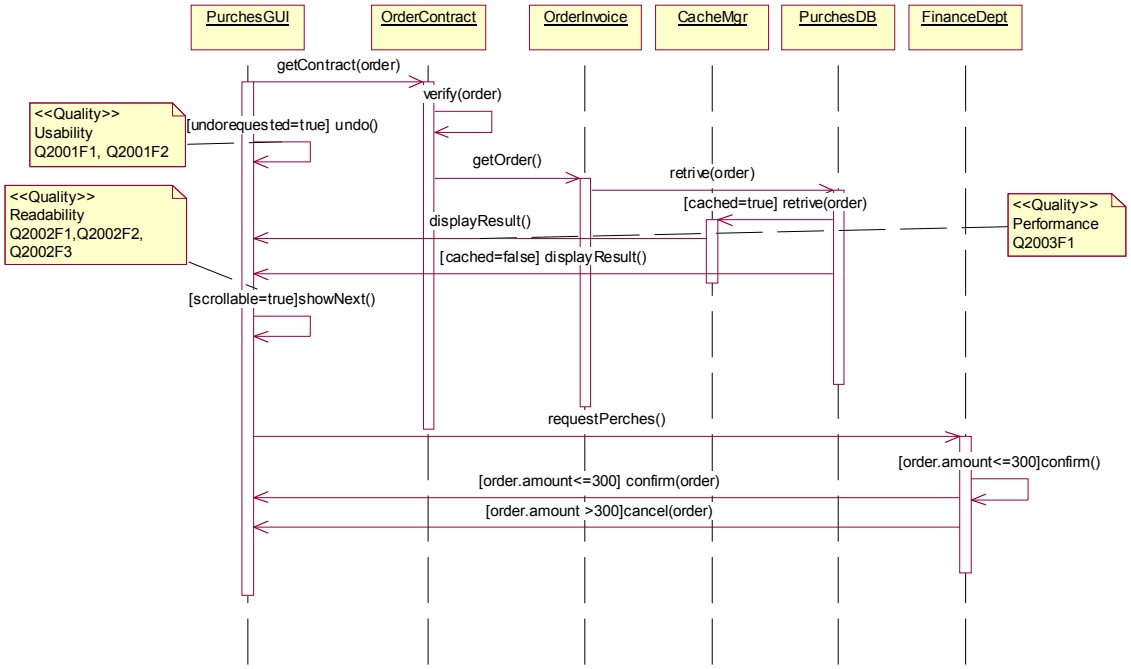


(그림 6) 초기 품질 확장 시퀀스 다이어그램

품질 목표-구현 이벤트 참조표의 가장 큰 용도는 품질 이벤트와 구현 부분에 대한 추적 링크를 제공해 준다는 점이다. 만일 품질 이벤트에 변경이 발생할 경우 개발자는 이벤트 참조표와 품질 요구 특성 그래프를 토대로 모듈을 수정할 수 있다. 또한 수정을 통해 다른 품질요

(표 4) 품질 목표 구현 이벤트 참조표

관련 기능	품질이벤트 명	구현모듈
Q2001F01: 하이퍼링크의 제공	Q1F01: setHyperlink()	PurchesGUI
Q2001F02: 입력 자료의 undo 기능 제공	Q1F02: undo()	PurchesGUI
Q2002F1: 중요 자료의 하이라이팅 기능 제공	Q2F01: setHighlighting()	PurchesGUI
Q2002F2: 창 길이를 조절하는 스크롤바 제공	Q2F02: enableScroll()	PurchesGUI
Q2002F3: 계약별 자료 표현시 별도 창을 제공	Q2F03: showContractFrame()	PurchesGUI
Q2003F01: 빈번히 검색되는 자료는 캐쉬에 보관	Q1F02: getRecFromCache()	CacheMgr



(그림 7) 정제된 품질확장 시퀀스 다이어그램

소에는 어떤 변경이 생기는지 추정할 수 있다. 정제된 품질확장 시퀀스 다이어그램에서 표현된 품질 이벤트와 품질 목표와의 관계는 표 4에서와 같이 표현되어 상호 참조 관계를 추적할 수 있도록 한다.

이상에서 설명된 과정을 통해 확보한 품질요구 사항은 그림 8과 같은 형태로 화면 설계에 반영이 된다.

(그림 8) 설계된 구매요구 처리 화면 예시

5. 결 론

본 논문에서는 정보시스템을 구축할 때 적용될 수 있는 품질 요구의 표현과 설계 과정에 있어서 품질 조건을 만족시키는지를 확인할 수 있는 절차를 제안하였다. 본 연구 결과는 기존의 개발기법과 구분되는 다음의 특징을 갖는다. 먼저, 요구 명세 단계에서 파악된 목표지향 품질 요구의 내용을 설계에 반영하기 위해서는 설계 단계의 배려가 필요하다. 이를 위해 파악된 품질 조건은 유스케이스의 사전조건, 사후조건과 더불어 시스템 설계에 반영할 수 있도록 확장한다. 또한, 설계 단계에서부터 품질 요구와 직접 관련이 되는 이벤트를 분리 정의하는 것은 이후의 품질 요소 구현이나 변경의 추적에 직접적인 도움을 줄 수 있게 하였다. 마지막으로, 품질 요구에 정의된 품질 요소와 이를 구현하는 이벤트 간에 관계를 시뮬레이션을 통해 가장 적합한 수준의 품질 요구 수준을 설정할 수 있다.

본 논문에서 제안한 품질 요소의 표현과 설계 방법은 품질 관련 이벤트들이 명확히 정의될 수 있는 경우에 한정한다. 따라서 하드웨어 환경에 종속되는 품질 특성들, 예를 들면 강인성, 고장 감내성과 같은 조건들, 신뢰성과 같이 특정한 몇 개의 함수로는 구현시킬 수 없는 부분들에 관한 검증에는 적용할 수 없는 한계를 가지고 있다.

참 고 문 헌

[1] Karl E. Wiegers, Software Requirements, Microsoft GmbH, 2006

[2] 프로젝트 성공 잠재력 기준표, Standish Group, 2006

[3] 최은만, 소프트웨어공학론, 정보문화사 2006

[4] L. Chung, B A. Nixon, E Yu, J. Mylopoulos, Non-Functional Requirements in Software Engineering, Kluwer Academic Publishers, 2000

[5] M. Glinz, Rethinking the Notion of Non-Functional Requirement, Proc. of Third World Congress for Software Quality, Munich, Germany, pp55-64, 2005

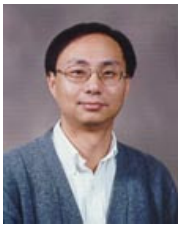
[6] J. Arlow, I. Neustadt, UML2 and the Unified Process, 2nd ed. Addison Wesley, 2005

[7] L. Chung, B. A. Nixon and E. Yu, Dealing with Change: An Approach Using Non-Functional Requirements, Requirements Engineering Journal, Vol 1 No4, pp. 238-259, 1996

[8] D. Moody, Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions, Data & Knowledge Engineering, Vol 55, p243-276, 2005

[9] D. Leffingwell, D. Widrig, Managing Software Requirements: A Use Case Approach, 2nd ed. Addison Wesley, 2003

◐ 저 자 소 개 ◑



서 동 수

1987년 중앙대학교 컴퓨터공학과(이학사)
1990년 맨체스터대학교 대학원 Department of Computation(공학석사)
1994년 맨체스터대학교 대학원 Department of Computation(공학석사)
1998～현재 성신여자대학교 컴퓨터정보학부 교수
관심분야 : 소프트웨어공학, 정형기법
E-mail : dseo@sungshin.ac.kr