



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0135773  
(43) 공개일자 2015년12월03일

(51) 국제특허분류(Int. Cl.)

G06F 9/54 (2006.01)

(52) CPC특허분류

G06F 9/542 (2013.01)

(21) 출원번호 10-2015-7026260

(22) 출원일자(국제) 2014년03월18일

심사청구일자 없음

(85) 번역문제출일자 2015년09월23일

(86) 국제출원번호 PCT/EP2014/055424

(87) 국제공개번호 WO 2014/154534

국제공개일자 2014년10월02일

(30) 우선권주장

13305390.0 2013년03월28일

유럽특허청(EPO)(EP)

(71) 출원인

톰슨 라이센싱

프랑스 92130 이씨레물리노 잔 다르크 뢰 1-5

(72) 발명자

마르텐스, 크리스토프

벨기에 2650 에데쎄 프린스 부데베인란 47 테크니컬러 - 톰슨 텔레콤 벨기에

차넷, 도미니크

벨기에 2650 에데쎄 프린스 부데베인란 47 테크니컬러 - 톰슨 텔레콤 벨기에

드 스몐트, 알렉스

벨기에 2650 에데쎄 프린스 부데베인란 47 테크니컬러 - 톰슨 텔레콤 벨기에

(74) 대리인

양영준, 전경석, 백만기

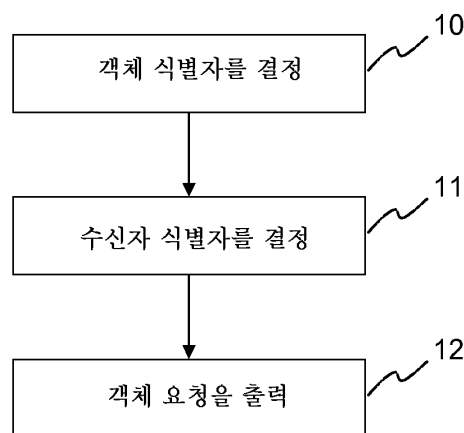
전체 청구항 수 : 총 15 항

(54) 발명의 명칭 발행-가입 환경 내에서의 객체 수명 주기 관리

### (57) 요약

발행-가입 환경 내에서의 객체들을 위한 해결법이 설명된다. 객체 결정 유닛(22)은 객체에 대한 객체 식별자를 결정하고(10), 반면 수신자 결정 유닛(23)은 객체를 조작할 수신자에 대한 수신자 식별자를 결정한다(11). 적어도 객체 식별자 및 수신자 식별자를 포함하는 객체 요청은 다음으로 발행-가입 환경에 출력된다(12). 객체 요청의 수신자들은 객체의 조작이 요구되는지 여부를 검사하고(31), 요구된 대로 객체를 조작한다(32).

대표도 - 도2



## 명세서

### 청구범위

#### 청구항 1

객체(object)의 조작(manipulation)을 요청하기 위한 방법으로서,

- 상기 객체에 대한 객체 식별자(object identifier)를 결정하는 단계(10);
  - 상기 객체를 조작할 수신자(recipient)에 대한 수신자 식별자(recipient identifier)를 결정하는 단계(11); 및
  - 적어도 상기 객체 식별자 및 상기 수신자 식별자를 포함하는 객체 요청(object request)을 발행-가입 환경(publish-subscribe environment)에 출력하는 단계(12)
- 를 포함하는 방법.

#### 청구항 2

제1항에 있어서,

상기 객체의 조작은 상기 객체의 생성, 상기 객체의 수정, 상기 객체의 삭제, 및 상기 객체의 이동 중 하나를 포함하는, 방법.

#### 청구항 3

제1항 또는 제2항에 있어서,

상기 객체 요청은 상기 객체의 이름 및 상기 객체의 특징들을 더 포함하는, 방법.

#### 청구항 4

객체의 조작을 요청하도록 구성된 장치(20)로서,

- 상기 객체에 대한 객체 식별자를 결정하도록(10) 구성된 객체 결정 유닛(object determining unit)(22);
  - 상기 객체를 조작할 수신자에 대한 수신자 식별자를 결정하도록(11) 구성된 수신자 결정 유닛(recipient determining unit)(23); 및
  - 적어도 상기 객체 식별자 및 상기 수신자 식별자를 포함하는 객체 요청을 발행-가입 환경에 출력하도록(12) 구성된 출력(24)
- 을 포함하는, 장치(20).

#### 청구항 5

제4항에 있어서,

상기 객체의 조작은 상기 객체의 생성, 상기 객체의 수정, 상기 객체의 삭제, 및 상기 객체의 이동 중 하나를 포함하는, 장치(20).

#### 청구항 6

제4항 또는 제5항에 있어서,

상기 객체 요청은 상기 객체의 이름 및 상기 객체의 특징들을 더 포함하는, 장치(20).

#### 청구항 7

객체의 조작을 요청하는 것을 가능하게 하는 명령어들이 저장되어 있는 컴퓨터 판독가능 저장 매체로서,

상기 명령어들은 컴퓨터에 의해 실행되는 경우, 상기 컴퓨터가

- 상기 객체에 대한 객체 식별자를 결정하고(10);
- 상기 객체를 조작할 수신자에 대한 수신자 식별자를 결정하며(11);
- 적어도 상기 객체 식별자 및 상기 수신자 식별자를 포함하는 객체 요청을 발행-가입 환경에 출력하게 하는(12),

컴퓨터 판독가능 저장 매체.

#### 청구항 8

제7항에 있어서,

상기 객체의 조작은 상기 객체의 생성, 상기 객체의 수정, 상기 객체의 삭제, 및 상기 객체의 이동 중 하나를 포함하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 9

제7항 또는 제8항에 있어서,

상기 객체 요청은 상기 객체의 이름 및 상기 객체의 특징들을 더 포함하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 10

객체를 조작하기 위한 방법으로서,

- 적어도 객체 식별자 및 수신자 식별자를 포함하는 객체 요청을 발행-가입 환경으로부터 수신하는 단계(30);
- 상기 객체 요청을 수신하는 것(30)에 대한 응답으로, 객체의 조작이 요구되는지 여부를 검사하는 단계(31); 및
- 상기 객체를 요구된 대로 조작하는 단계(32)

를 포함하는, 방법.

#### 청구항 11

제10항에 있어서,

상기 객체의 조작은 상기 객체의 생성, 상기 객체의 수정, 상기 객체의 삭제, 및 상기 객체의 이동 중 하나를 포함하는, 방법.

#### 청구항 12

객체를 조작하도록 구성된 장치(40)로서,

- 적어도 객체 식별자 및 수신자 식별자를 포함하는 객체 요청을 발행-가입 환경으로부터 수신하도록(30) 구성된 입력(41);
- 상기 객체 요청을 수신하는 것(30)에 대한 응답으로, 객체의 조작이 요구되는지 여부를 검사하도록(31) 구성된 평가 유닛(42); 및
- 상기 객체를 요구된 대로 조작하도록(32) 구성된 조작 유닛(43)

을 포함하는, 장치(40).

#### 청구항 13

제12항에 있어서,

상기 객체의 조작은 상기 객체의 생성, 상기 객체의 수정, 상기 객체의 삭제, 및 상기 객체의 이동 중 하나를 포함하는, 장치(40).

#### 청구항 14

객체의 조작을 가능하게 하는 명령어들이 저장되어 있는 컴퓨터 판독가능 저장 매체로서,  
상기 명령어들은 컴퓨터에 의해 실행되는 경우, 상기 컴퓨터가

- 적어도 객체 식별자 및 수신자 식별자를 포함하는 객체 요청을 발행-가입 환경으로부터 수신하고(30);
  - 상기 객체 요청을 수신하는 것(30)에 대한 응답으로, 객체의 조작이 요구되는지 여부를 검사하며(31);
  - 상기 객체를 요구된 대로 조작하게 하는(32),
- 컴퓨터 판독가능 저장 매체.

#### 청구항 15

제14항에 있어서,

상기 객체의 조작은 상기 객체의 생성, 상기 객체의 수정, 상기 객체의 삭제, 및 상기 객체의 이동 중 하나를 포함하는, 컴퓨터 판독가능 저장 매체.

### 발명의 설명

#### 기술 분야

[0001] 본 발명은 객체 수명 주기 관리(object life cycle management)를 위한 해결법에 관한 것이다. 더 구체적으로, 본 발명은 발행-가입 환경(publish-subscribe environment) 내에서의 객체 수명 주기 관리를 위한 해결법에 관한 것이다.

#### 배경 기술

[0002] 소프트웨어 아키텍처 도메인에서, 발행-가입(P/S)은 발행자들(publishers)로 지칭되는 메시지 발송자들이 메시지들을 가입자들(subscribers)로 지칭되는 특정 수신자들에게 직접적으로 전송되도록 프로그래밍하지 않는 메시징 패턴(messaging pattern)이다. 대신, 발행된 데이터는 만일 가입자들이 있다면 어떠한 가입자들이 있을 것인지의 지식 없이 멀티캐스팅된다. 유사하게, 가입자들은 특정한 데이터를 구독하고, 만일 발행자들이 있다면 어떠한 발행자들이 있는지의 지식 없이 관심이 있는 메시지들만을 수신한다.

[0003] 예시적인 P/S 프레임워크가 도 1에서 도시된다. P/S 개체들(P/S entities) 외에도, P/S 토픽들(P/S Topics)과 애플리케이션들에 의해 사용되는 것과 같은 객체 인스턴스들 사이의 관계가 도시되어 있다. DDS 용어(DDS: Data Distribution Service)가 도 1에서 사용되는 것을 유의한다.

[0004] 그들이 연결되어 있는 객체, 즉 토픽 인스턴스가, 예를 들어, 애플리케이션이 그것을 소정의 방식으로 핸들링하는 것으로 인하여 또는 변화된 환경으로 인하여 변화를 경험하는 경우, 발행자들로부터 가입자들로의 발행 액션(publish action)이 있다.

[0005] 위에서 설명된 것과 같은 P/S 환경에서, 객체들을 핸들링하는 것, 구체적으로 객체 수명 주기의 문제가 있다.

[0006] 후술하는 객체 수명 주기 메소드들이 고려된다:

[0007] · 객체 생성;

[0008] · 객체 업데이트;

[0009] · 객체 삭제;

[0010] · 다른 컨텍스트(context)로의 객체 이동, 즉 하나의 컨텍스트 내에서의 객체 삭제 및 다른 컨텍스트 내에서의 동일한 객체 생성;

[0011] · 가능한 경우, 위의 메소드들의 조합들을 동시에 수행, 예를 들어 객체 이동 및 업데이트를 조합.

[0012] 사실, P/S 환경은 데이터-지향적(data-oriented)인 반면, 위의 액션들은 상당히 메소드-지향적(method-oriented)이다. 추가적으로, 메소드들은 객체가 생성되지 않는 한, 객체에 영향을 줄 수 없다. 다른 문제는 메소드를 발부하는(issuing) 개체들(역할들) 및 메소드를 수행하는 개체들(역할들)이 있어야 한다는 점이다.

## 발명의 내용

- [0013] 발행-가입 환경 내에서 객체를 조작하기(manipulating) 위한 개선된 해결법을 제안하는 것이 본 발명의 목적이다.
- [0014] 본 발명의 일 태양에 따르면, 객체의 조작을 요청하기 위한 방법은 다음의 단계들을 포함한다:
- [0015] - 객체에 대한 객체 식별자(object identifier)를 결정하는 단계;
- [0016] - 객체를 조작할 수신자(recipient)에 대한 수신자 식별자(recipient identifier)를 결정하는 단계; 및
- [0017] - 적어도 객체 식별자 및 수신자 식별자를 포함하는 객체 요청(object request)을 발행-가입 환경에 출력하는 단계.
- [0018] 이에 따라, 객체의 조작을 요청하도록 구성된 장치는 다음을 포함한다:
- [0019] - 객체에 대한 객체 식별자를 결정하도록 구성된 객체 결정 유닛(object determining unit);
- [0020] - 객체를 조작할 수신자에 대한 수신자 식별자를 결정하도록 구성된 수신자 결정 유닛(recipient determining unit); 및
- [0021] - 적어도 객체 식별자 및 수신자 식별자를 포함하는 객체 요청을 발행-가입 환경에 출력하도록 구성된 출력.
- [0022] 유사하게, 객체의 조작을 요청하는 것을 가능하게 하는 명령어들이 그 안에 저장되어 있는 컴퓨터 판독가능 저장 매체에서, 명령어들은 컴퓨터에 의해 실행되는 경우, 컴퓨터가
- [0023] - 객체에 대한 객체 식별자를 결정하고;
- [0024] - 객체를 조작할 수신자에 대한 수신자 식별자를 결정하며;
- [0025] - 적어도 객체 식별자 및 수신자 식별자를 포함하는 객체 요청을 발행-가입 환경에 출력하게 하게 한다.
- [0026] 바람직하게, 객체 요청은 객체의 이름 및 객체의 특징들을 더 포함한다.
- [0027] 본 발명의 추가적인 태양에 따르면, 객체를 조작하기 위한 방법은 다음의 단계들을 포함한다:
- [0028] - 적어도 객체 식별자 및 수신자 식별자를 포함하는 객체 요청을 발행-가입 환경으로부터 수신하는 단계;
- [0029] - 객체 요청을 수신하는 것에 대한 응답으로, 객체의 조작이 요구되는지 여부를 검사하는 단계; 및
- [0030] - 객체를 요구된 대로 조작하는 단계.
- [0031] 이에 따라, 객체를 조작하도록 구성된 장치는 다음을 포함한다:
- [0032] - 적어도 객체 식별자 및 수신자 식별자를 포함하는 객체 요청을 발행-가입 환경으로부터 수신하도록 구성된 입력;
- [0033] - 객체 요청을 수신하는 것에 대한 응답으로, 객체의 조작이 요구되는지 여부를 검사하도록 구성된 평가 유닛; 및
- [0034] - 객체를 요구된 대로 조작하도록 구성된 조작 유닛.
- [0035] 유사하게, 객체의 조작을 가능하게 하는 명령어들이 그 안에 저장되어 있는 컴퓨터 판독가능 저장 매체에서, 명령어들은 컴퓨터에 의해 실행되는 경우, 컴퓨터가
- [0036] - 적어도 객체 식별자 및 수신자 식별자를 포함하는 객체 요청을 발행-가입 환경으로부터 수신하고;
- [0037] - 객체 요청을 수신하는 것에 대한 응답으로, 객체의 조작이 요구되는지 여부를 검사하며;
- [0038] - 객체를 요구된 대로 조작하게 한다.
- [0039] 본 발명은 P/S 환경 내에서의 객체 수명 주기를 위한 프레임워크를 설명한다. 그것은 그 프레임워크 내의 관련 역할들을 정의하고, 모든 객체 수명 주기 메소드들을 수행하기 위한 단일 요청 메커니즘(one-request mechanism)을 정의한다. 더 구체적으로, 2개의 역할, 즉 메소드들을 발부하는 객체 관리자(Object Manager)의 역할 및 메소드들을 수행하는 객체 제공자(Object Provider)의 역할이 설명된다. CREATE, UPDATE, DELETE, 및 MOVE 메소드들이 단일 OBJECT\_REQUEST 메소드 내에 포함된다.

[0040] 프레임워크는 객체 관리자가, 그것이 자기 자신의 제어 환경 내에서 관리하는 것이 아니라 객체 제공자에 의해 제어되는 환경 내에서 관리하는, 객체들을 제어하거나 관리하는 것을 허용한다.

[0041] 제안된 해결법은 동적 객체 생성 및 핸들링을 요구하는 환경의 데이터 지향적 제어 시스템 내에서 객체 수명 주기 관리를 위한 메소드들을 사용하는 것을 허용한다. 단 하나의 메소드만으로, 완전한 객체 수명 주기가 커버될 수 있고, 따라서 해결책은 상당히 간단하다. 이와 동시에, 제안된 해결법은 프레임워크를 설명한다. 이처럼, 그것은 동적 객체 핸들링을 요구하는 많은 상이한 애플리케이션들에 대하여 재-사용될 수 있다. 그것은 단지 일반적인 객체 수명 주기 역할들을 특정 애플리케이션들에 의해 제공된 특정한 기능들로 매핑하는 것의 문제이다.

[0042] 더 나은 이해를 위하여, 본 발명은 이제부터 도면들을 참조하여 후술하는 설명에서 더 상세히 설명될 것이다. 본 발명이 이 예시적 실시예로 한정되지 않는 것이고, 상세한 특징들은 첨부되는 청구항들에서 정의된 것과 같은 본 발명의 범주를 벗어남이 없이도 또한 방편으로 결합되고/결합되거나 수정될 수 있는 것이 이해될 것이다.

### 도면의 간단한 설명

[0043] 도 1은 예시적인 P/S 프레임워크를 개략적으로 나타낸다;  
 도 2는 본 발명에 따른 객체의 조작을 요청하기 위한 방법을 도시한다;  
 도 3은 도 2의 방법을 수행하도록 구성된 장치를 개략적으로 나타낸다;  
 도 4는 본 발명에 따른 객체를 조작하기 위한 방법을 도시한다;  
 도 5는 도 4의 방법을 수행하도록 구성된 장치를 개략적으로 나타낸다;  
 도 6은 본 발명에 따른 P/S 프레임워크를 객체 관리자 및 객체 제공자들과 함께 도시한다;  
 도 7은 객체 관리자로부터 객체 제공자로의 OBJECT\_REQUEST의 전달을 나타낸다;  
 도 8은 본 발명에 따른 객체 핸들링 메커니즘에 관한 개관을 제공한다;  
 도 9는 제안된 객체 수명 주기 관리를 사용하여 실현된 CREATE 액션을 도시한다;  
 도 10은 제안된 객체 수명 주기 관리를 사용하여 실현된 UPDATE 액션을 도시한다;  
 도 11은 제안된 객체 수명 주기 관리를 사용하여 실현된 MOVE 액션을 도시한다;  
 도 12는 제안된 객체 수명 주기 관리를 사용하여 실현된 DELETE 액션을 도시한다.

### 발명을 실시하기 위한 구체적인 내용

[0044] 도 2는 본 발명에 따른 객체의 조작을 요청하기 위한 방법을 도시한다. 제1 단계에서, 객체에 대한 객체 식별자가 결정된다(10). 객체를 조작할 수신자에 대한 수신자 식별자도 결정된다(11). 적어도 결정된 객체 식별자 및 수신자 식별자는 다음으로 발행-가입 환경에 출력되는(12) 객체 요청 내에 포함된다. 의도된 조작에 따라, 객체 요청은 객체의 이름 및 객체의 특징들을 더 포함한다.

[0045] 도 2의 방법을 수행하도록 구성된 장치(20)가 도 3에서 개략적으로 나타내어진다. 장치(20)는 객체의 조작을 위한 요청을 게시하기 위해 요청을 수신하기 위한 입력(21)을 포함한다. 입력(21)은 예를 들어, 사용자 인터페이스 또는 애플리케이션이다. 객체 결정 유닛(22)은 객체에 대한 객체 식별자를 결정하는(10) 반면, 수신자 결정 유닛(23)은 객체를 조작할 수신자에 대한 수신자 식별자를 결정한다(11). 결정된 객체 식별자 및 수신자 식별자는 다음으로 요구되는 경우, 추가적인 데이터와 함께 객체 요청 내에 포함되고, 이는 출력(24)을 통해 발행-가입 환경에 출력된다(12). 물론, 장치(20)의 상이한 요소들은 또한 단일 유닛으로 결합될 수 있거나, 프로세서 상에서 실행되는 소프트웨어로서 구현될 수 있다.

[0046] 도 4는 본 발명에 따른 객체를 조작하기 위한 방법을 도시한다. 적어도 객체 식별자 및 수신자 식별자를 포함하는 객체 요청은 발행-가입 환경으로부터 수신된다(30). 객체 요청을 수신하는 것(30)에 대한 응답으로, 객체 요청의 수신자는 객체의 조작이 요구되는지 여부를 검사한다(31). 만약 그러한 경우, 객체는 요구되는 대로 조작된다(32).

[0047] 도 4의 방법을 수행하도록 구성된 장치(40)가 도 5에서 개략적으로 나타내어진다. 장치(40)는 적어도 객체 식

별자 및 수신자 식별자를 포함하는 객체 요청을 발행-가입 환경으로부터 수신(30)하도록 구성된 입력(41)을 포함한다. 평가 유닛(42)은 객체 요청을 수신하는 것(30)에 대한 응답으로 객체의 조작이 요구되는지 여부를 검사한다(31). 그러한 경우, 조작 유닛(43)은 요구되는 대로 객체를 조작한다(32). 적절한 경우, 조작된 객체는 출력(44)을 통해 발행-가입 환경에 출력된다. 물론, 장치(40)의 상이한 요소들은 또한 단일 유닛으로 결합될 수 있거나, 프로세서 상에서 실행되는 소프트웨어로서 구현될 수 있다.

[0048] 본 발명에 따른 P/S 객체 수명 주기 프레임워크는 동적으로 생성, 수정, 삭제, 이동된 객체들이 P/S 환경 내에서 핸들링되는 방법을 설명한다. 도 6에서 나타내어진 것과 같이, 프레임워크는 2개의 역할을 예상한다. 객체 관리자는 수명 주기 요청을 객체 제공자로 전송하는 역할을 실현한다. 객체 제공자는 요청에 반응하고, 액션 즉, 특정 객체에 대해 요청된 생성, 수정, 삭제 또는 이동을 수행한다. 객체 제공자는 객체들의 저장소이고, 객체 관리자의 요청에 따라 객체들의 수명 주기를 핸들링한다. 다수의 객체 관리자는 다수의 객체 제공자에게 요청들을 발부할 수 있다. 또한, 다수의 객체 제공자는 다수의 객체 관리자로부터 요청들을 수신할 수 있다.

[0049] 통상적으로, 객체 관리자는 사용자 인터페이스를 통해 사용자에게 의해 제어되는 애플리케이션이고, 이는 디바이스 상에 위치된 데이터 또는 디바이스 상의 애플리케이션 내의 데이터에 영향을 미친다. 객체 제공자는 애플리케이션 또는 디바이스 내의 제어 유닛이거나, 객체들의 집합을 그룹화하고 제어하는 디바이스 그 자체이다. 객체 제공자는 객체 관리자의 요청에 따른 조치를 취한다. 상이한 객체 제공자들이 서로의 존재의 인식을 가지는 것은 요구되지 않는다. 그들은 서로간에 독립적이다.

[0050] 노트 핸들링 애플리케이션('스티키 노트들(sticky notes)')에서, 어떠한 객체들일 수 있는지의 소정의 예시들을 제공하기 위해, 객체들은 임의의 특정한 시간에서 사용가능한 상이한 노트들이다. 문서 핸들링 애플리케이션에서, 객체들은 상이한 문서들의 표시들이다. 멀티미디어 애플리케이션에서, 객체들은 애플리케이션에 의해 제어되는 미디어 최종지점들(endpoints)이다.

[0051] 이하에서, 모든 객체 수명 주기 메소드들을 나타내는 단일 객체 수명 주기 요청은 도 7을 참조하여 설명될 것이다. 이 요청의 예시적인 포맷은 다음과 같다:

[0052] OBJECT\_REQUEST (Object\_ID, Object\_Provider\_ID, Name, parameters (value))

[0053] 객체는 '토픽' 이름 및 Object\_ID에 의해 식별된다. 특정한 객체의 특징들은 값을 가지는 객체와 관련된 파라미터들에 의해 설명된다. P/S 도메인 내에서, Object\_ID 숫자들 및 Object\_Provider\_ID 숫자들은 고유하다.

[0054] 예를 들어, 객체가 노트인 경우, 각각의 노트는 숫자 또는 제목인 이름을 가질 수 있고, 특징들은 치수, 글씨체, 최대 글자수 등을 나타낼 수 있다.

[0055] P/S 환경 내에서, OBJECT\_REQUEST는 객체/토픽 그 자체로서 고려되고, 그것의 파라미터 값들은 객체 관리자가 값들을 변화시키는 경우 발행되는 것을 유의한다. OBJECT\_REQUEST들에 가입된 모든 객체 제공자들은 요청을 수신할 것이다. 어드레스된 객체 제공자만이 요청에 대한 조치를 취할 것이다.

[0056] 상이한 수명 주기 액션들에 대한 객체 수명 주기 요청의 사용은 다음과 같다:

[0057] CREATE

[0058] OBJECT\_REQUEST (Object\_ID=j, Object\_Provider\_ID=p, Name=n, parameters=value)

[0059] Object\_ID j는 (임의의 객체 제공자들에서) 존재하지 않는 것이다. 객체 제공자 p는 요청에 대한 반응으로서 그것의 영역 내에 객체를 생성한다. OBJECT\_REQUEST 내의 값들은 객체의 파라미터들에 할당된다.

[0060] DELETE

[0061] OBJECT\_REQUEST (Object\_ID=j, Object\_Provider\_ID=void)

[0062] 객체 제공자 p는 그것이 그것의 영역 내에 객체 j를 가지는 것을 인식한다. 객체 제공자 p는 요청에 대한 반응으로서 객체 j를 삭제한다.

[0063] UPDATE

[0064] OBJECT\_REQUEST (Object\_ID=j, Object\_Provider\_ID=p, Name=n, parameters=other value)

[0065] 객체 제공자 p는 그것이 객체 j를 그것의 영역 내에 가지는 것을 인식한다. 객체 제공자 p는 요청에 대한 반응으로서 객체 j의 파라미터들을 나타내어진 것과 같은 새로운 값들로 업데이트한다.



- [0066] MOVE
- [0067] OBJECT\_REQUEST (Object\_ID=j, Object\_Provider\_ID=q, Name=n, parameters=value)
- [0068] 객체 제공자 p는 그것이 객체 j를 그것의 영역 내에 가지는 것을 인식한다. 요청이 객체 j가 이제 객체 제공자 q에 속함을 나타내기 때문에, 객체 제공자 p는 객체 j를 그것의 영역으로부터 삭제한다. 객체 제공자 q는 그것의 영역 내에 나타내어진 것과 같은 파라미터들의 값들을 가진 객체 j를 생성한다.
- [0069] 도 8은 본 발명에 따른 객체 핸들링 메커니즘의 개관을 제공한다.
- [0070] 객체 관리자 내의 사용자 인터페이스 및/또는 애플리케이션을 통해, 객체 요청 액션(생성, 업데이트 등)이 입력된다(1). 객체 관리자 내의 OBJECT\_REQUEST 객체 내의 OBJECT\_REQUEST 객체 데이터는 대응하여 변화된다. 다음으로, OBJECT\_REQUEST 객체는 P/S 도메인 상에 발행된다(2). OBJECT\_REQUEST 객체는 OBJECT\_REQUEST 토픽에 가입되는 모든 객체 제공자들에 의하여 관독된다. 목표 객체 제공자는 요청을 보고, 어드레스된 객체를 요청된 대로 생성하고 업데이트하며 삭제한다(3). 이동은 하나의 객체 제공자 내에서의 생성 및 다른 객체 제공자 내에서의 삭제를 시사하는 것임을 유의한다. 이 객체는 그것의 상태 및 특징들(생성됨, 업데이트됨, 삭제됨)을 발행한다(4). P/S 도메인 상에 이동을 위한 2개의 통지, 즉 삭제됨 및 생성됨이 있음을 유의한다. 최종적으로, 요청의 결과가 객체 관리자 사용자 인터페이스 또는 애플리케이션 상에 도시된다(5).
- [0071] 이하에서, 발명은 스틱 노트 애플리케이션의 예시를 사용하여 추가로 설명될 것이다. 스틱 노트들은 가상의 벽들(walls)에 부착된다. 스틱 노트 애플리케이션은 그것의 환경 내에서 존재하는 임의의 벽으로부터 노트들을 보여주고 조작(생성, 업데이트, 삭제, 이동)할 수 있다. 추가로, 스틱 노트 애플리케이션 그 자체는 그것 소유의 가상의 벽을 제공할 수 있다.
- [0072] 이전에 이미 나타내어진 것과 같이, 2개의 역할이 스틱 노트 애플리케이션의 상호작용에서 식별된다. 벽 제공자는 가상의 벽으로서 역할하고, 그것의 벽에 부착된 노트들을 소유할 것이며, 다시 말하면 발행에 대한 책임이 있을 것이다. 노트 사용자 인터페이스는 노트들을 사용자에게 보여주고, 그에게 노트들을 조작하기 위한 제어들을 제시한다. 스틱 노트 애플리케이션은 동시에 양자의 역할을 수행할 수 있다. 일부 헤드리스(headless) 디바이스, 예컨대 게이트웨이가 벽 제공자로서만 역할하는 애플리케이션을 구현할 수 있다는 점이 고려될 수 있다.
- [0073] 스틱 노트 애플리케이션의 데이터 모델은 3개의 주요 부분, 즉 노트들이 부착될 수 있는 가상의 벽들의 표시, 노트들 자체의 표시, 및 노트 조작 요청들의 표시를 포함한다. 이들 노트 조작 요청들은 노트 생성, 노트의 텍스트 수정, 노트의 다른 벽으로의 이동, 노트의 삭제를 포함한다.
- [0074] (위에서 설명된 OBJECT\_REQUEST에 대응하는) NoteRequest의 필드들에 대한 값들의 모든 가능한 조합은 의미를 가진다. 모든 요청은 주어진 벽 제공자와 무관하거나, 모든 요청은 생성, 삭제 또는 업데이트 요청으로서 해석 가능하다.
- [0075] 노트 사용자 인터페이스의 관점에서, 해석은 동등하게 간단하다. 새로운 노트를 생성하기 위하여, (OBJECT\_ID에 대응하는) 새로운 고유 노트 ID가 생성되고, (OBJECT\_PROVIDER\_ID에 대응하는) 그것의 정확한 벽 ID를 가진 벽이 선택되며, 요청이 발행된다. 기존의 노트를 업데이트하기 위하여, 요청은 대응하는 노트 ID, 현재 벽 ID, 및 새로운 메시지와 함께 발행된다. 노트를 이동하기 위하여, 요청은 노트 ID, 이동하기를 원하는 벽의 벽 ID, 및 기존의 메시지와 함께 발행된다. 실질적으로, 이것은 하나의 요청에 결합된 새로운 벽에 대한 생성 액션 및 기존의 벽에 대한 삭제 액션이다. 노트를 삭제하기 위하여, 요청은 노트 ID 및 보이드 벽 ID와 함께 발행된다. ('특징들'에 대응하는) 메시지는 이 경우에는 무관하다.
- [0076] 도 9는 CREATE 액션을 나타낸다:
- [0077] 1. 사용자는 ID 1234를 가진 벽에 부착될 메시지 "Hello"를 가진 새로운 노트를 생성한다. 노트 사용자 인터페이스는 (이 경우에는 1인) 노트에 대한 고유 ID를 선택하고, NoteRequest를 발행한다.
- [0078] 2. NoteRequest 이벤트의 수신 시에, 벽 제공자 1234는 그것을 노트의 생성을 위한 요청으로서 해석한다. 벽 제공자 3456은 요청이 자신의 벽에 대해 무관하기 때문에 요청을 무시한다.
- [0079] 3. 벽 제공자 1234는 다음으로 새로운 노트 인스턴스를 노트 토픽에 발행한다.
- [0080] 4. 노트 사용자 인터페이스는 토픽에 대한 변화들을 통지받고, 새로운 노트를 발견하며, 그것의 사용자 인터페이스



이스를 업데이트한다.

[0081] 도 10은 UPDATE 액션을 나타낸다:

[0082] 1. 사용자는 노트의 메시지를 "Hello"에서 "World"로 수정할 것을 결정한다. 노트 사용자 인터페이스는 적절한 NoteRequest를 발행한다.

[0083] 2. NoteRequest 이벤트의 수신 시에, 벽 제공자 1234는 그것을 기존의 노트에 대한 업데이트 요청으로서 해석한다. 벽 제공자 3456은 요청이 자신의 벽에 대해 무관하기 때문에 요청을 무시한다.

[0084] 3. 벽 제공자 1234는 업데이트된 노트 인스턴스를 노트 토픽에 발행한다.

[0085] 4. 노트 사용자 인터페이스는 토픽에 대한 변화들을 통지받고, 업데이트된 노트를 발견하며, 그것의 사용자 인터페이스를 업데이트한다.

[0086] 도 11은 MOVE 액션을 나타낸다:

[0087] 1. 사용자는 노트를 벽 3456으로 이동시킨다. 노트 사용자 인터페이스는 적절한 NoteRequest를 발행한다.

[0088] 2. NoteRequest 이벤트의 수신 시에, 벽 제공자 1234는 그것을 기존의 노트에 대한 삭제 요청으로 해석한다. 벽 제공자 3456은 동일한 이벤트를 새로운 노트에 대한 생성 요청으로서 해석한다.

[0089] 3. 벽 제공자 1234는 그것의 노트 인스턴스를 토픽으로부터 삭제하고, 벽 제공자 3456은 새로운 노트 인스턴스를 토픽에 발행한다.

[0090] 4. 노트 사용자 인터페이스는 토픽에 대한 변화들을 통지받고, 업데이트된 노트를 발견하며, 그것의 사용자 인터페이스를 업데이트한다.

[0091] 마지막으로, 도 12는 DELETE 액션을 나타낸다:

[0092] 1. 사용자는 노트를 삭제한다. 노트 사용자 인터페이스는 적절한 NoteRequest를 발행한다: 그것은 보이드 벽 ID(0)를 선택한다.

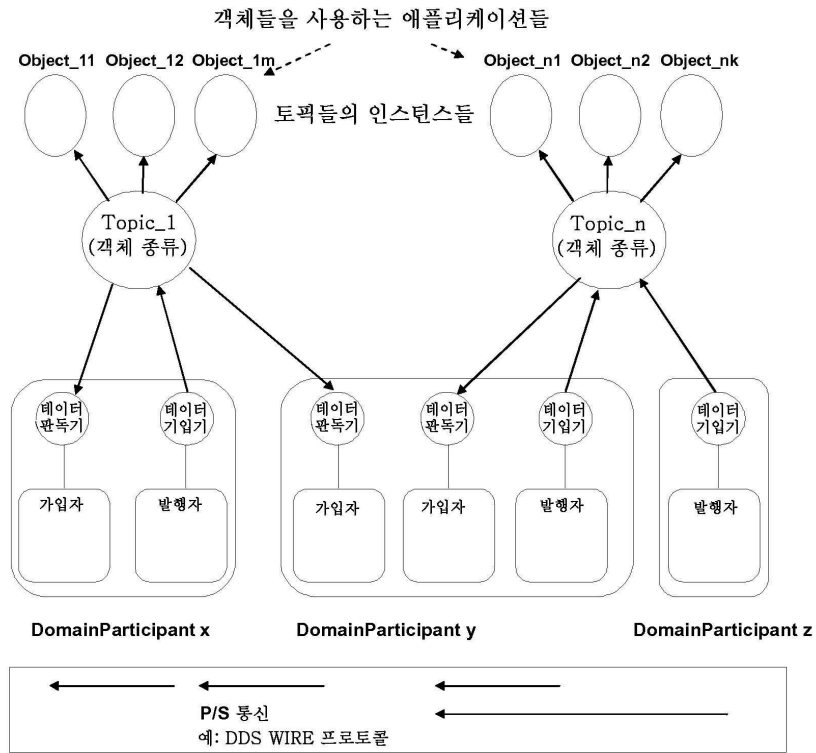
[0093] 2. NoteRequest 이벤트의 수신 시에, 벽 제공자 1234는 요청이 자신의 벽에 대하여 무관하기 때문에 요청을 무시한다. 벽 제공자 3456은 요청을 삭제 요청으로서 해석한다.

[0094] 3. 벽 제공자 3456은 그것의 노트를 토픽으로부터 삭제한다.

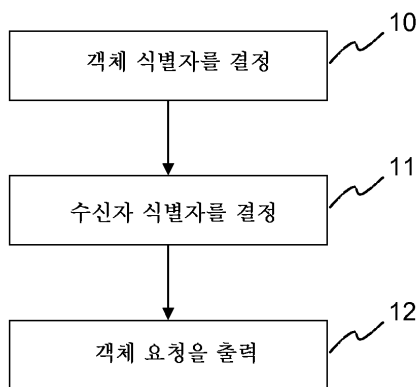
[0095] 4. 노트 사용자 인터페이스는 토픽에 대한 변화들을 통지받고, 노트 인스턴스가 토픽으로부터 삭제된 것을 발견하며, 그것의 사용자 인터페이스를 업데이트한다.

도면

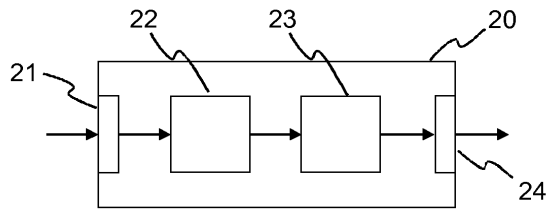
도면1



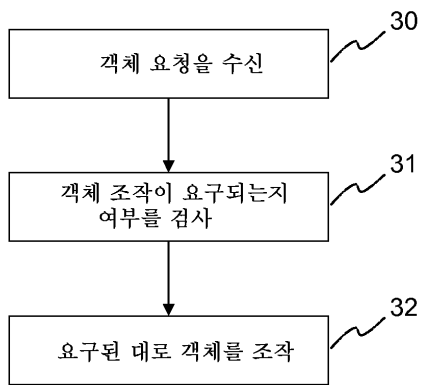
도면2



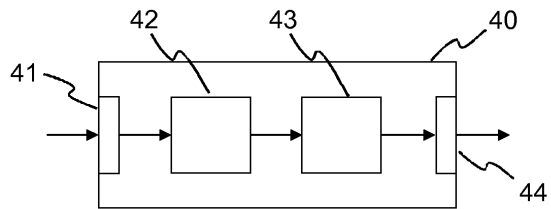
도면3



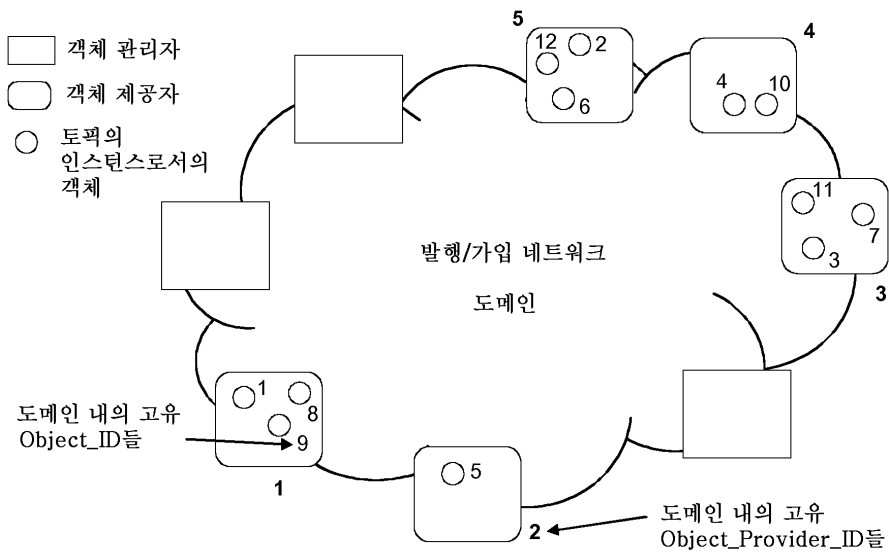
도면4



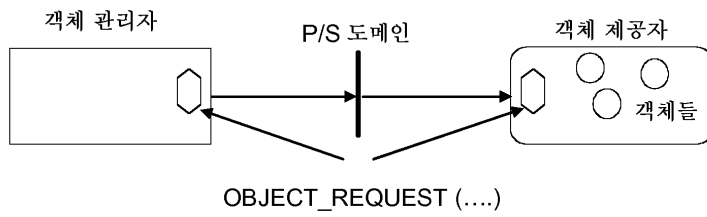
도면5



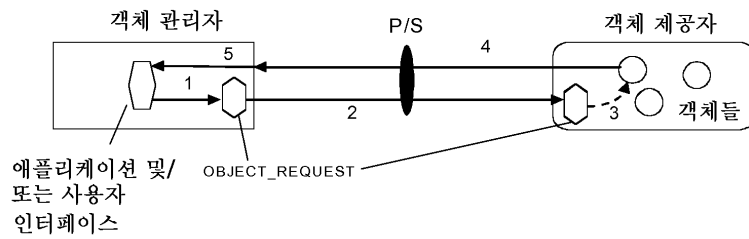
도면6



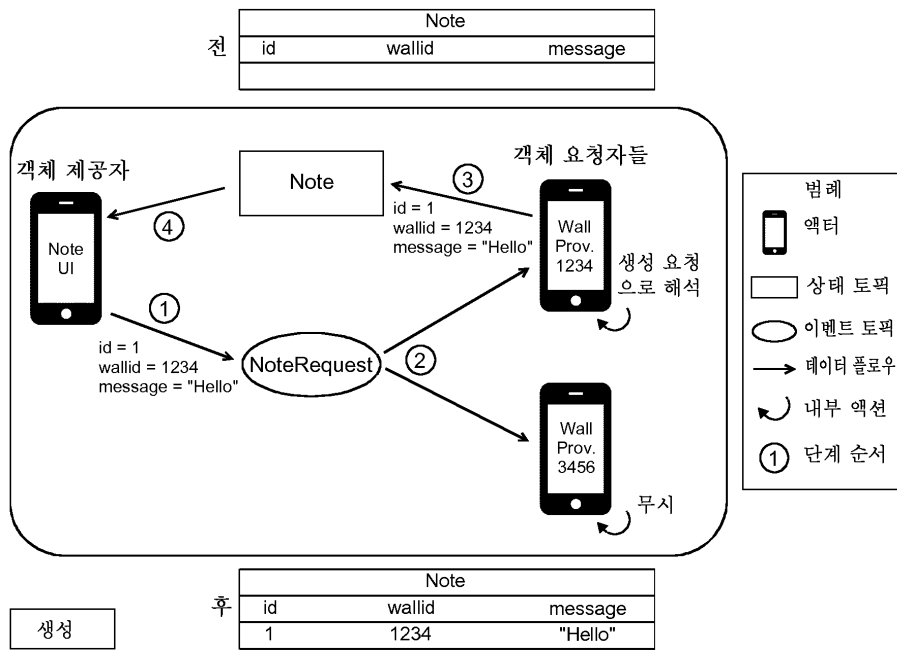
도면7



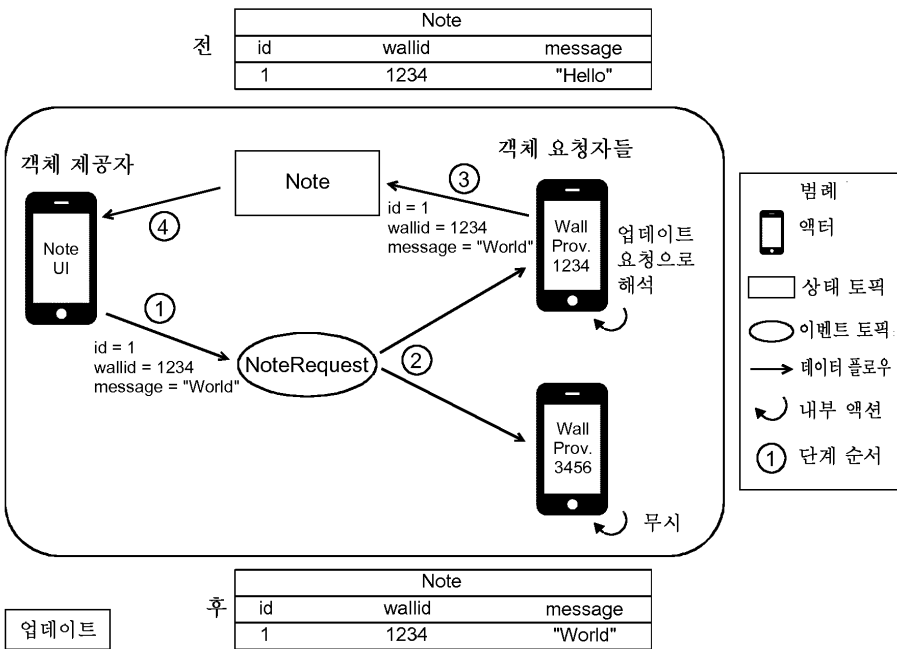
도면8



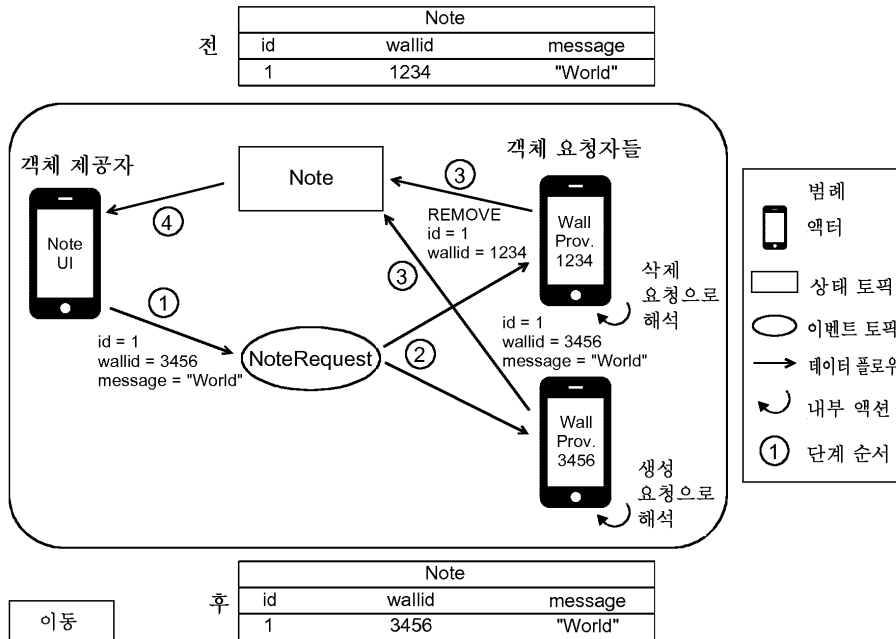
도면9



도면10



도면11



도면12

