



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0019745
(43) 공개일자 2012년03월07일

(51) 국제특허분류(Int. Cl.)

G06F 9/44 (2006.01)

(21) 출원번호 10-2010-0083148

(22) 출원일자 2010년08월26일

심사청구일자 2010년08월26일

(71) 출원인

서강대학교산학협력단

서울특별시 마포구 백범로 35 (신수동, 서강대학교)

(72) 발명자

김순태

서울특별시 강서구 공항대로55길 19, 코오롱오투빌 3차아파트 302동 1103호 (등촌동)

박수용

서울특별시 마포구 백범로 35 (신수동, 서강대학교)

(74) 대리인

특허법인충현

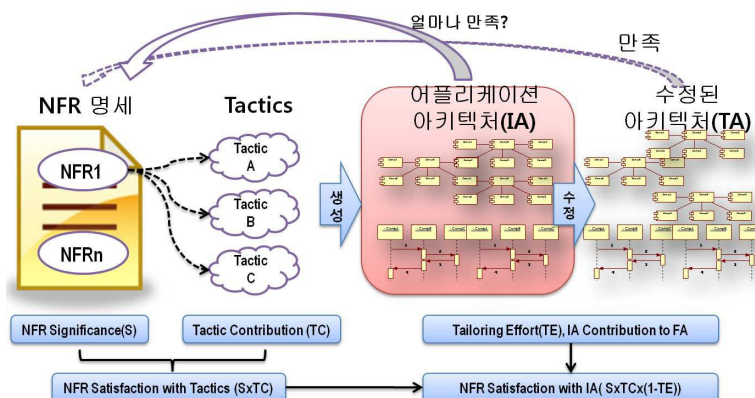
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 자동 실체화된 아키텍처에 대한 정량적 평가 장치, 방법 및 그 기록매체

(57) 요약

자동 실체화된 아키텍처에 대한 정량적 평가 장치, 방법 및 그 기록매체가 개시된다. 본 발명의 일 실시 예에 따른 자동 실체화된 아키텍처에 대한 정량적 평가 장치는 사용자에게 의해 아키텍처를 위한 전술들이 선택되면, 각각의 전술마다 전술 기여 팩터를 입력받고, 상기 전술 기여 팩터에 기반하여 상기 선택된 전술들이 미리 주어진 비기능적 요구사항을 만족하는 정도를 나타내는 NFR 만족도를 계산하는 전술 평가부; 상기 전술들을 통합한 후 구성요소로 실체화하여 어플리케이션 아키텍처를 생성하고, 상기 어플리케이션 아키텍처의 구성요소들을 수정할 수 있는 사용자 인터페이스를 제공하는 소프트웨어 아키텍처 생성부; 상기 어플리케이션 아키텍처가 사용자에게 의해 수정되어 수정된 아키텍처가 생성된 후, 수정된 구성요소의 수를 계산하고, 상기 수정된 구성요소의 수에 기반하여 사용자에게 의한 수정 노력도를 계산하는 테일러링 평가부; 및 상기 NFR 만족도 및 상기 수정 노력도에 기반하여 상기 어플리케이션 아키텍처가 상기 비기능적 요구사항을 만족하는 정도를 나타내는 수정 NFR 만족도를 계산하는 아키텍처 평가부를 포함한다.

대표도 - 도1b



이 발명을 지원한 국가연구개발사업

과제고유번호 C1090 - 1031 - 0008

부처명 정보통신산업진흥원

연구사업명 대학IT연구센터지원사업

연구과제명 융합소프트웨어를 위한 요구 및 품질검증 기술 개발

주관기관 서강대학교 산학협력단

연구기간 2009.03.01 ~ 2012.12.31

특허청구의 범위

청구항 1

사용자에 의해 아키텍처를 위한 전술들이 선택되면, 각각의 전술마다 전술 기여 팩터를 입력받고, 상기 전술 기여 팩터에 기반하여 상기 선택된 전술들이 미리 주어진 비기능적 요구사항을 만족하는 정도를 나타내는 NFR 만족도를 계산하는 전술 평가부;

상기 전술들을 통합한 후 구성요소로 실체화하여 어플리케이션 아키텍처를 생성하고, 상기 어플리케이션 아키텍처의 구성요소들을 수정할 수 있는 사용자 인터페이스를 제공하는 소프트웨어 아키텍처 생성부;

상기 어플리케이션 아키텍처가 사용자에 의해 수정되어 수정된 아키텍처가 생성된 후, 수정된 구성요소의 수를 계산하고, 상기 수정된 구성요소의 수에 기반하여 사용자에게 의한 수정 노력도를 계산하는 테일러링 평가부; 및

상기 NFR 만족도 및 상기 수정 노력도에 기반하여 상기 어플리케이션 아키텍처가 상기 비기능적 요구사항을 만족하는 정도를 나타내는 수정 NFR 만족도를 계산하는 아키텍처 평가부

를 포함하는, 자동 실체화된 아키텍처에 대한 정량적 평가 장치.

청구항 2

제 1 항에 있어서,

상기 전술 평가부는

각각의 전술마다 중요도 팩터를 입력받고, 상기 중요도 팩터를 상기 전술 기여 팩터에 곱하여 상기 NFR 만족도를 계산하는 것을 특징으로 하는, 자동 실체화된 아키텍처에 대한 정량적 평가 장치.

청구항 3

제 1 항에 있어서,

상기 테일러링 평가부는

상기 어플리케이션 아키텍처에 추가된 구성요소의 수 및 상기 어플리케이션 아키텍처로부터 삭제된 구성요소의 수에 구성요소별 가중치를 반영하여 상기 수정 노력도를 계산하는 것을 특징으로 하는, 자동 실체화된 아키텍처에 대한 정량적 평가 장치.

청구항 4

제 1 항에 있어서,

상기 테일러링 평가부는

상기 어플리케이션 아키텍처의 구성요소 중 상기 수정된 아키텍처에 남아있는 구성요소를 상기 수정된 아키텍처의 구성요소의 총수로 나누어 상기 어플리케이션 아키텍처의 기여도를 계산하는 것을 특징으로 하는, 자동 실체화된 아키텍처에 대한 정량적 평가 장치.

청구항 5

사용자에 의해 아키텍처를 위한 전술들이 선택되면, 각각의 전술마다 전술 기여 팩터를 입력받는 단계;

상기 전술 기여 팩터에 기반하여 상기 선택된 전술들이 미리 주어진 비기능적 요구사항을 만족하는 정도를 나타내는 NFR 만족도를 계산하는 단계;

상기 전술들을 통합한 후 구성요소로 실체화하여 어플리케이션 아키텍처를 생성하는 단계;

상기 어플리케이션 아키텍처가 사용자에게 의해 수정되어 수정된 아키텍처가 생성된 후, 수정된 구성요소의 수를 계산하는 단계;

상기 수정된 구성요소의 수에 기반하여 사용자에게 의한 수정 노력도를 계산하는 단계; 및

상기 NFR 만족도 및 상기 수정 노력도에 기반하여 상기 어플리케이션 아키텍처가 상기 비기능적 요구사항을 만족하는 정도를 나타내는 수정 NFR 만족도를 계산하는 단계

를 포함하는, 자동 실체화된 아키텍처에 대한 정량적 평가 방법.

청구항 6

제 5 항에 있어서,

상기 NFR 만족도를 계산하는 단계는

각각의 전술마다 중요도 팩터를 입력받고, 상기 중요도 팩터를 상기 전술 기여 팩터에 곱하여 상기 NFR 만족도를 계산하는 단계인 것을 특징으로 하는, 자동 실체화된 아키텍처에 대한 정량적 평가 방법.

청구항 7

제 5 항에 있어서,

상기 수정 노력도를 계산하는 단계는

상기 어플리케이션 아키텍처에 추가된 구성요소의 수 및 상기 어플리케이션 아키텍처로부터 삭제된 구성요소의 수에 구성요소별 가중치를 반영하여 상기 수정 노력도를 계산하는 단계인 것을 특징으로 하는, 자동 실체화된 아키텍처에 대한 정량적 평가 방법.

청구항 8

제 5 항에 있어서,

상기 어플리케이션 아키텍처의 구성요소 중 상기 수정된 아키텍처에 남아있는 구성요소를 상기 수정된 아키텍처의 구성요소의 총수로 나누어 상기 어플리케이션 아키텍처의 기여도를 계산하는 단계를 더 포함하는 것을 특징으로 하는, 자동 실체화된 아키텍처에 대한 정량적 평가 방법.

청구항 9

제 5 항에 있어서,

상기 전술 기여 팩터는

각각의 전술이 상기 비기능적 요구사항에 기여하는 정도를 나타내는 것을 특징으로 하는, 자동 실체화된 아키텍처에 대한 정량적 평가 방법.

청구항 10

제 5 항에 있어서,

상기 어플리케이션 아키텍처를 생성하는 단계는

상기 전술들의 상호 관계 및 상기 전술들의 구조와 행위에 기반하여 상기 전술들을 통합하고, 통합된 전술들을 구성요소로 실체화하여 어플리케이션 아키텍처를 생성하는 단계인 것을 특징으로 하는, 자동 실체화된 아키텍처에 대한 정량적 평가 방법.

청구항 11

제 10 항에 있어서,

상기 어플리케이션 아키텍처를 생성하는 단계는

사용자에 의해 아키텍처를 위한 전술들이 선택되면, 전술 그룹 내의 제약 조건 및 전술 그룹 간의 제약 조건이 정의된 전술 설정, 전술의 구조와 행위가 RBML로 기술된 전술 시맨틱, 및 전술의 구조와 행위를 파라미터로 하여 전술들을 통합하기 위한 통합 규칙에 기반하여 상기 선택된 전술들을 통합하여 통합 전술을 생성하는 단계; 및

상기 사용자에게 의해 아키텍처의 실체화 다중도가 결정되면, 아키텍처 실체화부에서 상기 실체화 다중도에 따라

상기 통합 전술을 구성요소 단위로 실체화하여 어플리케이션 아키텍처를 생성하는 단계를 포함하는 것을 특징으로 하는, 자동 실체화된 아키텍처에 대한 정량적 평가 방법.

청구항 12

제5항 내지 제11항 중 어느 한 항의 방법을 컴퓨터 시스템에서 실행하기 위한 프로그램이 기록된, 컴퓨터 시스템이 판독할 수 있는 기록매체.

명세서

기술분야

- [0001] 본 발명은 컴포넌트가 아닌 전술에 기반하여 소프트웨어 아키텍처를 설계하는 경우에 비기능적 요구사항과 관련하여 아키텍처를 정량적으로 평가하는 기술에 관한 것으로, 특히, 자동 실체화된 아키텍처에 대한 정량적 평가 장치, 방법 및 그 기록매체에 관한 것이다.

배경기술

- [0002] 과거의 전통적인 소프트웨어 개발 방법론은 프로세스 중심적인 소프트웨어 개발방법에 대한 잘 설명해 주고 있다. 그러나 소프트웨어의 규모가 복잡해짐에 따라 이러한 방법론들은 아키텍처의 설계에 대해 추가적인 고려를 필요로 하게 되었다. 소프트웨어 아키텍처는 소프트웨어 시스템의 구조나 구조들의 집합이다. 이 구조는 소프트웨어의 구성요소와 각 요소의 가시적 특성, 그리고 요소들 간의 관계를 설명한다. 아키텍처 중심의 개발방법은 소프트웨어 개발에서 아키텍처를 먼저 정립한 후, 아키텍처 베이스라인을 토대로 하여 구현하는 것을 말한다. 아키텍처 설계의 핵심이라면, 기능적 요구사항이 되는 설계요소를 분해하고 설계할 때 요구품질에 적합한 패턴을 찾아 이 패턴에 분해결과를 매핑해 나간다는 것이다.
- [0003] 아키텍처는 표현되어야 하고 문서화되어야 한다. 그러나 오랫동안 아키텍처의 설계를 위한 표준화된 도구 없이, 원이나 선, 화살표 등을 이용한 비정형적인 방법이 사용되어 왔다. 비정형적인 방법은 아키텍처의 표현에 대해 이해관계자들에게 통일된 이해를 주지 못하고, 다양한 관점에서 아키텍처를 표현하지 못하여 명확히 아키텍처를 제시하는 것은 불가능하였다.
- [0004] 그런데 실제로 아키텍처를 설계하는 작업에서 시급한 문제는 일단 설계된 아키텍처가 비기능적 요구사항(non-functional requirements; NFRs)을 얼마나 만족시키는지를 평가할만한 정량적인 방법을 찾는 일이다. 아키텍처를 생성하는 주목적이 주어진 NFR을 만족시키는 것이므로, 최초로 실체화된 아키텍처가 얼마나 성공적인 것인지 평가하는 작업 즉, NFR 만족도를 계산하는 작업은 중요할 수밖에 없다. 그러나 종래의 소프트웨어 아키텍처의 설계 작업은 아키텍처를 정량적으로 평가하지 못하고 정성적으로 평가하는 데에 머물러 있었기 때문에, 아키텍처를 생성해가는 일련의 작업이 타당한 것인지 객관적으로 판단하기 어려운 문제점이 있었다.

발명의 내용

해결하려는 과제

- [0005] 본 발명이 이루고자 하는 첫 번째 기술적 과제는 아키텍처를 생성하는 과정의 타당성을 객관적으로 평가하고, 생성된 아키텍처가 비기능적 요구사항을 만족하는 정도를 정량적으로 평가할 수 있는 자동 실체화된 아키텍처에 대한 정량적 평가 장치를 제공하는 데 있다.
- [0006] 본 발명이 이루고자 하는 두 번째 기술적 과제는 상기의 자동 실체화된 아키텍처에 대한 정량적 평가 장치에 적용되는 자동 실체화된 아키텍처에 대한 정량적 평가 방법을 제공하는 데 있다.
- [0007] 본 발명이 이루고자 하는 세 번째 기술적 과제는 상기 자동 실체화된 아키텍처에 대한 정량적 평가 방법을 컴퓨터 시스템에서 실행하기 위한 프로그램이 기록된 매체로서, 컴퓨터 시스템이 판독할 수 있는 기록매체를 제공하는 데 있다.

과제의 해결 수단

- [0008] 상기의 첫 번째 기술적 과제를 이루기 위하여, 본 발명의 일 실시 예에 따른 자동 실체화된 아키텍처에 대한 정량적 평가 장치는 사용자에게 의해 아키텍처를 위한 전술들이 선택되면, 각각의 전술마다 전술 기여 팩터를 입력 받고, 상기 전술 기여 팩터에 기반하여 상기 선택된 전술들이 미리 주어진 비기능적 요구사항을 만족하는 정도

를 나타내는 NFR 만족도를 계산하는 전술 평가부; 상기 전술들을 통합한 후 구성요소로 실체화하여 어플리케이션 아키텍처를 생성하고, 상기 어플리케이션 아키텍처의 구성요소들을 수정할 수 있는 사용자 인터페이스를 제공하는 소프트웨어 아키텍처 생성부; 상기 어플리케이션 아키텍처가 사용자에게 의해 수정되어 수정된 아키텍처가 생성된 후, 수정된 구성요소의 수를 계산하고, 상기 수정된 구성요소의 수에 기반하여 사용자에게 의한 수정 노력도를 계산하는 테일러링 평가부; 및 상기 NFR 만족도 및 상기 수정 노력도에 기반하여 상기 어플리케이션 아키텍처가 상기 비기능적 요구사항을 만족하는 정도를 나타내는 수정 NFR 만족도를 계산하는 아키텍처 평가부를 포함한다.

[0009]

상기의 두 번째 기술적 과제를 이루기 위하여, 본 발명의 일 실시 예에 따른 자동 실체화된 아키텍처에 대한 정량적 평가 방법은 사용자에게 의해 아키텍처를 위한 전술들이 선택되면, 각각의 전술마다 전술 기여 팩터를 입력받는 단계; 상기 전술 기여 팩터에 기반하여 상기 선택된 전술들이 미리 주어진 비기능적 요구사항을 만족하는 정도를 나타내는 NFR 만족도를 계산하는 단계; 상기 전술들을 통합한 후 구성요소로 실체화하여 어플리케이션 아키텍처를 생성하는 단계; 상기 어플리케이션 아키텍처가 사용자에게 의해 수정되어 수정된 아키텍처가 생성된 후, 수정된 구성요소의 수를 계산하는 단계; 상기 수정된 구성요소의 수에 기반하여 사용자에게 의한 수정 노력도를 계산하는 단계; 및 상기 NFR 만족도 및 상기 수정 노력도에 기반하여 상기 어플리케이션 아키텍처가 상기 비기능적 요구사항을 만족하는 정도를 나타내는 수정 NFR 만족도를 계산하는 단계를 포함한다.

발명의 효과

[0010]

본 발명의 실시 예들에 의하면, 아키텍처를 생성하는 과정의 타당성을 객관적으로 평가하고, 생성된 아키텍처가 비기능적 요구사항을 만족하는 정도를 정량적으로 평가할 수 있으며, 비기능적 요구사항이 일정 수준 이상으로 충족되는 것을 보장할 수 있다.

도면의 간단한 설명

[0011]

도 1a는 전술에 기반하여 어플리케이션 아키텍처가 생성된 후, NFR을 만족하도록 아키텍처가 수정되는 과정을 도시한 것이다.

도 1b는 도 1a에서 어플리케이션 아키텍처를 평가하기 위한 요소들을 도시한 것이다.

도 2는 본 발명의 일 실시 예에 따른 자동 실체화된 아키텍처에 대한 정량적 평가 장치의 블록도이다.

도 3은 도 2의 소프트웨어 아키텍처 생성부(100)의 예시적인 상세 블록도이다.

도 4a는 도 3에 적용될 수 있는 피쳐 모델링 방법의 예를 도시한 것이다.

도 4b는 도 4a의 피쳐 모델링 방법을 적용하여 생성한 가용성 측면에서의 피쳐 모델의 예를 도시한 것이다.

도 5a는 전술의 시맨틱을 상술하기 위한 전술 메타모델의 예를 도시한 것이다.

도 5b는 통합 규칙과 전술에 대한 메타모델의 예를 도시한 것이다.

도 5c는 실체화의 예를 도시한 것이다.

도 6은 본 발명의 일 실시 예에 따른 자동 실체화된 아키텍처에 대한 정량적 평가 방법의 흐름도이다.

도 7a는 전술 기여 팩터 및 중요도 팩터의 개념을 도시한 것이다.

도 7b는 아키텍처 설계에 요구되는 비기능적 요구사항의 예를 도시한 것이다.

도 7c는 도 7b의 예에서 중요도 팩터를 결정한 예를 도시한 것이다.

도 7d는 도 7b의 예에서 전술 기여 팩터와 추가 노력도를 계산한 예를 도시한 것이다.

도 8a는 최초 생성된 어플리케이션 아키텍처와 수정된 아키텍처 사이의 관계를 도시한 것이다.

도 8b는 수정 노력의 개념을 도시한 것이다.

도 8c는 수정 노력도와 기여도의 계산을 위한 가중치의 값을 예시한 것이다.

도 9a는 최초 생성된 어플리케이션 아키텍처의 예를 도시한 것이다.

도 9b는 도 9a를 기반으로 수정된 아키텍처의 예를 도시한 것이다.

도 10은 도 7b 및 도 7c의 예에 기반하여 수정 NFR 만족도를 계산한 예를 도시한 것이다.

발명을 실시하기 위한 구체적인 내용

- [0012] 이하에서는 도면을 참조하여 본 발명의 바람직한 실시 예를 설명하기로 한다. 그러나, 다음에 예시하는 본 발명의 실시 예는 여러 가지 다른 형태로 변형될 수 있으며, 본 발명의 범위가 다음에 상술하는 실시 예에 한정되는 것은 아니다.
- [0013] 도 1a는 전술에 기반하여 어플리케이션 아키텍처가 생성된 후, NFR을 만족하도록 아키텍처가 수정되는 과정을 도시한 것이다.
- [0014] 본 발명의 실시 예들은 아키텍처 전술 기반의 접근법(architectural tactic-based approach)을 사용하여 소프트웨어 아키텍처를 자동으로 실체화한 경우에 실체화된 아키텍처가 주어진 비기능적 요구사항을 얼마나 만족시키는가를 측정하는 장치 및 방법에 관한 것이다.
- [0015] 여기서, 아키텍처 전술(이하 '전술')은 시스템의 구조를 설계할 때 재사용이 가능한 블록으로서, 품질 속성(quality attributes)과 관련하여 공통의 과제를 해결하기 위한 일반적인 솔루션을 제공한다. 본 발명의 실시 예들은 전술을 사용하여 비기능적 요구사항을 소프트웨어 아키텍처에 구현하기 위한 품질 주도적 접근법(quality-driven approach)을 제공한다.
- [0016] NFR 만족도를 측정하기 위해 AHP(Analytical Hierarchical Process)에 기반한 일련의 행렬을 사용할 수 있는데, AHP는 아이템 쌍마다 비교를 수행하여 아이템들의 상대적 중요도를 측정하는 정량화 방법이다. 이하에서는 NFR 명세(specifications)는 미리 주어지고, 전술들은 아키텍처를 실체화하기 전에 각각의 NFR을 만족시키도록 선택되어졌다고 가정한다.
- [0017] 선택된 전술들은 아키텍처로 실체화되는데, 실체화된 구성요소의 이름을 재정의하거나 어플리케이션의 문맥을 반영하고 기능적 요구사항을 충족하기 위해 기능적 컴포넌트를 추가하는 등으로 아키텍처에 수정이 가해지면, 최종 아키텍처(FA) 또는 수정된 아키텍처(TA)를 얻을 수 있다.
- [0018] 도 1b는 도 1a에서 어플리케이션 아키텍처를 평가하기 위한 요소들을 도시한 것이다.
- [0019] NFR 만족도의 계산은 다음의 3단계로 이루어질 수 있다.
- [0020] 먼저, 전술들이 아키텍처로 실체화되기 전에, 중요도 팩터(Significance Factor; SF)(도 1b에서는 'S')와 각각의 전술마다 전술 기여 팩터(Tactic Contribution Factor; TCF)를 입력받는다. 중요도 팩터는 각각의 NFR이 어플리케이션에서 얼마나 중요한 요소인지를 나타내는 것이고, 전술 기여 팩터는 주어진 NFR을 만족시키는데 각각의 전술이 얼마나 기여하는지를 나타내는 것이다. 전술 기여 팩터는 AHP를 이용하여 결정될 수 있다. 중요도 팩터나 전술 기여 팩터는 아키텍처 아키텍처 평가 전문가 또는 전술 관련 전문가가 결정하는 값이다. 중요도 팩터와 전술 기여 팩터에 기반하여 선택된 전술들이 NFR을 만족시키는 정도($NFRS_T$)('NFR 만족도')를 측정할 수 있다.
- [0021] 다음, 수정 노력도(Tailoring Effort; TE)와 실체화된 아키텍처의 기여도(Contribution; C)를 측정한다. 수정 노력도와 기여도는 실체화된 아키텍처의 NFR 만족도($NFRS_T$)를 측정하는데에 사용될 수 있다. 이들 요소의 값은 실체화된 아키텍처에서 최종 아키텍처로 수정되는 과정에서 삭제된 구성요소 또는 추가된 구성요소의 수, 그리고 최종 아키텍처에 남아있는 구성요소의 수를 계산하여 얻어질 수 있다.
- [0022] 마지막으로, NFR 만족도($NFRS_T$), 수정 노력도 및 기여도를 이용하여 실체화된 아키텍처의 NFR 만족도($NFRS_T$)를 계산한다.
- [0023] 도 2는 본 발명의 일 실시 예에 따른 자동 실체화된 아키텍처에 대한 정량적 평가 장치(10)의 블록도이다.
- [0024] 전술 평가부(210)는 사용자에게 의해 아키텍처를 위한 전술들이 선택되면, 각각의 전술마다 전술 기여 팩터를 입력받고, 전술 기여 팩터에 기반하여 선택된 전술들이 미리 주어진 비기능적 요구사항을 만족하는 정도를 나타내는 NFR 만족도를 계산한다. 전술 기여 팩터는 TCF 데이터베이스에 저장되어 있을 수 있다. 본 발명의 다른 실시 예에서, 전술 평가부(210)는 중요도 팩터를 전술 기여 팩터와 함께 고려하여 NFR 만족도를 계산한다. 중요도 팩터도 또한 중요도 팩터 데이터베이스에 저장되어 있을 수 있다.
- [0025] 소프트웨어 아키텍처 생성부(100)는 선택된 전술들을 통합한 후 구성요소로 실체화하여 어플리케이션 아키텍처

를 생성한다. 또한, 소프트웨어 아키텍처 생성부(100)는 어플리케이션 아키텍처의 구성요소들을 수정할 수 있는 사용자 인터페이스를 제공한다.

- [0026] 테일러링 평가부(220)는 소프트웨어 아키텍처 생성부(100)에서 생성된 어플리케이션 아키텍처가 사용자에게 의해 수정되어 수정된 아키텍처가 생성된 후, 수정된 구성요소의 수를 계산한다. 테일러링 평가부(220)는 수정되지 않고 그대로 남아있는 구성요소의 수를 계산할 수도 있다.
- [0027] 테일러링 평가부(220)는 수정된 구성요소의 수에 기반하여 사용자에게 의한 수정 노력도를 계산한다. 수정 노력도에는 구성요소별 가중치가 반영될 수 있다. 본 발명의 다른 실시 예에서, 테일러링 평가부(220)는 수정되지 않고 그대로 남아있는 구성요소의 수를 이용하여 기여도를 계산한다. 기여도에는 구성요소별 가중치가 반영될 수 있다.
- [0028] 아키텍처 평가부(230)는 NFR 만족도 및 수정 노력도에 기반하여 수정 NFR 만족도를 계산한다. 수정 NFR 만족도는 소프트웨어 아키텍처 생성부(100)에 의해 최초로 실체화된 어플리케이션 아키텍처가 비기능적 요구사항을 만족하는 정도를 나타낸다.
- [0029] 도 3은 도 2의 소프트웨어 아키텍처 생성부(100)의 예시적인 상세 블록도이다.
- [0030] 도 3에서는 일련의 NFR이 주어지면, 전술이 선택되고 통합된 후, 통합된 전술이 어플리케이션을 위한 초기 아키텍처로 실체화된다.
- [0031] 전술은 피쳐 모델(feature models)을 이용하여 표현되고, 시맨틱(semantics)은 RBML(Role-Based Metamodeling Language)을 이용하여 정의된다. RBML은 UML 기반의 패턴 명세 표기법(pattern specification notation)이다. RBML을 이용하면 아키텍처의 구조 측면과 행위 측면을 동시에 다룰 수 있다.
- [0032] 전술 피쳐 모델은 전술 그룹 내의 제약 조건 및 전술 그룹 간의 제약 조건이 정의된 전술 설정을 저장한다. 전술들은 FMP(Feature Model Plug-in)를 이용하여 설정될 수 있다. FMP는 피쳐 모델을 지원하는 플러그인이다. FMP를 사용하여 전술 전문가는 전술에 관하여 새로운 피쳐 모델을 설계할 수도 있다.
- [0033] 전술 시맨틱에는 전술의 구조와 행위가 RBML로 기술되어 있다. 피쳐 모델의 말단 노드에서 전술들은 구조와 행위의 시맨틱을 포함할 수 있다. 본 발명에서는 이를 위해 RBML을 사용한다. RBML은 확장된 UML로서 구조와 행위를 구체화할 수 있고, 실체화 다중도에 따른 구현 예를 포괄할 수 있다.
- [0034] RBML로 전술의 시맨틱을 표현하기 위해, 전술 설계자는 다음의 과정을 거친다. 먼저, 전술 습득이다. 전술들은 자연어로 표현된 전술들, 정형화되지 않은 프로젝트, 시스템 또는 경험에 산재되어 있는 전술들을 수집해야 한다. 다음, 패턴 명세 프레임에 따라 전술들을 파악한다. 패턴 명세 프레임은 문맥, 문제점, 솔루션, 결과 및 변수 등으로 구성된다. 이러한 프레임은 구조와 행위를 포착하기에 용이한 접근 방법이 될 수 있다. 다음, 전술의 구조와 행위를 UML 다이어그램으로 모델링한다. 전술의 솔루션 부분은 전술의 목표를 달성하기 위한 구조와 행위를 제공한다. 전술 설계자는 UML 클래스와 시퀀스 다이어그램을 이용하여 구조와 행위에 대한 전술의 시맨틱을 정의한다. 마지막으로, UML 다이어그램을 RBML SPS와 IPS로 변환하고, 전술 정보를 기재하는 작업이다. 이 과정에서 UML 다이어그램의 각각의 분류자(classifier)의 앞에는 롤 표시 'l'가 추가된다. 또한 각 구성요소의 실체화 다중도가 정해질수도 있다.
- [0035] 전술 통합부(110)는 사용자에게 의해 아키텍처를 위한 전술들이 선택되면, 전술 설정을 전술 피쳐 모델로부터 읽어들이고, 통합 규칙 관리자로부터 통합 규칙을 읽어들이는다. 전술 피쳐 모델은 FMP를 통해 전술 통합부(110)에 연결될 수 있다.
- [0036] 전술 통합부(110)는 통합 규칙, 전술 설정, 및 술 시맨틱에 기반하여 앞서 선택된 전술들을 통합하여 통합 전술을 생성한다. 상호 관련이 없는 전술들에는 통합 규칙이 제공되지 않는다.
- [0037] 통합 규칙 관리자(120)는 전술의 구조와 행위를 파라미터로 하여 전술들을 통합하기 위한 통합 규칙을 저장한다.
- [0038] 여기서, 통합 규칙은 전술의 구조와 행위를 파라미터로 하여 통합 전술에 신규 구성요소를 추가하는 추가(add) 연산자, 전술의 구조와 행위를 파라미터로 하여 통합 전술의 구성요소를 갱신하는 갱신(update) 연산자, 전술의 구조와 행위를 파라미터로 하여 통합 전술의 구성요소를 삭제하는 삭제(delete) 연산자 등을 포함할 수 있다.
- [0039] 통합 규칙은 추가(add) 연산자, 갱신(update) 연산자, 삭제(delete) 연산자 이외에, 전술의 구조와 행위를 파라미터로 하여 서로 다른 전술의 롤을 병합하여 병합된 롤을 가지는 새로운 전술을 생성하는 병합(merge) 연산자,

전술의 구조와 행위를 파라미터로 하여 전술의 구성요소를 통합 전술에 복사하는 사용(use) 연산자, 및 전술의 구조와 행위를 파라미터로 하여 기존 통합 규칙을 참조하여 새로운 통합 규칙을 정의하기 위한 포함(include) 연산자 등을 추가로 포함할 수 있다.

[0040] 전술 통합부(110)에 의해 통합된 통합 전술들은 아키텍처 실체화부(130)에 전달되는데, 아키텍처 실체화부(130)는 통합 전술을 실체화하여 어플리케이션을 위한 초기 구조를 생성한다. 보다 구체적으로, 아키텍처 실체화부(130)는 사용자에게 의해 아키텍처의 실체화 다중도가 결정되면, 실체화 다중도에 따라 통합 전술을 구성요소 단위로 실체화하여 어플리케이션 아키텍처를 생성한다. 아키텍처 실체화부(130)는 전술이 가지고 있는 물의 실체화 다중화도를 고려하여 생성할 전술 실체화의 수를 결정한다.

[0041] 도 4a는 도 3에 적용될 수 있는 피처 모델링 방법의 예를 도시한 것이다.

[0042] 전술들 사이의 관계를 나타내기 위해 피처 모델링 방법이 사용된다. 전술들은 품질 속성(quality attribute)을 전제로 특정 목적을 위해 그룹으로 묶여질 수 있다. 또한 그룹 내의 전술들뿐만 아니라 전술 그룹은 전술들간의 의존도에 따라 필수적인 전술 그룹(Essential Tactic Group) 또는 부가적인 전술 그룹(Optional Tactic Group)로 구분될 수 있다. 필수적인 전술(Essential Tactic)은 다른 전술들을 구동시키기 위한 코어 전술로 고려되는 것이고, 부가적인 전술(Optional Tactic)은 필수적인 전술에 보충적으로 사용되는 전술로 정의될 수 있다.

[0043] 전술 그룹과 전술들 사이에 함께 사용하면 협동적일 수 있는 관계, 또는 동시에 함께 사용될 수 없는 관계 등이 존재할 수 있다. 이들 관계는 개발자가 NFR을 만족시키는 것을 도와주고 부적절한 전술들을 결합시키는 것을 방지한다.

[0044] 피처 모델을 사용하여 전술 관계를 나타내기 전에 전술들은 3가지 측면 즉, 전술과 관련된 품질 속성, 전술이 달성해야 하는 목표, 목표를 달성하기 위해 필수적인 전술 또는 부가적인 전술의 측면에서 여러 그룹으로 분류된다. 전술 또는 전술 그룹이 필수적인지 부가적인지에 따라 전술 또는 전술 그룹은 피처 모델에서 필수적인 피처(Mandatory Feature) 또는 부가적인 피처(Optional Feature)로 표현될 수 있다.

[0045] 전술들이 분류된 후에는 전술 관계가 정해진다. 전술 관계는 그룹내 관계(In Group Relationship)와 그룹간 관계(Between Group Relationship)로 구분된다. 그룹내 관계는 같은 그룹에 분류된 전술 사이의 관계를 나타낸다. 이는 "함께 사용(Use Together)" 또는 "따로 사용(Use Exclusively)"과 같은 관계를 표현한다. 그룹간 관계는 다른 전술 그룹 또는 품질 속성에 관계된 전술들 사이의 관계를 나타낸다. 이들은 일방이 타방을 "요구하는(Required)"관계, 일방이 타방을 "제안하는(Suggested)" 관계, 일방이 타방을 "배제하는(Exclusive)" 관계 등을 나타낼 수 있다.

[0046] 이와 같이 표현되는 관계들은 전술들을 통합할 때 제약 조건으로 사용될 수 있다.

[0047] 도 4b는 도 4a의 피처 모델링 방법을 적용하여 생성한 가용성 측면에서의 피처 모델의 예를 도시한 것이다.

[0048] 도 4b에서 전술들은 오류 검출(Fault Detection), 회복 준비 및 보정(recovery-preparation and repair), 회복 재도입(recovery reintroduction)의 카테고리로 분류될 수 있다. 오류 검출 전술은 핑/에코(Ping/Echo) 전술, 심박(Heartbeat) 전술, 제외(Exception) 전술에 한정된다. 상태 중복(State Redundancy) 전술은 상태 재동기화(State Resynchronization) 전술을 요구하는 관계에 있다. 도 4b에 도시된 나머지 전술들도 도 4a의 피처 모델링 방법에 기반하여 해석될 수 있다.

[0049] FMP로 정의되는 피처 모델은 가용성, 성능, 보안성 등으로 구분될 수 있다. 가용성 피처 모델에서 오류 검출 피처 아래의 개체수 0-2는 포함 가능 그룹(inclusive group)을 나타내고, 상태 중복 피처 아래의 개체수 0-1은 배타 그룹(exclusive group)을 나타낸다. 피처가 설정되면, FMP는 그 설정에서 선택된 피처들만을 포함하는 새로운 피처 모델을 생성할 수 있다. 설정들은 전술 통합부(110)에 전달되어 통합 과정에 사용될 수 있다.

[0050] 모든 물은 실체화 다중도(realization multiplicity)라고 불리우는 속성을 가지는데, 이는 해당 물을 수행하기 위해 생성될 수 있는 예시의 수를 나타낸다. 실체화 다중도의 값은 어플리케이션의 문맥에 기반하여 결정된다. 예를 들어, 캐시 클라이언트 물은 실체화 다중도로 2의 값을 가질 수 있는데, 이 경우, 캐시 클라이언트 물을 위해 적어도 2개의 예시가 생성될 것이 요구된다. 실체화 다중도는 각각의 물의 기록 필드(documentation field)에 기재되고, 다이어그램 상에 특별히 표시되지 않을 수 있다.

[0051] 사용자가 선택된 전술들을 참고하여 물 실체화 다중도 즉, 생성할 물 예시의 수를 확정하면, 아키텍처 실체화부(130)는 실체화 다중도에 따라 물을 구성요소로 실체화한다. 이후 사용자는 어플리케이션의 문맥에 상응하도록 실체화된 구성요소들의 이름을 재정의하고, 기능적 구성요소를 이용하여 실체화된 아키텍처를 완성하는 방법으

로 통합 전술이 구체화할 수 있다.

- [0052] 도 5a는 전술의 시맨틱을 상술하기 위한 전술 메타모델의 예를 도시한 것이다.
- [0053] 도 5a에서, "Tactic"은 하나의 SPS와 0 이상의 IPS로 구성되는데, 이는 전술이 하나의 구조와 0 이상의 행위를 가짐을 의미한다.
- [0054] RBML은 롤(roles)의 측면에서 전술 시맨틱을 구체화한다. 여기서 롤은 UML 메타모델 레벨(UML metamodel level)에서 정의되는데, 상기 롤은 UML 모델 구성요소(UML model elements)(예를 들어, 클래스(classes), 메시지(messages) 등)에 의해 수행된다. 롤은 그 기저로서 UML 메타클래스(UML metaclass)를 가지며, UML 메타클래스에 대한 일련의 제약 조건을 규정한다. 상기 제약 조건을 만족시키는 기저 메타클래스(base metaclass)의 예시(instances)만이 상기 롤을 수행할 수 있다. 모든 롤은 자신의 롤을 수행할 수 있는 구성요소의 수를 제약하는 실체화 다중도(realization multiplicity)를 가진다.
- [0055] 도 5a에서 행위의 구성요소들은 구조의 구성요소에 기반하여 표현되고, IPS는 SPS에 대해 "+sps"라는 레퍼런스를 갖는다. RBML의 구성요소로 표시되어 있는 "ClassRole", "Association", "Lifelie" 등의 SPS 및 IPS 구성요소들은 "RBML::Role"로부터 상속된 피처인데, "RBML::Role"에는 실체화 다중도의 상한과 하한이 정의되어 있다. 각각의 롤은 UML 메타 클래스로부터 상속된 것이다.
- [0056] 전술들을 통합하는 것은 소프트웨어 아키텍처를 구성할 때 필수적인 단계 중의 하나이다. 전술들은 특정의 비기능적 요구 사항(NFR)을 달성하기 위한 하나의 솔루션으로서 각자의 목표를 달성하기 위해 서로 상호작용을 하기 때문이다. 하나의 전술이 주어진 NFR을 달성하기 어려운 경우, 선택된 전술들 사이에 상호 작용이 있다면 전술들이 통합될 필요가 있다.
- [0057] 특정의 NFR을 만족하는 전술들을 선택하는 경우, 전술들을 효과적으로 통합하기 위해 전술들 사이에 관계(예를 들어, 협동적인 관계, 요구되는 관계 등)가 분석될 수 있다. 이하에서는 주어진 NFR을 만족하기 위해 전술들을 자동으로 통합하여 통합 전술을 생성하는 방법에 대해 설명한다.
- [0058] 전술들은 SPS와 IPS의 동작 단계에 따라 통합된다. 통합을 자동으로 수행하기 위해서, SPS와 IPS의 구성요소들을 조작할 수 있다. 통합 규칙에는 XML 기반의 통합 연산자 즉, add, update, delete, merge, use, include가 포함될 수 있다. 통합 과정은 모델 구성요소에 대한 일련의 조작 과정인데, add, update, delete 등의 3가지 연산자로 모든 구성요소를 조작하는 것이 가능하다. 2 이상의 전술을 통합할 때 모델 구성요소들의 조작을 용이하게 하기 위해 merge, use, include의 추가적인 연산자가 사용될 수 있다.
- [0059] 도 5b는 통합 규칙과 전술에 대한 메타모델의 예를 도시한 것이다.
- [0060] "compositionrule"은 전술을 통합하는 규칙들을 규정한다. "Composition Rule"은 "SPS::CompositionRule", "IPS::CompositionRule" 및 "CR::Include"로 구성되는데, 각각 SPS를 위한 통합 규칙, IPS를 위한 통합 규칙, include 연산자를 의미한다.
- [0061] 도 5c는 실체화의 예를 도시한 것이다.
- [0062] SPS의 클래스 롤과 IPS의 라이프라인 롤은 실체화 다중도에 의해 규정된다. 예를 들어, "|ClsA" 롤은 2이상의 실체화 다중도를 가지므로, 적어도 2이상의 롤 예시를 생성하게 한다. "|ClsA" 클래스 롤이 2번 실체화되므로, "|ClsA"와 "|ClsB" 사이의 연결 롤(association role)도 2번 실체화될 필요가 있다.
- [0063] 두 개의 전술이 통합된 통합 전술을 이용하여 아키텍처를 실체화하는 경우를 예를 설명한다. 구성요소의 이름에 표시된 "G_0", "G_1" 등은 실체화 과정에서 생성된 구성요소의 인덱스를 의미하는 경우, "|CacheClient" 롤이 2이상의 실체화 다중도를 가지므로, 어플리케이션 아키텍처에는 그 롤로부터 실체화된 "CacheClientG_0" 클래스와 "CacheClientG_1" 클래스가 존재하게 된다.
- [0064] 소프트웨어 아키텍처 생성부(100)를 통해 실체화된 아키텍처에 대해, 설계자는 기능적 요구 사항을 만족시키기 위해 구성요소들의 이름을 재정의하거나 새로운 구성요소들을 추가할 수 있다. 전술로부터 실체화된 구성요소들은 상호간에 어떠한 관계도 갖지 않을 수 있는데, 이 경우 설계자는 실체화된 구성요소들 사이에 링크를 설정하게 된다.
- [0065] 도 6은 본 발명의 일 실시 예에 따른 자동 실체화된 아키텍처에 대한 정량적 평가 방법의 흐름도이다.
- [0066] 먼저, 사용자에게 의해 아키텍처를 위한 전술들이 선택되면, 각각의 전술마다 전술 기여 팩터를 입력받는다(S21).

이 과정(S21)에서 중요도 팩터를 입력받을 수도 있다. 도 7a는 전술 기여 팩터 및 중요도 팩터의 개념을 도시한 것이다. 중요도 팩터는 전술과 관계없이 NFR 자체를 평가한 값이다. 전술 기여 팩터는 각각의 전술이 특정 NFR을 얼마나 만족시키는지를 평가한 값이다. 추가 노력도(AE)는 NFR 만족도와 반비례하는 값으로, 전술 자체가 NFR을 만족하는 정도를 기준으로 NFR을 완전히 만족하기 위해 추가적으로 들여야 하는 노력의 정도를 나타낸다. 도 7b의 예는 온라인 거래 시스템(Online Trading System)을 위한 아키텍처를 설계하기 위해 주어진 비기능적 요구사항의 예를 보여준다. 도 7b의 예에서, NFR1은 가용성, NFR2와 NFR3은 성능, NFR4는 보안성 측면의 비기능적 요구사항이다. 가장 우측에는 해당 NFR을 만족시키는 전술들이 기재되어 있다. 도 7c는 도 7b의 예에서 중요도 팩터를 결정한 예를 도시한 것이다. 중요도 팩터는 NFR들을 쌍마다 비교하여 얻어진다. 예를 들어, NFR1은 NFR2와 비교했을 때 동일한 정도의 중요도를 가지지만, NFR3과 비교했을 때는 1.5, NFR4와 비교했을 때는 3, NFR5와 비교했을 때는 5의 중요도를 가진다. NFR2, NFR3, NFR4, NFR5에 대해서는 다른 NFR과 비교한 결과를 정리한 후, 각 NFR의 상대적인 중요도를 결정하면, 중요도 팩터를 구할 수 있다. 도 7d는 도 7b의 예에서 전술 기여 팩터와 추가 노력도를 계산한 예를 도시한 것이다. 전술 기여 팩터는 특정 NFR과 관계되는 전술들의 기여도를 모두 더한 값이다. NFR 만족도(88.49%)와 추가 노력도(11.51%)를 합하면 100%가 된다.

[0067] 다음, 전술 기여 팩터에 기반하여 선택된 전술들이 미리 주어진 비기능적 요구사항을 만족하는 정도를 나타내는 NFR 만족도를 계산한다(S22). NFR 만족도는 수학적 식 1과 같이 계산될 수 있다.

수학적 식 1

$$NFRS_T = \sum_{i=1}^n SF_i \times \left(\sum_{j=1}^m TCF_{ij} \right)$$

[0068]

[0069] 여기서, i는 NFR의 인덱스이고, j는 전술의 인덱스이다. m은 전술의 총 개수이고, n은 NFR의 총 개수이다. TCF_{ij} 는 i번째 NFR을 만족하는 j번째 전술의 전술 기여 팩터를 나타내고, SF_i 는 i번째 NFR의 중요도 팩터를 나타낸다. 참고로, 추가 노력도는 수학적 식 2와 같이 계산될 수 있다.

수학적 식 2

Additional Effort(AE)

$$AE = \sum_{i=1}^n (SF_i \times (1 - \sum_{j=1}^m TCF_{ij}))$$

[0070]

[0071] 다음, 전술들을 통합한 후 구성요소로 실체화하여 어플리케이션 아키텍처를 생성한다(S23). 본 발명의 다른 실시 예에서, 이 과정(S23)은 전술들의 상호 관계 및 전술들의 구조와 행위에 기반하여 전술들을 통합하고, 통합된 전술들을 구성요소로 실체화하여 어플리케이션 아키텍처를 생성하는 과정일 수 있다. 보다 구체적으로, 어플리케이션 아키텍처를 생성하는 과정은 사용자에게 의해 아키텍처를 위한 전술들이 선택되면 전술 설정, 전술 시맨틱, 및 통합 규칙에 기반하여 선택된 전술들을 통합하여 통합 전술을 생성하는 단계, 사용자에게 의해 아키텍처의 실체화 다중도가 결정되면 아키텍처 실체화부에서 실체화 다중도에 따라 통합 전술을 구성요소 단위로 실체화하여 어플리케이션 아키텍처를 생성하는 단계를 포함할 수 있다. 어플리케이션 아키텍처는 NFR을 만족시키기 위해 추가적인 작업을 거쳐야 한다.

[0072] 다음, 어플리케이션 아키텍처가 사용자에게 의해 수정되어 수정된 아키텍처가 생성된 후, 수정된 구성요소의 수를 계산한다(S24). 도 8a에서 보듯이, 최초 생성된 어플리케이션 아키텍처(IA)와 수정된 아키텍처(TA)는 동일하지 않다. 어플리케이션 아키텍처(IA)에 구성요소가 추가/삭제되기 때문이다.

[0073] 다음, 수정된 구성요소의 수에 기반하여 사용자에게 의한 수정 노력도를 계산한다(S25). 도 8b에서 보듯이, 수정 노력(Tailoring Effort)은 구성요소의 추가나 삭제를 합한 개념이다. 또한, 기여도는 최초 생성된 어플리케이션 아키텍처(IA)와 수정된 아키텍처(TA)에서 공통된 부분이다.

[0074] 수정 노력도는 수학적 식 3과 같이 계산될 수 있다.

수학식 3

$$TE_i = \sum_{k=1}^l \left(\left(\frac{(IA_k - IA'_k) + (TA_k - IA'_k)}{IA_k + (TA_k - IA'_k)} \right) \times w_k \right)$$

[0075]

[0076]

k는 구성요소의 타입을 의미한다. 예를 들어, k=1이면 컴포넌트, k=2이면 연결, k=3이면 상호작용, k=4이면 라이브러리, k=5이면 메시지 등으로 구성요소의 타입이 정해질 수 있다. l은 구성요소 타입의 총수이다. IA_k는 최초로 생성된 어플리케이션 아키텍처의 구성요소의 수, IA'_k는 최초의 구성요소 중 수정된 아키텍처에 남아있는 구성요소의 수, TA_k는 수정된 아키텍처의 구성요소의 수, w_k는 구성요소에 대한 가중치이다. 참고로 기여도는 수학식 4와 같이 계산될 수 있다.

수학식 4

$$C_i = \sum_{k=1}^l \left(\left(\frac{IA'_k}{TA_k} \right) \times w_k \right)$$

[0077]

[0078]

도 8c는 수정 노력도와 기여도의 계산을 위한 가중치의 값을 예시한 것이다.

[0079]

도 9a 및 도 9b를 참고로 하여 수정 노력도와 기여도를 계산하는 방법을 설명한다. 도 9a는 최초 생성된 어플리케이션 아키텍처의 예를 도시한 것이다. 최초로 실체화된 아키텍처의 구성요소(컴포넌트)는 A, B, C, D로, IA_{comp}는 4이다.

[0080]

도 9b는 도 9a를 기반으로 수정된 아키텍처의 예를 도시한 것이다. 삭제된 구성요소(컴포넌트)는 D, 추가된 구성요소(컴포넌트)는 E, F이다. 따라서, IA'_{comp}는 3, TA_{comp}는 5이다. 이 값들을 이용하여 수정 노력도를 구하면,

$$TE_i = \sum \left(\left(\frac{(4-3)+(5-3)}{4+(5-3)} \right) \times 0.3 \right) \dots$$

면, $C_i = \sum \left(\left(\frac{3}{5} \right) \times 0.3 \right) \dots$ 와 같이 되고, 기여도를 구하면, $C_i = \sum \left(\left(\frac{3}{5} \right) \times 0.3 \right) \dots$ 와 같이 된다.

[0081]

마지막으로, NFR 만족도 및 수정 노력도에 기반하여 최초의 어플리케이션 아키텍처가 비기능적 요구사항을 만족하는 정도를 나타내는 수정 NFR 만족도를 계산한다(S26). 수정 NFR 만족도는 수학식 5와 같이 계산될 수 있다.

수학식 5

$$NFRS_i = \sum_{i=1}^n \left(SF_i \times \left(\sum_{j=1}^m TCF_{ij} \right) \times (1 - TE_i) \right)$$

[0082]

[0083]

수정된 구성요소의 모든 타입을 분석하기 위해서 평균 수정 노력을 계산할 수 있다. 평균 수정 노력도(Avg.TE)는 수학식 6과 같이 계산된다. 또한, 전술 시맨틱의 유효성과 커버리지를 검증하기 위해 평균 기여도를 계산할 수도 있다. 평균 기여도(Avg.C)는 수학식 7과 같이 계산될 수 있다.

수학식 6

$$Avg.TE = \left(\sum_{i=1}^n TE_i \right) / n$$

[0084]

수학식 7

$$Avg.C = (\sum_{i=1}^n C_i) / n$$

[0085]

[0086]

여기서, n은 모든 NFR의 개수이다.

[0087]

도 10은 도 7b, 7c 및 7d에 도시된 NFR, 중요도 팩터, 전술 기여 팩터 등에 기반하여 수정 NFR 만족도를 계산한 결과를 도시한 것이다.

[0088]

도 10에서 실체화하기 전의 NFR 만족도 즉, 전술 자체만으로 계산한 NFR 만족도는 88.49%인데, 아키텍처로 실체화한 후 최종 아키텍처로 수정하기 위해서는 12.74%의 수정 노력도가 필요하므로, 수정 NFR 만족도는 77.49%가 된다. 전술 자체로 계산 NFR 만족도가 88.49%에 불과한 점을 고려하면 아키텍처 자동 실체화의 효율성이 높다고 해석된다. 이러한 점은 평균 기여도가 91.01%인 점을 참고하여도 명백하다.

[0089]

본 발명은 소프트웨어를 통해 실행될 수 있다. 바람직하게는, 본 발명의 실시 예들에 따른 자동 실체화된 아키텍처에 대한 정량적 평가 방법을 컴퓨터에서 실행시키기 위한 프로그램을 컴퓨터로 읽을 수 있는 기록매체에 기록하여 제공할 수 있다. 소프트웨어로 실행될 때, 본 발명의 구성 수단들은 필요한 작업을 실행하는 코드 세그먼트들이다. 프로그램 또는 코드 세그먼트들은 프로세서 판독 가능 매체에 저장되거나 전송 매체 또는 통신망에서 반송파와 결합된 컴퓨터 데이터 신호에 의하여 전송될 수 있다.

[0090]

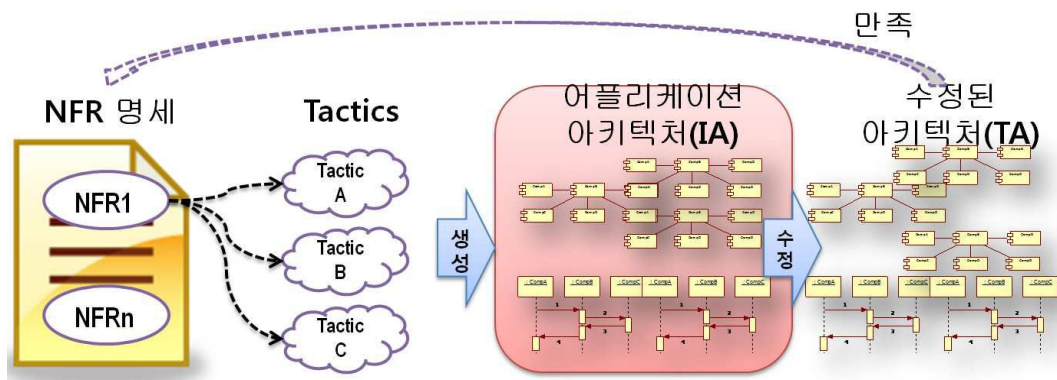
컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다. 컴퓨터가 읽을 수 있는 기록 장치의 예로는 ROM, RAM, CD-ROM, DVD±ROM, DVD-RAM, 자기 테이프, 플로피 디스크, 하드 디스크(hard disk), 광데이터 저장장치 등이 있다. 또한, 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 장치에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

[0091]

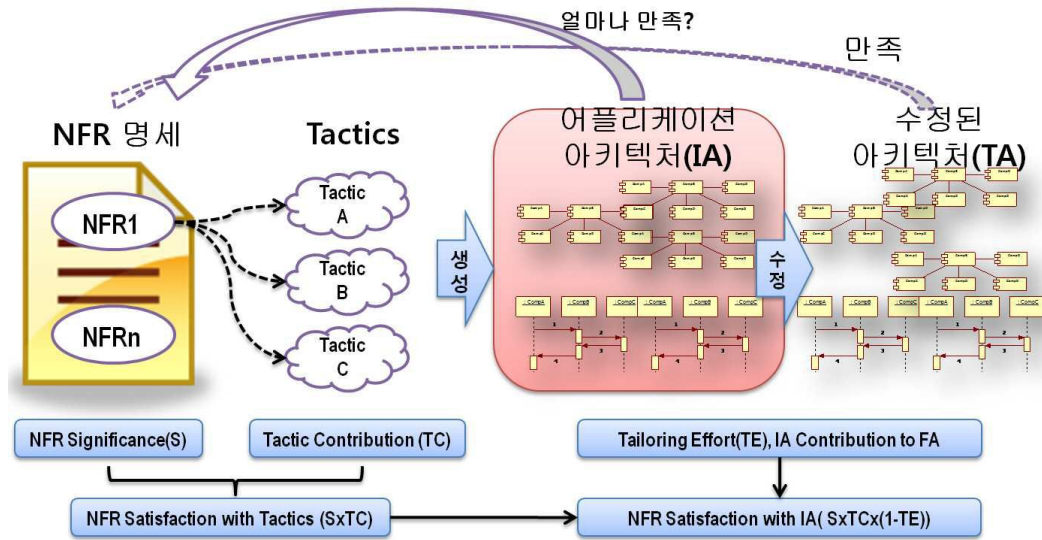
본 발명은 도면에 도시된 일 실시 예를 참고로 하여 설명하였으나 이는 예시적인 것에 불과하며 당해 분야에서 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 실시 예의 변형이 가능하다는 점을 이해할 것이다. 그리고, 이와 같은 변형은 본 발명의 기술적 보호범위 내에 있다고 보아야 한다. 따라서, 본 발명의 진정한 기술적 보호범위는 첨부된 특허청구범위의 기술적 사상에 의해서 정해져야 할 것이다.

도면

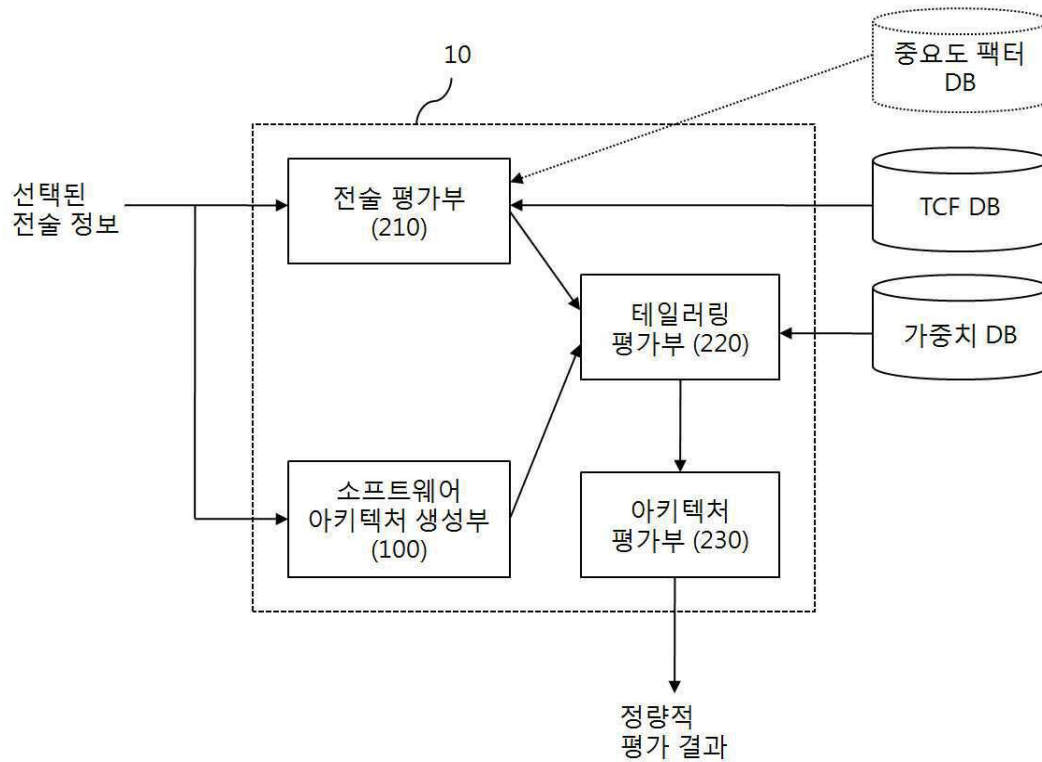
도면1a



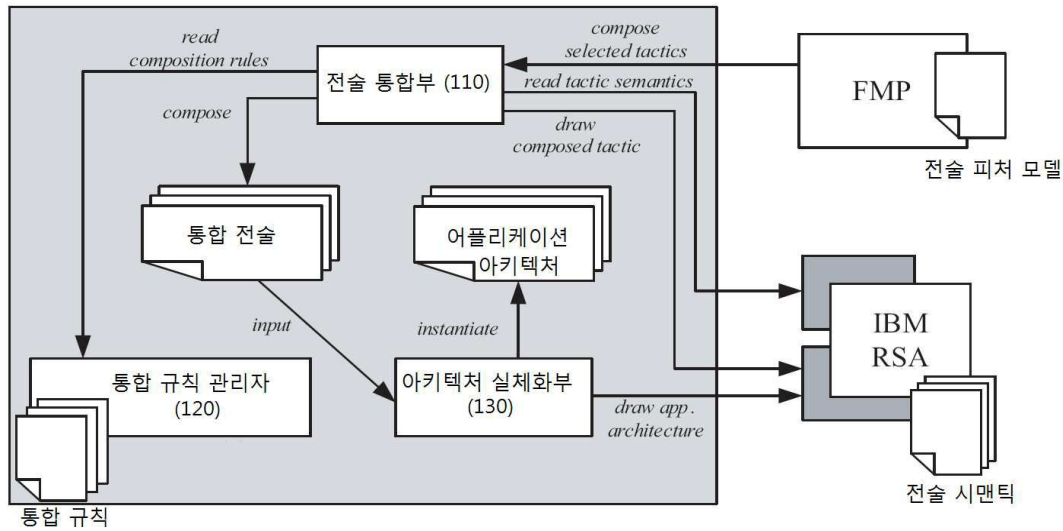
도면1b



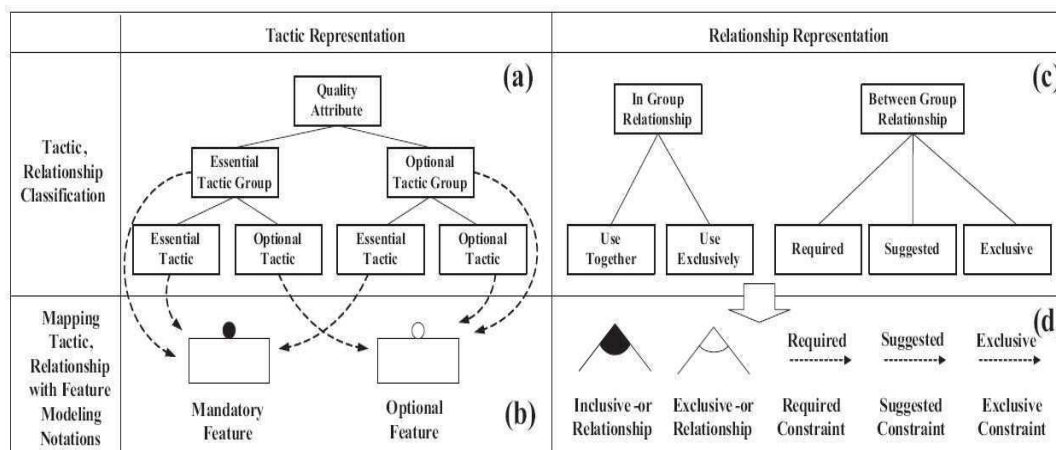
도면2



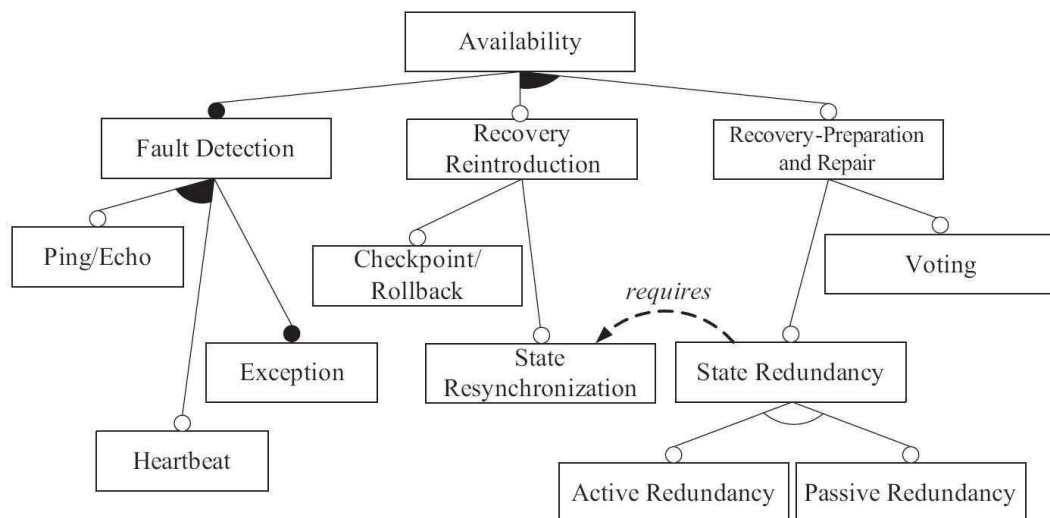
도면3



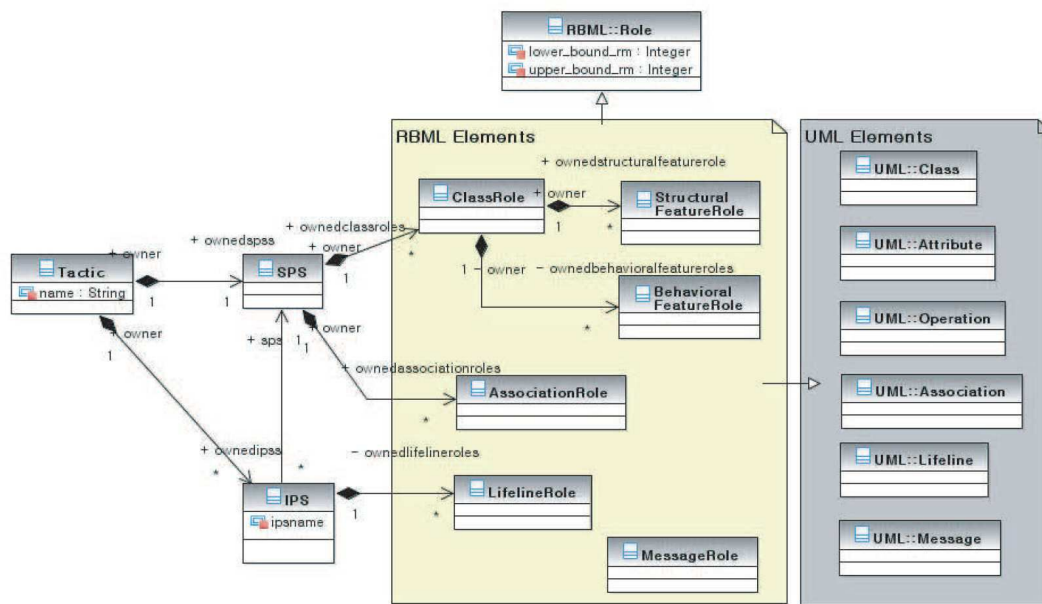
도면4a



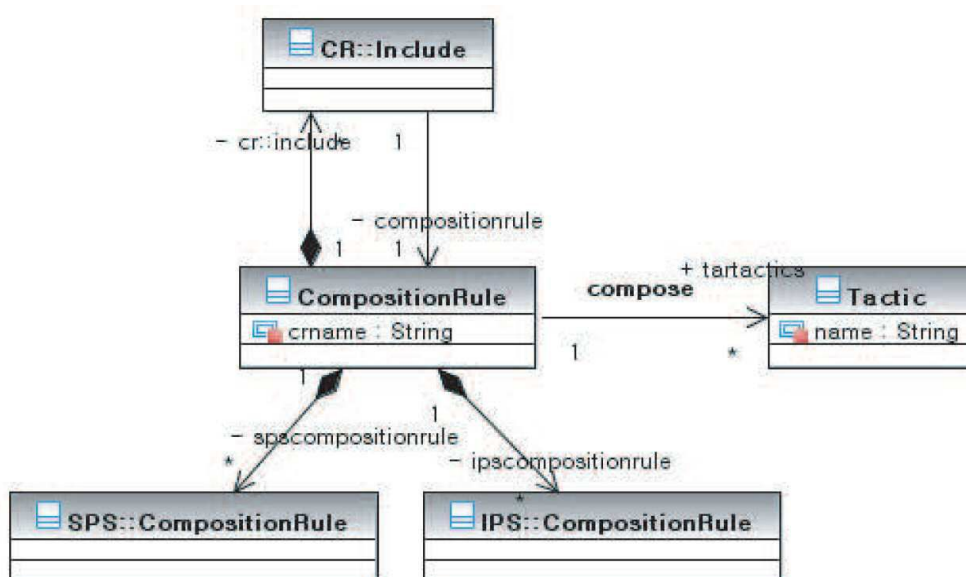
도면4b



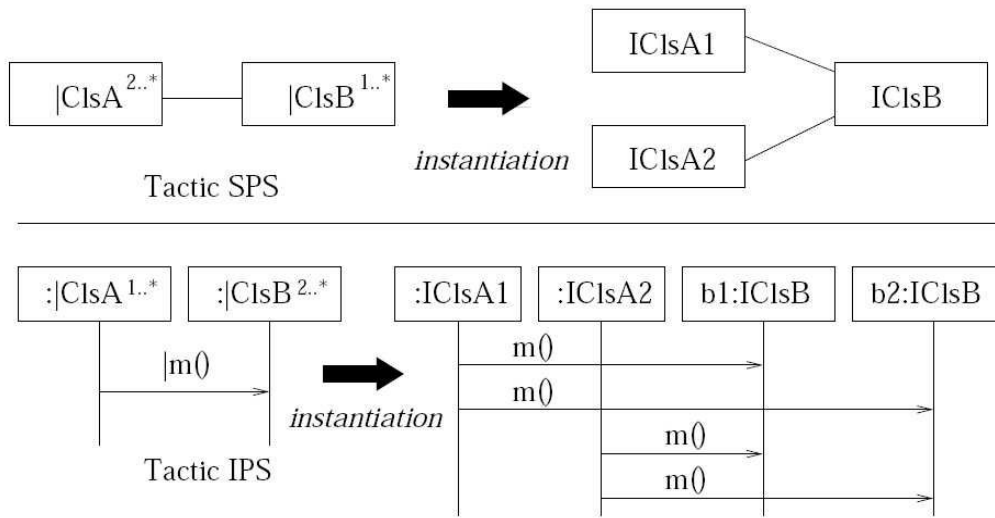
도면5a



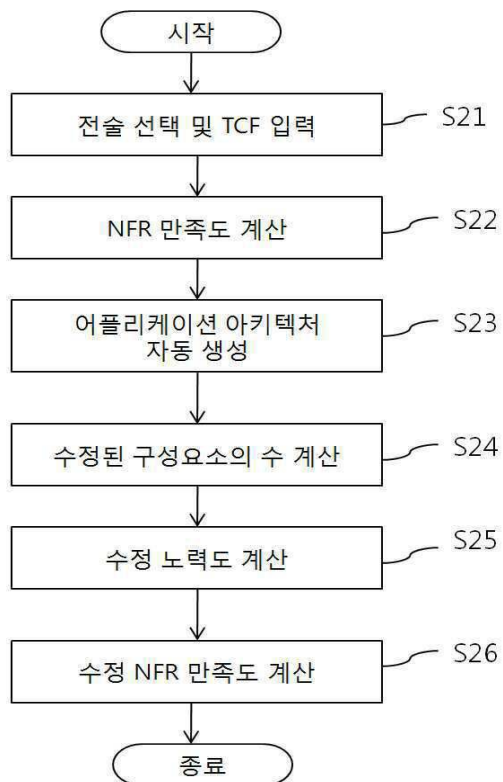
도면5b



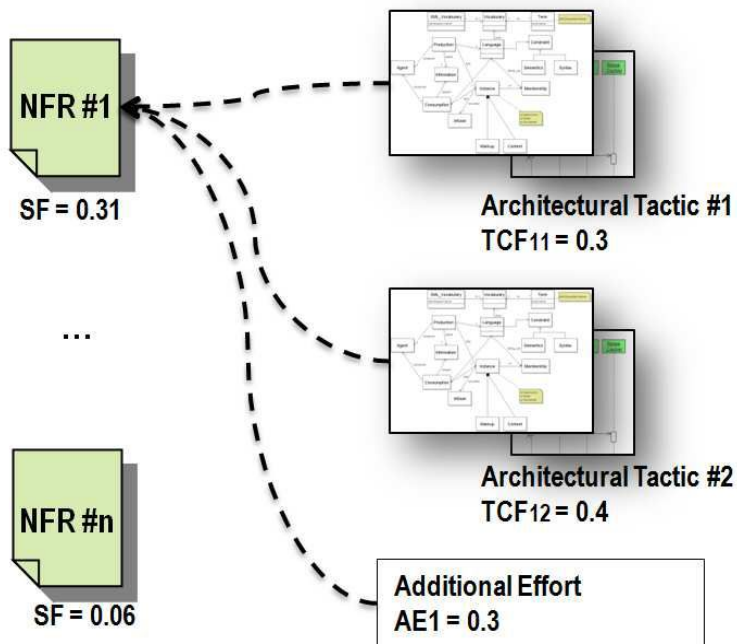
도면5c



도면6



도면7a



도면7b

NFR1	The Monitoring Subsystem should <u>detect faults within 5 second</u> and notify it to administrator. - Availability	Ping/Echo, Heartbeat
NFR2	The system should be able to <u>process 300 transactions per second</u> 400,000 transactions per day and allow multiple orders - Performance	FIFO, Introduce Concurrency
NFR3	For the representative quotes, the server should <u>send the information to the clients within 1 seconds.</u> - Performance	Maintain Multiple Copies
NFR4	Only <u>authenticated</u> users can access the system. The user credentials <u>must not be seen to unauthorized personnel.</u> - Security	ID/Password, Maintain Data Confidentiality

도면7c

Significance Factor (Consistency Ratio = 9.18%)

	NFR1	NFR2	NFR3	NFR4	SF
NFR1	1.00	1.73	3.87	7.94	0.47
NFR2	0.58	1.00	4.58	5.92	0.35
NFR3	0.26	0.22	1.00	4.58	0.14
NFR4	0.13	0.17	0.22	1.00	0.05

도면7d

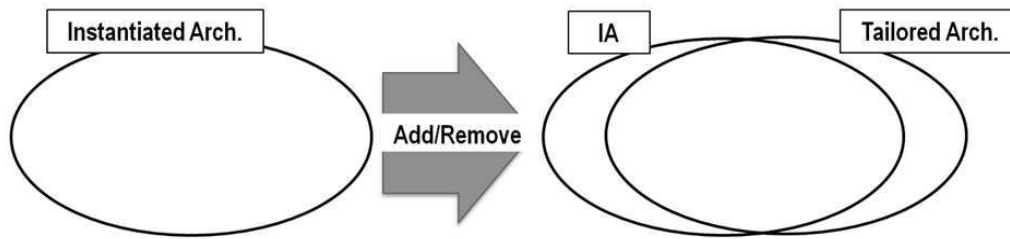
Tactic Contribution Factor and Additional Effort

	TCF	TCFij	AE
NFR1	0.95	Ping/Echo(0.47), Heartbeat(0.47)	0.05
NFR2	0.86	FIFO(0.14), Intro. Concurrency(0.71)	0.14
NFR3	0.75	M.M. Copies(0.75)	0.25
NFR4	0.86	ID/Password(0.14), M.D. Confidentiality(0.71)	0.14

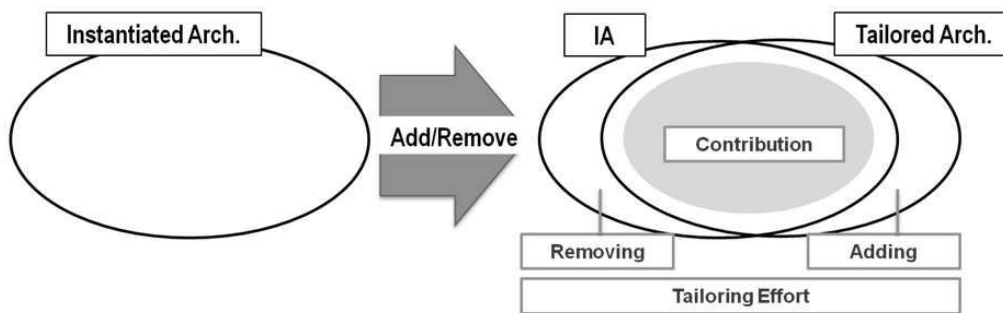
$$NFRS_r = 88.49\%$$

$$AE = 11.51\%$$

도면8a



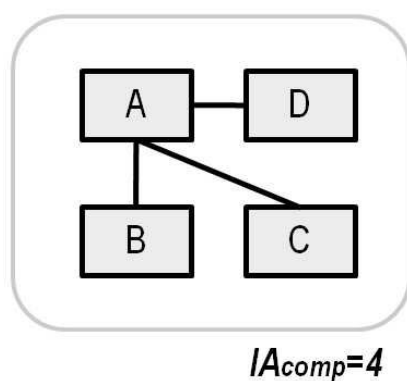
도면8b



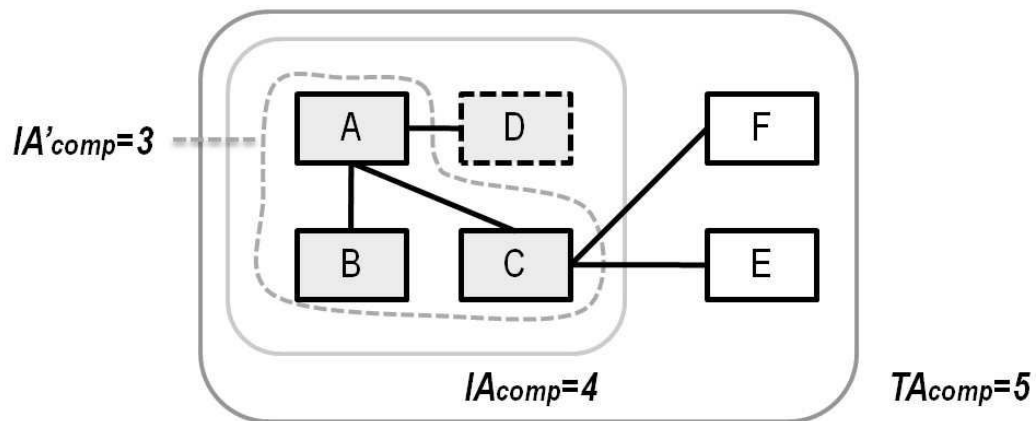
도면8c

W_k	Elements	Weight
1	Component	0.30
2	Association	0.15
3	Interaction	0.11
4	Lifeline	0.06
5	Message	0.06

도면9a



도면9b



도면10

Degree of NFR Satisfaction with Tactics

