



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0108105

(43) 공개일자 2015년09월25일

(51) 국제특허분류(Int. Cl.)

G06F 9/44 (2006.01)

(21) 출원번호 10-2014-0030862

(22) 출원일자 2014년03월17일

심사청구일자 2014년03월17일

(71) 출원인

서강대학교산학협력단

서울특별시 마포구 백범로 35 (신수동, 서강대학교)

(72) 발명자

윤성혜

서울특별시 은평구 서오릉로25길 22 502호(구산동, 이안아파트)

박수진

서울특별시 마포구 독막로18길 5 102동 101호(상수동, 상수두산위브아파트)

(뒷면에 계속)

(74) 대리인

특허법인충현

전체 청구항 수 : 총 11 항

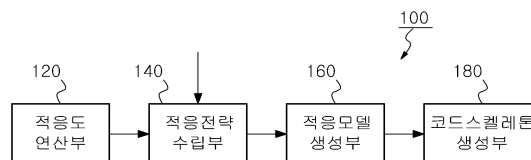
(54) 발명의 명칭 적응형 소프트웨어의 적응화 방법 및 시스템

(57) 요약

본 발명은 적응형 소프트웨어의 적응화 방법 및 시스템에 관한 것으로서, 보다 바람직하게는 적응도연산부가 소프트웨어의 소스코드 내 가변 요소가 존재하는지 검색하여 상기 소프트웨어에 대하여 상황 변화에 대처하는 능력을 나타내는 적응도를 연산하는 단계; 적응전략수립부가 검색된 상기 소프트웨어의 소스코드 내 가변 요소에 대응하는 수행과정을 나타내는 적응 전략을 수립하여 수립된 적응 전략에 해당하는 적응 패턴을 선택하는 단계; 적응모델생성부가 선택된 적응 패턴을 위빙(waving)하여 적응형 소프트웨어의 설계 모델인 적응 모델을 생성하는 단계; 코드 스켈레톤 생성부가 생성된 적응 모델에 기초하여 코드 스켈레톤을 생성하는 단계;를 포함한다.

이러한 구성에 의해, 본 발명의 적응형 소프트웨어의 적응화 방법 및 시스템은 모바일 환경에서 환경 변화 또는 사용자의 요구 사항을 인식하여 신속하고 능동적으로 대응하여 사용자 만족도를 향상시킬 수 있는 효과가 있다.

대표도 - 도1



(72) 발명자

이채영

경기도 안산시 단원구 광덕3로 201 314동1602호(고잔동, 안산고잔3차푸르지오)

박수용

서울특별시 마포구 독막로 98 101동1001호(상수동, 상수두산위브아파트)

이 발명을 지원한 국가연구개발사업

과제고유번호 2012M3C4A7033348

부처명 미래창조과학부

연구관리전문기관 한국연구재단

연구사업명 차세대정보컴퓨팅기술개발사업

연구과제명 동적 자가 적응을 위한 집단 지성 기반 컴포넌트 설계 및 검증 기술

기 여 율 1/1

주관기관 서강대학교

연구기간 2013.07.01 ~ 2014.06.30

명세서

청구범위

청구항 1

적응도연산부가 소프트웨어의 소스코드 내 가변 요소가 존재하는지 검색하여 상기 소프트웨어에 대하여 상황 변화에 대처하는 능력을 나타내는 적응도를 연산하는 단계;

적응전략수립부가 검색된 상기 소프트웨어의 소스코드 내 가변 요소에 대응하는 수행과정을 나타내는 적응 전략을 수립하여 수립된 적응 전략에 해당하는 적응 패턴을 선택하는 단계;

적응모델생성부가 선택된 적응 패턴을 위빙(weaving)하여 적응형 소프트웨어의 설계 모델인 적응 모델을 생성하는 단계; 및

코드 스켈레톤 생성부가 생성된 적응 모델에 기초하여 코드 스켈레톤을 생성하는 단계;

를 포함하는 적응형 소프트웨어의 적응화 방법.

청구항 2

제1항에 있어서,

상기 적응도연산부가 소프트웨어의 소스코드 내 가변 요소가 존재하는지 검색하여 적응도를 평가하는 단계는

상기 소프트웨어의 소스코드 내 적어도 하나의 가변 요소가 존재하는지 여부를 검색하는 과정; 및

상황 변화에 자가 적응하도록 기설정된 가변 요소의 개수를 상기 소프트웨어의 소스코드로부터 검색된 가변요소로 나누어 적응도를 연산하는 과정;

을 포함하는 것을 특징으로 하는 적응형 소프트웨어의 적응화 방법.

청구항 3

제1항에 있어서,

상기 가변 요소는

프레임워크로부터 입력받은 이벤트 또는 상기 프레임워크로부터 입력받은 이벤트에 대한 대상 객체들과 관련된 API(Application Programming Interface)를 나타내는 것을 특징으로 하는 적응형 소프트웨어의 적응화 방법.

청구항 4

제1항에 있어서,

상기 적응전략수립부가 적응 전략을 수립하여 수립된 적응 전략에 해당하는 적응 패턴을 선택하는 단계는

상기 소프트웨어의 소스 코드 내 가변 요소가 검색된 지점의 전후 지점에서 상기 소프트웨어가 상황 변화에 따라 적응해야 하는지 여부를 입력받는 과정; 및

적응하고자 하는 가변 요소의 종류에 따라 각각 대응하도록 기설정된 복수 개의 적응 패턴 중 검색된 가변 요소에 대응하는 적어도 하나의 적응 패턴을 선택하는 과정;

을 포함하는 것을 특징으로 하는 적응형 소프트웨어의 적응화 방법.

청구항 5

제1항에 있어서,

상기 적응모델생성부가 선택된 적응 패턴을 위빙하여 적응형 소프트웨어의 설계 모델인 적응 모델을 생성하는 단계는

역공학 기법에 기초하여 소스 모델을 생성하는 과정;

선택된 적응 패턴의 참여단말들과 상기 소스 모델의 참여단말들을 질의 응답을 통해 서로 연결하거나, 새로운 참여단말과 연결하는 과정; 및

상기 적응 패턴을 상기 적응 패턴과 연결된 소스 모델에 적용하여 적응 모델로 생성하는 과정;

을 포함하는 것을 특징으로 하는 적응형 소프트웨어의 적응화 방법.

청구항 6

제1항 내지 제5항 중 어느 한 항에 따른 방법을 컴퓨터로 실행하기 위한 프로그램이 기록된 컴퓨터 판독가능 기록매체.

청구항 7

소프트웨어의 소스코드 내 가변 요소가 존재하는지 검색하여 상기 소프트웨어에 대하여 상황 변화에 대처하는 능력을 나타내는 적응도를 연산하는 적응도연산부;

검색된 상기 소프트웨어의 소스 코드 내 가변 요소에 대응하는 수행과정을 나타내는 적응 전략을 수립하여 수립된 적응전략에 해당하는 적응 패턴을 선택하는 적응전략수립부;

선택된 적응 패턴을 위빙(weaving)하여 적응형 소프트웨어의 설계 모델인 적응 모델을 생성하는 적응모델생성부; 및

생성된 적응 모델에 기초하여 코드 스켈레톤을 생성하는 코드 스켈레톤 생성부;

를 포함하는 적응형 소프트웨어의 적응화 시스템.

청구항 8

제7항에 있어서,

상기 적응도연산부는

상기 소프트웨어의 소스코드 내 적어도 하나의 가변 요소가 존재하는지 여부를 검색하고, 상황 변화에 자가 적응하도록 기설정된 가변 요소의 개수를 상기 소프트웨어의 소스코드로부터 검색된 가변요소로 나누어 적응도를 연산하는 것을 포함하는 것을 특징으로 하는 적응형 소프트웨어의 적응화 시스템.

청구항 9

제7항에 있어서,

상기 가변 요소는

프레임워크로부터 입력받은 이벤트 또는 상기 프레임워크로부터 입력받은 이벤트에 대한 대상 객체들과 관련된 API(Application Programming Interface)를 나타내는 것을 특징으로 하는 적응형 소프트웨어의 적응화 시스템.

청구항 10

제7항에 있어서,

상기 적응전략수립부는

상기 소프트웨어의 소스 코드 내 가변 요소가 검색된 지점의 전후 지점에서 상기 소프트웨어가 상황 변화에 따라 적응해야 하는지 여부를 입력받고, 적응하고자 하는 가변 요소의 종류에 따라 각각 대응하도록 기설정된 복수 개의 적응 패턴 중 검색된 가변 요소에 대응하는 적어도 하나의 적응 패턴을 선택하는 것을 특징으로 하는 적응형 소프트웨어의 적응화 시스템.

청구항 11

제7항에 있어서,

상기 적응모델생성부는

역공학 기법에 기초하여 소스 모델을 생성하고, 선택된 적응 패턴의 참여단말들과 상기 소스 모델의 참여단말들을 질의 응답을 통해 서로 연결하거나, 새로운 참여단말과 연결하며, 상기 적응 패턴을 상기 적응 패턴과 연결된 적응 모델로 생성하는 것을 특징으로 하는 적응형 소프트웨어의 적응화 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 적응형 소프트웨어의 적응화 방법 및 시스템에 관한 것으로, 특히 사용자 요구 및 환경 변화에 따라 능동적으로 소프트웨어의 적응성을 향상시킬 수 있는 적응형 소프트웨어의 적응화 방법 및 시스템에 관한 것이다.

배경 기술

[0002] 최근 들어, 휴대 단말의 사용이 남녀노소를 불문하고 널리 보편화됨에 따라, 상기 휴대 단말에서 실행되는 모바일 어플리케이션에 대하여 사용자의 요구사항 또한 점차 다양화되고 있다.

[0003] 이러한 모바일 어플리케이션의 종류와 복잡도는 스마트폰의 등장으로 인하여 급격히 증가하는 추세에 있으며, 다양한 서비스들이 모바일 어플리케이션의 형태로 제공되고 있다. 스마트폰 사용자의 요구사항은 기능적인 관점뿐만 아니라 다양한 품질 요소 또한 요구되고 있으며, 이와 같이 요구되는 품질 요소 중 하나는 사용자가 처한 상황에 따라 적절한 서비스를 제공하는 상황 인식(ambient awareness) 기반의 서비스이다. 예를 들어, 가맹점에 대한 검색을 제공하는 모바일 어플리케이션을 살펴보면, 가나다순에 의한 검색 결과와, 사용자의 현재 위치를 기반으로 가까운 지점을 우선하여 검색한 결과를 놓고 보면, 사용자의 현재 위치를 기반으로 가까운 지점을 우선 검색한 결과를 알려주는 것이 사용자의 품질의 만족도를 향상시킬 수 있다. 이처럼, 사용자에게 제공되는 품질의 향상을 위해 사용자가 처한 상황을 먼저 인지하고, 인지된 상황을 활용하여 차별화된 서비스를 지원하기 위해서는 사용자 단말에 구비되는 소프트웨어가 발생하는 상황 변화에 능동적으로 대처할 능력을 갖추어야 하는데, 이러한 소프트웨어를 적응형 소프트웨어(adaptive software)라고 한다.

[0004] 이러한 적응형 소프트웨어는 적응 행위(behavior)의 추가 또는 변경을 용이하게 하기 위해, 다양한 적응 메커니즘을 사용하여 적응 상황에 대한 조건과 행위를 공통모델로부터 분리한 형태를 가진다. 또한, 지속적으로 주변 환경의 상태를 모니터링하고, 적응하기 위한 적응 프레임워크(adaptation framework)를 제공한다.

[0005] 하지만 기존의 모바일 어플리케이션은 하나의 로직 내에서 사용자에게 주어지는 상황을 인식하고 그에 따른 적응 행위를 실행함에 따라, 다양한 환경에 따른 변화를 수렴하기 어려운 문제점이 발생했다. 예를 들어, 앞서 기재한 바와 같이, 위치 기반의 검색 서비스를 제공하는 것과 더불어, 사용자단말이 이동 중인 위치에서 가까운 지점에서 주회하거나 발급하는 행사 또는 쿠폰을 푸쉬(push)하는 서비스를 추가하고자 하면, 위치를 인식하는 로직이 포함된 모듈을 수정한 후 다시 배포해야 하는데, 이러한 경우 발생한 상황 변화를 빠르게 인식하고, 능동적으로 변화에 대응하기 어려운 문제점이 발생했다.

선행기술문헌

특허문헌

- [0006] (특허문헌 0001) KR 10-1998-037148 (컴퓨터 시스템에서의 적응 어플리케이션의 구성방법, 삼성전자 주식회사) 1998.08.05.

발명의 내용

해결하려는 과제

- [0007] 상기와 같은 종래 기술의 문제점을 해결하기 위해, 본 발명은 모바일 환경에서 발생하는 다양한 변화 또는 사용자 요구사항을 신속하게 인지하고 수립하여 적응적으로 구동할 수 있는 적응형 소프트웨어의 적응화 방법 및 시스템을 제공하고자 한다.

과제의 해결 수단

- [0008] 위와 같은 과제를 해결하기 위한 본 발명의 한 실시 예에 따른 적응형 소프트웨어의 적응화 방법은 적응도연산부가 소프트웨어의 소스코드 내 가변 요소가 존재하는지 검색하여 상기 소프트웨어에 대하여 상황 변화에 대처하는 능력을 나타내는 적응도를 평가하는 단계; 적응전략수립부가 검색된 상기 소프트웨어의 소스코드 내 가변 요소에 대응하는 수행과정을 나타내는 적응 전략을 수립하여 수립된 적응 전략에 해당하는 적응 패턴을 선택하는 단계; 적응모델생성부가 선택된 적응 패턴을 위빙(weaving)하여 적응형 소프트웨어의 설계 모델인 적응 모델을 생성하는 단계; 코드 스켈레톤 생성부가 생성된 적응 모델에 기초하여 코드 스켈레톤을 생성한 후, 코드를 구현하는 단계;를 포함한다.
- [0009] 보다 바람직하게는 상기 소프트웨어의 소스코드 내 적어도 하나의 가변 요소가 존재하는지 여부를 검색하는 과정; 및 상황 변화에 자가 적응하도록 기설정된 가변 요소의 개수를 상기 소프트웨어의 소스코드로부터 검색된 가변요소로 나누어 백분율 연산을 통해 적응도를 연산하는 과정; 을 포함하는 적응도연산부가 소프트웨어의 소스코드 내 가변 요소가 존재하는지 검색하여 적응도를 평가하는 단계를 포함할 수 있다.
- [0010] 특히, 프레임워크로부터 입력받은 이벤트 및 상기 프레임워크로부터 입력받은 이벤트에 대한 대상 객체들과 관련된 API 중 적어도 하나를 나타내는 가변 요소를 포함할 수 있다.
- [0011] 보다 바람직하게는 상기 소프트웨어의 소스 코드 내 가변 요소가 검색된 지점의 전후 지점에서 상기 소프트웨어가 상황 변화에 따라 적응해야 하는지 여부를 입력받는 과정; 및 적응하고자 하는 가변 요소의 종류에 따라 각각 대응하도록 기설정된 복수 개의 적응 패턴 중 검색된 가변 요소에 대응하는 적어도 하나의 적응 패턴을 선택하는 과정; 을 포함하는 적응전략수립부가 적응 전략을 수립하여 수립된 적응 전략에 해당하는 적응 패턴을 선택하는 단계를 포함할 수 있다.
- [0012] 보다 바람직하게는 역공학 기법에 기초하여 소스 모델을 생성하는 과정; 선택된 적응 패턴의 참여단말들과 상기 소스 모델의 참여단말들을 질의 응답을 통해 서로 연결하거나, 새로운 참여단말과 연결하는 과정; 및 상기 적응 패턴을 상기 적응 패턴과 연결된 소스 모델에 적용하여 적응 모델로 생성하는 과정; 을 포함하는 적응모델생성부가 선택된 적응 패턴을 위빙하여 적응형 소프트웨어의 설계 모델인 적응 모델을 생성하는 단계를 포함할 수 있다.
- [0013] 위와 같은 과제를 해결하기 위한 본 발명의 다른 실시 예에 따른 적응형 소프트웨어의 적응화 시스템은 소프트웨어의 소스코드 내 가변 요소가 존재하는지 검색하여 상기 소프트웨어에 대하여 상황 변화에 대처하는 능력을 나타내는 적응도를 평가하는 적응도연산부; 검색된 상기 소프트웨어의 소스 코드 내 가변 요소에 대응하는 수행과정을 나타내는 적응 전략을 수립하여 수립된 적응전략에 해당하는 적응 패턴을 선택하는 적응전략수립부; 선택된 적응 패턴을 위빙(weaving)하여 적응형 소프트웨어의 설계 모델인 적응 모델을 생성하는 적응모델생성부; 및 생성된 적응 모델에 기초하여 코드 스켈레톤을 생성한 후, 코드를 구현하는 코드 스켈레톤 생성부;를 포함한

다.

- [0014] 보다 바람직하게는 상기 소프트웨어의 소스코드 내 적어도 하나의 가변 요소가 존재하는지 여부를 검색하고, 상황 변화에 자가 적응하도록 기설정된 가변 요소의 개수를 상기 소프트웨어의 소스코드로부터 검색된 가변요소로 나누어 적응도를 연산하는 적응도연산부를 포함한다.
- [0015] 특히, 프레임워크로부터 입력받은 이벤트 및 상기 프레임워크로부터 입력받은 이벤트에 대한 대상 객체들과 관련된 API 중 적어도 하나를 나타내는 가변 요소를 포함할 수 있다.
- [0016] 보다 바람직하게는 상기 소프트웨어의 소스 코드 내 가변 요소가 검색된 지점의 전후 지점에서 상기 소프트웨어가 상황 변화에 따라 적응해야 하는지 여부를 입력받고, 적응하고자 하는 가변 요소의 종류에 따라 각각 대응하도록 기설정된 복수 개의 적응 패턴 중 검색된 가변 요소에 대응하는 적어도 하나의 적응 패턴을 선택하는 적응 전략수립부를 포함할 수 있다.
- [0017] 보다 바람직하게는 역공학 기법에 기초하여 소스 모델을 생성하고, 선택된 적응 패턴의 참여단말들과 상기 소스 모델의 참여단말들을 질의 응답을 통해 서로 연결하거나, 새로운 참여단말과 연결하며, 상기 적응 패턴을 상기 적응 패턴과 연결된 적응 모델로 생성하는 적응모델생성부를 포함할 수 있다.

발명의 효과

- [0018] 본 발명의 적응형 소프트웨어의 적응화 방법 및 시스템은 모바일 환경에서 환경 변화 또는 사용자의 요구 사항을 인식하여 신속하고 능동적으로 대응하여 사용자 만족도를 향상시킬 수 있는 효과가 있다.
- [0019] 또한 본 발명의 적응형 소프트웨어의 적응화 방법 및 시스템은 미리 예상하지 못한 주변 상황의 변화가 발생하는 경우에 상기 소프트웨어가 저장된 모듈을 수집하여, 상기 소프트웨어의 주변 상황 변화에 따른 적응패턴을 포함시킨 후, 다시 모듈에 저장하여 재배포하지 않고, 소프트웨어가 주변 환경 변화에 능동적으로 적응함에 따라 신속하게 변화에 대응할 수 있는 효과가 있다.

도면의 간단한 설명

- [0020] 도 1은 본 발명의 일 실시 예에 따른 적응형 소프트웨어의 적응화 시스템의 블록도이다.
- 도 2는 본 발명의 다른 실시 예에 따른 적응형 소프트웨어의 적응화 방법의 순서도이다.
- 도 3은 소스 코드의 일 예를 나타낸 도면이다.
- 도 4는 배터리 잔여량을 모니터링한 시퀀스 다이어그램이다.

발명을 실시하기 위한 구체적인 내용

- [0021] 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.
- [0022] 이하, 도 1을 참조하여 본 발명에 따른 적응형 소프트웨어의 적응화 시스템에 대하여 자세히 살펴보도록 한다.
- [0023] 도 1은 본 발명의 일 실시 예에 따른 적응형 소프트웨어의 적응화 시스템의 블록도이다.
- [0024] 도 1에 도시된 바와 같이, 본 발명의 적응형 소프트웨어의 적응화 시스템(100)은 적응도연산부(120), 적응전략수립부(140), 적응모델생성부(160) 및 코드 스켈레톤 생성부(180)를 포함한다.
- [0025] 적응도연산부(120)는 적응도를 판단하고자 하는 소프트웨어의 소스코드 내 가변 요소가 존재하는지 검색하여 상기 소프트웨어에 대하여 상황 변화에 대처하는 능력을 나타내는 적응도를 연산한다. 이러한 적응도연산부(120)는 먼저, 상기 소프트웨어의 소스코드 내 적어도 하나의 가변 요소가 존재하는지 여부를 검색하고, 상황 변화에 자가 적응하도록 기설정된 가변 요소의 개수를 상기 소프트웨어의 소스코드로부터 검색된 가변요소로 나누어 적응도를 연산한다. 이때, 상기 가변요소는 프레임워크로부터 입력받은 이벤트 또는 프레임워크로부터 입력받은 이벤트의 대상 객체들과 관련된 API(Application Programming Interface)를 나타낼 수 있다.

- [0026] 적응전략수립부(140)는 검색된 상기 소프트웨어의 소스 코드 내 가변 요소에 대응하는 수행과정을 나타내는 적응 전략을 수립하고, 수립된 적응전략에 해당하는 적응 패턴을 선택한다. 이러한 적응전략수립부(140)는 상기 소프트웨어의 소스 코드 내 가변 요소가 검색된 지점의 전후 지점에서 상기 소프트웨어가 상황 변화에 따라 적응해야 하는지 여부를 입력받고, 적응해야한다는 신호를 입력받으면 적응하고자 하는 가변 요소의 종류에 따라 각각 대응하도록 기설정된 복수 개의 적응 패턴 중 검색된 가변 요소에 대응하는 적어도 하나의 적응 패턴을 선택할 수 있다.
- [0027] 적응모델생성부(160)는 선택된 적응 패턴을 위빙(waving)하여 적응형 소프트웨어의 설계 모델인 적응 모델을 생성한다. 이러한 적응모델생성부(160)는 기계어 코드로부터 원시 소스 코드로 변환하는 기술을 나타내는 역공학 기법에 기초하여 소스 모델을 생성하고, 선택된 적응 패턴의 참여단말들과 상기 소스 모델의 참여단말들을 질의 응답을 통해 서로 연결하거나, 새로운 참여단말과 연결하는 위빙을 통해 상기 적응 패턴을 상기 적응 패턴과 연결된 적응 모델로 생성할 수 있다.
- [0028] 코드 스케leton 생성부(180)는 생성된 적응 모델에 기초하여 코드 스케leton을 생성한다. 이후, 생성된 코드를 구현함으로써, 소프트웨어가 환경 변화에 능동적으로 적응화된다.
- [0029] 이하, 도 2를 참조하여 본 발명의 다른 실시 예에 따른 적응형 소프트웨어의 적응화 방법에 대하여 자세히 살펴 보도록 한다.
- [0030] 도 2는 본 발명의 다른 실시 예에 따른 적응형 소프트웨어의 적응화 방법의 순서도이다.
- [0031] 도 2에 도시된 바와 같이, 본 발명의 적응형 소프트웨어의 적응화 방법은 먼저, 적응도연산부(120)가 소프트웨어의 소스코드 내 가변 요소가 존재하는지 검색하여 상기 소프트웨어에 대하여 상황 변화에 대처하는 능력을 나타내는 적응도를 연산한다(S210). 이러한 적응도 연산과정을 자세히 살펴보도록 한다. 먼저, 상기 적응도연산부(120)가 상기 소프트웨어의 소스코드 내 적어도 하나의 가변 요소가 존재하는지 여부를 검색한다. 이때, 상기 가변 요소는 프레임워크로부터 입력받은 이벤트 또는 상기 프레임워크로부터 입력받은 이벤트에 대한 대상 객체들과 관련된 API(Application Programming Interface)를 나타낼 수 있다.
- [0032] 이후, 상기 적응도연산부(120)가 하기의 수학적 식 1에 기재된 바와 같이, 상황 변화에 자가 적응하도록 기설정된 가변 요소의 개수를 상기 소프트웨어의 소스코드로부터 검색된 가변요소로 나누어 적응도를 연산한다.
- [0033] [수학적 식 1]

$$\text{적응도}(\%) = \frac{\text{자가 적응 가능한 가변성 개수}}{\text{소스 코드에서 감지된 가변성 개수}} \times 100$$

- [0034]
- [0035] 이러한 적응도는 해당 소프트웨어가 얼마나 상황에 적합하게 소프트웨어의 동작을 변경하는지를 나타내는 지표이므로, 적응도가 높으면 주어지는 상황이 변화할 때 대처할 수 있는 능력이 높다는 것을 나타낸다.
- [0036] 이러한 가변 요소는 해당 소프트웨어의 도메인이나 개발 환경에 의존적이다. 안드로이드 기반의 모바일 소프트웨어의 경우, 안드로이드 프레임워크가 소프트웨어로 주는 이벤트, 안드로이드 프레임워크로부터 받은 이벤트 대상 객체들과 관련된 API들 등이 가변 요소에 포함된다. 예를 들면, GPS 상태 변경을 기다리는 Location Listener의 onLocationChanged() 함수나 GPS가 시작될 때 발생하는 GPS_EVENT_STARTED 이벤트 등이 가변 요소로 검색될 수 있다. 이처럼, 소스코드에서 감지된 가변 요소 중 이미 상황변화에 자가 적응하도록 구현된 가변 요소의 백분율을 적응도로 산출해 낸다.
- [0037] 특히, 소스 코드에서 가변 요소로 감지되었으나, 자가 적응 가능한 가변성이 아닌 경우, 적응형 메커니즘 구현이 필요한 적응화 가능한 가변 요소 (adaptizable variation)로 검색된다.
- [0038] 예를 들어, GPS로 사용자단말의 현재 위치를 고려하여 상기 사용자단말을 사용하는 사용자가 회의 시작 시각 5분 전에 회의실에 도착할 수 있도록 알림을 주는 기능에 대하여 도 3을 참조하여 살펴보도록 한다.
- [0039] 도 3은 소스 코드의 일 예를 나타낸 도면이다.
- [0040] 도 3에 도시된 바와 같이, 적응도연산부(120)가 소스코드를 탐색하여 장소를 조회하는 안드로이드 API인 locationManager.getLastKnownLocation()와 locationManager.getBestProvider()를 자동 식별하여 추천하고, 이러한 API들의 결과값을 이용하여 조건문을 수행하는 로직을 검토한다.

- [0041] 적응전략수립부(140)가 검색된 상기 소프트웨어의 소스코드 내 가변 요소에 대응하는 수행과정을 나타내는 적응 전략을 수립하고, 수립된 적응 전략에 해당하는 적응 패턴을 선택한다(S220). 이때, 적응 전략 수립이란, 가변 요소에 의한 행위들이 어떤 방식에 의해 동적으로 수행될지를 결정하는 것을 나타낸다. 이러한 적응 패턴의 선택과정을 살펴보면, 먼저, 상기 소프트웨어의 소스 코드 내 가변 요소가 검색된 지점의 전후 지점에서 상기 소프트웨어가 상황 변화에 따라 적응해야 하는지 여부를 입력받는다.
- [0042] 상기 적응전략수립부(140)가 가변 요소가 검색된 지점의 전후 지점에서 상기 소프트웨어가 상황 변화에 따라 적응해야 한다는 내용을 포함하는 신호를 입력받으면, 이후, 적응하고자 하는 가변 요소의 종류에 따라 각각 대응하도록 미리 설정되어 저장된 복수 개의 적응 패턴 중 검색된 가변 요소에 대응하는 적어도 하나의 적응 패턴을 선택한다.
- [0043] 이어서, 적응시키고자 하는 가변 요소의 종류 및 행위의 독립성 여부 등에 따라 달라지는 구현 메커니즘을 결정한다. 이를 위해서 분석된 가변 요소에 적합한 적응 패턴들을 선택하는 것만으로 전략을 수립할 수 있게 돕는다. 이하, 적응형 모바일 소프트웨어의 구현 시 필수적으로 고려되어야 하는 상황들 중 일부를 패턴으로 정의한 예를 살펴보도록 한다.
- [0044] 먼저, 적응 패턴이 실패 투명성 보장인 경우에는, 소프트웨어가 사용하는 외부 컴포넌트가 비활성화 상태이거나, 오류로 인하여 제대로된 동작이 불가능한 경우에도 소프트웨어의 동작에 영향을 주지 않도록 예외 처리코드를 삽입하여 동작을 완료할 수 있도록 한다.
- [0045] 또는 적응 패턴이 예외 보증을 하지 않는 안전성 확보인 경우에는, 소프트웨어가 사용하는 외부 컴포넌트에서 오류가 발생하면 동작을 중단하고, 사용자에게 이를 알림으로써, 예상치 못한 부작용을 방지한다. 예를 들면, 접근 제어 컴포넌트 또는 보안 컴포넌트 등이 될 수 있다.
- [0046] 또한 적응 패턴이 리소스 절약을 위한 대체 적응 행위 선택인 경우에는, 모바일 환경에서는 리소스가 제한적이므로, 제한적 리소스로 인하여 소프트웨어가 동작할 수 없는 것보다는 제한적 리소스 내에서 가장 근접하게 동작할 수 있는 대체 적응 행위를 선택할 수 있어야 한다. 따라서, 지속적으로 리소스를 모니터링하고, 리소스에 따라 행위를 결정하는 행위가 이루어져야 한다. 예를 들면, 배터리 잔여량이 최소값 이하가 되면, GPS 대신에 WLAN의 위치 정보로 대신하는 경우 등이 될 수 있다.
- [0047] 예를 들어, 상술한 GPS로 사용자단말의 현재 위치를 고려하여, 상기 사용자단말을 사용하는 사용자가 회의 시작 5분 전에 회의실에 도착할 수 있도록 알림을 주는 기능에 더하여, 위치 정보의 정확도를 높이기 위해 차량의 GPS가 블루투스를 통해 연동되는 경우에 해당 정보를 사용하는 요구사항이 추가된다고 가정한다.
- [0048] 이러한 요구사항을 분석한 결과, 알람이 푸쉬되는 서비스는 향후 조건도 다양해질 가능성이 높으며, 해당 조건에 대한 센서나 컴포넌트 또한 다양해질 것으로 예측된다. 이러한 가변 요소에 대해 선택된 적응 전략은 첫째, 센서나 외부 컴포넌트의 활성화 여부나 오동작에 대해 Fail-Safe 해야하고, 둘째, 배터리 잔여량이 일정 수준 이하가 되면, 배터리 소모가 많은 센서는 사용하지 않는다는 것이다.
- [0049] 이와 같이, 수립된 적응 전략을 만족시키기 위해, 적응전략수립부가 이에 해당하는 다양한 적응패턴 중, 실패의 투명성을 보장하는 적응패턴과, 리소스 절약을 위한 대체 적응 행위 선택 적응 패턴을 선택한다.
- [0050] 이에 따라, 적응모델생성부(160)가 선택된 적응 패턴을 위빙(waving)하여 적응형 소프트웨어의 설계 모델인 적응 모델을 생성한다(S230). 이를 위해, 먼저 기계어 코드로부터 원시 소스코드로 변환하는 기술을 나타내는 역공학 기법에 기초하여 소스 모델을 생성한다.
- [0051] 이어서, 선택된 적응 패턴의 참여단말들과 상기 소스 모델의 참여단말들을 질의 응답을 통해 서로 연결하거나, 새로운 참여단말과 연결한다.
- [0052] 이후, 상기 적응 패턴을 상기 적응 패턴과 연결된 소스 모델에 적용하여 적응 모델로 생성한다.
- [0053] 보다 구체적인 과정을 예로 들면, 기존 소프트웨어의 적응모델이 존재하지 않는다면, 역공학 기법을 사용하여 소스 모델을 생성해준다. 이때, 상기 소스 모델은 UML 다이어그램으로 표현되며, 소프트웨어의 정적인 구조를 파악할 수 있는 클래스 다이어그램과 소프트웨어의 특정 동작들이 수행되는 행위(interaction)를 표현하는 시퀀스 다이어그램이 각각 생성된다.
- [0054] 이어서, 적응 패턴의 참여자들(participants)이 사용하는 참여단말과 소스 모델의 참여자들이 사용하는 참여단말들을 연결하거나, 새로운 참여자가 사용하는 참여단말을 생성시켜 나가는 과정을 패턴 위빙이라고 한다.

이때, 적응 패턴이 간단하더라도 적응 패턴을 소스 모델에 적용 시키는 것은 어려우므로, 질의응답 방식의 위빙 과정을 통해 적응 패턴을 소스모델에 적용하도록 한다.

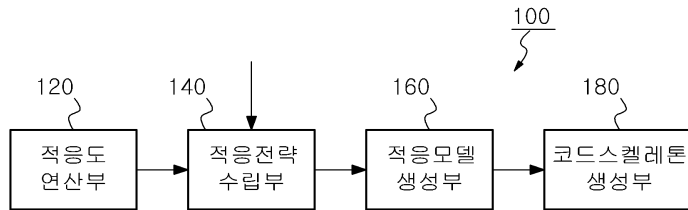
- [0055] 예를 들면, 앞서 선택된 리소스 절약을 위한 대체 적응 행위 선택 적응 패턴의 경우에는 하기와 같은 질의응답이 필요하다.
- [0056] 만약, '모니터링 대상이 되는 리소스와, 모니터링의 종류를 선택하세요'라는 질의 응답이 발생하면 모니터링 대상이 되는 리소스가 배터리 소비량이 되고, 모니터링의 종류는 배터리의 최대 소비량과, 최소 소비량이 될 수 있다.
- [0057] 이어서, '적응대상의 행위를 선택하세요'라는 질의 응답이 발생하면, 사용자단말의 GPS 또는 차량의 GPS가 적응대상의 행위로서 선택될 수 있다.
- [0058] 또한, '각 행위별 리소스 모니터링 조건을 등록하세요'라는 질의 응답이 발생하면, 사용자 단말의 배터리 소비량 중 최대 소비량, 최소 소비량을 모니터링하고, 또는 차량의 GPS 행위 시 배터리 소비량 중 최대 소비량, 최소 소비량이 모니터링의 조건이 될 수 있다.
- [0059] 이하, 적응 전략 수립을 위한 패턴 종류에 대하여 예를 들어 살펴보도록 한다.
- [0060] 앞서 질의에 대한 응답에 따라 미리 정의된 구현 패턴이 소스 모델에 위빙 되어 적응 모델이 생성된다.
- [0061] 이하, 도 4를 참조하여 적응 패턴을 적용한 적응 모델의 일부를 살펴보도록 한다. 이러한 시퀀스 다이어그램을 통해 적응형 프레임워크에서 배터리 잔여량에 따라 적합한 적응 행위 번들을 활성화 또는 비활성화시키는 동작을 확인할 수 있다.
- [0062] 코드 스켈레톤 생성부가 생성된 적응 모델에 기초하여 코드 스켈레톤을 생성한다(S240). 이후, 생성된 코드를 구현함으로써, 소프트웨어가 환경 변화에 능동적으로 적응화된다.
- [0063] 만약, 기존에 행위 구현 로직이 구현되어 있다면 질의응답 과정에서 행위 모듈로 이동할 수 있다. 하지만 실질적으로 행위를 구현한 코드가 독립된 형태로 존재하지 않기 때문에 완전한 이관이 불가능하거나, 또는 구현 코드가 미비하여 개발자가 보완해야 할 가능성이 높다.
- [0064] 이후, 구현된 적응형 모바일 소프트웨어를 적응형 프레임워크에서 실행하여 테스트한다. 이때, 적응형 프레임워크에서의 동작 이상 여부와 기능적 테스트를 수행하여 변환된 모바일 소프트웨어의 적응성을 테스트하게 된다.
- [0065] 상술한 과정을 수행한 후, 다시 적응도 연산을 수행하고, 변환된 소프트웨어의 적응도가 원하는 적응도보다 낮다고 판단되면 다시 변환과정을 시작한다.
- [0066] 이처럼, 모바일 소프트웨어는 점점 자신이 실행되는 다양한 환경의 변화를 감지하고 대응할 수 있는 적응형 소프트웨어로 진화하고 있다. 따라서, 본 발명을 통해 적응이 발생할 수 있는 상황을 소스 코드에서 검색하여 적응 패턴을 선택함으로써 적응 전략을 수립할 수 있다.
- [0067] 본 발명의 적응형 소프트웨어의 적응화 방법 및 시스템은 모바일 환경에서 환경 변화 또는 사용자의 요구 사항을 인식하여 신속하고 능동적으로 대응하여 사용자 만족도를 향상시킬 수 있는 효과가 있다.
- [0068] 또한 본 발명의 적응형 소프트웨어의 적응화 방법 및 시스템은 미리 예상하지 못한 주변 상황의 변화가 발생하는 경우에 상기 소프트웨어가 저장된 모듈을 수집하여, 상기 소프트웨어의 주변 상황 변화에 따른 적응패턴을 포함시킨 후, 다시 모듈에 저장하여 재배포하지 않고, 소프트웨어가 주변 환경 변화에 능동적으로 적응함에 따라 신속하게 변화에 대응할 수 있는 효과가 있다.
- [0069] 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

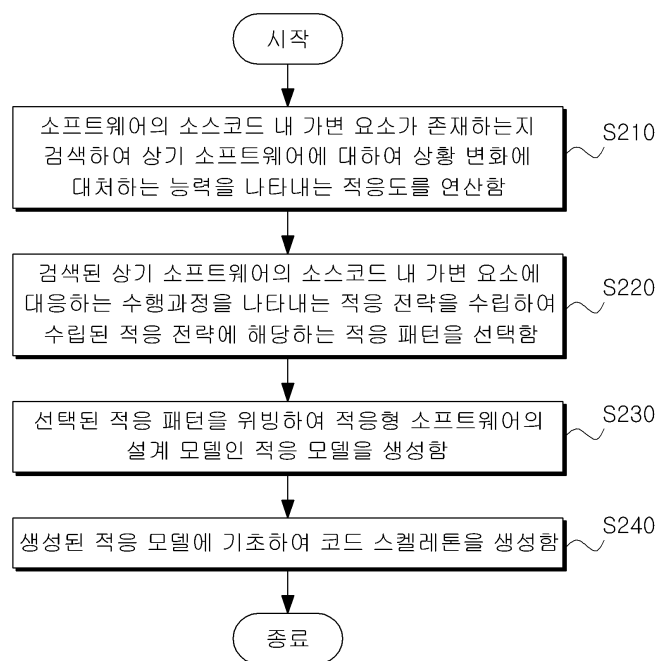
- [0070] 120: 적응도연산부 140: 적응전략수립부
160: 적응모델생성부 180: 코드 스켈레톤 생성부

도면

도면1



도면2



도면3

```

String provider = locationManager.getBestProvider(new Criteria(),
true);

Location src = locationManager.getLastKnownLocation(provider);
Location dest = new Location("dest");

dest.setLongitude(schedule.getLongitude());
dest.setLatitude(schedule.getLatitude());

double distance = (Double) compareDistance(src, dest);

if (distance > 0 && distance <= 100) {
    ....
}
Else
{
    ....
}
    
```

도면4

