(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0044634
(43) 공개일자 2013년05월03일

(51) 국제특허분류(Int. Cl.)
G06F 9/44 (2006.01)
(21) 출원번호 10-2011-0108791
(22) 출원일자 2011년10월24일
심사청구일자 2011년10월24일

(71) 출원인
고려대학교 산학협력단
서울 성북구 안암동5가 1
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
(72) 발명자
인호
서울특별시 성북구 인촌로10길 66, 201호 (안암동3가)
김능회
경기도 부천시 오정구 상오정로7번길 43, 101동 209호 (오정동, 누리안아파트)
(뒷면에 계속)
(74) 대리인
특허법인충현

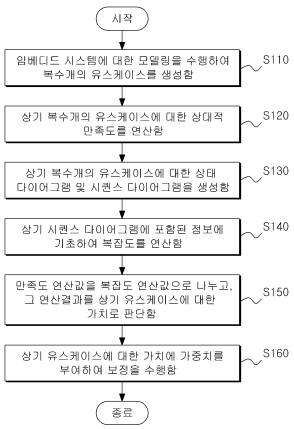
전체 청구항 수 : 총 5 항

(54) 발명의 명칭 **임베디드 시스템 내 유스케이스에 대한 가치연산방법**

(57) 요 약

본 발명은 임베디드 시스템 내 유스케이스에 대한 가치연산방법에 관한 것으로, 보다 구체적으로는 유스케이스생성모듈이 임베디드 시스템에 대한 모델링을 수행하여 복수 개의 유스케이스를 생성하는 생성단계; 만족도연산모듈이 상기 복수 개의 유스케이스를 상호 비교하고, 기설정된 중요도테이블에 기초하여 상기 복수 개의 유스케이스에 대한 상대적 만족도를 연산하여 만족도 연산값을 생성하는 만족도연산단계; 다이어그램생성모듈이 각 유스케이스별 상태 다이어그램(state diagram) 및 시퀀스 다이어그램(sequence diagram)을 생성하는 다이어그램생성단계; 복잡도연산모듈이 상기 유스케이스에 대한 시퀀스 다이어그램에 포함된 정보에 기초하여 상기 유스케이스의 복잡도를 연산하여 복잡도 연산값을 생성하는 복잡도연산단계; 가치연산모듈이 상기 만족도 연산값을 상기 복잡도 연산값으로 나눈 값을 상기 유스케이스에 대한 가치로 판단하는 가치연산단계;를 포함한다. 이러한 구성에 의해, 본 발명의 임베디드 시스템 내 유스케이스에 대한 가치연산방법은 사용자 만족도 및 유스케이스의 복잡도를 각각 연산한 후, 그 연산결과에 기초하여 유스케이스에 대한 가치를 연산함으로써, 다수의 유스케이스 중 가치가 높은 유스케이스를 용이하게 선별할 수 있는 효과가 있다.

대 표 도 - 도1



(72) 발명자

이윤규

서울특별시 성북구 안암로9가길 76, 제헌빌딩 609호 (안암동5가)

서동민

서울특별시 노원구 공릉로58다길 8, 202호 (하계동, 설악빌라트)

특허청구의 범위

청구항 1

유스케이스생성모듈이 임베디드 시스템에 대한 모델링을 수행하여 복수 개의 유스케이스를 생성하는 생성단계;

만족도연산모듈이 상기 복수 개의 유스케이스를 상호 비교하고, 기설정된 중요도테이블에 기초하여 상기 복수 개의 유스케이스에 대한 상대적 만족도를 연산하여 만족도 연산값을 생성하는 만족도연산단계;

다이어그램생성모듈이 각 유스케이스별 상태 다이어그램(state diagram) 및 시퀀스 다이어그램(sequence diagram)을 생성하는 다이어그램생성단계;

복잡도연산모듈이 상기 유스케이스에 대한 시퀀스 다이어그램에 포함된 정보에 기초하여 상기 유스케이스의 복잡도를 연산하여 복잡도 연산값을 생성하는 복잡도연산단계;

가치연산모듈이 상기 만족도 연산값을 상기 복잡도 연산값으로 나눈 값을 상기 유스케이스에 대한 가치로 판단하는 가치연산단계;

를 포함하는 임베디드 시스템 내 유스케이스에 대한 가치연산방법.

청구항 2

제1항에 있어서,

상기 복잡도연산단계는

상기 시퀀스 다이어그램의 오브젝트 개수에 기설정된 가중치를 곱한 값에, 컨트롤러로부터 발생되는 메시지의 개수에 기설정된 가중치를 곱한 값을 합산한 후, 그 합산결과에 사용자를 나타내는 액터의 개수를 합산하여 유스케이스에 대한 복잡도를 연산하는 것을 특징으로 하는 임베디드 시스템 내 유스케이스에 대한 가치연산방법.

청구항 3

제1항에 있어서,

상기 가치연산단계 수행 후,

상기 복수 개의 유스케이스 중 선행개발되어야만 다른 유스케이스의 개발이 가능한 유스케이스의 가치에 가중치를 부여하는 제1보정과정을 수행하는 보정단계;

를 더 포함하는 것을 특징으로 하는 임베디드 시스템 내 유스케이스에 대한 가치연산방법.

청구항 4

제3항에 있어서,

상기 보정단계는

사용자에 의해 요구되는 완성도 수준에 따라 상기 유스케이스의 가치에 가중치를 부여하는 제2보정과정;을

더 포함하는 것을 특징으로 하는 임베디드 시스템 내 유스케이스에 대한 가치연산방법.

청구항 5

제1항 내지 제4항 중 어느 한 항에 따른 방법을 컴퓨터로 실행하기 위한 프로그램이 기록된 컴퓨터 판독가능 기록매체.

명 세 서

기 술 분 야

[0001] 본 발명은 임베디드 시스템 내 유스케이스에 대한 가치연산방법에 관한 것으로, 특히 다수의 유스케이스 중 사용자 만족도가 높으면서, 개발비용이 낮은 유스케이스를 먼저 개발하도록 상기 다수의 유스케이스에 대한 가치를 연산하는 임베디드 시스템 내 유스케이스에 대한 가치연산방법에 관한 것이다.

배 경 기 술

[0002] IT기술의 급속한 발전에 따라, 각 상황 또는 사용자의 요구에 맞게 다양한 기능을 수행하는 소프트웨어를 탑재하는 임베디드 시스템이 다양한 분야에서 널리 사용되고 있다.

[0003] 이처럼 다양한 분야에서 사용되는 임베디드 시스템은 내부에 탑재되는 소프트웨어가 중요한 역할을 하는데 특히, 상기 소프트웨어의 각 반복과정에서 수행되는 유스케이스(use case)를 선정하여 개발하는 것이 점차 중요해지고 있다.

[0004] 특히, 초기의 반복과정 상에서 수행되는 유스케이스는 연속하여 반복되는 과정을 통해 전체 임베디드시스템의 후반부에 배치되는 유스케이스에 비하여 상대적으로 많은 검증과정을 거치기 때문에, 더욱 높은 양질을 보유하는 특징을 갖는다.

[0005] 이에 따라, 상기 임베디드 시스템을 구성하는 다수의 유스케이스들 중에서 가장 핵심적인 유스케이스를 선별하여 초기 반복과정에 배치시킴으로써, 상기 임베디드 시스템에 대한 질을 향상시키도록 하고 있다.

[0006] 이와 같이, 가장 핵심적인 유스케이스를 선별하는 과정은 여러 기준을 통해서 유스케이스를 선별할 수 있는데 먼저, 사용자가 가장 중요하다고 판단하는 유스케이스를 기준으로 우선순위를 부여하는 방법이 있고, 개발자 입장에서 구현하기 어렵다고 판단하는 즉, 복잡도가 높은 유스케이스를 기준으로 우선순위를 부여하는 방법도 있다.

[0007] 하지만 이와 같은 방법은 사용자가 가장 중요한 유스케이스라고 판단하더라도 개발에 따른 많은 시간과 비용이 소모되는 경우, 핵심적인 유스케이스라고 판단하기는 어려움에 따라 이러한 유스케이스를 초기 반복과정에 배치하는 경우에는 임베디드 시스템의 질 또한 저하된다는 문제점이 발생했다.

[0008] 상술한 바와 같이, 임베디드 시스템 내 유스케이스에 대한 가치연산방법에 대한 선행기술을 살펴보면 다음과 같다.

[0009] 선행기술 1은 제2003-0041040호(2003.05.23)로서, 소프트웨어 규모 및 복잡도 측정방법 및 시스템에 관한 것이다. 이러한 선행기술 1은 소프트웨어 요구사항 명세서로부터 주요 이벤트 흐름과, 대안적 이벤트 흐름의 수, 유스케이스 간의 포함관계 및 확장관계를 분석하여 정보를 추출하고, 소프트웨어설계서로부터 클래스를 구성하는 속성, 오퍼레이션을 분석하여 정보를 추출하고, 소스코드를 분석하여 정보를 추출한 후, 추출된 정보에 의해 소프트웨어의 규모 및 복잡도를 측정한다.

[0010] 또한 선행기술 2는 한국공개특허 제2007-0058954호(2007.06.11)로서, 임베디드 시스템 프로토타입 개발 지원 장치 및 방법에 관한 것이다. 이러한 선행기술 2는 사용자로부터 입력된 임베디드 시스템의 요구 사항 정보 및 유스케이스 정보를 분석하여 유스케이스 분석표를 생성하고, 상기 사용자로부터 입력된 임베디드 시스템의 소프트웨어 및/또는 하드웨어 구조와 각 구조에 대한 구성 요소를 분석하여 시스템 구조 그래프를 생성하고, 상기 생성한 시스템 구조 그래프를 유스케이스 분석표에 반영하여 상기 유스케이스 분석표를 업데이트시키며, 상기 구성 요소 간의 수행 흐름과 호출 관계를 포함하는 행위 정보를 정의하고, 상기 구성 요소가 정의된 상기 행위 정보에 따라 동작되도록 코드를 생성하고, 구성 요소의 기능 수행 여부를 확인해 나가며 상기 시스템을 구현하고, 구현된 시스템의 기능과 성능을 테스트함으로써, 하드웨어와 소프트웨어가 함께 설계되고 검증되도록 한다.

발명의 내용

해결하려는 과제

[0011] 상기와 같은 종래 기술의 문제점을 해결하기 위해, 본 발명은 임베디드 시스템의 복수 개의 유스케이스에 대하여 사용자만족도 및 개발비용에 따라 가치를 연산함으로써, 높은 가치를 갖는 유스케이스가 먼저 개발되도록 하는 임베디드 시스템 내 유스케이스에 대한 가치연산방법을 제공하고자 한다.

과제의 해결 수단

[0012] 위와 같은 과제를 해결하기 위한 본 발명의 한 실시 예에 따른 임베디드 시스템 내 유스케이스에 대한 가치연산방법은 유스케이스생성모듈이 임베디드 시스템에 대한 모델링을 수행하여 복수 개의 유스케이스를 생성하는 생성단계; 만족도연산모듈이 상기 복수 개의 유스케이스를 상호 비교하고, 기설정된 중요도테이블에 기초하여 상기 복수 개의 유스케이스에 대한 상대적 만족도를 연산하여 만족도 연산값을 생성하는 만족도연산단계; 다이어그램생성모듈이 각 유스케이스별 상태 다이어그램(state diagram) 및 시퀀스 다이어그램(sequence diagram)을 생성하는 다이어그램생성단계; 복잡도연산모듈이 상기 유스케이스에 대한 시퀀스 다이어그램에 포함된 정보에 기초하여 상기 유스케이스의 복잡도를 연산하여 복잡도 연산값을 생성하는 복잡도연산단계; 가치연산모듈이 상기 만족도 연산값을 상기 복잡도 연산값으로 나눈 값을 상기 유스케이스에 대한 가치로 판단하는 가치연산단계;를 포함한다.

[0013] 보다 바람직하게는 상기 시퀀스 다이어그램의 오브젝트 개수에 기설정된 가중치를 곱한 값에, 컨트롤러로부터 발생되는 메시지의 개수에 기설정된 가중치를 곱한 값을 합산한 후, 그 합산결과에 사용자를 나타내는 액터의 개수를 합산하여 유스케이스에 대한 복잡도를 연산하는 복잡도연산단계를 포함할 수 있다.

[0014] 보다 바람직하게는 상기 가치연산단계 수행 후, 상기 복수 개의 유스케이스 중 선행조건을 갖는 유스케이스가 존재하는 경우, 상기 유스케이스의 가치에 선행조건에 따른 가중치를 부여하는 제1보정과정을 수행하는 보정단계;를 더 포함할 수 있다.

[0015] 특히, 사용자에 의해 요구되는 완성도 수준에 따라 상기 유스케이스의 가치에 가중치를 부여하는 제2보정과정 또는 상기 복수 개의 유스케이스 중 선행개발되어야만 다른 유스케이스의 개발이 가능한 유스케이스의 가치에 가중치를 부여하는 제3보정과정 중 적어도 하나의 보정과정을 더 포함하는 보정단계를 포함할 수 있다.

발명의 효과

[0016] 본 발명의 임베디드 시스템 내 유스케이스에 대한 가치연산방법은 사용자 만족도 및 유스케이스의 복잡도를 각각 연산한 후, 그 연산결과에 기초하여 유스케이스에 대한 가치를 연산함으로써, 다수의 유스케이스 중 가치가 높은 유스케이스를 용이하게 선별할 수 있는 효과가 있다.

[0017] 또한, 본 발명의 임베디드 시스템 내 유스케이스에 대한 가치연산방법은 사용자 만족도 및 유스케이스의 복잡도를 연산한 결과에 기초하여 가치가 가장 높은 유스케이스를 용이하게 판단함에 따라, 전체 임베디드 시스템의 질을 향상시킬 수 있는 효과가 있다.

도면의 간단한 설명

[0018] 도 1은 본 발명의 일 실시 예에 따른 임베디드 시스템 내 유스케이스에 대한 가치연산방법의 순서도이다.

도 2는 본 발명의 다른 실시 예에 따른 유스케이스에 대한 상태 다이어그램 및 시퀀스 다이어그램을 나타낸 도면이다.

도 3은 본 발명의 다른 실시 예에 따른 유스케이스의 시퀀스 다이어그램을 나타낸 도면이다.

도 4는 본 발명의 또 다른 실시 예에 따른 휴대폰 카메라 시스템의 유스케이스에 대한 유스케이스 모델링을 나타내는 도면이다.

도 5는 본 발명의 휴대폰 카메라 시스템의 동영상촬영 유스케이스에 대한 상태 다이어그램을 나타낸 도면이다.

도 6은 본 발명의 휴대폰 카메라 시스템의 동영상촬영 유스케이스에 대한 시퀀스 다이어그램을 나타낸 도면이다.

발명을 실시하기 위한 구체적인 내용

[0019] 이하, 본 발명을 바람직한 실시 예와 첨부한 도면을 참고로 하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며, 여기에서 설명하는 실시 예에 한정되는 것은 아니다.

[0020] 임베디드 시스템을 구성하는 다수의 유스케이스들 중 가장 핵심적인 유스케이스를 선별하기에 앞서, 핵심적인 유스케이스를 선별하는데 기준으로 사용되는 가치에 대하여 자세히 알아보도록 한다.

[0021] 핵심적인 유스케이스를 선별하는데 기준으로 사용되는 가치란, 임베디드 시스템에 탑재되는 소프트웨어 개발의 투자수익률 즉 ROI(Return On Investment)를 말한다. 이와 같은 가치를 각 유스케이스별로 판단하여, 상기 임베디드 시스템에 탑재되는 다수의 유스케이스 중에서 사용자만족도가 높으면서 개발비용이 적게 소용되는 가치를 갖는 유스케이스를 먼저 개발하도록 하여, 임베디드 시스템의 효율성을 높이고자 하는 것이다.

[0022] 이하, 도 1을 참조하여, 본 발명의 임베디드 시스템 내 유스케이스에 대한 가치연산방법에 대하여 자세히 살펴보도록 한다.

[0023] 도 1은 본 발명의 일 실시 예에 따른 임베디드 시스템 내 유스케이스에 대한 가치연산방법의 순서도이다.

[0024] 도 1에 도시된 바와 같이, 본 발명의 임베디드 시스템 내 유스케이스에 대한 가치연산방법은 먼저 유스케이스생성모듈이 임베디드 시스템에 대한 모델링을 수행하여 상기 임베디드 시스템에 포함되는 복수 개의 유스케이스를 생성한다(S110).

[0025] 만족도연산모듈이 앞서 생성된 복수 개의 유스케이스를 상호 비교하고 기설정된 중요도테이블에 기초하여 상기 복수 개의 유스케이스에 대한 상대적 만족도를 연산하여 만족도연산값을 생성한다(S120).

[0026] 즉, 복수 개의 유스케이스를 행렬한 후, 각각의 쌍별로 상호 비교하고, 그 비교결과에 기설정된 중요도테이블에 따라 중요도를 부과한 후, 이를 연산하여 각 유스케이스에 대한 상대적 만족도(Relative Satisfaction Degree)를 연산하는 것이다. 이때, 상기 중요도테이블은 임베디드 시스템에 대한 공헌도 또는 선호도에 따라 차등된 점수가 할당된 테이블을 말한다. 예를 들어, 비교하고자 하는 한 쌍의 유스케이스(U1, U2)가 서로 비슷한 공헌도를 갖는다고 판단하는 경우, 상기 한 쌍의 유스케이스(U1, U2)는 중요도 1이 부과되고, 이후, 공헌도가 높아질수록 높은 중요도가 2 내지 9까지의 형태로 부과되도록 한다.

[0027] 이에 따라, 상기 중요도테이블에 기초하여 각 유스케이스에 중요도가 부과되고, 행렬상태로 배치되는 각 유스케이스에 대하여 열간 부여된 중요도를 합산한 후, 각 열의 중요도를 열간 부여된 중요도 합산결과로 나눈다.

[0028] 이와 같이 행렬상태로 배열된 각 유스케이스에 대하여 행간의 합을 연산하고 각 행간의 합을 유스케이스의 수로 나누어 각 유스케이스간의 상대적 만족도를 연산한다.

[0029] 이후, 다이어그램생성모듈이 임베디드 시스템을 모델링하여 생성된 상기 복수 개의 유스케이스에 대하여 각 유스케이스별 상태 다이어그램(state diagram) 및 시퀀스 다이어그램(sequence diagram)을 생성한다(S130).

[0030] 이하, 도 2를 참조하여, 상기 상태 다이어그램 및 시퀀스 다이어그램에 대하여 보다 자세히 살펴보도록 한다.

[0031] 도 2는 본 발명의 다른 실시 예에 따른 유스케이스에 대한 상태 다이어그램 및 시퀀스 다이어그램을 나타낸 도면이다.

[0032] 도 2(a)에 도시된 바와 같이, 임베디드 시스템 내에서 발생하는 각각의 이벤트에 대응하여 구현되어야 하는 적어도 하나의 특정상태가 다이어그램으로서 표현된다.

[0033] 또한 도 2(b)에 도시된 바와 같이, 도 2(a)에 도시화된 다수의 각 상태(state)가 컨트롤러의 라이프라인 상에 기재되는 시퀀스 다이어그램(sequence diagram)이 생성되어, 다수의 각 상태들이 임베디드 시스템의 내부 컴포넌트와 어떠한 영향을 상호 미치는지 알 수 있게 된다.

[0034] 이후, 복잡도연산모듈이 상기 유스케이스에 대한 시퀀스 다이어그램에 포함된 적어도 하나의 정보에 기초하여 상기 유스케이스의 복잡도를 연산함으로써, 복잡도 연산값을 생성한다(S140).

[0035] 이하, 도 3 및 표 1을 참조하여, 시퀀스 다이어그램에 포함된 적어도 하나의 정보에 기초하여 유스케이스의 복잡도를 연산하는 과정에 대하여 보다 자세히 설명하도록 한다.

[0036] 도 3은 본 발명의 다른 실시 예에 따른 유스케이스의 시퀀스 다이어그램을 나타낸 도면이고, 표 1은 도 3의 시

퀀스 다이어그램의 복잡도를 나타낸 표이다.

[0037] 도 3에 도시된 바와 같이, 시퀀스 다이어그램은 점선으로 표시되는 A영역은 임베디드 시스템의 구성요소 중 액터를 나타내고, B영역은 컨트롤러가 제어하는 컴포넌트를 포함하는 오브젝트를 나타내며, C영역은 상기 컨트롤러에서 발생되는 메시지를 나타낸다.

[0038] 이와 같이, 복수 개의 유스케이스에 해당하는 시퀀스 다이어그램 중 하나의 시퀀스 다이어그램에 대하여, 상기 오브젝트와 메시지의 개수에 미리 할당된 가중치를 곱한값을 상기 액터의 개수와 합산함으로써, 상기 시퀀스 다이어그램에 대응하는 유스케이스에 대한 복잡도를 연산한다.

[0039] [표 1]

[0040]

복잡도 요소	요소 이름	형태	가중치
액터	User	-	1
액터복잡도= 1			1
오브젝트	DirectionCalculatorIF	Complex	2.0
	WheelActuatorIF	Average	1.5
	SonarSensorIF	Average	1.5
	IntrusionDetectorIF	Complex	2.0
	SpeakerIF	Simple	1.0
오브젝트 복잡도= (1*2.0)+(1*1.5)+(1*1.5)+(1*2.0)+(1*1.0)			8
메시지	setRange()	Asynchronous	1.0
	getDirection()	Asynchronous	1.0
	turnDirection()	Asynchronous	1.0
	startWheel()	Asynchronous	1.0
	getSonarSensor()	Asynchronous	1.0
	checkIntrusion()	Asynchronous	1.0
	stopWheel()	Asynchronous	1.0
	alarm()	Asynchronous	1.0
메시지 복잡도= (1*1.0)+(1*1.0)+(1*1.0)+(1*1.0)+(1*1.0)+(1*1.0)+(1*1.0)+(1*1.0)			8
유스케이스 복잡도= 1 + 8 + 8			17

[0041] 예를 들어, 표 1에 도시된 바와 같이, 액터의 개수는 1개이고, 오브젝트는 5개이며, 메시지는 8개로 구성되는 경우, 상기 오브젝트에 해당하는 요소를 복잡한 난이도에 따라 complex, average, simple 의 세 단계로 분류할 수 있으며, 이때, 분류된 상기 오브젝트의 가중치는 complex가 2.0이고, average가 1.5이며, simple은 1.0으로 설정한다. 이에 따라, 앞서 기술한 복잡도 연산과정에 따라 복잡도를 연산하면, 액터의 개수는 1개이므로, 상기 액터의 복잡도는 1이된다.

[0042] 이어서, 상기 오브젝트는 DirectionCalculatorIF, WheelActuatorIF, SonarSensorIF, IntrusionDetectorIF, SpeakerIF의 총 5개이며, 이때 상기 DirectionCalculatorIF 및 IntrusionDetectorIF의 난이도가 상임에 따라 2.0의 가중치가 부여되고, 상기 WheelActuatorIF 및 SonarSensorIF의 난이도가 중임에 따라 1.5의 가중치가 부여되며, 마지막으로 SpeakerIF의 난이도는 하임에 따라 1.0의 가중치가 부여되는 것을 알 수 있다. 따라서, 상기 오브젝트의 복잡도는 (1×2.0)+(1×1.5)+(1×1.5)+(1×2.0)+(1×1.0)인 8이 된다.

[0043] 이어서, 상기 오브젝트의 개수가 5이고, 각 분류된 상태에 할당된 가중치는 곱하여, 상기 오브젝트의 복잡도는 8이된다. 더불어, 상기 메시지의 개수는 8개이고, 가중치는 8개 모두 1.0이 할당됨에 따라 상기 메시지의 복잡도는 8이된다. 따라서, 상기 시퀀스 다이어그램의 복잡도는 상기 액터의 복잡도 1과 상기 오브젝트의 복잡도 8 및 상기 메시지의 복잡도 8을 모두 합산한 17이 된다. 이러한 복잡도 연산과정은 하나의 유스케이스에 하나의 시퀀스 다이어그램이 포함된 경우를 나타낸다. 하지만 이와 달리, 하나의 유스케이스에 복수 개의 시퀀스 다이

어그램이 포함된 경우에는 상기 복수 개의 시퀀스 다이어그램의 복잡도를 모두 합산함으로써, 상기 유스케이스에 대한 복잡도를 산정할 수 있다.

[0044] 이에 따라, 가치연산모듈이 앞서 연산한 상기 만족도 연산값을 상기 복잡도 연산값으로 나누고, 나눈 값을 상기 유스케이스에 대한 가치로 판단한다(S150).

[0045] 예를 들어, 유스케이스의 상대적 만족도를 나타내는 만족도 연산값이 10이고, 앞서 연산한 상기 유스케이스의 복잡도 연산값이 17인 경우, 상기 만족도 연산값인 10을 상기 복잡도 연산값 17로 나누고, 이에 100을 곱하여 정수화시킨 후, 소수점 첫째 자리에서 반올림하면 59가 된다. 이러한 연산과정을 거쳐, 상기 유스케이스에 대한 가치는 59가 되는 것을 알 수 있다.

[0046] 뿐만 아니라, 이러한 상기 유스케이스의 가치연산 과정 이후, 상기 복수 개의 유스케이스 중 선행개발되어야만 다른 유스케이스의 개발이 가능한 특정 유스케이스에 대하여 가치에 가중치를 부여하거나, 사용자에 의해 요구되는 수준에 따라 특정 유스케이스의 가치에 가중치를 부여하는 보정과정을 수행할 수 있다(S160).

[0047] 특히, 복수 개의 유스케이스의 개발 시, 먼저 특정 유스케이스(이하, 제1 유스케이스라고 한다.)가 개발되어야, 앞서 개발된 제1유스케이스를 통해 이어서 다른 유스케이스(이하, 제2 유스케이스라고 한다.)가 개발가능한 경우가 존재할 수 있는데, 이때 상기 보정과정을 통해 상기 제1 유스케이스에 가중치를 부과하여, 상기 제1 유스케이스의 가치를 보정하는 것이다.

[0048] 또한 이러한 보정 외에도, 사용자 또는 개발을 의뢰한 발주자가 일정 수준 이상을 갖는 유스케이스를 요구하는 경우, 상기 유스케이스에 대하여 상기 사용자 또는 발주자에 의해 요구되는 완성도 수준을 상, 중, 하의 세 단계로 나누고, 요구되는 완성도의 수준에 따라 서로 다른 가중치를 해당 유스케이스의 가치에 부여함으로써, 보정을 수행한다.

[0049] 이하, 도 4 내지 도 6 및 표 2 내지 표 6을 이용하여 본 발명의 또 다른 실시 예에 따른 휴대폰 카메라 시스템의 유스케이스의 가치연산에 대하여 살펴보기로 한다.

[0050] 도 4는 본 발명의 또 다른 실시 예에 따른 휴대폰 카메라 시스템의 유스케이스에 대한 유스케이스 모델링을 나타내는 도면이다.

[0051] 도 4에 도시된 바와 같이, 유스케이스생성모듈이 휴대폰 카메라 시스템에 대한 모델링을 수행하여 이전보기, 스냅샷촬영, 영상녹화, 이후보기, 영상재생, 앨범관리, 사진영상편집, 사진영상전송, 사진출력, 앨범보기와 같은 총 10개의 유스케이스를 생성한다. 또한, 상기 휴대폰 카메라 시스템은 상기 10개의 유스케이스를 수행하는 주체인 캠 센서, 마이크, 스피커, 사용자, 엘씨디, 월페이퍼 매니저, MMS 매니저, 블루투스 매니저, 프린터의 총 9개의 액터로 구성된다.

[0052] 이어서, 만족도연산모듈이 상기 10개의 유스케이스에 대한 복잡도를 상호 비교하고, 미리 설정된 중요도테이블에 따라 상대적 만족도를 연산하며 각 유스케이스별 만족도연산값을 테이블화한다.

[0053] 표 2는 휴대폰 카메라 시스템의 복수 개의 유스케이스에 대한 상대적 만족도를 연산하여 만족도연산값을 테이블화한 표이다.

[0054] [표 2]

유스케이스 ID	유스케이스 이름	만족도 연산값
U_2	스냅샷 촬영	24
U_3	영상녹화	16
U_1	이전보기	11
U_4	이후보기	11
U_5	영상재생	10
U_6	앨범관리	7
U_{10}	앨범보기	7
U_8	사진영상 전송	6
U_7	사진영상 편집	5
U_9	사진출력	4

[0055]

[0056] 표 2에 도시된 바와 같이, 복수 개의 유스케이스에 대하여 미리 부여된 고유한 ID와, 각 유스케이스에 따른 만족도연산값을 알 수 있다. 예를 들어, 유스케이스 ID가 U2에 해당하는 스냅샷 촬영 유스케이스의 만족도 연산값이 24로서, 휴대폰 카메라 시스템의 다른 유스케이스보다 가장 높은 만족도 연산값을 가짐에 따라, 상기 스냅샷 촬영 유스케이스가 휴대폰 카메라 시스템의 다수의 유스케이스 중 가장 높은 상대적 만족도를 갖는 것을 알 수 있다.

[0057] 이처럼, 복수 개의 각 유스케이스에 대한 상대적 만족도를 연산한 후, 상기 복수 개의 각 유스케이스에 대한 상태 다이그램 및 시퀀스 다이어그램을 생성한다.

[0058] 도 5는 본 발명의 휴대폰 카메라 시스템의 동영상촬영 유스케이스에 대한 상태 다이어그램을 나타낸 도면이다.

[0059] 도 5에 도시된 바와 같이, 휴대폰 카메라 시스템의 동영상촬영 유스케이스에 대하여, 먼저 명령어를 기다리는 아이들 상태(Idle state)가 지속되다 카메라시작(Startcamera) 이벤트가 발생하게 되면, 상기 아이들 상태(Idle state)에서 초기화 상태(Initialized state)가 되어, 휴대폰 카메라의 초기화 작업을 수행한다. 이어서, 이전보기 이벤트가 발생하는 경우에는 상기 초기화 상태(Initialized state)가 이전보기 상태(Preview state)로 변경되며, 이전보기 상태에서 이후보기, 녹화, 스냅샷 등과 같은 작업과정을 수행하며, 이러한 상태변화가 다이어그램으로 표현된다.

[0060] 이어서, 상기 상태 다이어그램 상에서 나타난 이벤트 및 작업과정을 수행하는 주체별로 나누고, 이를 다시 시계열 순으로 표현하는 시퀀스 다이어그램을 생성한다.

[0061] 도 6은 본 발명의 휴대폰 카메라 시스템의 동영상촬영 유스케이스에 대한 시퀀스 다이어그램을 나타낸 도면이다.

[0062] 도 6에 도시된 바와 같이, 이전보기 상태를 갖는 카메라컨트롤러 Cameracontroller가 사용자로부터 startRecord() 이벤트를 입력받으면, oper_set_state() 및 get_sensor_data() 명령어를 Camsensor IF에 호출하고, 상기 카메라컨트롤러는 oper_record()를 수행함에 따라, 이전보기상태에서 녹화상태로 상태가 변환되는 것을 알 수 있다. 이에 따라, 각각의 컴포넌트들이 어떠한 작업과정을 수행하고 있으며, 현재 상태가 어떠한지 용이하게 파악할 수 있다.

[0063] 이와 같이, 도 6의 시퀀스 다이어그램을 통해, 영상녹화 유스케이스를 수행하는 액터가 사용자, 캠센서 및 마이크임에 따라 총 3개임을 알 수 있으며, 이때 사용되는 오브젝트는 Camsensor IF 및 AudioController IF 인 총 2개인 것을 알 수 있다. 또한 이때 발생하는 메시지는 oper_set_state(), get_sensordata(), oper_record(), oper_take_movie(), set_audio_path(), get_audio_input_data(), check_free_space(), save_recorded_data(), get_temp_filename(), save_file(), stop_record(), restartPreview() 와 같이 총 12개인 것을 알 수 있다.

[0064] 이와 같이, 복수 개의 유스케이스에 대한 상태 다이어그램 및 시퀀스 다이어그램을 생성함에 따라, 복잡도연산 모듈이 앞서 확인한 상기 시퀀스 다이어그램에 포함된 적어도 하나의 정보를 통해 상기 복수 개의 유스케이스에

대한 각각의 복잡도를 연산한다.

[0065] 표 3은 본 발명의 휴대폰 카메라 시스템의 영상녹화 유스케이스에 대한 복잡도를 연산하여 복잡도연산값을 테이블화한 표이다.

[0066] [표 3]

[0067]

복잡도 요소	요소이름	형태	가중치
액터	사용자	-	1
	캠센서	-	1
	마이크	-	1
액터 복잡도= 1+1+1			3
오브젝트	CamsensorIF	Complex	2.0
	AudioControlIF	Average	1.5
오브젝트 복잡도= (1*2)+(1*1.5)			3.5
메시지	oper_set_state()	Asynchronous	1.0
	get_sensordata()	Synchronous	2.0
	oper_record()	Asynchronous	1.0
	oper_take_movie()	Asynchronous	1.0
	set_autdio_path()	Asynchronous	1.0
	get_audio_input_data()	Synchronous	2.0
	check_free_space()	Asynchronous	1.0
	save_recorded_data()	Asynchronous	1.0
	get_temp_filename()	Asynchronous	1.0
	save_file()	Asynchronous	1.0
	stop_record()	Asynchronous	1.0
	restartPreview()	Asynchronous	1.0
메시지 복잡도= (1*1)+(1*2)+(1*1)+(1*1)+(1*1)+(1*2)+(1*1)+(1*1) +(1*1)+(1*1)+(1*1)+(1*1)			14
유스케이스 복잡도= 20.5			20.5

[0068] 표 3에 도시된 바와 같이, 영상녹화 유스케이스에 대한 복잡도를 연산하기 위해, 먼저 액터에 대한 복잡도를 연산하면 상기 액터의 수가 3개임에 따라, 상기 액터의 복잡도 연산값은 3이된다. 또한, 오브젝트 중 Camsensor IF가 난이도 상에 해당하고, AudioControl IF는 난이도 중에 해당함에 따라, 상기 Camsensor IF는 2.0의 가중치를 가지며, 상기 AudioControl IF는 1.5의 가중치를 갖는다. 이에 따라, 상기 오브젝트의 복잡도 연산값은 상기 오브젝트의 수에 미리 할당된 가중치를 곱하여 합산한 값 즉, (1×2.0)+(1×1.5)으로 연산된 3.5가 된다.

[0069] 더불어, 상기 오브젝트의 수행에 따라 발생하는 메시지에 대한 복잡도는 동기화 여부에 따라 Asynchronous 또는 synchronous로 그 상태를 분류한다. 이때, 상기 Asynchronous는 가중치가 1.0이고, 상기 synchronous는 가중치가 2.0로 설정됨에 따라, 발생한 메시지의 수에 가중치를 곱한값을 모두 합산한 값 즉, (1×1.0)+(1×2.0)+(1×1.0)+(1×1.0)+(1×1.0)+(1×2.0)+(1×1.0)+(1×1.0)+(1×1.0)+(1×1.0)+(1×1.0)인 14가 상기 메시지에 대한 복잡도 연산값이 된다.

[0070] 결국, 영상녹화 유스케이스에 대한 복잡도 연산값은 앞서 연산한 액터의 복잡도 연산값 3과 오브젝트의 연산값 3.5 및 메시지의 연산값 14를 모두 합산한 값인 20.5가 된다.

[0071] 상술한 바와 같이, 휴대폰 카메라 시스템의 복수 개의 유스케이스에 대하여 각각의 복잡도를 연산하고, 이를 테이블화할 수 있다.

[0072] 표 4는 휴대폰 카메라 시스템의 각 유스케이스별 복잡도 연산값을 나타낸 표이다.

[0073] [표 4]

유스케이스 ID	유스케이스 이름	복잡도 연산값
U_1	이전보기	28
U_3	영상녹화	20.5
U_4	이후보기	18.5
U_6	앨범관리	16.5
U_8	사진영상전송	15
U_9	사진출력	14
U_5	영상재생	12.5
U_2	스냅샷 촬영	12
U_7	사진영상 편집	11
U_{10}	앨범보기	10

[0074]

[0075] 표 4에 도시된 바와 같이, 휴대폰 카메라 시스템의 복수 개의 유스케이스 중에서 이전보기 유스케이스, 영상녹화 유스케이스, 이후보기 유스케이스 등의 순으로 복잡도 연산값이 높은 것을 알 수 있다.

[0076] 이와 같이 연산한 각 유스케이스에 대한 상대적 만족도에 따른 복잡도 연산값을 복잡도 연산값으로 나누고, 나눈 값을 각 유스케이스에 대한 가치로서 판단하며, 이에 따라 휴대폰 카메라 시스템의 복수 개의 유스케이스를 가치 순으로 테이블화할 수 있다.

[0077] 표 5는 본 발명의 휴대폰 카메라 시스템의 복수 개의 유스케이스에 대한 보정 전 가치를 나타낸 표이다.

[0078] [표 5]

유스케이스 ID	유스케이스 이름	만족도 연산값(A)	복잡도 연산값(B)	유스케이스 가치 (A/B) * 100
U_2	스냅샷 촬영	24	12	200
U_5	영상재생	10	12.5	80
U_3	영상녹화	16	20.5	78
U_{10}	앨범보기	7	10	70
U_4	이후보기	11	18.5	59
U_7	사진영상 편집	5	11	45
U_6	앨범관리	7	16.5	42
U_8	사진영상 전송	6	15	40
U_1	이전보기	11	28	39
U_9	사진출력	4	14	29

[0079]

[0080] 표 5에 도시된 바와 같이, 스냅샷 촬영 유스케이스의 만족도 연산값이 24이고 복잡도 연산값이 12임에 따라 상기 스냅샷 촬영 유스케이스의 가치는 200이 되어, 휴대폰 카메라 시스템의 복수 개의 유스케이스 중 가장 높은 가치를 갖는 것을 알 수 있다. 또한 상기 스냅샷 촬영 유스케이스의 뒤를 이어 영상재생 유스케이스, 영상녹화 유스케이스, 앨범보기 유스케이스 등의 순으로 가치가 높은 것을 알 수 있다. 하지만 표 5에 나타난 복수 개의 유스케이스에 대한 가치는 보정을 수행하기 전의 상태를 나타낸 것으로, 추후 보정과정을 거치는 경우, 표 5에 기재된 각 유스케이스의 가치는 달라지게 된다.

[0081] 표 6은 본 발명의 휴대폰 카메라 시스템의 복수 개의 유스케이스에 대한 보정 후 가치를 나타낸 표이다.

[0082] [표 6]

유스케이스ID	유스케이스 이름	제 1 보정	제 2 보정	유스케이스 가치
U_2	스냅샷 촬영	Dependency (Not Related)	200*(1+0.2+0.3+0.2+0.3+0.1)	420
U_{10}	앨범보기	70*(1+0.1+0.1+0.1+0.1+0.1)	105*(1+0.0+0.3+0.1+0.1+0.3)	189
U_3	영상녹화	Dependency (Not Related)	78*(1+0.2+0.3+0.3+0.3+0.1)	172
U_1	이전보기	39*(1+0.1+0.1+0.1)	51*(1+0.3+0.3+0.3+0.3+0.3)	128
U_5	영상재생	Dependency (Not Related)	80*(1+0.0+0.3+0.0+0.0+0.0)	104
U_4	이후보기	Dependency (Not Related)	59*(1+0.0+0.1+0.0+0.0+0.0)	65
U_6	앨범관리	Dependency (Not Related)	42*(1+0.2+0.3+0.0+0.0+0.0)	63
U_7	사진영상 편집	Dependency (Not Related)	45*(1+0.0+0.1+0.0+0.0+0.0)	50
U_8	사진영상 전송	Dependency (Not Related)	40*(1+0.0+0.2+0.0+0.0+0.0)	48
U_9	사진출력	Dependency (Not Related)	29*(1+0.0+0.2+0.0+0.0+0.0)	35

[0083]

[0084] 표 6에 도시된 바와 같이, 휴대폰 카메라 시스템의 복수 개의 유스케이스 중 특정 유스케이스가 선행개발되어야만 다른 유스케이스의 개발이 가능한지 여부를 확인하여 상기 특정 유스케이스에 가중치를 부여하는 제1보정과정 또는 사용자에 의해 요구되는 완성도 수준에 따라 서로 다른 가중치를 부여하는 제2보정과정을 수행하고, 각 보정을 반영하는 상기 유스케이스에 대한 가치를 재연산한다.

[0085] 이와 같이, 복수 개의 각 유스케이스에 대하여 제1보정과정 또는 제2보정과정을 수행하고, 해당 보정과정을 각 유스케이스별 가치에 반영한 결과, 휴대폰 카메라 시스템의 다수의 유스케이스 중에서 스냅샷 촬영 유스케이스, 앨범보기 유스케이스, 영상녹화 유스케이스, 이전보기 유스케이스의 순으로 가치가 높은 것을 알 수 있다.

[0086] 즉, 앞서 표 5를 통해 알 수 있는 보정 전 가치가 높은 순서의 유스케이스와, 표 6을 알 수 있는 보정 후 가치가 높은 순서의 유스케이스를 비교하여 살펴보면, 가장 높은 가치를 갖는 유스케이스는 스냅샷촬영 유스케이스로 동일하지만, 그 뒤를 위어 두 번째로 높은 가치를 갖는 유스케이스는 보정전의 경우 영상재생 유스케이스인 반면, 보정 후 경우 앨범보기 유스케이스임을 알 수 있다.

[0087] 따라서, 복수 개의 유스케이스에 대하여 가치를 연산하고, 이에 더하여, 각 유스케이스별 보정과정이 수행되어 상기 가치에 반영되면, 보다 정확하고 사용자 요구에 적합한 각 유스케이스별 가치를 확인할 수 있게 된다.

[0088] 또한, 이러한 임베디드 시스템 내 유스케이스에 대한 가치연산방법은 컴퓨터로 실행하기 위한 프로그램이 기록된 컴퓨터 판독가능 기록매체에 저장될 수 있다. 이때, 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다. 컴퓨터가 읽을 수 있는 기록 장치의 예로는 ROM, RAM, CD-ROM, DVD±ROM, DVD-RAM, 자기 테이프, 플로피 디스크, 하드 디스크(hard disk), 광데이터 저장장치 등이 있다. 또한, 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 장치에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

[0089] 본 발명의 임베디드 시스템 내 유스케이스에 대한 가치연산방법은 사용자 만족도 및 유스케이스의 복잡도를 각각 연산한 후, 그 연산결과에 기초하여 유스케이스에 대한 가치를 연산함으로써, 다수의 유스케이스 중 가치가 높은 유스케이스를 용이하게 선별할 수 있는 효과가 있다.

[0090] 또한, 본 발명의 임베디드 시스템 내 유스케이스에 대한 가치연산방법은 사용자 만족도 및 유스케이스의 복잡도를 연산한 결과에 기초하여 가치가 가장 높은 유스케이스를 용이하게 판단함에 따라, 전체 임베디드 시스템의 질을 향상시킬 수 있는 효과가 있다.

[0091] 상기에서는 본 발명의 바람직한 실시 예에 대하여 설명하였지만, 본 발명은 이에 한정되는 것이 아니고 본 발명의 기술 사상 범위 내에서 여러 가지로 변형하여 실시하는 것이 가능하고 이 또한 첨부된 특허청구범위에 속하는 것은 당연하다.

부호의 설명

[0092] S110: 임베디드 시스템에 대한 모델링을 수행하여 복수 개의 유스케이스를 생성함

S120: 상기 복수 개의 유스케이스에 대한 상대적 만족도를 연산함

S130: 상기 복수 개의 유스케이스에 대한 상태 다이어그램 및 시퀀스 다이어그램을 생성함

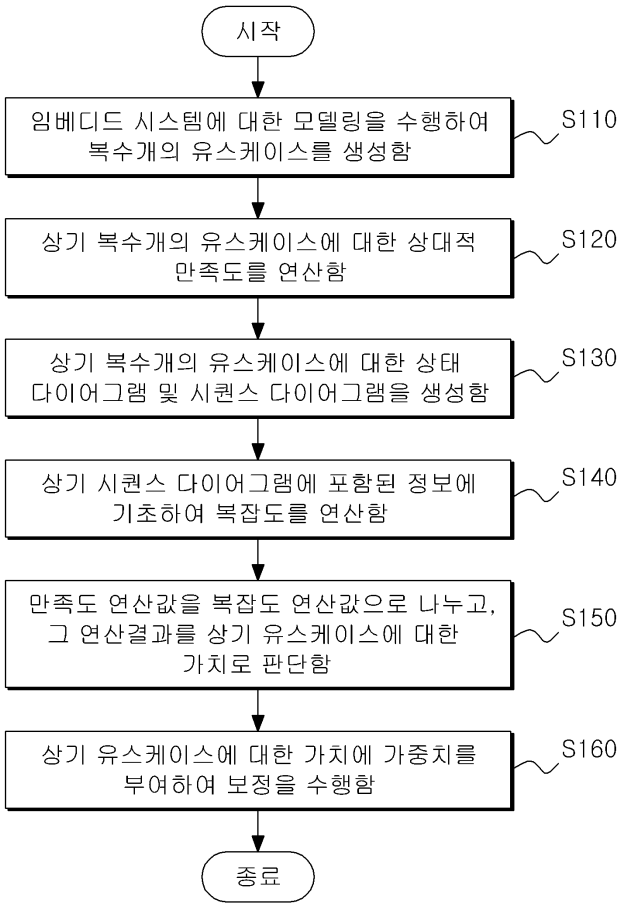
S140: 상기 시퀀스 다이어그램에 포함된 정보에 기초하여 복잡도를 연산함

S150: 만족도 연산값을 복잡도 연산값으로 나눈 값을 상기 유스케이스에 대한 가치로 판단함

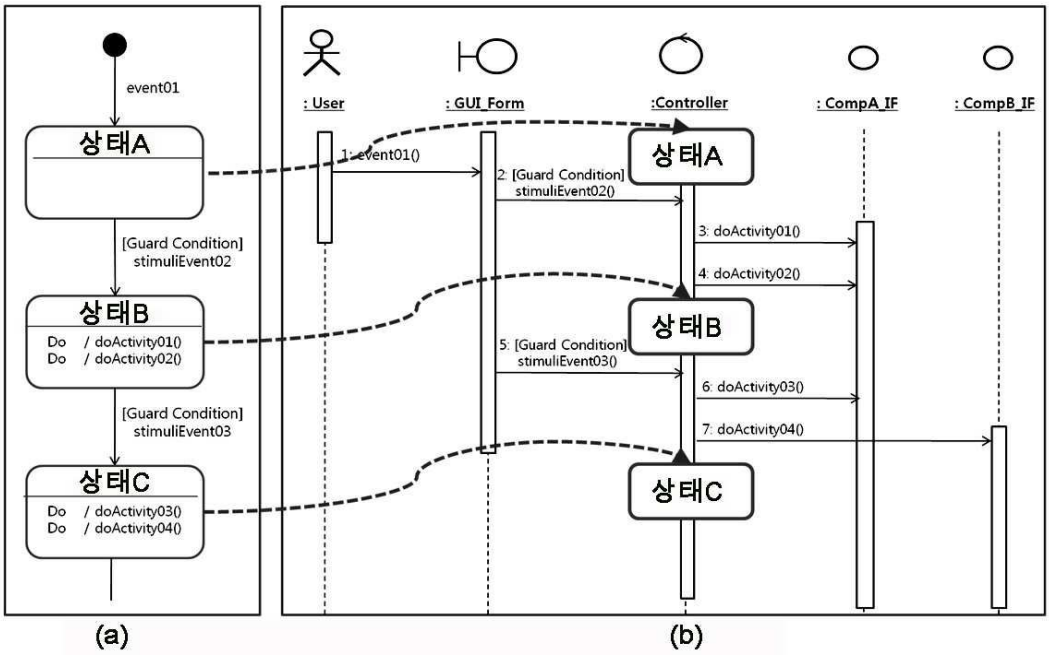
S160: 상기 유스케이스에 대한 가치에 가중치를 부여하여 보정을 수행함

도면

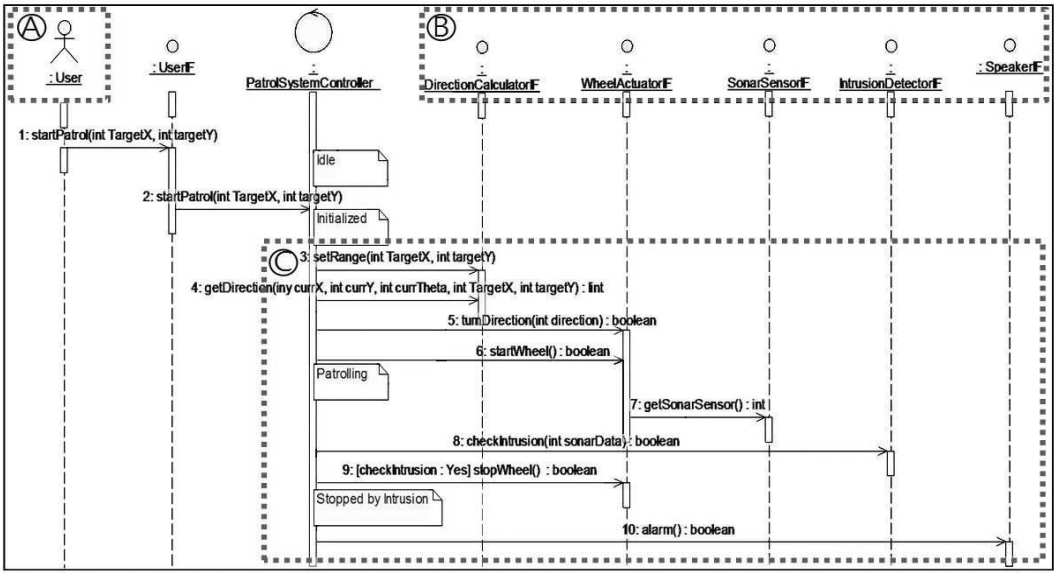
도면1



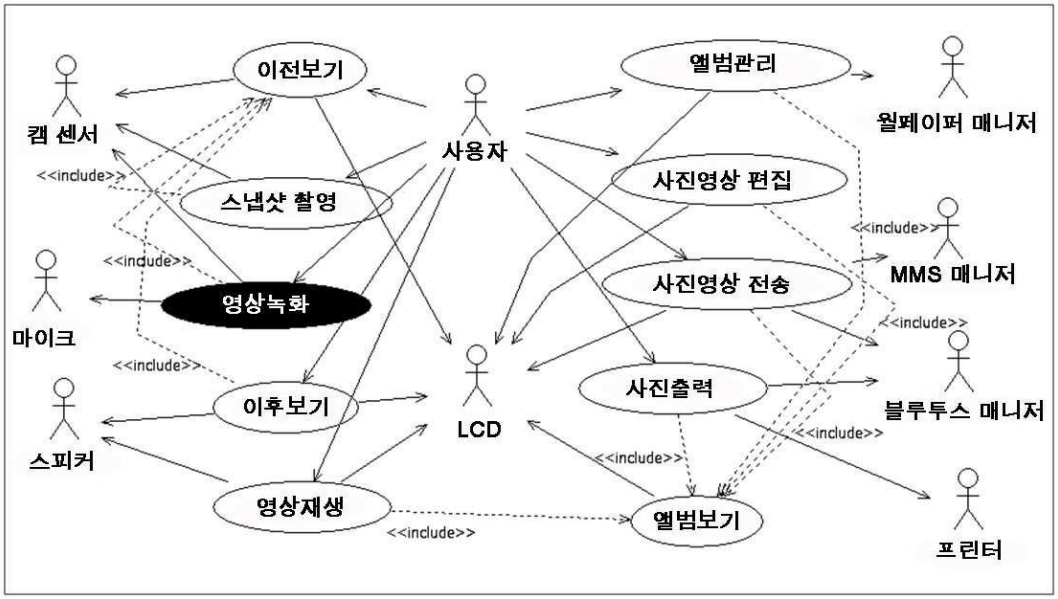
도면2


event01
상태A
[Guard Condition]
stimuliEvent02
상태B
Do / doActivity01()
Do / doActivity02()
[Guard Condition]
stimuliEvent03
상태C
Do / doActivity03()
Do / doActivity04()
(a)
: User
: GUI_Form
:Controller
: CompA_IF
: CompB_IF
1: event01()
2: [Guard Condition]
stimuliEvent02()
3: doActivity01()
4: doActivity02()
5: [Guard Condition]
stimuliEvent03()
6: doActivity03()
7: doActivity04()
(b)

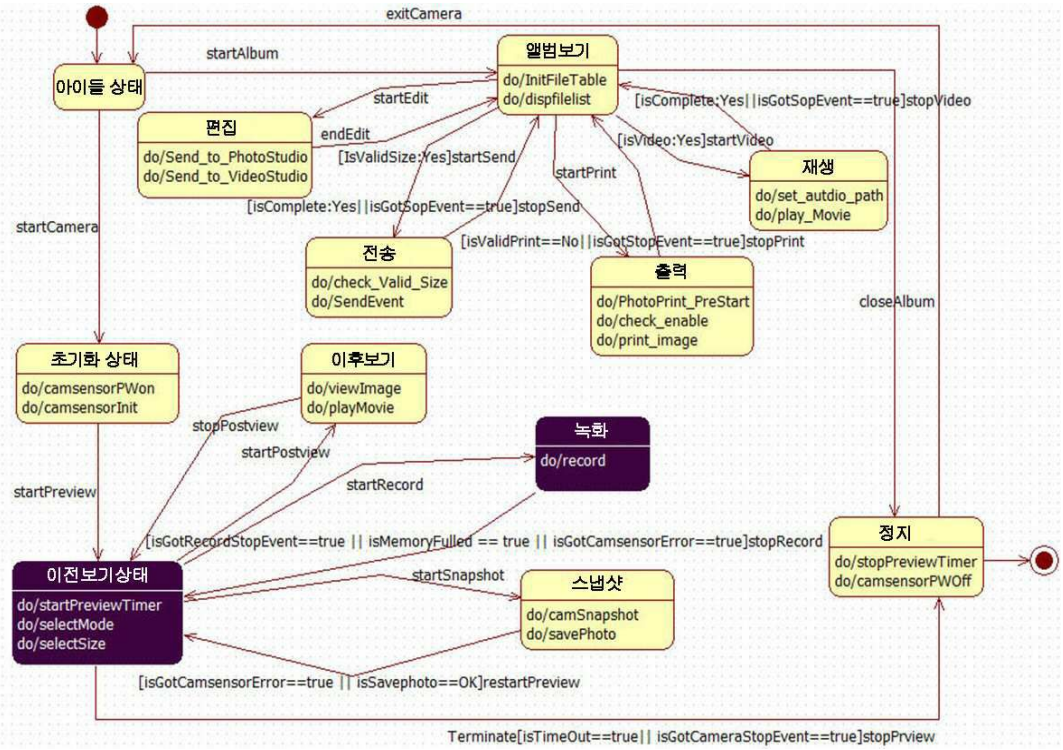
도면3


Ⓐ
: User
: UserIF
: PatrolSystemController
Ⓑ
: DirectionCalculatorIF
: WheelActuatorIF
: SonarSensorIF
: IntrusionDetectorIF
: SpeakerIF
1: startPatrol(int TargetX, int targetY)
Idle
2: startPatrol(int TargetX, int targetY)
Initialized
Ⓒ
3: setRange(int TargetX, int targetY)
4: getDirection(iny currX, int currY, int currTheta, int TargetX, int targetY) : int
5: turnDirection(int direction) : boolean
6: startWheel() : boolean
Patrolling
7: getSonarSensor() : int
8: checkIntrusion(int sonarData) : boolean
9: [checkIntrusion : Yes] stopWheel() : boolean
Stopped by Intrusion
10: alarm() : boolean

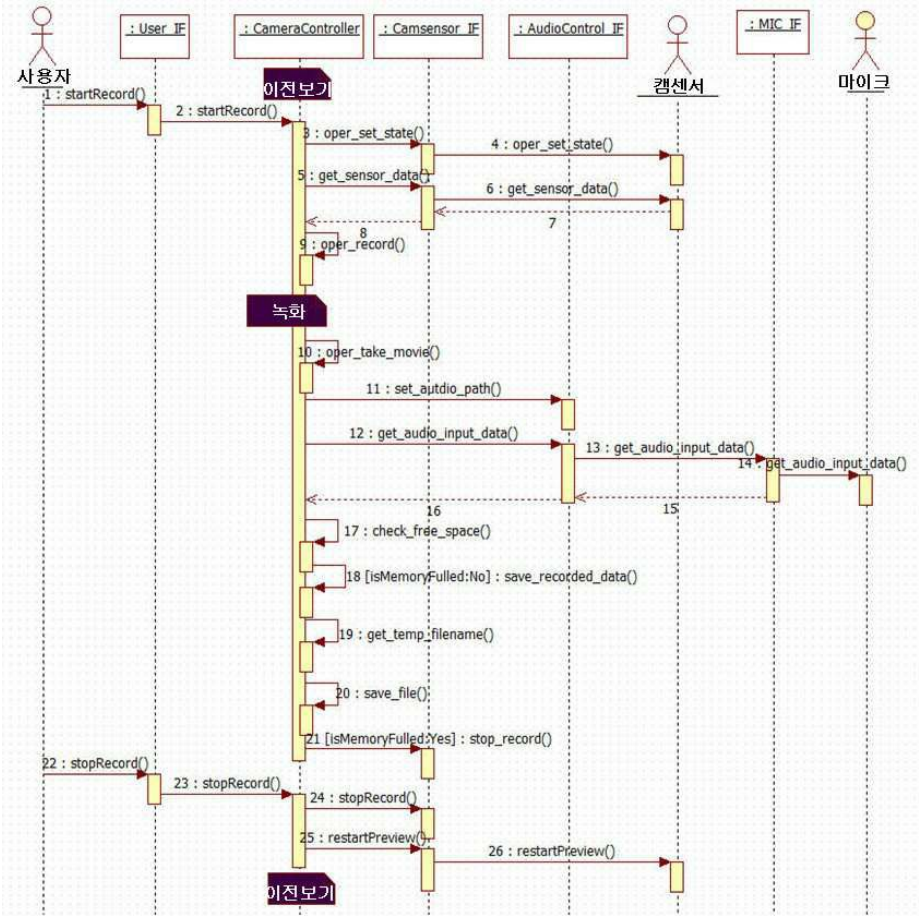
도면4


이전보기
앨범관리
캠 센서
사용자
윌페이퍼 매니저
<<include>>
스냅샷 촬영
사진영상 편집
<<include>>
<<include>>
MMS 매니저
사진영상 전송
마이크
영상녹화
<<include>>
<<include>>
사진출력
블루투스 매니저
이후보기
LCD
스피커
<<include>>
<<include>>
영상재생
앨범보기
<<include>>
프린터

도면5


exitCamera
startAlbum
앨범보기
do/InitFileTable
do/dispfilelist
아이들 상태
startEdit
[isComplete:Yes||isGotSopEvent==true]stopVideo
편집
endEdit
do/Send_to_PhotoStudio
do/Send_to_VideoStudio
[IsValidSize:Yes]startSend
[isVideo:Yes]startVideo
startPrint
재생
do/set_autdio_path
do/play_Movie
[isComplete:Yes||isGotSopEvent==true]stopSend
startCamera
[isValidPrint==No||isGotStopEvent==true]stopPrint
전송
do/check_Valid_Size
do/SendEvent
출력
do/PhotoPrint_PreStart
do/check_enable
do/print_image
closeAlbum
초기화 상태
do/camsensorPWon
do/camsensorInit
이후보기
do/viewImage
do/playMovie
stopPostview
녹화
do/record
startPostview
startPreview
startRecord
[isGotRecordStopEvent==true || isMemoryFulled == true || isGotCamsensorError==true]stopRecord
정지
do/stopPreviewTimer
do/camsensorPWOff
이전보기상태
do/startPreviewTimer
do/selectMode
do/selectSize
startSnapshot
스냅샷
do/camSnapshot
do/savePhoto
[isGotCamsensorError==true || isSavephoto==OK]restartPreview
Terminate[isTimeOut==true|| isGotCameraStopEvent==true]stopPrview

도면6


사용자
: User_IF
: CameraController
: Camsensor_IF
: AudioControl_IF
캠센서
: MIC_IF
마이크
이전보기
1 : startRecord()
2 : startRecord()
3 : oper_set_state()
4 : oper_set_state()
5 : get_sensor_data()
6 : get_sensor_data()
7
8
9 : oper_record()
녹화
10 : oper_take_movie()
11 : set_autdio_path()
12 : get_audio_input_data()
13 : get_audio_input_data()
14 : get_audio_input_data()
15
16
17 : check_free_space()
18 [isMemoryFulled:No] : save_recorded_data()
19 : get_temp_filename()
20 : save_file()
21 [isMemoryFulled:Yes] : stop_record()
22 : stopRecord()
23 : stopRecord()
24 : stopRecord()
25 : restartPreview()
26 : restartPreview()
이전보기