

(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。 Int. Cl. (11) 공개번호 10-2006-0106641
G06Q 99/00 (2006.01) (43) 공개일자 2006년10월12일

(21) 출원번호 10-2006-0012424
(22) 출원일자 2006년02월09일

(30) 우선권주장 11/094,926 2005년03월31일 미국(US)

(71) 출원인 마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이

(72) 발명자 아펠, 데이비드 벤자민
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이
마이클로소프트 코포레이션
데이비드슨, 엘리자베스 엘.
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이
마이클로소프트 코포레이션
메리필드, 에릭 사바튼 주니어.
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이
마이클로소프트 코포레이션
레비, 마르크
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이
마이클로소프트 코포레이션
아이직스, 스콧 엠.
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이
마이클로소프트 코포레이션
호만, 올리치
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이
마이클로소프트 코포레이션

(74) 대리인 주성민
백만기
이중희

심사청구 : 없음

(54) 비즈니스 모델들을 비교 및 대조하는 방법

요약

본 발명은 비즈니스 모델들을 비교하고 대조하는 것에 관한 것이다. 모델 프로세싱 모듈들은 비즈니스 모델들을 조작하는 데에 사용될 수 있는 형식 연산자들을 구현한다. 비교 연산자는 유사한 유형의 비즈니스 모델들을 비교하는 데에 사용될 수 있다. 대조 연산자는 상이한 유형들의 비즈니스 모델들을 대조하는 데에 사용될 수 있다. 제약 연산자는 제약들의 준수

와 관련하여 비즈니스 모델들을 검사하는 데에 사용될 수 있다. 개선 연산자는 산업 특유 (또는 그렇지 않으면 좀더 구체적인) 데이터에 기초해 비즈니스 모델들을 개선하는 데에 사용될 수 있다. 구성 연산자는 다른 비즈니스 모델들의 일부들로부터 새로운 비즈니스 모델들을 구성하는 데에 사용될 수 있다.

대표도

도 1

색인어

비즈니스 모델, 모델링, 형식 연산자, 비교, 대조, 제약, 개선, 구성

명세서

도면의 간단한 설명

도 1은 비즈니스 모델들을 비교하고 대조하는 데에 사용될 수 있는 예시적인 컴퓨터 아키텍처를 도시한다.

도 2는 비즈니스 모델들에서 수행될 수 있는 연산들의 예시적 아키텍처를 도시한다.

도 3은 구조화된 비즈니스 기능들에 기초해 효율적으로 그리고 융통성있게 비즈니스 모델링하는 데에 사용될 수 있는 예시적인 기능 모델링 스키마(capability modeling schema)를 도시한다.

도 4는 비즈니스 기능 모델과 서비스 네트워크 모델간의 예시적인 매핑을 도시한다.

도 5는 유사한 유형의 비즈니스 모델들을 비교하기 위한 방법의 예시적인 흐름도를 도시한다.

도 6은 비즈니스 아키텍처의 대응하는 상이한 계층들을 표현하는 비즈니스 모델들의 상이한 유형들을 대조하기 위한 방법의 예시적인 흐름도를 도시한다.

도 7은 비즈니스 모델이 하나 이상의 제약들에 따른다는 것을 확인하기 위한 방법의 예시적인 흐름도를 도시한다.

도 8은 비즈니스 모델을 개선하기 위한 방법의 예시적인 흐름도를 도시한다.

도 9는 비즈니스 모델을 구성하기 위한 방법의 예시적인 흐름도를 도시한다.

도 10은 본 발명의 원리들을 위한 적당한 동작 환경을 도시한다.

<도면의 주요 부분에 대한 부호의 설명>

100 : 컴퓨터 아키텍처

101 : 컴퓨터 시스템

111 : 사용자 인터페이스

112 : 모델 프로세싱 모듈들

121 : 비즈니스 기능 계층

131 : 서비스 네트워크 계층

141 : 비즈니스 프로세스 흐름 계층

151 : 비즈니스 편성 계층

161 : 결과들

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 비즈니스 모델링에 관한 것으로서, 보다 구체적으로는, 비즈니스 모델들을 비교하고 대조하는 것에 관한 것이다.

비즈니스들은 복잡할 수 있으며 대개는 서로 관련되어 있으며 상호 의존적인 다수의 비즈니스 평선들로부터 (어쩌면 특별한 방식으로) 어셈블링된다. 일부 환경들에서, 비즈니스 평선들(예를 들어, 아웃소싱된 비즈니스 평선들)은, 어셈블리 및 분석의 복잡도를 증가시키며, 심지어 다수 기업들에 미친다. 불행스럽게도, 비즈니스 평선들의 상호 작용들과 상호 의존성들은 (동일한 기업들에서든 상이한 기업들에서든) 대부분 만족스럽게 정의되지 않으며 그리고/또는 만족스럽게 문서화되지 않는다. 다시 말해, 비즈니스의 고레벨 뷰는 대부분, 비즈니스가 동작하는 방법의 무수한 세부 사항들에 만족스럽게 연결되지 않는다. 예를 들어, 관리자 또는 소유자는, 비즈니스가 어셈블링되는 방식에 관한 지식을 일률적으로 전달되기 어려운 방식으로 (예를 들어, 비공식적인 "후계자 방식으로") 보유할 수 있다. 따라서, (예를 들어, 기업의 다른 부서들의) 다른 관리자 또는 피고용자들은 이러한 지식을 액세스하기가 어려울 수 있고, 그들이 그러한 지식을 액세스할 수 있다고 하더라도, 그것은 그들이 획득할 가치가 있는 맥락이 아닐 수도 있다.

그러나, 비즈니스 기능의 변화들을 구현하는 것은 대부분 다수 개개인들의 작업을 포함하는데, 다수 개개인들 중 대다수는 비즈니스가 전반적으로 어셈블링되는 방식을 알지 못하며 판정할 수 없고, 맥락 및 언어의 차이들로 인해, 그들만의 뷰들은 다른 사람들의 뷰들에 "연결"되거나 관련될 수 없다. 그에 따라, 비즈니스 평선을 계획된 이벤트로서 추가하거나 삭제하는 경우라 하더라도, 비즈니스 평선의 추가 또는 삭제가, 서로 관련되어 있으며 상호 의존적인 다른 비즈니스 평선들에 어떻게 영향을 미칠 것인지를 판정하기 위한 효율적인 메커니즘이 존재하지 않을 수도 있다. 따라서, 일단 구현된 비즈니스에 대한 변화들이 예상치 못한 (그리고 대개는 부정적인) 영향들을 초래할 수도 있다.

그에 따라, 비즈니스 평선들의 일반적인 이해는 판정하기 어려울 수 있지만, 예를 들어, 비용들, 이점들, 위험들, 대안들, 및 의존성들의 관점에서의 변화를 준비하기 위해서는, 비즈니스에 대한 이러한 이해가 중요하다. 예를 들어, 관리자는, 규범적인 안내(예를 들어, 최상 관행(best practices))가 이용 가능하고 비즈니스 실행을 모니터링하며 비즈니스 실행에 따를 것을 강제할 수 있을 경우, 규범적인 안내에 따라 비즈니스 평선들을 전개하고 관리할 수 있기를 원할 수 있다. 따라서, 비즈니스 평선 정보를 좀더 효율적으로 유포할 수 있게 하는 다양한 기술들이 개발되어 비즈니스들을 모델링하고 표현하여 왔다.

일부 모델링 기술들은, 작업이 수행되는 방식을 설명하는 비즈니스 프로세스들을 표현하는 도면들의 수동적인 생성을 포함한다. 예를 들어, 훈련된 개개인들은 비즈니스의 모든 태양들을 분석하여 비즈니스 평선들을 식별하고 비즈니스 평선들 간의 상호 관계들 및 상호 의존성들을 식별할 수 있다. 분석에 기초해, 훈련된 개개인들은 대표적인 도면들을 발생시킬 수 있다. 그러나, 비즈니스 평선 관점으로부터의 정확한 비즈니스 분석은 더 긴 시간이 걸릴 수 있다. 또한, 일단 대표적인 도면들이 발생되고 나면, 이러한 도면들은 쉽게 변경되지 않는다.

그러나, 많은 비즈니스 프로세스들은 동적이므로(즉, 시간에 따라 빈번하게 변하므로), 비즈니스 프로세스들의 수동적으로 발생된 표현은, 그것이 완료되기도 전에, 대부분 시대에 뒤진 것이 된다. 또한, 비즈니스 프로세스들의 수동적으로 발생된 표현이 그것이 발생된 당시에는 정확했다고 하더라도, 비즈니스 표현이 발생된 이후의 비즈니스 관련 프로세스들의 모든 변화는 비즈니스 표현을 부정확하게 할 것이고, 알아채지 못한 상태로 시대에 뒤진 표현에 기초하는 모든 판정들은 위험할 것이다. 그에 따라, 수동적으로 발생된 표현들은, 다양한 비즈니스 기능들에 대한 시뮬레이션된 그리고/또는 가정에 기초한 변화들이 비즈니스에 어떻게 영향을 미칠 것인지를 판정하기 위한, 존재한다면, 신뢰할 수 있지만 제약된 기능을 비즈니스에 제공한다.

수동적으로 발생된 비즈니스 표현들에서의 결함들에 대한 적어도 부분적인 결과로서, 비즈니스 표현들을 발생시키기 위한 일부 컴퓨터화된 기술들이 개발되어 왔다. 이러한 컴퓨터화된 기술들은 다양한 기술들을 사용해 비즈니스를 표현하고, 필

요한 비즈니스 평선들은 대부분, 그러한 평선들을 지원하는 비즈니스 프로세스들 및 세부 절차들의 모델링에 초점을 둔다. 예를 들어, 일부 컴퓨터화된 기술들은 사용자-인터페이스에서 비즈니스 프로세스들의 그래픽 뷰를 표현한다. 다소 제약된 범위에서, 이들 그래픽 뷰들은 상이한 비즈니스 기능들의 비즈니스에 대한 영향을 시뮬레이션하도록 변경될 수 있다.

그러나, 수동 생성 모델 및 컴퓨터 생성 모델 모두가 통상적으로 비조직적이므로, 변경된 레벨들의 세부 사항을 제공하기 위한 메커니즘이 부족하다. 예를 들어, 동일한 비즈니스 평선의 (예를 들어, 상급 관리를 위한) 더 높은 레벨의 뷰 및 동시에 (예를 들어, 비즈니스 평선을 구현 중인 피고용자들을 위한) 더 낮은 레벨의 뷰 모두를 제공할 수 있는 단일 모델을 효율적으로 발생시키기는 어렵다(그리고 수동적으로 발생된 모델들로는 본질적으로 불가능하다). 또한, 이러한 기술들은 통상적으로 형식 연산자들(formal operators)이 부족한 비즈니스 모델들을 발생시킨다. 그에 따라, 컴퓨터 생성 모델들을 조작하기 위한 방법이 존재하지 않으므로, 컴퓨터 생성 모델들조차 한정된 유용성을 가질 수 있다. 예를 들어, 형식 연산자들없이, 유사한 유형의 비즈니스 모델들(similarly typed business models)을 비교하거나, 상이한 유형의 비즈니스 모델들을 대조하거나, 비즈니스 모델의 인스턴스를 제약하거나, 예를 들어, 특정 산업을 위해, 기초가 되는 비즈니스 모델을 개선하거나, 다른 비즈니스 모델들의 집합으로부터 비즈니스 모델을 구성할 방법이 없을 수도 있다.

모델 비교와 관련하여, 종래의 모델링 기술들은 통상적으로, 비즈니스가 스스로를, 그것의 연산 모델(비즈니스 모델)을, 또는 그것의 구성을 다른 비즈니스들, 연산의 다른 모델들, 또는 다른 비즈니스 구성들과 비교하기 위한 형식 연산자들을 포함하지 않는다. 예를 들어, 비즈니스들의 일부분들은 말할 필요도 없고, 통상적으로는 유사한 비즈니스들을 서로 비교하거나 비즈니스를 최상 관행의 비즈니스 모델들 및/또는 구성들과 비교하기 위한 방법이 존재하지 않는다. 이처럼 비즈니스들을 비교할 수 없는 것이, 비즈니스가 시장에서 얼마나 효과적인지에 대한 그리고/또는 비즈니스가 (예를 들어, 추가적인 비즈니스 평선들을 구현하거나, 중복되는 비즈니스 평선들을 삭제하거나, 비즈니스 평선들의 소싱과 같은 연산 관정들을 변경하는 것을 통해) 연산들을 얼마나 향상시킬 수 있는지에 대한 비즈니스의 이해 및 분석을 한정할 수 있다.

제약들(constraints)과 관련하여, 종래의 모델링 기술들은 통상적으로, 비즈니스 모델들을 제약하여 소정 제약들을 충족시키기 위한 또는 비즈니스 모델이 소정 제약들에 따르는지를 검사하기 위한 형식 연산자들을 포함하지 않는다. 그에 따라, 비즈니스 모델이 소정 비즈니스 평선을 포함하는 경우라 하더라도, 비즈니스 평선을 특정 용도로 제약하거나 비즈니스 평선이 특정 용도를 위해 충분히 제약되었는지를 판정하기 위한 방법은 존재하지 않을 수도 있다. 예를 들어, 운송 외부 위탁자(transportation outsourcer)는 크로스-도킹 기능(cross-docking functionality)(도크의 한쪽에 내려지는 일 유형의 제품으로 채워진 트럭들, 도크의 다른 쪽으로부터 실려지는 다양한 제품들을 선별하는 트럭들)을 포함하는 다수의 비즈니스 기능들을 포함할 수 있다. 그러나, 비즈니스가, 운송 외부 위탁자가 크로스-도킹 기능을 가지는지를 판정하기 위한 방법이 존재하지 않을 수도 있다. 그에 따라, 비즈니스는 알지 못하는 상태에서 운송 외부 위탁자와 계약한 다음 나중에야 운송 외부 위탁자가 크로스-도킹 기능이 없다는 것을 알게 될 수도 있다.

구성과 관련하여, 종래의 모델링 기술들은 통상적으로 모델들의 풀로부터의 상이한 모델들(또는 상이한 모델들의 일부분들)을 액세스하고 액세스된 모델들(또는 상이한 모델들의 일부분들)을 새로운 모델로 구성하기 위한 형식 연산자들을 포함하지 않는다. 예를 들어, 소매 बैं킹 모델, 트랜잭션 बैं킹 모델, 및 제품 बैं킹 모델의 소정 모델들의 요소들로부터 혁신적인 새로운 बैं킹 비즈니스용 비즈니스 모델을 구성하기 위한 방법이 존재하지 않을 수도 있다. 그에 따라, 새로운 비즈니스는 비즈니스 또는 산업의 기존 노력들을 신뢰할 수 있는 정확한 방식으로 조합하는 것에 의해 혁신적인 비즈니스 모델을 생성하기 위한 방법을 가지고 있지 않을 수도 있다.

개선과 관련하여, 종래의 모델링 기술들은 통상적으로 좀더 일반적인 비즈니스 모델을 (어쩌면 하나 이상의) 좀더 특수한 비즈니스 모델들로 개선하기 위한 형식 연산자들을 포함하지 않는다. 예를 들어, 미수금 계정 계정 평선(accounts receivable function)은 상이한 산업들(건강 관리, बैं킹, 소매 등)에 걸쳐, 상이한 규모의 비즈니스(예를 들어, 단독 소유권, 중견 기업, 엔터프라이즈 등)에 걸쳐, 그리고 상이한 지리적 위치들(예를 들어, 홍콩, 유럽, 미국 등)에서 약간의 차이점들을 가질 수 있다. 그러나, 미수금 계정 계정 모델이 특정 산업에, 특정 규모의 비즈니스에, 또는 특정한 지리적 위치에 적합하도록, 일반적인 미수금 계정 계정 모델을 개선하기 위한 방법은 존재하지 않을 수도 있다. 그에 따라, 좀더 일반적인 미수금 계정 계정 평선에 기초해 비즈니스를 모델링하는 것은 부정확한 (또는 적어도 불완전한) 비즈니스 모델을 발생시킬 수 있다. 그 결과, (기능이 정확하게 모델링되지 못할 수도 있으므로) 비즈니스는 그것의 실제적인 비즈니스 기능을 다른 모델들과 비교하기 위한 방법을 가지고 있지 않을 수도 있다.

일부 환경들에서, (수동적이든 컴퓨터화되었든) 모델링 기술들은, 예를 들어, 편성 구조들, 절차들, 프로세스 흐름들, 및 지원 기술과 같은, 비즈니스 데이터의 상이한 계층들(또는 유형들)을 모델링(통상적으로 조합)하는 데에 사용된다. 그에 따라, 다양하고 상이한 비즈니스 모델들을 통해 비즈니스가 표현될 수 있다. 그러나, 비즈니스 모델링 메커니즘들은 통상적

으로, 동일한 비즈니스를 위한 상이한 유형들의 관련 모델들간의 관계들을 대조하거나 식별하기 위한 형식 연산자들을 포함하지 않는다. 예를 들어, 통상적으로는 비즈니스 평선들의 모델과 그러한 비즈니스 평선들을 구현하는 데에 사용되는 서비스 네트워크(예를 들어, 웹 기반 서비스들)의 대응하는 모델간의 관계들을 식별하기 위한 메커니즘이 존재하지 않는다.

이처럼 상이한 유형들의 모델들을 대조할 수 없는 것이, 상이한 비즈니스 태양들(예를 들어, 연산들, 서비스들, 직원들 등)이 서로 상호 동작하고 의존하는 방식에 대한 비즈니스의 이해를 한정할 수 있다. 예를 들어, 비즈니스는, 특정 비즈니스 연산의 추가가 하나 이상 서비스들의 대응하는 추가를 획득할 것인지를 알 수 있는 방법을 가지고 있지 않을 수도 있다. 마찬가지로, 비즈니스는, 서비스가 대응하는 비즈니스 연산들에 의해 충분히 활용되고 있지 않다는 것(그리고 그에 따라 어찌면 다른 비즈니스 연산들을 위해 사용될 수도 있다는 것)을 알 수 있는 방법을 가지고 있지 않을 수도 있다. 또한, 비즈니스는, 특정 비즈니스 평선의 아웃소싱 관정이 비즈니스 또는, (예를 들어, 웹 기반 서비스들의) 서비스들의 네트워크와 같은, 그것의 관련 모델들의 연산에 어떤 영향들을 미칠 수 있는지를 신뢰할 수 있게 그리고 정확하게 관정하기 위한 방법을 가지고 있지 않을 수도 있다.

따라서, 비즈니스 모델들의 융통성 있는 조작을 용이하게 하는 시스템들, 방법들, 컴퓨터 프로그램 제품들, 및 데이터 구조들이 바람직하다.

발명이 이루고자 하는 기술적 과제

종래 기술 상태와의 상기 문제들은, 비즈니스 모델들을 비교하고 대조하기 위한 방법들, 시스템들, 컴퓨터 프로그램 제품들, 및 데이터 구조들에 관한 본 발명의 원리들에 의해 극복된다. 일부 환경들에서, 컴퓨터 시스템은, 제1의 구조화된 비즈니스 모델인, 기존 비즈니스 아키텍처를 표현하는 (데이터 모델에 따라 구조화된) 구조화된 비즈니스 모델의 제1 인스턴스를 액세스한다. 컴퓨터 시스템은 제2 비즈니스 아키텍처를 표현하는 (데이터 모델에 따라 구조화된) 구조화된 비즈니스 모델의 제2 인스턴스를 액세스한다. 컴퓨터 시스템은 구조화된 비즈니스 모델의 제1 인스턴스를 구조화된 비즈니스 모델의 제2 인스턴스와 비교한다. 컴퓨터 시스템은 기존 비즈니스 아키텍처와 제2 비즈니스 아키텍처간의 어떤 차이들을 식별한다. 컴퓨터 시스템은 기존 비즈니스 아키텍처와 제2 비즈니스 아키텍처간의 식별된 차이들에 기초해, 기존 비즈니스 아키텍처의 충분성(sufficiency)을 평가한다.

다른 실시예들에서, 컴퓨터 시스템은 비즈니스 아키텍처의 제1 비즈니스 계층을 표현하는 구조화된 비즈니스 모델을 액세스한다. 구조화된 비즈니스 모델은 구조화된 데이터 모델에 따라 제1 비즈니스 계층의 하나 이상의 제1 비즈니스 계층 컴포넌트들을 모델링한다. 컴퓨터 시스템은 제2 비즈니스 계층을 표현하는 제2의 구조화된 모델을 액세스한다. 제2의 구조화된 모델은 상이한 제2 비즈니스 계층의 하나 이상의 제2 비즈니스 계층 컴포넌트들을 모델링한다. 컴퓨터 시스템은, 제1 비즈니스 계층의 구성 변화들이 어떻게 제2 비즈니스 계층에 영향을 미치며 제2 비즈니스 계층의 구성 변화들이 어떻게 제1 비즈니스 계층에 영향을 미치는지가 관정될 수 있도록, 하나 이상의 제1 비즈니스 계층 컴포넌트들과 하나 이상의 제2 비즈니스 계층 컴포넌트들간의 관계들을 매핑한다. 컴퓨터 시스템은 매핑된 관계들을 이용해 구조화된 비즈니스 모델을 제2의 구조화된 모델과 대조한다.

추가적인 실시예들에서, 컴퓨터 시스템은 유효한 비즈니스 아키텍처를 표현하는 제1 비즈니스 컴포넌트들의 (데이터 모델에 따라 구조화된) 구조화된 비즈니스 모델을 액세스한다. 제1 비즈니스 컴포넌트들은, 하나 이상의 제약들을 준수함을 나타내는 대응하는 제1 특성 값들을 가진다. 컴퓨터 시스템은 기존 비즈니스 아키텍처를 표현하는 제2 비즈니스 컴포넌트들의 제2의 구조화된 비즈니스 모델을 액세스하는데, 제2 비즈니스 컴포넌트들은 제2 특성 값들을 가진다. 컴퓨터 시스템은, 제2 특성 값들이 하나 이상의 제약들을 준수하는지를 검사하기 위해, 적어도 준수하는 제1 특성 값들의 부분집합을 적어도 제2 특성 값들의 대응하는 부분집합과 비교한다. 컴퓨터 시스템은, 비교에 기초해, 제2의 구조화된 비즈니스 모델이 유효한지를 관정한다.

또 다른 실시예들에서, 컴퓨터 시스템은 비즈니스 아키텍처를 표현하는 (데이터 모델에 따라 구조화된) 구조화된 비즈니스 모델을 액세스한다. 컴퓨터 시스템은, 구조화된 비즈니스 모델에 기초해 모델링될 특정 비즈니스 아키텍처를 식별하는데, 특정 비즈니스 아키텍처는 비즈니스 아키텍처보다 좀더 구체적이다. 컴퓨터 시스템은 특정된 비즈니스 아키텍처를 위한 비즈니스 특유 데이터를 액세스한다. 컴퓨터 시스템은, 액세스되는 비즈니스 특유 데이터에 기초해, 구조화된 비즈니스 모델을 비즈니스 특유의 구조화된 데이터 모델로 개선한다. 컴퓨터 시스템은 특정된 비즈니스 아키텍처를 비즈니스 특유의 구조화된 데이터 모델에 따라 모델링한다.

또 다른 실시예들에서, 컴퓨터 시스템은, 한 집합의 비즈니스 컴포넌트들을 포함하는 비즈니스 아키텍처가 모델링될 것이라는 표시를 수신한다. 컴퓨터 시스템은, 비즈니스 컴포넌트들의 집합의 제1 부분이, 복수개의 상이한 비즈니스 모델들 중에서 선택되는 제1의 구조화된 비즈니스 모델에 포함되어 있다는 것을 식별한다. 컴퓨터 시스템은 제1의 구조화된 비즈니스

스 모델로부터 비즈니스 컴포넌트들의 집합의 제1 부분을 추출한다. 컴퓨터 시스템은, 비즈니스 컴포넌트들의 집합의 제2 부분이, 복수개의 상이한 비즈니스 모델들 중에서 선택되는 제2의 구조화된 비즈니스 모델에 포함되어 있다는 것을 식별한다. 컴퓨터 시스템은 제2의 구조화된 비즈니스 모델로부터 비즈니스 컴포넌트들의 집합의 제2 부분을 추출한다. 컴퓨터 시스템은 비즈니스 컴포넌트들의 집합의 적어도 제1 부분 및 제2 부분을 비즈니스 아키텍처의 모델로 구성한다.

본 발명의 이들 및 다른 사양들이 다음의 명세서 및 첨부된 청구항들로부터 좀더 명백해지거나, 다음에서 기술되는 바와 같은 본 발명의 실시예에 의해 학습될 수 있을 것이다.

발명의 구성 및 작용

본 발명의 상기 및 다른 이점들과 사양들을 좀더 명료하게 하기 위해, 첨부된 도면들에 도시되어 있는 본 발명의 구체적인 실시예들을 참조하여 본 발명에 대한 좀더 구체적인 설명이 제공될 것이다. 이들 도면들은 본 발명의 통상적인 실시예들만을 도시할 뿐이므로 본 발명의 범위를 한정하는 것으로 간주되어서는 안된다는 것을 알 수 있을 것이다. 첨부 도면들을 사용해, 본 발명을 좀더 구체적으로 그리고 상세하게 묘사하고 설명할 것이다.

본 발명의 원리들은 비즈니스 모델들의 비교 및 대조를 제공한다. 일부 실시예들에서, 컴퓨터 시스템은, 제1의 구조화된 비즈니스 모델인, 기존 비즈니스 아키텍처를 표현하는 (데이터 모델에 따라 구조화된) 구조화된 비즈니스 모델의 제1 인스턴스를 액세스한다. 컴퓨터 시스템은 제2 비즈니스 아키텍처를 표현하는 (데이터 모델에 따라 구조화된) 구조화된 비즈니스 모델의 제2 인스턴스를 액세스한다. 컴퓨터 시스템은 구조화된 비즈니스 모델의 제1 인스턴스를 구조화된 비즈니스 모델의 제2 인스턴스와 비교한다. 컴퓨터 시스템은 기존 비즈니스 아키텍처와 제2 비즈니스 아키텍처간의 어떤 차이들을 식별한다. 컴퓨터 시스템은, 기존 비즈니스 아키텍처와 제2 비즈니스 아키텍처간의 식별된 차이들에 기초해, 기존 비즈니스 아키텍처의 충분성을 평가한다.

다른 실시예들에서, 컴퓨터 시스템은 비즈니스 아키텍처의 제1 비즈니스 계층을 표현하는 구조화된 비즈니스 모델을 액세스한다. 구조화된 비즈니스 모델은 구조화된 데이터 모델에 따라 제1 비즈니스 계층의 하나 이상의 제1 비즈니스 계층 컴포넌트들을 모델링한다. 컴퓨터 시스템은 제2 비즈니스 계층을 표현하는 제2의 구조화된 모델을 액세스한다. 제2의 구조화된 모델은 상이한 제2 비즈니스 계층의 하나 이상의 제2 비즈니스 계층 컴포넌트들을 모델링한다. 컴퓨터 시스템은, 제1 비즈니스 계층의 구성 변화들이 어떻게 제2 비즈니스 계층에 영향을 미치며 제2 비즈니스 계층의 구성 변화들이 어떻게 제1 비즈니스 계층에 영향을 미치는지가 판정될 수 있도록, 하나 이상의 제1 비즈니스 계층 컴포넌트들과 하나 이상의 제2 비즈니스 계층 컴포넌트들간의 관계들을 매핑한다. 컴퓨터 시스템은 매핑된 관계들을 이용해 구조화된 비즈니스 모델을 제2의 구조화된 모델과 대조한다.

추가적인 실시예들에서, 컴퓨터 시스템은 유효한 비즈니스 아키텍처를 표현하는 제1 비즈니스 컴포넌트들의 (데이터 모델에 따라 구조화된) 구조화된 비즈니스 모델을 액세스한다. 제1 비즈니스 컴포넌트들은, 하나 이상의 제약들을 준수함을 나타내는 대응하는 제1 특성 값들을 가진다. 컴퓨터 시스템은 기존 비즈니스 아키텍처를 표현하는 제2 비즈니스 컴포넌트들의 제2의 구조화된 비즈니스 모델을 액세스하는데, 제2 비즈니스 컴포넌트들은 제2 특성 값들을 가진다. 컴퓨터 시스템은, 제2 특성 값들이 하나 이상의 제약을 준수하는지를 검사하기 위해, 적어도 준수하는 제1 특성 값들의 부분집합을 적어도 제2 특성 값들의 대응하는 부분집합과 비교한다. 컴퓨터 시스템은, 비교에 기초해, 제2의 구조화된 비즈니스 모델이 유효한지를 판정한다.

또 다른 실시예들에서, 컴퓨터 시스템은 비즈니스 아키텍처를 표현하는 (데이터 모델에 따라 구조화된) 구조화된 비즈니스 모델을 액세스한다. 컴퓨터 시스템은, 구조화된 비즈니스 모델에 기초해 모델링될 특정 비즈니스 아키텍처를 식별하는데, 특정 비즈니스 아키텍처는 비즈니스 아키텍처보다 좀더 구체적이다. 컴퓨터 시스템은 특정된 비즈니스 아키텍처를 위한 비즈니스 특유 데이터를 액세스한다. 컴퓨터 시스템은, 액세스되는 비즈니스 특유 데이터에 기초해, 구조화된 비즈니스 모델을 비즈니스 특유의 구조화된 데이터 모델로 개선한다. 컴퓨터 시스템은 특정된 비즈니스 아키텍처를 비즈니스 특유의 구조화된 데이터 모델에 따라 모델링한다.

또 다른 실시예들에서, 컴퓨터 시스템은, 한 집합의 비즈니스 컴포넌트들을 포함하는 비즈니스 아키텍처가 모델링될 것이라는 표시를 수신한다. 컴퓨터 시스템은, 비즈니스 컴포넌트들의 집합의 제1 부분이, 복수개의 상이한 비즈니스 모델들 중에서 선택되는 제1의 구조화된 비즈니스 모델에 포함되어 있다는 것을 식별한다. 컴퓨터 시스템은 제1의 구조화된 비즈니스 모델로부터 비즈니스 컴포넌트들의 집합의 제1 부분을 추출한다. 컴퓨터 시스템은, 비즈니스 컴포넌트들의 집합의 제2 부분이, 복수개의 상이한 비즈니스 모델들 중에서 선택되는 제2의 구조화된 비즈니스 모델에 포함되어 있다는 것을 식별한다. 컴퓨터 시스템은 제2의 구조화된 비즈니스 모델로부터 비즈니스 컴포넌트들의 집합의 제2 부분을 추출한다. 컴퓨터 시스템은 비즈니스 컴포넌트들의 집합의 적어도 제1 부분 및 제2 부분을 비즈니스 아키텍처의 모델로 구성한다.

본 발명의 범위내의 실시예들은 컴퓨터-실행 가능 명령어들 또는 데이터 구조들을 전달하기 위한 또는 컴퓨터-실행 가능 명령어들 또는 데이터 구조들이 저장되어 있는 컴퓨터-판독 가능 매체들을 포함한다. 이러한 컴퓨터-판독 가능 매체들은, 범용 또는 특수 목적 컴퓨터 시스템에 의해 액세스될 수 있는, 이용 가능한 임의의 매체들일 수 있다. 한정이 아닌 일례로써, 이러한 컴퓨터-판독 가능 매체들은, RAM, ROM, EPROM, CD-ROM이나 다른 광 디스크 저장 장치, 자기 디스크 저장 장치나 다른 자기 저장 장치들 또는, 소정의 프로그램 코드 수단들을 컴퓨터-실행 가능 명령어들, 컴퓨터-판독 가능 명령어들, 또는 데이터 구조들의 형태로 전달하거나 저장하는 데에 사용될 수 있으며 범용 또는 특수-목적 컴퓨터 시스템에 의해 액세스될 수 있는 임의의 다른 매체들과 같은, 물리적 저장 매체들을 포함할 수 있다.

본 명세서에서 그리고 후술되는 청구항들에서, "컴퓨터 네트워크"는 컴퓨터 시스템들 및/또는 모듈들 사이에서 전자 데이터의 전송을 가능하게 하는 하나 이상의 데이터 링크들로서 정의된다. 컴퓨터 네트워크 또는 다른 통신 접속(유선, 무선, 또는 유선과 무선의 조합)을 통해 컴퓨터 시스템으로 정보가 전송되거나 제공될 때, 이 접속은 컴퓨터-판독 가능 매체로 보는 것이 적당하다. 그에 따라, 임의의 이러한 접속은 컴퓨터-판독 가능 매체라고 명명되는 것이 적당하다. 상기 조합들 또한 컴퓨터-판독 가능 매체들의 범위내에 포함되어야 한다. 컴퓨터-실행 가능 명령어들은, 예를 들어, 범용 컴퓨터 시스템 또는 특수-목적 컴퓨터 시스템으로 하여금 소정 평선 또는 평선들의 그룹을 수행하게 하는 명령어들 및 데이터를 포함한다. 컴퓨터 실행 가능 명령어들은, 예를 들어, 바이너리들(binaries), 어셈블리어와 같은 중간 포맷 명령어들, 또는 심지어 소스 코드일 수도 있다.

본 명세서에서 그리고 후술되는 청구항들에서, "컴퓨터 시스템"은, 다같이 전자 데이터에 대한 연산들을 수행하는, 하나 이상의 소프트웨어 모듈들, 하나 이상의 하드웨어 모듈들, 또는 그들의 조합들로서 정의된다. 예를 들어, 컴퓨터 시스템의 정의는 개인용 컴퓨터의 하드웨어 컴포넌트들 뿐만 아니라, 개인용 컴퓨터의 운영 시스템과 같은, 소프트웨어 모듈들도 포함한다. 모듈들의 물리적 레이아웃은 중요하지 않다. 컴퓨터 시스템은 컴퓨터 네트워크를 통해 커플링되어 있는 하나 이상의 컴퓨터들을 포함할 수 있다. 마찬가지로, 컴퓨터 시스템은, (메모리 및 프로세서와 같은) 내장형 모듈들이 다같이 전자 데이터에 대한 연산들을 수행하는 (휴대폰 또는 PDA(Personal Digital Assistant)와 같은) 단일 물리 장치를 포함할 수도 있다.

본 명세서에서 그리고 후술되는 청구항들에서, "비즈니스 계층"은 비즈니스에 대한 소정 특징들의 뷰로서 정의된다. 예를 들어, 비즈니스는 그것의 편성 구조, 그것의 비즈니스 기능들, 그것의 비즈니스 프로세스들, 그것의 서비스 네트워크 등에 기초해 고찰될 수 있다. 그에 따라, 비즈니스는 대응하는 편성 계층, 기능 계층, 프로세스 흐름 계층, 서비스 네트워크 계층 등을 포함할 수 있다.

본 명세서에서 그리고 후술되는 청구항들에서, "비즈니스 아키텍처"는 비즈니스의 적어도 일부분에 대한 전반적인 설계(overall design)로서 정의된다. 회사 또는 회사의 하나 이상의 부분들을 위한 비즈니스 아키텍처는 회사의 내부 및/또는 외부의 다양한 경계들에 미치는 하나 이상의 비즈니스 계층들을 포함할 수 있다. 예를 들어, 회사의 비즈니스 아키텍처가 외부적으로는 물리 경계들(예를 들어, 벽들, 빌딩들 등)에 미칠 수 있고, 내부적으로는 물리 경계들(예를 들어, 구획들, 부서들 등) 및 논리 경계들(예를 들어, 회계 연도 끝, 감지되는 서비스 경계, 보안 등)에 미칠 수 있다. 그에 따라, 아웃소싱된 비즈니스 기능은, 비록 아웃소싱된 비즈니스 기능이 회사에 의해 수행되지 않는 경우라고 하더라도, 회사를 위한 비즈니스 아키텍처의 일부로 간주될 수 있다. 비즈니스 아키텍처들은 전체 비즈니스 또는 비즈니스의 하나 이상의 부분들에 대한 과거, 현재(있는 그대로), 또는 미래(장래)의 아키텍처들일 수 있다. 비즈니스의 일부분은 비즈니스 기능들의 소정 서브-네트워크 또는 서브-네트워크들의 집합을 포함할 수 있다.

일반적으로, 상이한 비즈니스 계층들에 대응하는 비즈니스 모델들의 상이한 유형들에 대한 안정성(또는 휘발성)은 달라질 수 있다. 다시 말해, 비즈니스 모델의 일 유형은 비즈니스 모델들의 다른 유형들에 비해 좀더 안정하거나 덜 안정할 수 있다. 예를 들어, 비즈니스 절차들을 모델링하는 비즈니스 절차 모델은 비즈니스 편성 구조를 모델링하는 비즈니스 편성 모델보다 더 안정할 수 있다. 한편, 비즈니스 절차들을 모델링하는 비즈니스 절차는 비즈니스 기능들을 모델링하는 비즈니스 기능 모델보다 덜 안정할 수 있다.

본 명세서 및 후술되는 청구항들에서, "스키마"는, 공유되는 표현된 어휘에 따라 복수개 컴퓨터 시스템들 또는 모듈들이 데이터를 프로세싱할 수 있게 하는, 복수개 컴퓨터 시스템들 또는 모듈들 사이에서 공유되는 어휘 표현으로서 정의된다. 스키마는 스키마 언어의 구문들(constructs: 예를 들어, 이름/값 쌍들)을 사용해 데이터의 클래스들을 정의하고 설명할 수 있다. 스키마 구문들은, 예를 들어, 비즈니스 기능 모델과 같은, 특정 어플리케이션에서 사용되는, 데이터 유형들, 요소들과 그들의 내용, 속성들과 그들의 값들, 엔티티들과 그들의 내용들, 및 표기 방법들의 의미, 용도, 및 관계들을 한정하고 문서

화하는 데에 사용될 수 있다. 그에 따라, 스키마를 액세스할 수 있는 임의의 컴퓨터 시스템 또는 모듈은 스키마에 따라 데이터를 프로세싱할 수 있다. 또한, 스키마를 액세스할 수 있는 임의의 컴퓨터 시스템 또는 모듈은 이 또한 스키마를 액세스할 수 있는 다른 컴퓨터 시스템들 및/또는 모듈들에 의한 사용을 위해 데이터를 구성하거나 변경할 수 있다.

스키마는 데이터 구조들을 정의하는 데에 사용되는 논리, 2진, 8진, 10진, 16진, 정수, 부동 소수점, 문자, 문자열, 사용자-정의의 데이터 유형들, 및 이러한 데이터 유형들의 조합들을 포함하는 임의의 데이터 유형을 가상적으로 정의하는데 이용될 수 있다. 사용자-정의의 데이터 유형들의 일부 예들로는 비즈니스 기능 특성들, 비즈니스 기능 입력들과 출력들, 비즈니스 기능 프로세스들, 비즈니스 기능 접속들, 및 비즈니스 기능 서비스 레벨 기대값들을 들 수 있다. 또한, 데이터 유형은 스키마 계층 구조의 다른 데이터 유형들로의 링크를 참조하도록 정의될 수도 있다.

XML(eXtensible Markup Language) 스키마는 스키마 유형의 일례이다. XML 스키마는 XML 스키마 언어의 스키마 구문들(예를 들어, 이름/값 쌍들)을 사용해 XML 문서들의 클래스를 정의하고 설명할 수 있다. 이들 스키마 구문들은, XML 문서들에서 사용되는, 데이터 유형들, 요소들과 그들의 내용, 속성들과 그들의 값들, 엔티티들과 그들의 내용들, 및 표기 방법들의 의미, 용도, 및 관계들을 한정하고 문서화하는 데에 사용될 수 있다. 그에 따라, 스키마는, 예를 들어, ".dtd" 확장자로 끝나는 DTD 파일들과 같은, DTD(Document Type Definitions) 및, 예를 들어, ".xsd" 확장자로 끝나는 XML 스키마 파일들과 같은, W3C(World Wide Web Consortium) XML 스키마들도 포함하도록 정의된다. 그러나, 특정한 DTD 또는 XML 스키마를 위한 실제파일 확장자는 중요하지 않다.

당업자들은, 본 발명이, 개인용 컴퓨터들, 랩탑 컴퓨터들, 핸드-헬드 장치들, 멀티-프로세서 시스템들, 마이크로프로세서-기반 또는 프로그램 가능한 상용 전자 장치들, 네트워크 PC들, 미니 컴퓨터들, 메인프레임 컴퓨터들, 휴대폰들, PDA들, 페이지들 등을 포함하는, 다수 유형들의 컴퓨터 시스템 구성들을 갖춘 컴퓨터 네트워크 환경들에서 실시될 수 있다는 것을 알 수 있을 것이다. 또한, 본 발명은, 컴퓨터 네트워크를 통해 (유선 데이터 링크들이나 무선 데이터 링크들에 의해, 또는 유선 및 무선 데이터 링크들의 조합에 의해) 링크되어 있는, 로컬 및 원격 컴퓨터 시스템들 모두가 태스크들을 수행하는 분산 시스템 환경들에서 실시될 수도 있다. 분산 시스템 환경에서, 프로그램 모듈들은 로컬 및 원격 메모리 저장 장치들 모두에 위치할 수 있다.

도 1은 비즈니스 모델들을 비교하고 대조하는 데에 사용될 수 있는 예시적인 컴퓨터 아키텍처(100)를 도시한다. 컴퓨터 아키텍처(100)에서 도시된 바와 같이, 컴퓨터 시스템(101)은 사용자-인터페이스(111) 및 모델 프로세싱 모듈들(112)을 포함한다. 사용자-인터페이스(111)는 컴퓨터 시스템 사용자와 컴퓨터 시스템(101) 사이를 인터페이스하도록 구성되어 있다. 사용자-인터페이스(111)는 컴퓨터 시스템 사용자에게, 모델 프로세싱 모듈들(112)로 데이터를 입력하기 위한(예를 들어, 비즈니스 모델들에서 수행할 형식 연산들을 선택하기 위한) 그리고 모델 프로세싱 모듈들(112)의 결과들을 살펴보기 위한 인터페이스를 제공할 수 있다.

일반적으로, 모델 프로세싱 모듈들은 비즈니스 모델들에서 형식 연산들을 수행하도록 구성되어 있는 모듈들을 포함할 수 있다. 예를 들어, 모델 프로세싱 모듈들(112)은 비즈니스 모델들을 비교하도록 구성되어 있는 비교 모듈(113), 비즈니스 모델들 사이를 매핑하도록 구성되어 있는 매핑 모듈(114), 비즈니스 모델들이 제약들을 충족시키는지를 판정하도록 구성되어 있는 제약 모듈(116), 개선 기준들에 기초해 비즈니스 모델들을 개선하도록 구성되어 있는 개선 모듈(117), 및 다른 비즈니스 모델들로부터 비즈니스 모델을 구성하도록 구성되어 있는 구성 모듈을 포함한다. 그러나, 모델 프로세싱 모듈들(112)은 비즈니스 모델들에서 다른 형식 연산들을 수행하도록 구성되어 있는 다른 모듈들도 포함할 수 있다.

일반적으로, 컴퓨터 시스템(101)은 적합한 데이터 모델들, 구조화되지 않은 비즈니스 모델들, 및/또는 구조화되지 않은 비즈니스 데이터에 따라 생성된 비즈니스 모델들을 수신하도록 구성되어 있다. 구조화되지 않은 비즈니스 모델들 및/또는 구조화되지 않은 비즈니스 데이터를 수신하는 것에 응답하여, 컴퓨터 시스템(101)은 적합한 데이터 모델들을 참조하여 데이터 모델들에 따른 비즈니스 모델들을 생성할 수 있다. 도 3은 다음에서 부연될 비즈니스 기능 모델링 스키마(300)의 일례를 도시한다.

비즈니스 아키텍처내의 비즈니스 계층은 단일 데이터 모델을 사용해 모델링될 수 있다. 예를 들어, 하나의 비즈니스 기능 데이터 모델이 비즈니스 아키텍처의 비즈니스 기능 계층을 모델링하는 데에 사용될 수 있다. 그러나, 비즈니스 계층은 복수개의 상이한 데이터 모델들 중 하나를 사용해 모델링될 수도 있다. 예를 들어, 복수개의 상이한 비즈니스 기능 데이터 모델들 중 하나가 비즈니스 아키텍처의 비즈니스 기능 계층을 모델링하는 데에 사용될 수 있다.

또한, 상이한 비즈니스 아키텍처들내의 동일한 비즈니스 계층은 동일한 데이터 모델 또는 유사한 데이터 모델들을 사용해 모델링될 수 있다. 예를 들어, 제1 비즈니스 아키텍처의 비즈니스 기능 계층을 모델링하는 데에 사용되는 동일한 데이터

모델이 제2 비즈니스 아키텍처의 대응하는 비즈니스 기능 계층을 모델링하는데도 사용될 수 있다. 본 명세서 및 후술되는 청구항들에서, "유사한 유형의 비즈니스 모델들"은 동일한 데이터 모델 또는 유사한 데이터 모델에 기초한 모델들로서 정의된다.

그러나, 상이한 데이터 모델들이 상이한 비즈니스 아키텍처들내의 동일한 비즈니스 계층을 모델링하는 데에 사용될 수도 있다. 예를 들어, 제1 비즈니스 기능 데이터 모델은 제1 비즈니스 아키텍처의 비즈니스 기능 계층을 모델링하는 데에 사용되고 제2 비즈니스 기능 데이터 모델은 제2 비즈니스 아키텍처의 비즈니스 기능 계층을 모델링하는 데에 사용될 수 있다. 부가적으로, 상이한 데이터 모델들이 동일한 비즈니스 아키텍처의 상이한 비즈니스 계층들을 모델링하는 데에 사용될 수도 있다. 예를 들어, 비즈니스 기능 데이터 모델은 비즈니스 아키텍처의 비즈니스 기능 계층을 모델링하는 데에 사용되고 서비스 네트워크 데이터 모델은 비즈니스 아키텍처의 서비스 네트워크 계층을 모델링하는 데에 사용된다.

그에 따라, 컴퓨터 시스템(101)은 상이한 비즈니스 계층들(예를 들어, 비즈니스 기능 계층(121), 서비스 네트워크 계층(131), 비즈니스 프로세스 흐름 계층(141), 비즈니스 편성 계층(151) 등)에 대응하는 비즈니스 모델들을 액세스할 수 있다. 예를 들어, 컴퓨터 시스템(101)은 기능 모델(122), 최상 관행 기능 모델(123), 제안된 이득 기능 모델(124), 서비스 모델들(132 및 133), 프로세스 흐름 모델들(142 및 143), 및 편성 모델들(152 및 153) 중 하나 이상을 액세스할 수 있다. 명시적으로 도시된 계층들 이전의, 사이의, 그리고 이후의 수직으로 이어진 2개의 연속적인 마침표들(수직 생략 부호)은, 컴퓨터 아키텍처(100)가 추가적인 다른 계층들을 포함할 수도 있다는 것을 표현한다. 각 계층의 명시적으로 도시된 모델들 이전의, 사이의, 그리고 이후의 수평으로 이어진 2개의 연속적인 마침표들(생략 부호)은, 각 계층이 추가적인 다른 모델들을 포함할 수도 있다는 것을 표현한다.

컴퓨터 시스템(101)은 액세스되는 모델들에 대해 하나 이상의 형식 연산들(예를 들어, 비교, 매핑, 대조, 개선, 구성)을 수행할 수 있고 대응하는 결과들을 발생시킬 수 있다. 발생된 결과들은 사용자-인터페이스(111)에서 출력될 수 있고, 추가적인 프로세싱을 위해 다른 프로세싱 모듈들로 송신될 수 있으며, 그리고/또는 전자 메시지들을 통해 다른 컴퓨터 시스템들로 송신될 수 있다.

도 2는 비즈니스 모델들에서 수행될 수 있는 연산들의 예시적 아키텍처(200)를 도시한다. 아키텍처(200)는 기능 모델(202), 기능 모델(203), 및 기능 모델(204)을 포함하는 기능 모델들(201)을 포함한다. 기능 모델들(202, 203, 및 204)은, 예를 들어, 일반적인 미수금 계정 계정 기능, 일반적인 구매 주문 프로세싱 기능, 일반적인 크로스-도킹 기능 등과 같은, 좀더 일반적이거나 핵심적인 기능 모델들을 표현한다.

(예를 들어, 구성 모듈(118)에 구현되는) 구성 연산자(231)는, 기능 모델들(202, 203, 및 204)이 다같이 구성되어 기능 모델(211)을 발생시킬 수 있다는 것을 표현한다. 구성은 새로운, 상이한, 그리고/또는 조합된 기능 모델을 형성하기 위해 기능 모델들 또는 기능 모델들의 일부분들을 다같이 조합하는 단계를 포함할 수 있다. 예를 들어, 상업적 대여 기능, 주택 대여 기능, 및 신용 검사 기능은 조합되어 주택 용자 신청 프로세싱 기능을 발생시킬 수 있다.

(예를 들어, 개선 모듈(117)에서 구현되는) 개선 연산자(232)는 기능 모델들(202, 203, 및 204) 중 하나(예를 들어, 하나의 기능 모델을 개선할 경우) 또는 그 이상(구성 연산(132)과 함께 사용될 경우)을 개선하여 기능 모델(211)을 발생시킬 수 있다는 것을 표현한다. 개선은 기능 모델을, 예를 들어, 특정 산업에서의 사용을 위한 것과 같은, 특정 사용을 위한 기능 모델로 적응시키는 단계를 포함할 수 있다. 예를 들어, 일반적인 미수금 계정 비즈니스 기능은 제조업에서의 사용을 위해 제1 방식으로 개선될 수 있고, 건강 관리업에서의 사용을 위해 제2 방식으로 개선될 수 있다.

(예를 들어, 제약 모듈(116)에서 구현되는) 제약 연산자(235)는, 기능 모델(211)의 인스턴스가 기능 모델 인스턴스(214)로서 제약될 수 있다는 것을 표현한다. 기능 모델 인스턴스는 실세계 전개형 비즈니스 기능(real world deployed business capability)을 표현한다. 예를 들어, 미수금 계정 비즈니스 기능의 인스턴스는 특정 비즈니스 아키텍처(예를 들어, 소프트웨어 회사, 식품점 등)를 위한 미수금 계정 비즈니스 기능으로 제약될 수 있다.

(예를 들어, 비교 모듈(113)에서 구현되는) 비교 연산자(233)는, 기능 모델(211)과 기능 모델(212)이 비교될 수 있다는 것을 표현한다. 기능 모델(211)과 기능 모델(212)은 (예를 들어, 동일한 스키마에 따라 모델링된) 동일하거나 유사한 유형의 기능 모델들일 수 있다. 예를 들어, 제1 비즈니스 아키텍처의 선적 기능은 제2 비즈니스 아키텍처의 선적 기능에 비교될 수 있다. 비교의 결과들은 비교되는 모델들의 차이점들 및 유사점들을 드러낼 수 있다. 예를 들어, 제1 및 제2 비즈니스 아키텍처의 선적 기능들을 비교하는 것은, 선적 기능들 중 하나가 크로스-도킹을 지원하지 않는다는 것을 드러낼 수 있다.

(예를 들어, 대조 모듈(114)에서 구현되는) 대조 연산자(234)는, 기능 모델(212)과 서비스 모델(213)이 서로 대조될 수 있다는 것을 표현한다. 기능 모델(212)과 서비스 모델(213)은 상이한 스키마들에 따라 모델링되어 상이한 비즈니스 계층들

을 모델링하는 상이한 유형들의 모델들일 수 있다. 예를 들어, 기능 모델(212)은 비즈니스 기능 모델링 스키마에 따라 비즈니스 기능 계층을 모델링할 수 있고 서비스 모델(213)은 서비스 네트워크 모델링 스키마에 따라 서비스 네트워크 계층을 모델링할 수 있다. 모델들을 대조하는 것은, 일 비즈니스 계층의 컴포넌트들이 다른 비즈니스 계층의 컴포넌트들에 의해 어떻게 지원되며 그리고/또는 그들과 어떻게 상호 작용하는지를 드러낼 수 있다. 예를 들어, 비즈니스 기능 모델을 서비스 네트워크 모델과 대조하는 것은 비즈니스 기능 계층의 대응하는 기능들을 지원하는 서비스 네트워크 계층의 서비스들을 드러낼 수 있다.

일부 실시예들에서, 대조 연산은 상이한 비즈니스 계층들의 모델들 사이에서 매핑을 발생시키는 단계를 포함한다. 예를 들어, 도 4는 (비즈니스 기능 계층의) 비즈니스 기능 모델과 (서비스 네트워크 계층의) 서비스 네트워크 모델간의 예시적인 매핑(400)을 도시한다. 매핑(400)은 비즈니스 기능 모델(401) 및 서비스 모델(451)을 포함한다. 비즈니스 기능 모델(401)은 (실선 경계로써 도시되는) 기존의 기능 컴포넌트들(402, 403, 404, 406, 407)을 포함한다. 서비스 모델(451)은 (실선 경계로써 도시되는) 기존의 서비스 컴포넌트들(452, 454, 456, 및 477)을 포함한다. 매핑들(421, 422, 423, 424, 426, 427, 428, 429, 431, 432)은, 대응하는 기능 컴포넌트들을 지원하는 서비스 컴포넌트들을 나타낸다.

서비스 컴포넌트는 복수개의 상이한 기능 컴포넌트들을 지원할 수 있다. 예를 들어, 서비스 컴포넌트(452)는 (매핑들(421 및 422)에 의해 각각 도시되는 바와 같이) 기능 컴포넌트들(402 및 403) 모두를 지원한다. 복수개의 서비스 컴포넌트들도 기능 컴포넌트를 지원할 수 있다. 예를 들어, 서비스 컴포넌트들(454 및 457) 모두는 (매핑들(423 및 424)에 의해 도시되는 바와 같이) 기능 컴포넌트(404)를 지원한다. 그에 따라, 다양한 일-대-일, 다-대-일, 일-대-다, 및 다-대-다 대응 관계가, 상이한 비즈니스 계층들의 비즈니스 모델들간의 매핑으로부터 기인할 수 있다.

상이한 비즈니스 계층들간의 매핑은, 일 비즈니스 계층의 기존 컴포넌트들이 다른 비즈니스 계층의 컴포넌트들에 의해 지원되는지를 판정하는데도 사용될 수 있다. 그에 따라, 예시적인 매핑(400)내에서, 기존의 서비스 컴포넌트들이 새롭게 제안된 비즈니스 기능에 대한 지원을 제공하는지가 판정될 수 있다. 예를 들어, (점선 경계에 의해 도시되는 바와 같은) 제안된 기능 컴포넌트(408)의 추가가 비즈니스 기능 모델(401)에서 시뮬레이션될 수 있다. 응답하여, 비즈니스 기능 모델(401)과 서비스 모델(452)간의 매핑들이 업데이트될 수 있다.

매핑(431)은, 서비스 컴포넌트(457)가 기능 컴포넌트(408)에 의해 표현되는 비즈니스 기능의 일부분에 대한 지원을 제공할 수 있다는 것을 나타낸다. 그러나, 매핑(432)은, 기능 컴포넌트(408)에 의해 표현되는 비즈니스 기능(예를 들어, 크로스-도킹)을 위한 지원이 (현재로는 포함되어 있지 않은, 사전-해칭으로써 도시되어 있는) 서비스 컴포넌트(458)도 필요로 할 것이라는 것을 나타낸다. 이것은, 대응하는 서비스 네트워크의 관리자에게 비즈니스 기능을 지원하기 위해 추가적인 인프라스트럭처(예를 들어, 파일 서버들, 저장 기능, 네트워크 액세스, 프로토콜들 등)가 필요하다는 것을 나타낸다.

매핑(400) 뿐만 아니라 (예를 들어, 다른 비즈니스 계층들의 모델들간의) 다른 유사 매핑들은, 사용자가 매핑들을 조작할 수 있도록, 사용자-인터페이스(111)에 제시될 수 있다. 사용자-인터페이스(111)를 통해, 사용자는 (예를 들어, 비즈니스 계층의) 일 모델에서의 컴포넌트 추가(또는 삭제)를 시뮬레이션할 수 있고 컴포넌트의 추가(또는 삭제)가 (예를 들어, 다른 비즈니스 계층들의) 다른 모델들에서의 컴포넌트들에 어떤 영향을 미칠 것인지를 판정할 수 있다.

상술된 바와 같이, 다양하고 상이한 데이터 모델들을 사용해 상이한 비즈니스 계층들을 모델링할 수 있다. 그에 따라, 일부 실시예들에서는, 데이터 모델들이 비즈니스 기능 계층을 모델링하기 위한 하나 이상의 비즈니스 기능 모델링 스키마, 비즈니스 편성 계층을 모델링하기 위한 하나 이상의 비즈니스 편성 스키마, 비즈니스 프로세스 흐름 계층을 모델링하기 위한 하나 이상의 비즈니스 프로세스 흐름 모델링 스키마, 서비스 네트워크 계층을 모델링하기 위한 하나 이상의 서비스 네트워크 계층 비즈니스 모델링 스키마 등을 포함할 수 있다.

일부 실시예들에서, 비즈니스 기능들을 위한 비즈니스 모델들 및 데이터 포맷 정의들은 일반적으로 표 1에서 나타내어진 바와 같이 설명된다.

[표 1]

	모델들은 기능들을 하나의 비즈니스를 설명하는 개개 그룹들로 분류하는 기능을 한다. 모델들은 비즈니스를 위해 정의된 모든 기능들 뿐만 아니라 정의된 임의의 기능들이 계층 구조적 분해 및 프로세스 흐름 관계들의 관점에서 서로 관련되는 방식을 포함한다. 모델들은 리포지토리의 데이터를, 서로 비교
--	--

	될 수 있지만 서로 독립적인 개개 비즈니스 모델들로 세그먼트화하는 것을 용이하게 한다. 또한, 기능 데이터는 모델내에서 정의되지만, 데이터 모델의 다른 데이터 요소들은 모델 외부에 위치하여, 상이한 모델들의 서로에 대한 비교를 용이하게 한다.
기능들	기능들은, 3가지 이상의 상이한 방식으로 모델링되는, 모델에서의 개개 비즈니스 기능 영역들이다. 기능들은, 그들의 고유한 특성들의 집합을 가진 개별적인 것들로서; 기능 영역들의 분해 계층 구조로서; 그리고 간단한 비즈니스 프로세스 흐름들에서의 접속들로서 모델링될 수 있다. 예를 들어, 더 높은 레벨의 기능이 그것의 구성 부분들로 분해될 때와 같이, 개략적인(또는 더 높은 레벨의) 기능들은 좀더 세분화된(또는 더 낮은 레벨의) 기능들을 포함할 수 있다. 기능들에 대한 특성들의 할당은 계층 구조의 다수 레벨들에서 발생할 수 있는데, 이것은 이후의 데이터 변환들을 제어하는 데에 사용될 수 있다. 예를 들어, 더 높은 레벨의 기능이 변환을 통해 조각될 경우, 대응하는 더 낮은 레벨의 기능들에 대한 특성들이 변환에서 고려될 수 있다.
기능 입력들과 출력들	기능 입력들과 출력들은 비즈니스 기능들에 의해 소비되고 그리고/또는 발생하는 아티팩트들 및 이벤트들이다. 이들은 외부로 향하는 것을 그리고 기능들의 거동에 관한 가시적인 것을 표현한다. 다른 입력들 및 출력들과 무관하게, 입력들은 소비될 수 있고 출력들은 발생할 수 있다. 예를 들어, 기능이 시작되기 전에, 기능을 위한 모든 입력들이 소비될 필요는 없다. 마찬가지로, 출력이 발생할 수 있기 전에, 기능의 모든 프로세싱이 완결될 필요는 없다.
프로세스들	프로세스들은, 비즈니스 프로세스를 도시하기 위해 플로우로 그리고 비즈니스 프로세스의 종단 뷰(end-to-end view)에 접속되어 있는 비즈니스 기능들의 네트워크들이다. 프로세스들은 더 큰 비즈니스 평면들을 가능하게 하는 기능들간의 접속들을 정의한다. 데이터 모델로 모델링된 프로세스들은 기능들간의 경계들에 대한 순회(traversal)를 표현하는 크로스-기능 프로세스들을 의미할 수 있다.
접속들	접속들은 비즈니스 기능들간의 관계들을 표현하는 데에 사용된다. 접속들은, 예를 들어, 비즈니스 문서와 같은, 데이터가 그러한 기능들 사이에서 이동할 수 있는 데이터 접속들일 수 있다. 그러나, 다른 유형들의 접속들도 가능할 수 있다. 접속들은, 비즈니스 활동의 규제된 영역들에서 흔히 발생하는, 비즈니스 평면의 감시 또는 관리를 의미할 수도 있다. 접속들은, 접속 유형들이 모든 모델들에 걸쳐 동일하도록, 유형화될 수 있다. 유형화된 접속들은 모델 비교들을 용이하게 하는 데에 사용될 수 있다.
서비스 레벨들	서비스 레벨들은 기능의 성능에 대한 일반적인 기대값을 의미한다. 서비스 레벨들은 성능 및 가측성 속성들을 다양한 정도들의 형식성(예를 들어, 계약) 및 시간(예를 들어, 역사적 현재, 현재, 타킷, 및 최대)으로 기능에 첨부한다. 일부 실시예들에서, 기능은 동사 및 명사구를 포함한다(또는 명사-동사구가 기능 설명으로부터 추론될 수 있다). 기능과 연관된 서비스 레벨 기술 데이터는, 기능이 구로써 함축된 동작을 얼마나 만족스럽게 수행하는지를 나타낸다. 예를 들어, Approve Loan Application은 이들의 서비스 레벨 기대값을 가질 수 있다.

도 3은, 구조화된 비즈니스 기능들에 기초해 효율적으로 그리고 융통성있게 모델링하는 데에 사용될 수 있는 예시적인 비즈니스 기능 모델링 스키마(300)를 도시한다. 비즈니스 기능 모델링 스키마(300)는 비즈니스 기능 특성들을 모델링하기 위한 데이터 포맷들, 비즈니스 기능 입력들과 출력들, 비즈니스 기능 프로세스들, 비즈니스 기능 접속들, 및 비즈니스 기능 서비스 레벨 기대값들을 포함할 수 있다. 비즈니스 기능 모델링 스키마(300)는 대응하는 복수개의 상이한 비즈니스 계층들을 모델링하기 위한 데이터 정의들을 포함하는 복수개 스키마들 중 하나일 수 있다는 것을 이해할 수 있어야 한다.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 모델 데이터 포맷(301)을 포함한다. 일반적으로, 모델 데이터 포맷(301)은 표 2에서 나타내어진 바와 같이 설명될 수 있다.

[표 2]

이름	데이터 유형	설명
ID	int	모델의 키이며, 다른 데이터 엔티티들을 이 모델과 관련짓는 데에 사용된다.
Name	varchar (150)	모델을 식별하는 고유한 이름.
Owner ID	int	모델의 소유자를 나타낸다. 소유자는 다수 모델들을 소유할 수 있다.
IsTemplate	bit	이 모델을 변경할 수 있는 모델 제작자의 기능을 제어한다. 이 필드가 참이면, 그것은 이 모델이 다른 모델들을 위한 템플릿으로 사용되어야 하므로, 특성들이, 유도된 모델들의 모델 제작자들에 의해 변경된 이후라 하더라도, 유도된 모델들을 비교하는 데에 사용될 수 있다는 것을 의미한다. 따라서, 이 모델은 모델들의 일반적인 편집기들에 의해서는 변경될 수 없다. 디폴트는 거짓이다.
Description	varchar (2000)	모델의 텍스트 기술.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 소유자 데이터 포맷(302)을 포함한다. 일반적으로, 소유자 데이터 포맷(302)은 표 3에서 나타내어진 바와 같이 설명될 수 있다.

[표 3]

이름	데이터 유형	설명
ID	int	소유자의 키이며, 다른 데이터 엔티티들을 이 소유자에 관련짓는 데에 사용된다.
Name	varchar (50)	소유자의 고유한 이름.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 기능 데이터 포맷(314)을 포함한다. 일반적으로, 기능 데이터 포맷(314)은 표 4에서 나타내어진 바와 같이 설명될 수 있다.

[표 4]

이름	데이터 유형	설명
ID	int	기능의 키이며, 다른 데이터 엔티티들을 이 기능에 관련짓는 데에 사용된다.
Name	varchar (256)	특정 모델내에서 고유한 이름.
Purpose	varchar (1000)	기능의 짧은 기술.
Description	varchar (8000)	기능의 좀더 상세한 기술로서, 이 기능의 관계들 및 특성들 뿐만 아니라 기능 자체를 설명할 수 있다.
SourcingType	int	이 필드는 3개의 값들: Internal, Outsourced, 또는 Both를 가질 수 있다. 이것은, 기능이 모델을 "소유하고" 있는 조직의 내부(일부) 조직에 의해; 모델의 "소유자"에게 기능을 공급하는 조직에 의해; 또는 내부 및 외부 공급자들 모두에 의해 수행되는지를 나타낸다.
Division	varchar (100)	기능이 수행되는 비즈니스 조직 영역을 식별한다.
Location	varchar (100)	기능이 수행되는 지리적 위치.
CopiedFromID	int	이 기능이 카피된 특정 기능(및 그에 따른 템플릿 모델)을 나타낸다. 시스템-설정 값일 수 있다.
ModelID	int	이 기능이 속한 모델을 나타낸다.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 기능 계층 구조 데이터 포맷(303)을 포함한다. 일반적으로, 기능 계층 구조 데이터 포맷(303)은 표 5에서 나타내어진 바와 같이 설명될 수 있다.

[표 5]

이름	데이터 유형	설명
CapabilityID	int	기능으로 링크한다.
ParentID	int	동일 모델의 기능으로 링크하고, 모델의 기능들에 대한 계층 구조도에서 이 기능의 부모를 나타낸다.
Generation	int	소정 쿼리들에서 사용되는 혈통 키의 일부이다.
Sequence	int	소정 쿼리들에서 사용되는 혈통 키의 일부이다.
Lineage	varchar (20)	기능의 전체 조상 혈통을 나타내며, 계층 구조적 정렬들을 수행하는 데에 사용된다.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 기능 특성 데이터 포맷(311)을 포함한다. 일반적으로, 기능 특성 데이터 포맷(311)은 표 6에서 나타내어진 바와 같이 설명될 수 있다.

[표 6]

이름	데이터 유형	설명
CapabilityID	int	기능으로 링크한다.
PropertyNameID	int	사용자-정의형 특성으로 링크한다.
Value	varchar (250)	이 기능을 위한 특성 값.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 특성 이름 데이터 포맷(312)을 포함한다. 일반적으로, 특성 이름 데이터 포맷(312)은 표 7에서 나타내어진 바와 같이 설명될 수 있다.

[표 7]

이름	데이터 유형	설명
ID	int	특성의 키이며, 이 특성을 기능들에 관련짓는 데에 사용된다.
Name	varchar (250)	특성의 이름이며, 사용자 정의형이다.
Description	varchar (4000)	특성이 무엇이며 그것이 기능들을 설명하는데 어떻게 사용되는지에 대한 기술이다.
DataType	int	DataType 엔티티로 링크하며, 모델 제작자가 기능을 위해 이 특성에 대한 값을 설정할 때 예상되는 데이터 유형을 나타낸다. 예를 들어, 모델 제작자가 "Fixed Cost Allocation"이라는 명칭의 특성을 정의한다면, 이 특성을 위한 데이터 유형은 "Currency"일 것이다.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 데이터 유형 데이터 포맷(313)을 포함한다. 일반적으로, 데이터 유형 데이터 포맷(313)은 표 8에서 나타내어진 바와 같이 설명될 수 있다.

[표 8]

이름	데이터 유형	설명
ID	int	데이터 유형의 키이며, 사용자 정의형 특성의 데이터 유형을 나타내는 데에 사용된다. 특성 값들의 확인들을 정확하게 수행하기 위해, 모델링 도구들은 "공치"되어 있는 값들에 의존하므로, 이것은, 우리가 모델 제작자들에 의해 변경되지 않는 것으로 가정하는 몇몇 도표들 중 하나이다.
Name	varchar (20)	데이터 유형의 친숙한 이름이다. 예들로는 "Integer", "String", "Currency" 등을 들 수 있다.
Description	varchar (4000)	사용자들이 정의하는 특성들에 대한 데이터 유형들의 사용자 선택을 안내할 때 특히 유용할 수 있는 데이터 유형에 관한 임의의 추가 정보.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 포트 데이터 포맷(324)을 포함한다. 비즈니스 기능에 대응하는 포트들은 대응하는 비즈니스 기능으로 입력을 전송하고 대응하는 비즈니스 기능으로부터 출력을 전송하는 데에 사용될 수 있다. 일반적으로, 포트 데이터 포맷(324)은 표 9에서 나타내어진 바와 같이 설명될 수 있다.

[표 9]

이름	데이터 유형	설명
ID	int	포트의 키이며, 이 포트를 다른 엔티티들에 관련짓는 데에 사용된다.
ModelID	int	이 포트가 관련 모델에 속한다는 것을 나타낸다. 특정 모델을 다룰 경우, 모델 제작자에게는 모델과 연관된 포트들만이 이용 가능하다. 포트는 기능으로 입력되어 기능에 의해 소비되거나 기능에 의해 발생되어 기능으로부터 출력되는 무엇이다.
Name	varchar (256)	특정 모델내에서의 고유한 이름.
ItemType	int	입력 또는 출력의 유형을 나타내는 ItemType 엔티티로 링크하는데, 이것은 전자 데이터, 물리 항목, 팩스, 이벤트 등일 수 있다.
SchemaID	int	ItemType이, 이것이 일종의 전자 데이터 레코드라고 나타내면, 이 필드는 데이터 레코드의 내용을 설명하는 스키마로 링크한다.
Description	varchar (4000)	입/출력 항목의 상세한 기술.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 기능 포트 데이터 포맷(319)을 포함한다. 일반적으로, 기능 포트 데이터 포맷(319)은 표 10에서 나타내어진 바와 같이 설명될 수 있다.

[표 10]

이름	데이터 유형	설명
ID	int	기능 포트의 키이며, 이 포트를 다른 엔티티들에 관련짓는 데에 사용된다.
CapabilityID	int	이 관계에 의해 참조되는 기능으로 링크한다.
PortID	int	이 관계에 의해 참조되는 포트로 링크한다.
Direction	int	3개의 값들을 가지며, 항목이 참조되는 기능으로 입력되는지, 참조되는 기능으로부터 출력되는지, 또는 양방향으로 이동하는지를 나타낸다.
UsageType	int	UsageType 엔티티로 링크하며, 기능이 이 항목을 어떻게 사용하는지를 나타낸다. 예 들로는 "Read only", "Read and update", "Create" 등을 들 수 있다.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 사용 유형 데이터 포맷(318)을 포함한다. 일반적으로, 사용 유형 데이터 포맷(318)은 표 11에서 나타내어진 바와 같이 설명될 수 있다.

[표 11]

이름	데이터 유형	설명
ID	int	UsageType의 키이며, 이 사용 유형을 입/출력 항목들(포트 엔티티)에 관련짓는 데에 사용된다. 도구들은 모델들을 프로세싱하기 위해 이 도표의 특정 값들에 의존하므로, 이 도표는 모델 제작자들에 의해 변경 불가능한 것으로 가정된다.
Name	varchar (150)	사용 유형을 식별하는 고유한 이름. 예 들로는 "Read only", "Read and update", "Create" 등을 들 수 있다.
Description	varchar (4000)	사용 유형에 대한 그리고 모델링 도구들이 특정 사용 유형을 다룰 때 어떻게 거동할 수 있는지에 대한 좀더 상세한 기술.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 항목 유형 데이터 포맷(316)을 포함한다. 일반적으로, 항목 유형 데이터 포맷(316)은 표 12에서 나타내어진 바와 같이 설명될 수 있다.

[표 12]

이름	데이터 유형	설명
ID	int	ItemType의 키이며, 이 항목 유형을 입/출력 항목들(포트 엔티티)에 관련짓는 데에 사용된다. 도구들은 모델들을 프로세싱하기 위해 이 도표의 특정 값들에 의존하므로, 이 도표는 모델 제작자들에 의해 변경 불가능한 것으로 가정된다.
ItemTypeName	varchar (150)	항목 유형을 식별하는 고유한 이름. 예들로는 "Electronic data", "Physical item", "Fax" 등을 들 수 있다.
Description	varchar (4000)	항목 유형에 대한 그리고 모델링 도구들이 특정 항목 유형을 다룰 때 어떻게 거동할 수 있는지에 대한 좀더 상세한 기술.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 스키마 데이터 포맷(317)을 포함한다. 일반적으로, 스키마 데이터 포맷(317)은 표 13에서 나타내어진 바와 같이 설명될 수 있다.

[표 13]

이름	데이터 유형	설명
ID	int	이것은 Schema 엔티티의 키이며, 이 항목 유형을 입/출력 항목들(포트 엔티티)에 관련짓는 데에 사용된다.
Name	varchar (250)	이것은 스키마를 위한 고유한 이름이다.
Description	varchar (4000)	이것은 XML 스키마 형태의 데이터 레코드에 대한 데이터 내용의 상세한 기술(또는 그것의 소정 간략화)일 수 있다.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 프로세스 데이터 포맷(327)을 포함한다. 일반적으로, 프로세스 데이터 포맷(327)은 표 14에서 나타내어진 바와 같이 설명될 수 있다.

[표 14]

이름	데이터 유형	설명
ID	int	Process 엔티티의 키이며, 이 항목 유형을 커넥터 엔티티들에 그리고 그들을 통해 ProcessCapability 엔티티의 관련 기능들에 관련짓는 데에 사용된다.
ModelID	int	이들 프로세스들이 속한 모델을 나타낸다.
Name	varchar (256)	이 모델내의 프로세스를 위한 고유한 이름.
Description	varchar (4000)	이 엔티티 및 ProcessCapability 엔티티에 의해 모델링되는 프로세스를 기술한다.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 프로세스 기능 데이터 포맷(326)을 포함한다. 일반적으로, 프로세스 데이터 포맷(326)은 표 15에서 나타내어진 바와 같이 설명될 수 있다.

[표 15]

이름	데이터 유형	설명
ProcessID	int	이 기능들 및 접속들이 속하는 프로세스를 나타낸다.
StepNumber	int	프로세스에서 이 접속의 시퀀스를 나타내며, 가상 모델에서의 렌더링을 위해 프로세스 단계들을 정렬하는 데에 사용된다.
Connector ID	int	Connector 엔티티로 그리고 그를 통해 소스 기능으로부터 타깃 기능으로의 프로세스 흐름에 대한 소스 및 타깃 기능들로 링크한다.

Sequence	int	단계내에서 접속의 시퀀스를 나타냄으로써, 그것을 통과하는 다수 경로들을 가진 프로세스 흐름들을 지원한다. 일 레그가 다른 레그보다 더 많은 단계들을 가진 (또는 더 많은 기능들을 통과하는) 경로를 정의하기 위해, 더 짧은 레그가, 커넥터는 동일하지만 StepNumber 들이 상이한 이 도표의 엔트리들에 의해 표현된다.
Condition	varchar (4000)	프로세스를 유도하는 조건들이 무엇인지에 관한 코멘트들을 저장한다.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 커넥터 데이터 포맷(323)을 포함한다. 일반적으로, 커넥터 데이터 포맷 (323)은 표 16에서 나타내어진 바와 같이 설명될 수 있다.

[표 16]

이름	데이터 유형	설명
ID	int	Connector 엔티티의 키이며, 2개 기능들간의 접속을 나타낸다. 이 키는 이 접속을 다른 엔티티들에 링크하는 데에 사용된다.
Name	varchar (256)	2개 기능들간의 이 접속과 연관되어 있는 코멘트
FromCapabilityID	int	소스 기능인 기능을 의미한다. ConnectorType에 따라, 소스 기능이라는 것의 의미는 약간 상이할 수 있다.
ToCapabilityID	int	타겟 기능인 기능을 의미한다. ConnectorType에 따라, 타겟 기능이라는 것의 의미는 약간 상이할 수 있다.
ConnectorType	int	ConnectorType 엔티티로 링크하며 참조되는 2개 기능들간의 관계가 실제로 의미하는 것이 무엇인지를 나타낸다. 예들로는 "Collaborative", "Controlling", "Dependent" 등을 들 수 있다.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 커넥터 유형 데이터 포맷(321)을 포함한다. 일반적으로, 커넥터 유형 데이터 포맷(321)은 표 17에서 나타내어진 바와 같이 설명될 수 있다.

[표 17]

이름	데이터 유형	설명
ID	int	ConnectorType 엔티티의 키이며, 특정 입/출력 항목을 갖춘 2개 기능들간의 특정 접속을 링크하는 기능을 한다.
TypeName	varchar (50)	접속의 유형을 설명하는 고유한 이름. 예들로는 "Collaborative", "Controlling", "Dependent" 등을 들 수 있다.
Description	varchar (4000)	접속 유형에 대한 상세한 설명으로서, 모델 제작자들의 모델들에서 접속들이 의미하는 바를 모델 제작자들이 이해할 수 있게 한다.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 커넥터 포트 데이터 포맷(322)을 포함한다. 일반적으로, 커넥터 포트 데이터 포맷(322)은 표 18에서 나타내어진 바와 같이 설명될 수 있다.

[표 18]

이름	데이터 유형	설명
Connector ID	int	Connector 엔티티로의 참조로서, 특정 입/출력 항목을 갖춘 2개 기능들간의 특정 접속을 링크하는 기능을 한다.
Port ID	int	Port 엔티티(입/출력 항목)로의 참조로서, 특정 접속을 따라 이동하는 입/출력 항목을 식별하는 기능을 한다.
Comments	varchar (4000)	이 접속에 따른 항목의 이러한 이동에 관한 좀더 상세한 코멘트.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 역할 데이터 포맷(309)을 포함한다. 일반적으로, 역할 데이터 포맷(309)은 표 19에서 나타내어진 바와 같이 설명될 수 있다.

[표 19]

이름	데이터 유형	설명
ID	int	Role 엔티티의 키이며, 이 역할을 Capability 엔티티들에 관련짓는 데에 사용된다.
ModelID	int	이 역할 엔티티가 어떤 모델에 속하는지를 나타낸다.
Name	varchar (100)	역할에 대한 이 모델내에서의 고유한 이름. 역할은 기능들을 수행하는 것에 관련된 사람 또는 사용자 유형을 설명한다.
Description	varchar (2000)	역할에 대한 기술을 제공하며, 모델 제작자들에게 기능들과 연관짓기 위한 모델 제작자들의 역할 선택에 관한 안내를 제공할 수도 있다.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 기능 역할 데이터 포맷(308)을 포함한다. 일반적으로, 기능 역할 데이터 포맷(308)은 표 20에서 나타내어진 바와 같이 설명될 수 있다.

[표 20]

이름	데이터 유형	설명
CapabilityID	int	특정 기능을 참조하며, 그 기능과 특정 역할을 링크하는 기능을 한다.
RoleID	int	특정 역할을 참조하며, 그것을 참조되는 기능에 링크하는 기능을 한다.
Count	int	이 기능을 수행하는데 필요한, 이 역할의 사람 수를 나타낸다. '0' 값은, 역할 참여가 정량화되지 않았다는 것을 나타낸다.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 SLE 유형 데이터 포맷(304)을 포함한다. 일반적으로, SLE 유형 데이터 포맷(304)은 표 21에서 나타내어진 바와 같이 설명될 수 있다.

[표 21]

이름	데이터 유형	설명
ID	int	SLEType 엔티티의 키이며, 이 역할을 CapabilitySLE 엔티티들에 관련짓는 데에 사용된다.
Name	varchar (100)	이 엔티티에서 설명되는 서비스 레벨의 유형을 고유하게 명명한다. 모델링 도구들이 서비스 레벨들을 가시화하기 위해서는 이 엔트리들의 값에 의존하기 때문에, 이 엔티티는 모델 제작자들에 의해서만 판독되는 것으로 가정된다. 서비스 레벨 유형들을 위한 일부 값들로는 "Duration", "Throughput", "Monetary Cost", "Time Cost", 및 "Concurrency"를 들 수 있다.
Description	varchar (4000)	서비스 레벨 유형에 대한 그리고 특정 서비스 레벨들이 기능들에 관련되는 방법에 대한 상세한 기술.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 기능 SLE 데이터 포맷(306)을 포함한다. 일반적으로, 기능 SLE 데이터 포맷(306)은 표 22에서 나타내어진 바와 같이 설명될 수 있다.

[표 22]

이름	데이터 유형	설명
ID	int	Role 엔티티의 키이며, 이 역할을 Capability 엔티티들에 관련짓는데 사용된다.
SLETypeID	int	SLEType 엔티티를 참조하며, 서비스 레벨을 측정하기 위한 특정 방식을 식별한다.
Name	varchar (50)	서비스 레벨 정의를 위한 고유한 이름.
CapabilityID	int	이 서비스 레벨이 적용되는 기능을 참조한다.
MeasurementPeriodType	varchar (50)	서비스 레벨을 위한 측정 유닛을 명명한다. "Duration" 유형 서비스 레벨들의 경우, 이것은 시간이어야 한다. "Momentary Cost" SLE 유형의 경우, "Dollars" 또는 "Thousands of dollars"가 적절할 것이다.
MeasurementPeriodLen	int	SLE 유형이 SLE의 "Throughput" 유형을 참조한다면, 이 필드는 처리율을 위한 측정 주기의 길이를 나타낸다.
MetricCount	int	구간의 날 수, 처리율을 위해 완결된 항목들의 수, 금융 비용을 위한 달러량 등과 같은, SLE의 실제(현재 상태/성능 또는 이력 성능) 측정치.
Goal	int	구간의 날 수, 처리율을 위해 완결된 항목들의 수, 금융 비용을 위한 화폐량 등과 같은 미래 성능을 위한 목표치.
VarianceThreshold	int	변동이 지적되거나 통지가 송신되기 전에, (예를 들어, 목표치로부터) 얼마나 큰 성능 변동이 허용 가능한지를 나타낸다. 예를 들어, 편차 임계치가 초과되면, 적당한 관리 담당자에게 전자 메일 메시지가 송신될 수 있다.
Description	varchar (2000)	이 기능을 위한 SLE의 상세한 기술.

도 3에 도시되어 있는 바와 같이, 스키마(300)는 기능 SLE 포트 데이터 포맷(307)을 포함한다. 일반적으로, 기능 SLE 포트 데이터 포맷(307)은 표 23에서 나타내어진 바와 같이 설명될 수 있다.

[표 23]

이름	데이터 유형	설명
CapabilitySLEID	int	CapabilitySLE 엔티티에서 설명되는 바와 같은 특정 기능을 위한 소정 서비스 레벨을 참조한다. 이것은 소정 서비스 레벨을 소정 입력 또는 출력 항목에 링크하는 기능을 한다.
PortID	int	기능의 소정 입력 또는 출력 항목을 참조하며 측정 중인 특정 항목에 서비스 레벨을 링크한다. 예를 들어, 이것은 주택 용자 프로세싱 기능에 대한 구간 서비스 레벨을 위해 주택 용자 승인들을 참조할 수 있고, 전체 서비스 레벨 정의는 그에 의해, 주택 용자 프로세싱 기능에 대해 매일 100건의 주택 용자 승인들이 완결된다는 것을 기술할 수 있다.

스키마(300)는 단지 비즈니스 기능 모델링 스키마의 일례일 뿐이라는 것을 이해해야 한다. 당업자에게는, 이 명세서를 살펴본 후, 본 발명의 실시예들이, 스키마(300) 이외에, 광범위한 다른 비즈니스 기능 모델링 스키마들과 함께 사용될 수 있다는 것이 명백할 것이다. 또한, 비즈니스 기능들의 모델링은, 스키마(300)의 모든 데이터 포맷들에 대한 기능 속성들이 액세스 가능할 것을 요하지 않는다. 예를 들어, 기능 및 커넥터는, 다른 데이터 포맷들에 대응하는 기능 속성들을 액세스할 필요없이, 기능 데이터 포맷(314) 및 커넥터 데이터 포맷(323)에 기초해 비즈니스 기능을 모델링하는 데에 사용될 수 있다. 그에 따라, 스키마(300)는 액세스되는 비즈니스 기능 속성들을 위한 데이터 포맷들을 정의하지만, 비즈니스 기능 모델을 발생시키기 위해 모든 데이터 포맷들이 채워질 것을 요하지는 않는다.

하나 이상의 비즈니스 계층들을 위한 스키마들이, 하나 이상의 비즈니스 계층들 사이에서 매핑하는 데에 사용되는 데이터를 위한 데이터 정의들도 포함할 수 있다는 것을 이해할 수 있어야 한다. 예를 들어, 비즈니스 기능 스키마는 모델링된 비즈니스 기능 컴포넌트를 지원하는데 필요한 서비스 네트워크 계층 컴포넌트들을 표현하는 데이터 정의들을 포함할 수 있다. 한편, 서비스 네트워크 스키마는 지원되는 비즈니스 기능 계층 컴포넌트들을 위한 데이터 정의들을 포함할 수 있다. 마찬가지로, 예를 들어, 비즈니스 프로세스 흐름 스키마 또는 비즈니스 편성 스키마와 같은, 다른 스키마들은 다른 비즈니스 계층들을 매핑하는 데에 사용되는 데이터를 위한 데이터 정의들을 포함할 수 있다.

도 5는 유사한 유형의 비즈니스 모델들을 비교하기 위한 방법(500)의 예시적인 흐름도를 도시한다. 방법(500)은 컴퓨터 아키텍처(100)의 모듈들 및 데이터를 참조하여 설명될 것이다.

방법(500)은 기존 비즈니스 아키텍처를 표현하는 구조화된 비즈니스 모델의 제1 인스턴스를 액세스하는 단계를 포함하는데, 제1의 구조화된 비즈니스 모델은 데이터 모델에 따라 구조화된다(단계 501). 예를 들어, 컴퓨터 시스템(101)은 비즈니스 아키텍처를 표현하는 기능 모델(122)을 액세스할 수 있다. 기능 모델(122)은 비즈니스 기능 스키마(예를 들어, 스키마(300))에 따라 구조화될 수 있다.

방법(500)은 제2 비즈니스 아키텍처를 표현하는 구조화된 비즈니스 모델의 제2 인스턴스를 액세스하는 단계를 포함하는데, 제2 구조화된 비즈니스 모델은 데이터 모델에 따라 구조화된다(단계 502). 예를 들어, 컴퓨터 시스템(101)은 최상 관행 기능 모델(123)을 액세스할 수 있다. 최상 관행 기능 모델(123)은 특정 산업을 위한 최상 관행 비즈니스 아키텍처의 비즈니스 기능들을 표현하는 모델일 수 있다. 예를 들어, 최상 관행 기능 모델(123)은 제조업, 건강 관리업, 소매업 등을 위한 최상 관행을 통합할 수 있다. 최상 관행은 산업 전문가들, 학원 등에 의해 합의될 수 있다. 그러나, 임의적인 가상의 최상 관행이 사용될 수도 있다.

다른 방법으로, 컴퓨터 시스템(101)은 제안된 이득 기능 모델(124)을 액세스할 수 있다. 제안된 이득 기능 모델(124)은 (예를 들어, 비교를 수행 중인 비즈니스에 의해) 획득될 비즈니스의 비즈니스 아키텍처를 표현하는 모델일 수 있다.

방법(500)은 구조화된 비즈니스 모델의 제1 인스턴스를 구조화된 비즈니스 모델의 제2 인스턴스와 비교하는 단계를 포함한다(단계 503). 예를 들어, 비교 모듈(113)은 (예를 들어, 비교 연산자(233)를 사용해) 기능 모델(122)을 최상 관행 기능 모델(123)과 비교할 수 있다. 다른 방법으로, 비교 모듈(113)은 기능 모델(122)을 제안된 이득 기능 모델(124)과 비교할 수 있다. 비교는 상이한 모델들로부터의 개개 비즈니스 기능들을 서로 비교하는 단계를 포함할 수 있다.

방법(500)은 기존 비즈니스 아키텍처와 제2 비즈니스 아키텍처간의 어떤 차이들을 식별하는 단계를 포함한다(단계 504). 예를 들어, 비교 모듈(113)은 기능 모델(122)과 최상 관행 기능 모듈(123)간의 어떤 차이들을 식별할 수 있다. 다른 방법으로, 비교 모듈(113)은 기능 모델(122)과 제안된 이득 기능 모델(124)간의 어떤 차이들을 식별할 수도 있다.

방법(500)은, 기존 비즈니스 아키텍처와 제2 비즈니스 아키텍처간의 식별된 차이들에 기초해, 기존 비즈니스 아키텍처의 충분성을 평가하는 단계를 포함한다(단계 505). 예를 들어, 122와 123 사이에서 식별된 차이들에 기초해, 컴퓨터 시스템(101)은, 기능 모델(122)로 표현된 비즈니스 기능들이 산업의 최상 관행에 얼마나 근접하게 대응하는지를 평가할 수 있다. 다른 방법으로는, 122와 124 사이에서 식별된 차이들에 기초해, 컴퓨터 시스템(101)은, 기능 모델(122)로 표현된 비즈니스 기능들이 제안된 이득 기능 모델(124)로 표현된 비즈니스 기능들에 얼마나 근접하게 대응하는지를 평가할 수 있다. 그에 따라, 이러한 획득이, 획득 중인 비즈니스에 임의의 상이한, 새로운, 증가된 등의 비즈니스 기능들을 제공할 것인지가 판정될 수 있다. 평가의 결과들은 사용자-인터페이스(111)에 제시되거나 그리고/또는 결과들(161)로서 소정의 다른 컴퓨터 시스템으로 전송될 수 있다.

도 6은 비즈니스 아키텍처의 대응하는 상이한 계층들을 표현하는 비즈니스 모델들의 상이한 유형들을 대조하기 위한 방법(600)의 예시적인 흐름도를 도시한다. 방법(600)은 컴퓨터 아키텍처(100)의 모듈들과 데이터 및 매핑(400)의 컴포넌트들과 매핑들을 참조하여 설명될 것이다.

방법(600)은 비즈니스 아키텍처의 제1 비즈니스 계층을 표현하는 구조화된 비즈니스 모델을 액세스하는 단계를 포함하는데, 구조화된 비즈니스 모델은 구조화된 데이터 모델에 따라 제1 비즈니스 계층의 하나 이상의 제1 비즈니스 계층 컴포넌트들을 모델링한다(단계 601). 예를 들어, 컴퓨터 시스템(101)은 비즈니스 아키텍처의 비즈니스 기능 계층을 표현하는 기능 모델(122)을 액세스할 수 있다. 기능 모델(122)은 기능 모델링 스키마(예를 들어, 스키마(300))에 따라 비즈니스 기능 계층(121)의 하나 이상의 비즈니스 기능 컴포넌트들을 모델링할 수 있다. 그러나, 예를 들어, 서비스 네트워크 계층(131),

비즈니스 프로세스 흐름 계층(141), 비즈니스 편성 계층(151)과 같은, 소정의 다른 제1 비즈니스 계층을 표현하는 구조화된 비즈니스 모델이 액세스될 수도 있다. 구조화된 비즈니스 모델은 적절한 스키마에 따라 대응하는 제1 비즈니스 계층의 하나 이상의 컴포넌트들을 모델링할 수 있다.

방법(600)은 제2 비즈니스 계층을 표현하는 제2의 구조화된 모델을 액세스하는 단계를 포함하는데, 제2의 구조화된 모델은 상이한 제2 비즈니스 계층의 하나 이상의 제2 비즈니스 계층 컴포넌트들을 모델링한다(단계 602). 예를 들어, 컴퓨터 시스템(101)은 비즈니스 아키텍처의 서비스 네트워크 계층을 표현하는 서비스 모델(132)을 액세스할 수 있다. 서비스 모델(132)은 서비스 모델링 스키마에 따라 서비스 네트워크 계층(131)의 하나 이상의 서비스 컴포넌트들을 모델링할 수 있다. 그러나, 예를 들어, 비즈니스 기능 계층(121), 비즈니스 프로세스 흐름 계층(141), 비즈니스 편성 계층(151)과 같은, 소정의 다른 제2 비즈니스 계층을 표현하는 제2의 구조화된 모델이 액세스될 수도 있다. 제2의 구조화된 비즈니스 모델은 적절한 스키마에 따라 대응하는 제2 비즈니스 계층의 하나 이상의 컴포넌트들을 모델링할 수 있다.

방법(600)은 하나 이상의 제1 비즈니스 계층 컴포넌트들과 하나 이상의 제2 비즈니스 계층 컴포넌트들간의 관계들을 매핑하는 단계를 포함한다(단계 603). 매핑에 기초해, 제1 비즈니스 계층의 구성 변화들이 어떻게 제2 비즈니스 계층에 영향을 미치며 제2 비즈니스 계층의 구성 변화들이 어떻게 제1 비즈니스 계층에 영향을 미치는지가 판정될 수 있다. 예를 들어, 대조 모듈(114)은 (매핑(400)과 유사하게) 기능 모델(122)의 하나 이상의 기능 컴포넌트들을 서비스 모델(131)의 하나 이상의 대응하는 서비스 컴포넌트들로 매핑할 수 있다. 그러나, 다른 비즈니스 계층들의 컴포넌트들간의 관계들이 매핑될 수도 있다. 예를 들어, 대조 모듈(114)은 프로세스 흐름 모델(142)의 하나 이상의 프로세스 흐름 컴포넌트들을 편성 모델(153)의 하나 이상의 대응하는 편성 컴포넌트들로 매핑할 수도 있다.

방법(600)은, 매핑된 관계들을 이용해, 구조화된 비즈니스 모델을 제2의 구조화된 모델과 대조하는 단계를 포함한다(단계 604). 예를 들어, 대조 모듈(114)은 기능 모델(122)을 서비스 모델(132)과 대조하기 위해 기능 모델(122)과 서비스 모델(132)의 컴포넌트들간의 매핑들을 대조할 수 있다. 상이한 비즈니스 계층들의 모델들을 대조하는 것은, 비즈니스 계층들이 어떻게 상호 작용하며 필요한 그리고/또는 중복되는 컴포넌트들의 표시를 비즈니스에 제공할 수 있는지를 드러낼 수 있다. 상이한 비즈니스 계층들의 컴포넌트들을 매핑하고 대조하는 결과들은 사용자-인터페이스(111)에 제시될 수 있고 그리고/또는 소정의 다른 컴퓨터 시스템으로 결과들(161)로서 전송될 수 있다.

도 7은 비즈니스 모델이 하나 이상의 제약들에 따르는지를 확인하기 위한 방법(700)의 예시적인 흐름도를 도시한다. 방법(700)은 컴퓨터 아키텍처(100)의 컴포넌트들과 관련하여 논의될 것이다.

방법(700)은 유효한 비즈니스 아키텍처를 표현하는 제1 비즈니스 컴포넌트들의 구조화된 비즈니스 모델을 액세스하는 단계를 포함하는데, 제1 비즈니스 컴포넌트들은, 하나 이상의 제약들을 준수함을 나타내는 대응하는 제1 특성 값들을 가지고, 구조화된 비즈니스 모델은 데이터 모델에 따라 구조화된(단계 701). 예를 들어, 컴퓨터 시스템(101)은 특정 기준들에 기초해 유효한 (기존의 또는 시뮬레이션된) 비즈니스 아키텍처를 표현하는 하나 이상의 프로세스 흐름 컴포넌트들을 가진 프로세스 흐름 모델(142)을 액세스할 수 있다. 프로세스 흐름 모델(142)은 산업의 최상 관행 또는 비즈니스에 유익한 소정의 다른 규칙들의 집합을 고수하는 프로세스 흐름 컴포넌트들의 구성을 포함할 수 있다.

프로세스 흐름 모델(142)은, 예를 들어, 제약들을 준수하는 좀더 일반적인 모델의 인스턴스를 발생시키는 데에 사용되는 좀더 일반적인 모델에 대한 제약들과 같은, 하나 이상의 제약들을 준수함을 나타내는 특성 값들을 포함할 수 있다. 제1 특성 값들은 프로세스 흐름 모델(142)에 포함되어 있으며 프로세스 흐름 스키마에 따라 포맷팅된 하나 이상의 이름/값 쌍들의 값 부분일 수 있다. 프로세스 흐름 모델(142)은 실제비즈니스 아키텍처를 위한 프로세스 흐름 모델이거나 시뮬레이션된 프로세스 흐름 모델일 수 있다.

방법(700)은 기존 비즈니스 아키텍처를 표현하는 제2 비즈니스 컴포넌트들의 제2의 구조화된 비즈니스 모델을 액세스하는 단계를 포함하는데, 제2 비즈니스 컴포넌트들은 제2 특성 값들을 가진다(단계 702). 예를 들어, 컴퓨터 시스템(101)은 기존 비즈니스 아키텍처를 표현하는 하나 이상의 프로세스 흐름 컴포넌트들을 가진 프로세스 흐름 모델(143)을 액세스할 수 있다. 제2 특성 값들은 프로세스 흐름 모델(143)에 포함되어 있으며 프로세스 흐름 스키마에 따라 포맷팅된 하나 이상의 이름/값 쌍들의 값 부분일 수 있다.

방법(700)은, 제2 특성 값들이 하나 이상의 제약들을 준수하는지를 검사하기 위해, 적어도 준수하는 제1 특성 값들의 부분 집합을 적어도 제2 특성 값들의 대응하는 부분집합과 비교하는 단계를 포함한다(단계 703). 예를 들어, 제약 모듈(116)은 프로세스 흐름 모델(142)의 특성 값들 중 일부를 프로세스 흐름 모델(143)의 특성 값들 중 대응하는 부분과 비교할 수 있다.

방법(700)은, 비교에 기초해, 제2의 구조화된 비즈니스 모델이 유효한지를 판정하는 단계를 포함한다(단계 704). 예를 들어, 제약 모듈(116)은, 특성 값 비교에 기초해, 프로세스 흐름 모델(143)이 유효한지를 판정할 수 있다. 다시 말해, 제약 모듈(116)은, 프로세스 흐름 모델(143)이 프로세스 흐름 모델(142)을 개시하는 데에 사용된 동일한 제약들에 따라 개시되었는지를 판정할 수 있다.

예를 들어, 제1 및 제2 특성 값 모두가 2인 경우와 같이, 제2 특성 값이 제1 특성 값과 매칭될 경우, 제약을 준수함을 나타낼 수 있다. 그러나, 제2 특성 값이, 예를 들어, 제1 특성 값의 10% 이내인 것과 같이, 제1 특성 값의 소정 임계치 이내일 경우에도, 제약을 준수함을 나타낼 수 있다. 제2 특성 값이, 예를 들어, 50과 100 사이와 같이, 특정 범위 이내인 경우에도, 제약을 준수함을 나타낼 수 있다. 제약들을 검사하는 단계의 결과들은 사용자-인터페이스(111)에 제시될 수 있고 그리고/또는 소정의 다른 컴퓨터 시스템으로 결과들(161)로서 전송될 수 있다.

도 8은 비즈니스 모델들을 개선하기 위한 방법(800)의 예시적인 흐름도를 도시한다. 방법(800)은 컴퓨터 아키텍처(100)의 컴포넌트들과 관련하여 논의될 것이다.

방법(800)은 비즈니스 아키텍처를 표현하는 구조화된 비즈니스 모델을 액세스하는 단계를 포함하는데, 구조화된 비즈니스 모델은 데이터 모델에 따라 구조화된다(단계 801). 예를 들어, 컴퓨터 시스템(100)은 구매 발주 의뢰(purchase order requisition)의 비즈니스 아키텍처를 지원하기 위한 일반적인 편성 계층 구조를 표현하는 편성 모델(152)을 액세스할 수 있다. 편성 모델(152)은 편성 스키마에 따라 구조화될 수 있다.

방법(800)은, 구조화된 비즈니스 모델에 기초해 모델링될 특정한 비즈니스 아키텍처를 식별하는 단계를 포함하는데, 특정한 비즈니스 아키텍처는 비즈니스 아키텍처보다 좀더 구체적이다(단계 802). 예를 들어, 개선 모듈(117)은, 편성 모델(152)에 기초해 건강 관리 (또는 소정의 다른 산업이나 그렇지 않으면 좀더 구체적인) 구매 발주 의뢰의 비즈니스 아키텍처가 모델링될 것이라는 것을 식별할 수 있다.

방법(800)은 특정한 비즈니스 아키텍처를 위한 비즈니스 특유 데이터를 액세스하는 단계를 포함한다(단계 803). 예를 들어, 개선 모듈(117)은 건강 관리 (또는 소정의 다른 산업이나 그렇지 않으면 좀더 구체적인) 구매 발주 의뢰의 비즈니스 아키텍처를 모델링하기 위해 건강 관리업 데이터(또는 소정의 다른 산업이나 그렇지 않으면 좀더 구체적인 데이터)를 액세스할 수 있다.

방법(800)은, 액세스된 비즈니스 특유 데이터에 기초해, 구조화된 비즈니스 모델을 비즈니스 특유의 구조화된 데이터 모델로 개선하는 단계를 포함한다(단계 804). 예를 들어, 개선 모듈(117)은 편성 모델(152)을 발생시키는 데에 사용되는 (예를 들어, 데이터 포맷들을 추가하고, 변경하며, 변경하는) 편성 스키마를 개선하기 위해 개선 연산자를 구현할 수 있다. 그에 따라, 개선 모듈(117)은, 건강 관리업 (또는 다른 산업이나 그렇지 않으면 좀더 구체적인) 데이터에 기초해, 편성 스키마를 건강 관리 특정의 (또는 소정의 다른 산업이나 그렇지 않으면 좀더 구체적인) 편성 스키마로 개선할 수 있다. 예를 들어, 건강 관리 구매 발주 의뢰의 비즈니스 아키텍처는, 다른 산업들에서는 요구되지 않는 다양한 보안 제어들을 가질 것이 요구될 수 있다. 이러한 추가적인 보안 제어들은 특수하게 훈련된 추가 인원들을 요할 수 있다. 그에 따라, 편성 스키마는 특수하게 훈련된 사람들을 표현하기 위한 데이터 포맷들을 포함하도록 변경될 수 있다.

방법(800)은 비즈니스 특유의 구조화된 데이터 모델에 따라 특정 비즈니스 아키텍처를 모델링하는 단계를 포함한다(단계 805). 예를 들어, 컴퓨터 시스템(101)은 건강 관리업 특정의 (또는 다른 산업이나 그렇지 않으면 좀더 구체적인) 편성 스키마에 따라 건강 관리 (또는 다른 산업이나 그렇지 않으면 좀더 구체적인) 구매 발주 의뢰의 비즈니스 아키텍처에 대한 편성 계층 구조를 표현하는 편성 모델을 발생시킬 수 있다. 좀더 구체적인 모델은 사용자-인터페이스(111)에 제시되거나 그리고/또는 소정의 다른 컴퓨터 시스템으로 결과들(161)로서 전송될 수 있다.

도 9는 비즈니스 모델을 구성하기 위한 방법(900)의 예시적인 흐름도를 도시한다. 방법(900)은 컴퓨터 아키텍처(100)의 컴포넌트들과 관련하여 논의될 것이다.

방법(900)은, 비즈니스 컴포넌트들의 집합을 포함하는 비즈니스 아키텍처가 모델링될 것이라는 표시를 수신하는 단계를 포함한다(단계 901). 예를 들어, 컴퓨터 시스템(101)은, 한 집합의 서비스들이 모델링될 것이라는 표시를 수신할 수 있다.

방법(900)은, 비즈니스 컴포넌트들의 집합 중 제1 부분이 제1의 구조화된 비즈니스 모델에 포함되어 있다는 것을 식별하는 단계를 포함하는데, 제1의 구조화된 비즈니스 모델은 복수개의 상이한 비즈니스 모델들 중에서 선택된다(단계 902). 예를 들어, 구성 모듈(118)은, 서비스 컴포넌트들의 제1 부분이 서비스 모델(132)에 포함되어 있다는 것을 식별할 수 있다.

방법(900)은, 제1의 구조화된 비즈니스 모델로부터 비즈니스 컴포넌트들의 집합 중 제1 부분을 추출하는 단계를 포함한다(단계 903). 예를 들어, 구성 모듈(118)은 서비스 모델(132)로부터 서비스 컴포넌트들의 일부 또는 전부를 추출할 수 있다.

방법(900)은, 비즈니스 컴포넌트들의 집합 중 제2 부분이 제2의 구조화된 비즈니스 모델에 포함되어 있다는 것을 식별하는 단계를 포함하는데, 제2의 구조화된 비즈니스 모델은 복수개의 상이한 비즈니스 모델들 중에서 선택된다(단계 904). 예를 들어, 구성 모듈(118)은, 서비스 컴포넌트들의 제2 부분이 서비스 모델(133)에 포함되어 있다는 것을 식별할 수 있다. 방법(900)은, 제2의 구조화된 비즈니스 모델로부터 비즈니스 컴포넌트들의 집합 중 제2 부분을 추출하는 단계를 포함한다(단계 905). 예를 들어, 구성 모듈(118)은 서비스 모델(133)로부터 서비스 컴포넌트들의 일부 또는 전부를 추출할 수 있다.

방법(900)은 적어도 비즈니스 컴포넌트들의 집합 중 제1 부분 및 제2 부분을 비즈니스 아키텍처의 모델로 구성하는 단계를 포함한다(단계 906). 예를 들어, 구성 모듈(118)은 서비스 모델들(132 및 133)로부터의 서비스 컴포넌트들을 새로운 서비스 모델로 구성할 수 있다. 새로운 서비스 모델은 사용자-인터페이스(111)에 제시되거나 그리고/또는 소정의 다른 컴퓨터 시스템으로 결과들(161)로서 전송될 수 있다.

본 발명의 실시예들이 형식 연산들의 조합들을 구현할 수 있다는 것도 이해할 수 있어야 한다. 예를 들어, 개선 연산자는, 복수개의 다른 비즈니스 모델들의 일부분들로부터 앞서 구성된 새로운 비즈니스 모델을 개선하는 데에 사용될 수 있다. 개선에 이어, 새로운 비즈니스 모델은 제약들을 준수하도록 검사될 수 있다.

도 10 및 다음의 논의는, 본 발명이 구현될 수 있는 적당한 컴퓨팅 환경에 대한 간략하고 일반적인 설명을 제공하기 위한 것이다. 반드시 그래야 하는 것은 아니지만, 본 발명은 컴퓨터 시스템들에 의해 실행되는, 프로그램 모듈들과 같은, 컴퓨터-실행 가능 명령어들의 일반적인 맥락으로 구현될 수 있다. 일반적으로, 프로그램 모듈들은 특정 태스크들을 수행하거나 특정한 추상 데이터형들을 구현하는 루틴들, 프로그램들, 객체들, 컴포넌트들, 데이터 구조들 등을 포함한다. 데이터 구조들과 연관된, 컴퓨터-실행 가능 명령어 및 프로그램 모듈들은 여기에 개시되어 있는 방법들의 단계들을 실행하기 위한 프로그램 코드 수단의 예들을 표현한다.

도 10을 참조하면, 본 발명을 구현하기 위한 예시적 시스템은, 프로세싱 유닛(1021), 시스템 메모리(1022), 및 시스템 메모리(1022)를 포함하는 다양한 시스템 컴포넌트들을 프로세싱 유닛(1021)에 커플링하는 시스템 버스(1023)를 포함하는, 컴퓨터 시스템(1020) 형태의 범용 컴퓨팅 장치를 포함한다. 프로세싱 유닛(1021)은, 본 발명의 사양들을 포함하여, 컴퓨터 시스템(1020)의 사양들을 구현하도록 설계된 컴퓨터-실행 가능 명령어들을 실행할 수 있다. 시스템 버스(1023)는 메모리 버스 또는 메모리 컨트롤러, 주변 장치 버스, 및 다양한 버스 아키텍처들 중 하나를 사용하는 로컬 버스를 포함하는 몇가지 유형들의 버스 구조들 중 하나일 수 있다. 시스템 메모리는 ROM(read only memory; 1024) 및 RAM(random access memory; 1025)을 포함한다. 스타트-업(start-up) 동안과 같은 때에, 컴퓨터 시스템(1020)내의 요소들 사이에서 정보 전달을 돕는 기본적인 루틴들을 포함하는 BIOS(basic input/output system; 1026)는 ROM(1024)에 저장될 수 있다.

또한, 컴퓨터 시스템(1020)은 자기 하드 디스크(1039)로부터 판독하고 그에 기입하기 위한 자기 하드 디스크 드라이브(1027), 분리형 자기 디스크(1029)로부터 판독하고 그에 기입하기 위한 자기 디스크 드라이브(1028) 및, 예를 들어, CD-ROM 또는 다른 광 매체와 같은, 분리형 광 디스크(1031)로부터 판독하고 그에 기입하기 위한 광 디스크 드라이브(1030)를 포함할 수도 있다. 자기 하드 디스크 드라이브(1027), 자기 디스크 드라이브(1028), 및 광 디스크 드라이브(1030)는, 각각, 하드 디스크 드라이브 인터페이스(1032), 자기 디스크 드라이브 인터페이스(1033), 및 광 드라이브 인터페이스(1034)에 의해 시스템 버스(1023)에 접속된다. 드라이브들 및 그들과 연관된 컴퓨터-판독 가능 매체들은 컴퓨터 시스템(1020)에 컴퓨터-실행 가능 명령어들, 데이터 구조들, 프로그램 모듈들, 및 다른 데이터의 비휘발성 저장을 제공한다. 여기에서 설명되는 예시적 환경은 자기 하드 디스크(1039), 분리형 자기 디스크(1029), 및 분리형 광 디스크(1031)를 이용하지만, 자기 카세트들, 플래시 메모리 카드들, DVD들, 베르누이 카트리지들, RAM들, ROM들 등을 포함하여, 데이터를 저장하기 위한 컴퓨터 판독 가능 매체들의 다른 유형들이 사용될 수도 있다.

운영 시스템(1035), 하나 이상의 어플리케이션 프로그램들(1036), 다른 프로그램 모듈들(1037), 및 프로그램 데이터(1038)를 포함하는, 하나 이상의 프로그램 모듈들을 포함하는 프로그램 코드 수단이 하드 디스크(1039), 자기 디스크(1029), 광 디스크(1031), ROM(1024), 또는 RAM(1025)에 저장될 수 있다. 사용자는 키보드(1040), 포인팅 장치(1042) 또는, 예를 들어, 마이크로폰, 조이스틱, 게임 패드, 스캐너 등과 같은, (나타내지 않은) 다른 입력 장치들을 통해 컴퓨터 시스템(1020)에 명령들 및 정보를 입력할 수 있다. 이들 및 다른 입력 장치들은 시스템 버스(1023)에 커플링되어 있는 입/출력 인터페이스(1046)를 통해 프로세싱 유닛(1021)에 접속될 수 있다. 입/출력 인터페이스(1046)는 논리적으로, 예를 들

어, 직렬 포트 인터페이스, PS/2 인터페이스, 병렬 포트 인터페이스, USB(Universal Serial Bus) 인터페이스, 또는 IEEE (Institute of Electrical and Electronics Engineers) 1394 인터페이스(즉, FireWire 인터페이스)와 같은, 광범위하고 상이한 인터페이스들 중 하나를 표현하거나, 심지어는 상이한 인터페이스들의 조합을 논리적으로 표현할 수 있다.

또한, 모니터(1047) 또는 다른 디스플레이 장치가 비디오 인터페이스(1048)를 통해 시스템 버스(1023)에 접속되어 있다. 또한, 스피커들 또는 다른 오디오 출력 장치가 오디오 인터페이스를 통해 시스템 버스(1023)에 접속되어 있다. 예를 들어, 프린터들과 같은, (나타내지 않은) 다른 주변 출력 장치들도 컴퓨터 시스템(1020)에 접속될 수 있다.

컴퓨터 시스템(1020)은, 예를 들어, 사무실-범위 또는 기업-범위의 컴퓨터 네트워크, 홈 네트워크, 인트라넷, 및/또는 인터넷과 같은, 컴퓨터 네트워크들에 접속될 수 있다. 컴퓨터 시스템(1020)은 이러한 컴퓨터 네트워크들을 통해, 예를 들어, 원격 컴퓨터 시스템들, 원격 어플리케이션들, 및/또는 원격 데이터베이스들과 같은, 외부 소스들과 데이터를 교환할 수 있다.

컴퓨터 시스템(1020)은, 그를 통해 컴퓨터 시스템(1020)이 외부 소스들로부터 데이터를 수신하고 그리고/또는 외부 소스들로 데이터를 전송하는, 네트워크 인터페이스(1053)를 포함한다. 도 1에 도시되어 있는 바와 같이, 네트워크 인터페이스(1053)는 링크(1051)를 통한 원격 컴퓨터 시스템(1083)과의 데이터 교환을 용이하게 한다. 네트워크 인터페이스(1053)는 논리적으로, 예를 들어, 네트워크 인터페이스 카드 및 대응하는 NDIS(Network Driver Interface Specification) 스택과 같은, 하나 이상의 소프트웨어 및/또는 하드웨어 모듈들을 표현할 수 있다. 링크(1051)는 컴퓨터 네트워크의 일부(예를 들어, 이더넷 세그먼트)를 표현하고, 원격 컴퓨터 시스템(1083)은 컴퓨터 네트워크의 노드를 표현한다.

마찬가지로, 컴퓨터 시스템(1020)은, 그를 통해 컴퓨터 시스템(1020)이 외부 소스들로부터 데이터를 수신하고 그리고/또는 외부 소스들로 데이터를 전송하는, 입/출력 인터페이스(1046)를 포함한다. 입/출력 인터페이스(1046)는, 그를 통해 컴퓨터 시스템(1020)이 외부 소스들로부터 데이터를 수신하고 그리고/또는 외부 소스들로 데이터를 전송하는, 모뎀(1054; 예를 들어, 표준 모뎀, 케이블 모뎀, 또는 DSL(digital subscriber line) 모뎀)에 커플링된다. 도 10에 도시된 바와 같이, 입/출력 인터페이스(1046) 및 모뎀(1054)은 링크(1052)를 통한 원격 컴퓨터 시스템(1093)과의 데이터 교환을 용이하게 한다. 링크(1052)는 컴퓨터 네트워크의 일부를 표현하고 원격 컴퓨터 시스템(1093)은 컴퓨터 네트워크의 노드를 표현한다.

도 10이 본 발명을 위한 적당한 동작 환경을 표현하기는 하지만, 본 발명의 원리들은, 필요하다면 적당히 변경되어, 본 발명의 원리들을 구현할 수 있는 임의의 시스템에 이용될 수 있다. 도 10에 도시된 환경은 단지 예시일 뿐이며, 결코 본 발명의 원리들이 구현될 수 있는 광범위한 환경들의 작은 부분조차도 표현하지 못한다.

본 발명에 따르면, 비교 모듈들, 매핑 모듈들, 제약 모듈들, 개선 모듈들, 구성 모듈들, 및 사용자-인터페이스들 뿐만 아니라 비즈니스 모델링 스키마들, 비즈니스 모델들, 및 비즈니스 모델 연산들에 의해 발생하는 결과들을 포함하는 관련 데이터가 컴퓨터 시스템(1020)과 연관된 컴퓨터-관독 가능 매체들 중 하나로부터 저장되어 액세스될 수 있다. 예를 들어, 이러한 모듈들의 일부분들 및 연관된 프로그램 데이터의 일부분들은, 시스템 메모리(1022)에서의 저장을 위해, 운영 시스템(1035), 어플리케이션 프로그램들(1036), 프로그램 모듈들(1037) 및/또는 프로그램 데이터(1038)에 포함될 수 있다.

예를 들어, 자기 하드 디스크(1039)와 같은, 대용량 저장 장치가 컴퓨터 시스템(1020)에 커플링될 경우, 이러한 모듈들 및 연관된 프로그램 데이터 또한 대용량 저장 장치에 저장될 수 있다. 컴퓨터 네트워크 환경에서, 컴퓨터 시스템(1020)과 관련하여 도시된 프로그램 모듈들 또는 그것의 일부분들은, 원격 컴퓨터 시스템(1083) 및/또는 원격 컴퓨터 시스템(1093)과 연관된 시스템 메모리 및/또는 대용량 저장 장치들과 같은, 원격 메모리 저장 장치들에 저장될 수 있다. 이러한 모듈들의 실행은 상술된 바와 같이 분산 환경에서 수행될 수도 있다.

본 발명은 그것의 취지 또는 본질적인 특징들을 벗어나지 않으면서 특정한 다른 형태들로 구체화될 수도 있다. 설명된 실시예들은, 모든 태양들에서, 한정이 아닌 일례로서만 간주되어야 한다. 따라서, 본 발명의 범위는 상기 명세서에 의해서가 아니라 첨부된 청구항들에 의해 나타내어진다. 청구항들과 등가인 의미 및 범위내에 해당되는 모든 변화들도 청구항들의 범위내에 포함되어야 한다.

발명의 효과

비즈니스 모델들을 비교하고 대조하기 위한 방법들, 시스템들, 컴퓨터 프로그램 제품들, 및 데이터 구조들에 관한 본 발명의 원리들에 따르면, 비즈니스 모델들의 융통성 있는 조작을 용이하게 하는 시스템들, 방법들, 컴퓨터 프로그램 제품들, 및 데이터 구조들이 제공될 수 있다.

(57) 청구의 범위

청구항 1.

컴퓨터 시스템에서, 유사한 유형의 비즈니스 모델들을 비교하는 방법으로서,

기존 비즈니스 아키텍처를 표현하는 구조화된 비즈니스 모델의 제1 인스턴스를 액세스하는 단계 - 제1의 구조화된 비즈니스 모델은 데이터 모델에 따라 구조화됨 - ,

제2 비즈니스 아키텍처를 표현하는 상기 구조화된 비즈니스 모델의 제2 인스턴스를 액세스하는 단계 - 제2의 구조화된 비즈니스 모델은 상기 데이터 모델에 따라 구조화됨 - ,

상기 구조화된 비즈니스 모델의 제1 인스턴스를 상기 구조화된 비즈니스 모델의 제2 인스턴스와 비교하는 단계,

상기 기존 비즈니스 아키텍처와 상기 제2 비즈니스 아키텍처간의 임의의 차이들을 식별하는 단계, 및

상기 기존 비즈니스 아키텍처와 상기 제2 비즈니스 아키텍처간의 식별된 차이들에 기초하여 상기 기존 비즈니스 아키텍처의 충분성(sufficiency)을 평가하는 단계를 포함하는 방법.

청구항 2.

제1항에 있어서,

상기 기존 비즈니스 아키텍처를 표현하는 구조화된 비즈니스 모델의 제1 인스턴스를 액세스하는 단계는 비즈니스 기능 모델의 제1 인스턴스를 액세스하는 단계를 포함하는 방법.

청구항 3.

제2항에 있어서,

상기 기존 비즈니스 아키텍처를 표현하는 구조화된 비즈니스 모델의 제1 인스턴스를 액세스하는 단계는 상기 비즈니스 기능 모델의 제2 인스턴스를 액세스하는 단계를 포함하는 방법.

청구항 4.

제1항에 있어서,

상기 구조화된 비즈니스 모델의 제1 인스턴스를 상기 구조화된 비즈니스 모델의 제2 인스턴스와 비교하는 단계는 기존의 구조화된 비즈니스 모델을 최상 관행(best practice)의 구조화된 비즈니스 모델과 비교하는 단계를 포함하는 방법.

청구항 5.

제1항에 있어서,

상기 구조화된 비즈니스 모델의 제1 인스턴스를 상기 구조화된 비즈니스 모델의 제2 인스턴스와 비교하는 단계는 기존의 구조화된 비즈니스 모델을 제안된 이득(proposed acquisition)의 구조화된 비즈니스 모델과 비교하는 단계를 포함하는 방법.

청구항 6.

제1항에 있어서,

상기 기존 비즈니스 아키텍처와 상기 제2 비즈니스 아키텍처간의 식별된 차이들에 기초하여 상기 기존 비즈니스 아키텍처의 충분성을 평가하는 단계는 상기 기존 비즈니스 아키텍처가 산업의 최상 관행을 충족시키는지를 판정하는 단계를 포함하는 방법.

청구항 7.

제1항에 있어서,

상기 기존 비즈니스 아키텍처와 상기 제2 비즈니스 아키텍처간의 식별된 차이들에 기초하여 상기 기존 비즈니스 아키텍처의 충분성을 평가하는 단계는 상기 제2 비즈니스 아키텍처가 상기 기존 비즈니스 아키텍처에 임의의 상이하거나, 새롭거나, 증가된 기능들을 제공할 것인지를 판정하는 단계를 포함하는 방법.

청구항 8.

컴퓨터 시스템에서, 비즈니스 아키텍처의 대응하는 상이한 계층들을 표현하는 상이한 유형들의 비즈니스 모델들을 대조하는 방법으로서,

비즈니스 아키텍처의 제1 비즈니스 계층을 표현하는 구조화된 비즈니스 모델을 액세스하는 단계 - 상기 구조화된 비즈니스 모델은 구조화된 데이터 모델에 따라 상기 제1 비즈니스 계층의 하나 이상의 제1 비즈니스 계층 컴포넌트들을 모델링함 - ,

제2 비즈니스 계층을 표현하는 제2의 구조화된 모델을 액세스하는 단계 - 상기 제2의 구조화된 모델은 상이한 상기 제2 비즈니스 계층의 하나 이상의 제2 비즈니스 계층 컴포넌트들을 모델링함 - ,

상기 제1 비즈니스 계층의 구성 변화들이 상기 제2 비즈니스 계층에 어떻게 영향을 미치는지, 및 상기 제2 비즈니스 계층의 구성 변화들이 상기 제1 비즈니스 계층에 어떻게 영향을 미치는지가 판정될 수 있도록, 상기 하나 이상의 제1 비즈니스 계층 컴포넌트들과 상기 하나 이상의 제2 비즈니스 계층 컴포넌트들간의 관계들을 매핑하는 단계, 및

상기 매핑된 관계들을 이용하여 상기 구조화된 비즈니스 모델을 상기 제2의 구조화된 모델과 대조하는 단계를 포함하는 방법.

청구항 9.

제8항에 있어서,

상기 비즈니스 아키텍처의 제1 비즈니스 계층을 표현하는 구조화된 비즈니스 모델을 액세스하는 단계는 비즈니스 기능 계층의 구조화된 비즈니스 모델을 액세스하는 단계를 포함하는 방법.

청구항 10.

제8항에 있어서,

상기 비즈니스 기능 계층의 구조화된 비즈니스 모델을 액세스하는 단계는 기능 스키마에 따라 구조화된 비즈니스 기능 모델을 액세스하는 단계를 포함하는 방법.

청구항 11.

제9항에 있어서,

상기 제2 비즈니스 계층을 표현하는 제2의 구조화된 비즈니스 모델을 액세스하는 단계는 서비스 네트워크 계층의 구조화된 비즈니스 모델을 액세스하는 단계를 포함하는 방법.

청구항 12.

제8항에 있어서,

상기 하나 이상의 제1 비즈니스 계층 컴포넌트들과 상기 하나 이상의 제2 비즈니스 계층 컴포넌트들간의 관계들을 매핑하는 단계는 비즈니스 기능 계층 컴포넌트들과 네트워크 서비스 계층 컴포넌트들 사이를 매핑하는 단계를 포함하는 방법.

청구항 13.

제8항에 있어서,

상기 매핑된 관계들을 이용하여 상기 구조화된 비즈니스 모델을 상기 제2의 구조화된 모델과 대조하는 단계는 상기 매핑된 관계들을 이용하여 비즈니스 기능 계층이 네트워크 서비스 계층과 어떻게 상호작용하는지를 식별하는 단계를 포함하는 방법.

청구항 14.

컴퓨터 시스템에서, 비즈니스 모델이 하나 이상의 제약을 준수하는지를 확인하기 위한 방법으로서,

유효한 비즈니스 아키텍처를 표현하는 제1 비즈니스 컴포넌트들의 구조화된 비즈니스 모델을 액세스하는 단계 - 상기 제1 비즈니스 컴포넌트들은 상기 하나 이상의 제약에 대한 준수를 나타내는 대응하는 제1 특성 값들을 가지며, 상기 구조화된 비즈니스 모델은 데이터 모델에 따라 구조화된 - ,

기존 비즈니스 아키텍처를 표현하는 제2 비즈니스 컴포넌트들의 제2의 구조화된 비즈니스 모델을 액세스하는 단계 - 상기 제2 비즈니스 컴포넌트들은 제2 특성 값들을 가짐 - ,

상기 하나 이상의 제약의 준수와 관련하여 상기 제2 특성 값들을 검사하기 위하여, 적어도 준수하는 제1 특성 값들의 부분 집합을 적어도 상기 제2 특성 값들의 대응하는 부분집합과 비교하는 단계, 및

상기 비교에 기초하여, 상기 제2의 구조화된 비즈니스 모델이 유효한지를 판정하는 단계를 포함하는 방법.

청구항 15.

제14항에 있어서,

상기 유효한 비즈니스 아키텍처를 표현하는 제1 비즈니스 컴포넌트들의 구조화된 비즈니스 모델을 액세스하는 단계는 상기 하나 이상의 제약을 준수하는 비즈니스 기능 모델을 액세스하는 단계를 포함하는 방법.

청구항 16.

제15항에 있어서,

상기 하나 이상의 제약을 준수하는 비즈니스 기능 모델을 액세스하는 단계는 기능 스키마에 따라 구조화된 비즈니스 기능 모델을 액세스하는 단계를 포함하는 방법.

청구항 17.

제15항에 있어서,

상기 하나 이상의 제약을 준수하는 비즈니스 기능 모델을 액세스하는 단계는 하나 이상의 이름/값 쌍들을 액세스하는 단계를 포함하는 방법.

청구항 18.

제14항에 있어서,

상기 기존 비즈니스 아키텍처를 표현하는 제2 비즈니스 컴포넌트들의 제2의 구조화된 비즈니스 모델을 액세스하는 단계는 상기 기존 비즈니스 아키텍처의 제2 비즈니스 기능 모델을 액세스하는 단계를 포함하는 방법.

청구항 19.

제14항에 있어서,

상기 비교에 기초하여 상기 제2의 구조화된 비즈니스 모델이 유효한지를 판정하는 단계는 상기 제1 특성 값들 중 하나 이상과 상기 제2 특성 값들 중 대응하는 하나 이상이 동일한 값들을 가지는지를 판정하는 단계를 포함하는 방법.

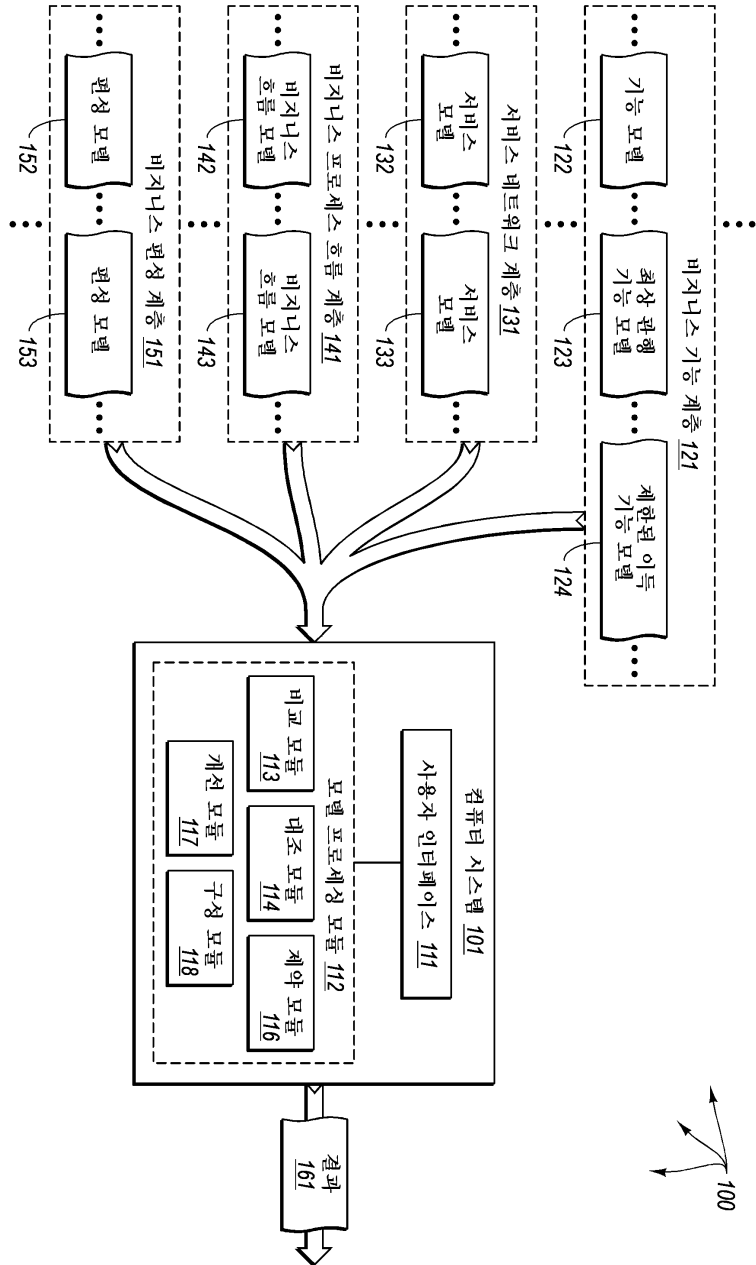
청구항 20.

제14항에 있어서,

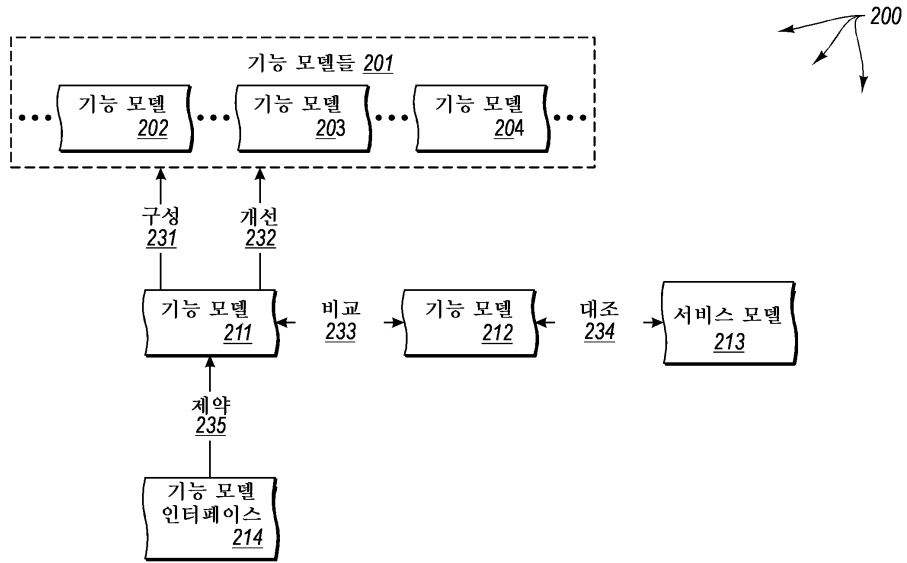
상기 비교에 기초하여 상기 제2의 구조화된 비즈니스 모델이 유효한지를 판정하는 단계는 상기 제1 특성 값들 중 하나 이상이 상기 제2 특성 값들 중 대응하는 하나 이상의 특정 임계치내에 속하는지를 판정하는 단계를 포함하는 방법.

도면

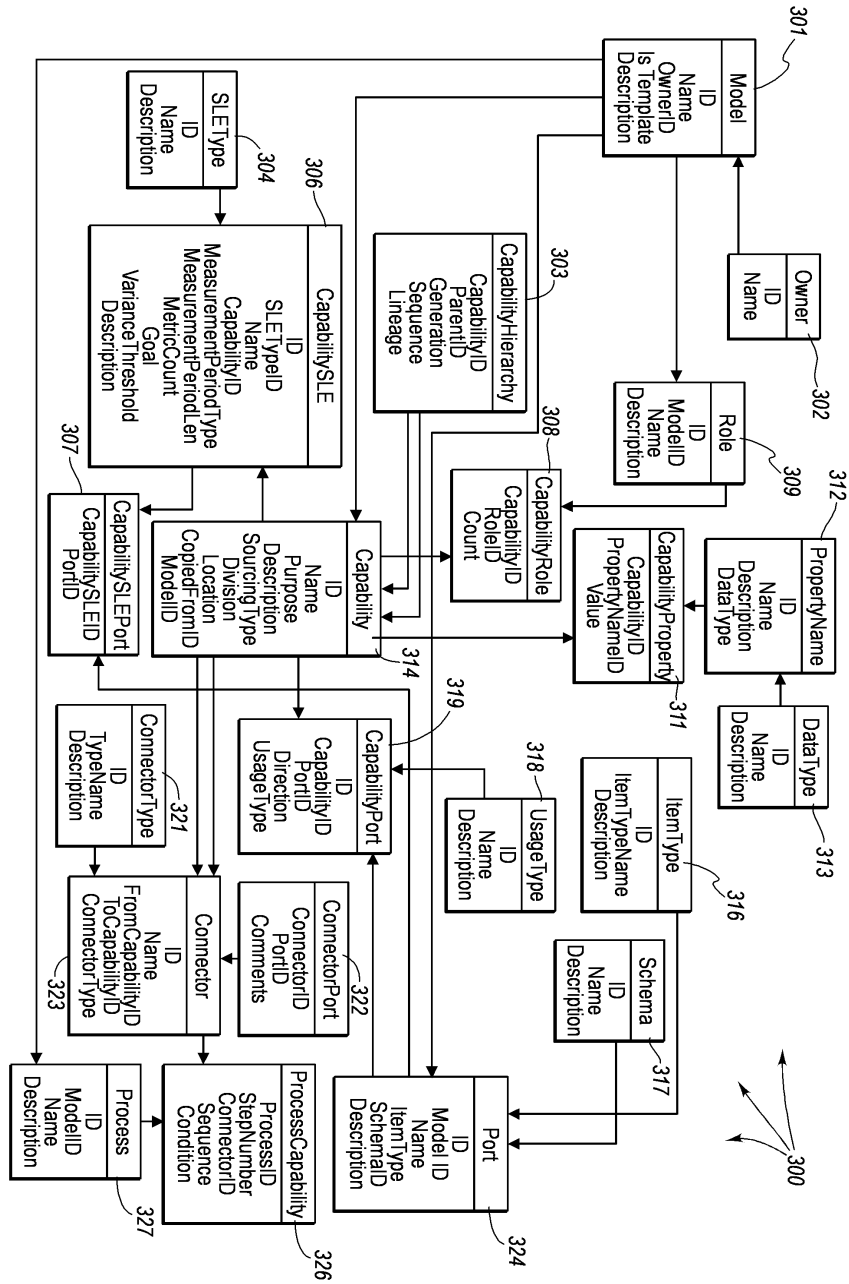
도면1



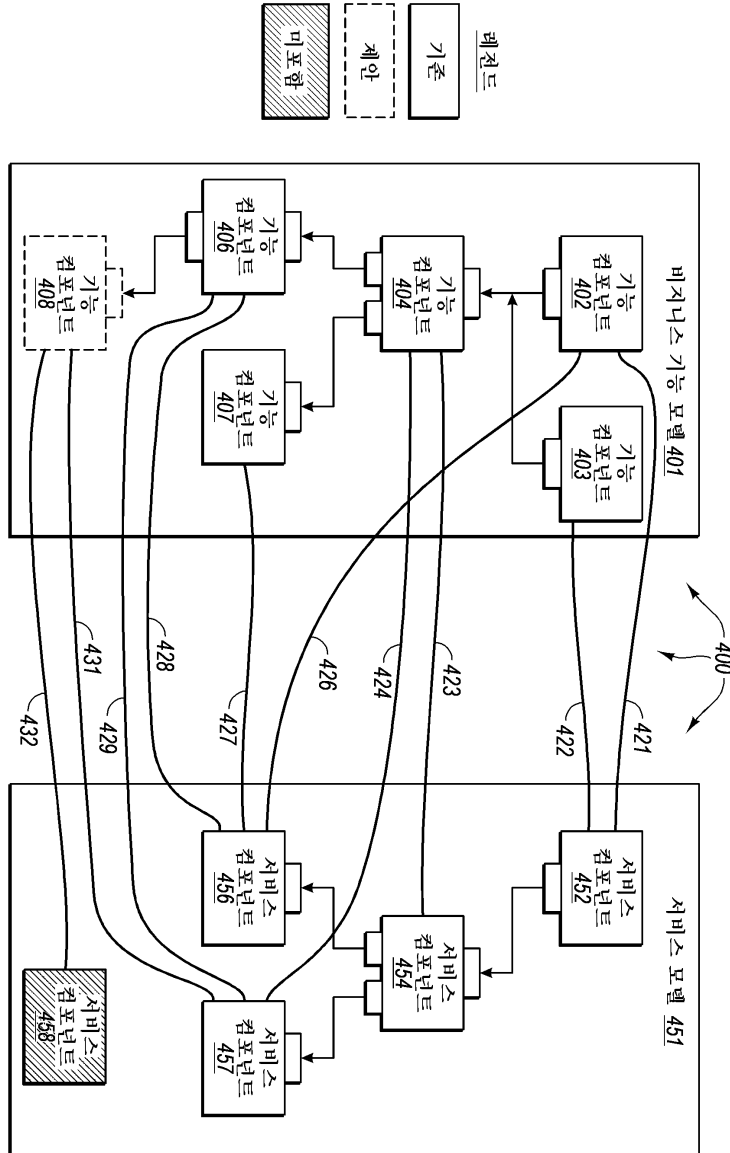
도면2



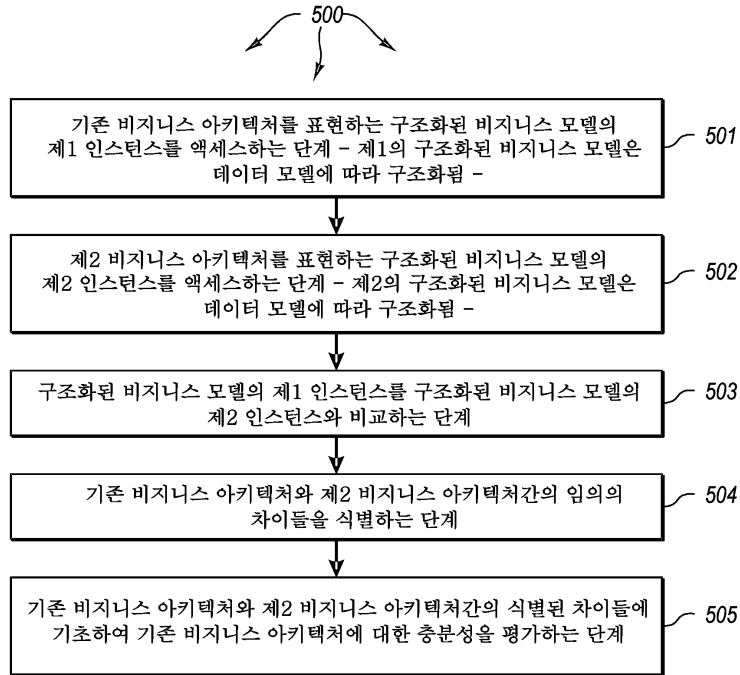
도면3



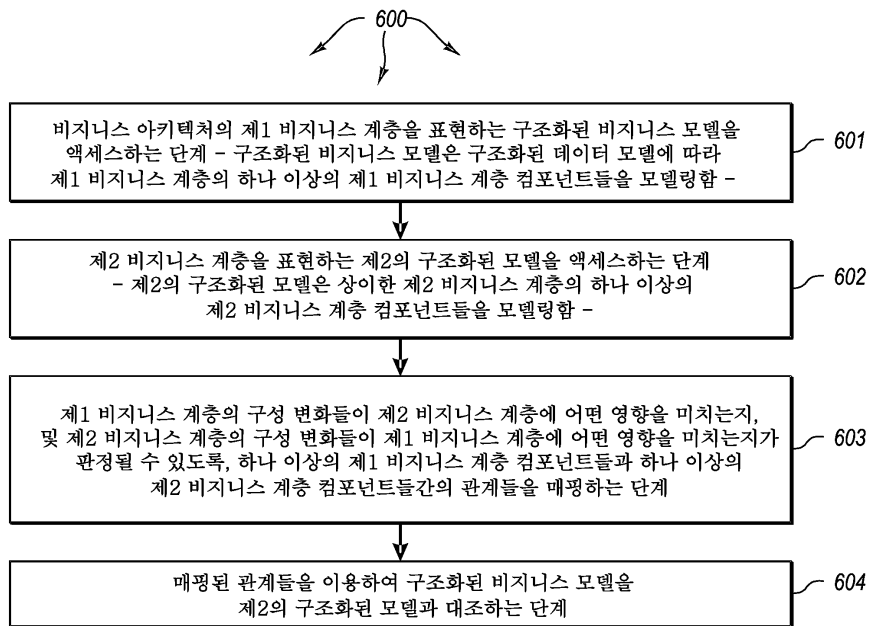
도면4



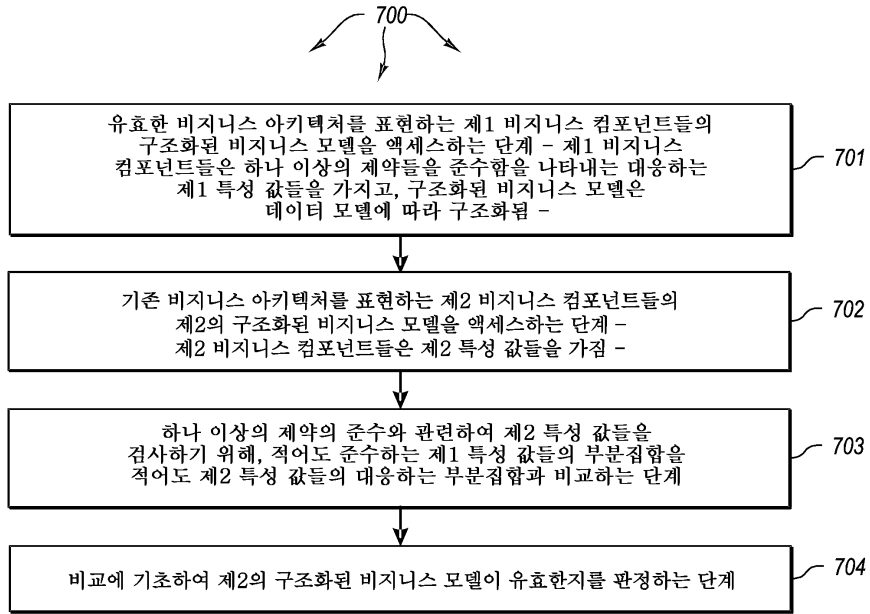
도면5



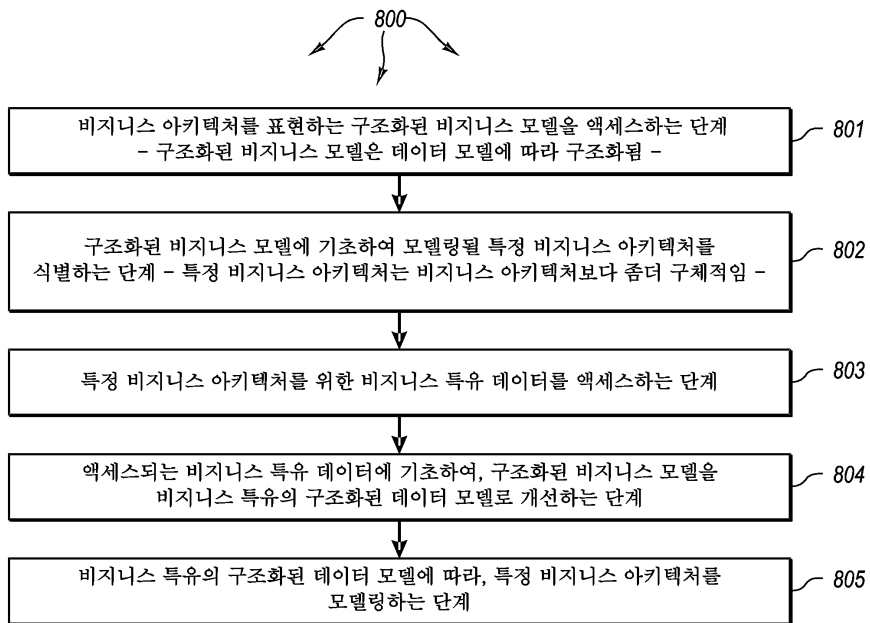
도면6



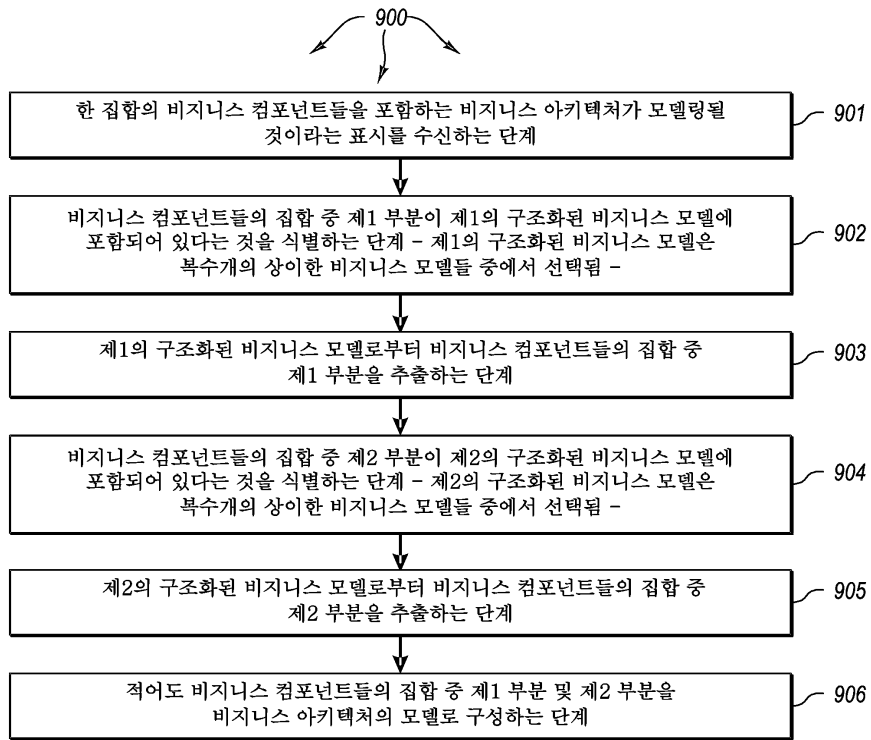
도면7



도면8



도면9



도면10

