



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0023882  
(43) 공개일자 2017년03월06일

(51) 국제특허분류(Int. Cl.)  
G06F 9/44 (2006.01) G06F 17/30 (2006.01)

(71) 출원인  
마이크로소프트 테크놀로지 라이센싱, 엘엘씨  
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원  
마이크로소프트 웨이

(52) CPC특허분류  
G06F 9/4443 (2013.01)  
G06F 17/30991 (2013.01)  
(21) 출원번호 10-2016-7036901

(72) 발명자  
**미탈 비자이**  
미국 워싱턴주 98052-6399 레드몬드 원 마이크로  
소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠  
(8/1172) 마이크로소프트 테크놀로지 라이센싱,  
엘엘씨

(21) 출원번호 10-2016-7036901

(22) 출원일자(국제) 2015년06월29일

심사청구일자      없음

(85) 번역문제출일자 2016

(86) 국제출원번호 PCT/US2015/038

(87) 국제공개번호 WO 2016/003848

국제공개일자 2016년01월07일

### (30) 우선권주장

14/320 083 2014년06월30일

14/320,000

14/320,000 2014년00월30일 미국(03)

**루빈 대릴**  
미국 워싱턴주 98052-6399 레드몬드 원 마이크로  
소프트 웨이 엘씨에이 - 인터내셔널 페이던츠  
(8/1172) 마이크로소프트 테크놀로지 라이센싱,  
엘엘씨  
**비엣트 비그라**

마파드 마크업  
미국 워싱턴주 98052-6399 레드몬드 원 마이크로  
소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠  
(8/1172) 마이크로소프트 테크놀로지 라이센싱,  
엘엘씨

(74) 대리인  
제일특허법인

전체 청구항 수 : 총 15 항

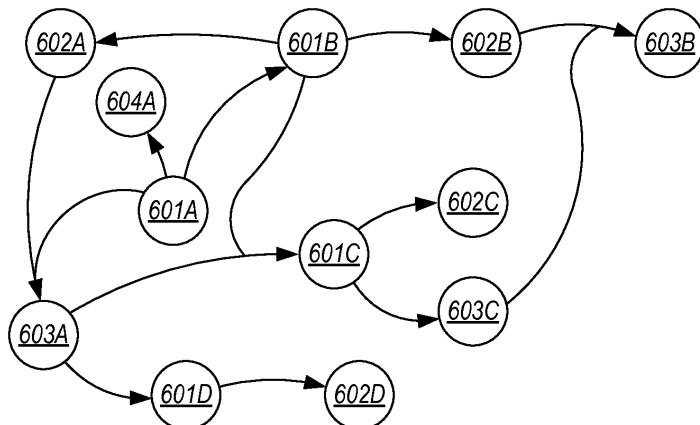
#### (54) 발명의 명칭 복합 장치 환경에서의 출력의 영화화 기법

(57) 요약

정보 사용자 인터페이스 요소와 애플리케이션 사용자 인터페이스 요소를 연속 스트림 형태로 합성하는 것이 제공된다. 이는, 정보의 세션이 제시되면, 해당 세션에 적용될 영화성(cinematicity)의 레벨을 조정함으로써 행해진다. 정보를 나타내는 사용자 인터페이스 요소를 제시함에 있어서 영화성의 레벨이 높을수록 보다 높은 정도의 움직임이 이용된다. 그에 반해서, 영화성의 레벨이 낮을수록 보다 낮은 정도의 움직임이 이용되거나, 또는, 정보를 나타내는 사용자 인터페이스 요소의 움직임이 없다.

대표도 - 도9

900



(52) CPC특허분류  
**G06T 11/40** (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

정보 사용자 인터페이스 요소와 애플리케이션 사용자 인터페이스 요소를 연속 스트림 형태로 합성하는 방법으로서,

정보의 세션이 제시되면, 해당 세션에 적용될 영화성(cinematicity)의 레벨을 조정하는 동작을 포함하고,

상기 영화성의 레벨이 높으면, 상기 정보를 나타내는 사용자 인터페이스 요소의 움직임을 제시할 때 높은 정도의 움직임을 이용하고,

상기 영화성의 레벨이 낮으면, 낮은 정도의 움직임을 이용하거나 또는 상기 정보를 나타내는 사용자 인터페이스 요소의 움직임이 없는

방법.

#### 청구항 2

제 1 항에 있어서,

상기 영화성의 레벨이 높으면, 상기 정보에 관련된 토픽의 합성된 배경을 이용하는

방법.

#### 청구항 3

제 1 항에 있어서,

상기 영화성의 레벨이 높으면, 상기 영화성의 레벨이 낮을 때보다 많은 점프 컷(jump cuts)을 이용하는

방법.

#### 청구항 4

제 1 항에 있어서,

상기 영화성의 레벨이 높다는 것은, 상기 사용자 인터페이스 내에 다수의 초점 레벨이 있는 것에 대응하고,

상기 영화성의 레벨이 낮다는 것은, 상기 사용자 인터페이스 내에 단일의 초점 레벨이 있는 것에 대응하는

방법.

#### 청구항 5

제 1 항에 있어서,

상기 영화성의 레벨이 높으면, 배경 음악을 제공하는 반면, 상기 영화성의 레벨이 낮으면, 배경 음악이 적거나 또는 전혀 없는

방법.

## 청구항 6

제 1 항에 있어서,

상기 영화성의 레벨이 높으면, 영화적인 조명을 더 제공하는 반면, 상기 영화성의 레벨이 낮으면, 영화적 조명을 덜 제공하거나 또는 전혀 제공하지 않는

방법.

## 청구항 7

제 1 항에 있어서,

상기 영화성의 레벨이 높은 경우에,

상기 방법은,

텍스트에서 아이템의 시각적 표현을 찾는 동작과,

시각화(visualization)를 디스플레이하는 동작

을 더 포함하는

방법.

## 청구항 8

제 1 항에 있어서,

상기 영화성의 레벨이 높은 경우에, 빨리감기 및 되감기가 제공되는

방법.

## 청구항 9

제 1 항에 있어서,

상기 방법은, 동일한 정보 열(train)에 대해 적어도 제 1 장치 및 제 2 장치에서 수행되고,

상기 정보 열에 적용되는 상기 영화성의 레벨은 상기 제 1 및 제 2 장치에 따라 달라지는

방법.

## 청구항 10

컴퓨팅 시스템의 하나 이상의 프로세서에 의해 실행될 때, 상기 컴퓨팅 시스템으로 하여금, 정보 사용자 인터페이스 요소와 애플리케이션 사용자 인터페이스 요소를 연속 스트림 형태로 합성하는 방법을 수행하게 하도록 구성되는 컴퓨터 실행가능 명령어를 구비하는 하나 이상의 컴퓨터 관독가능 저장 매체를 포함하는 컴퓨터 프로그램 제품으로서,

상기 방법은,

정보의 세션이 제시되면, 해당 세션에 적용될 영화성의 레벨을 조정하는 동작을 포함하고,

상기 영화성의 레벨이 높으면, 상기 정보를 나타내는 사용자 인터페이스 요소의 움직임을 제시할 때, 높은 정도의 움직임을 이용하고,

상기 영화성의 레벨이 낮으면, 낮은 정도의 움직임을 이용하거나 또는 상기 정보를 나타내는 사용자 인터페이스

요소의 움직임이 없는

컴퓨터 프로그램 제품.

### 청구항 11

제 10 항에 있어서,

상기 영화성의 레벨이 높으면, 상기 정보에 관련된 토픽의 합성된 배경을 이용하는

컴퓨터 프로그램 제품.

### 청구항 12

제 10 항에 있어서,

상기 영화성의 레벨이 높으면, 배경 음악을 제공하는 반면, 상기 영화성의 레벨이 낮으면, 배경 음악이 적거나 또는 없는

컴퓨터 프로그램 제품.

### 청구항 13

제 10 항에 있어서,

상기 영화성의 레벨이 높은 경우에, 빨리감기 및 되감기가 제공되는

컴퓨터 프로그램 제품.

### 청구항 14

제 10 항에 있어서,

상기 방법은, 동일한 정보 열에 대해 적어도 제 1 장치 및 제 2 장치에서 수행되고,

상기 정보 열에 적용된 상기 영화성의 레벨은 상기 제 1 및 제 2 장치에 따라 달라지는

컴퓨터 프로그램 제품.

### 청구항 15

시스템으로서,

하나 이상의 프로세서와,

상기 하나 이상의 프로세서에 의해 실행될 때, 컴퓨팅 시스템으로 하여금, 정보 사용자 인터페이스 요소와 애플리케이션 사용자 인터페이스 요소를 연속 스트림 형태로 합성하는 방법을 수행하게 하도록 구성되는 컴퓨터 실행가능 명령어를 구비하는 하나 이상의 컴퓨터 판독가능 저장 매체

를 포함하고,

상기 방법은,

정보의 세션이 제시되면, 해당 세션에 적용될 영화성의 레벨을 조정하는 동작을 포함하고,

상기 영화성의 레벨이 높으면, 상기 정보를 나타내는 사용자 인터페이스 요소의 움직임을 제시할 때, 높은 정도의 움직임을 이용하고,

상기 영화성의 레벨이 낮으면, 낮은 정도의 움직임을 이용하거나 또는 상기 정보를 나타내는 사용자 인터페이스 요소의 움직임이 없는 시스템.

### 발명의 설명

#### 배경 기술

[0001]

컴퓨팅 기술은 일, 놀이, 소통의 방식을 혁신시키고 있다. 컴퓨팅 기술이 진전함에 따라, 컴퓨팅 시스템을 구현하거나 또는 컴퓨팅 시스템으로부터의 콘텐츠를 디스플레이하는 장치는 다양해졌다. 예를 들어, 컴퓨팅 시스템은 서버 랙, 데스크탑 컴퓨터, 랩톱 컴퓨터, 태블릿, 스마트폰, 게임용 콘솔, 시계, 냉장고, 스마트 하우스 등의 형태를 취할 수 있다.

[0002]

컴퓨팅 시스템이 다양해지면서, 컴퓨터에 대하여 컴퓨팅 출력 및 입력 정보를 렌더링하는데 사용될 수 있는 장치의 유형도 마찬가지로 다양해지고 있다. 예를 들어, 출력 장치는 프로젝터, 텔레비전 모니터, 3차원 디스플레이, 랩톱, 태블릿 컴퓨터, 전화기 등과 같은 디스플레이를 포함할 수 있다. 출력 장치는 스피커와 같은 사운드 출력부를 포함할 수 있다. 출력 장치는 또한, 액추에이터, 발광부, 멜브 등을 포함할 수 있다. 입력 장치는 키보드, 포인터 장치(예컨대, 마우스), 터치스크린, 마이크, 비디오 카메라, 스틸 카메라, 3차원 위치 검출기, GPS 모니터, 광 센서, 가속도계, 온도계, 나침반 등을 포함할 수 있다.

[0003]

컴퓨팅 시스템 및 연관된 입력 및 출력 장치는 매우 많아지고 있으며, 또한 종종 모바일로 되고 있다. 보통, 어떤 장소에서든지, 다수의 다른 다양한 장치가 존재할 수 있다. 예를 들어, 누구든지 참석하게 되는 일반적인 회의실에서는, 오버헤드 프로젝터, 텔레비전 스크린, 랩톱, 태블릿, 스마트폰, 마이크, 카메라, 조명 등이 있을 수 있다. 각각의 장치가 자체 애플리케이션을 실행하거나 또는 단일의 애플리케이션으로부터 콘텐츠를 디스플레이하는 것은 종래의 패러다임이다. 애플리케이션들의 상호작용시에는, 이들은 보통 애플리케이션 프로그램 인터페이스를 통해 상호작용하는 별도의 애플리케이션으로서 상호작용한다.

[0004]

본 명세서에서 청구항의 청구대상은, 어떤 단점을 해결하는 실시예 또는 상술한 바와 같은 환경에서만 동작하는 실시예로 한정되는 것은 아니다. 오히려, 이러한 배경 기술은 본 명세서에 기술된 몇몇 실시예가 실행될 수 있는 예시적인 일 기술 분야를 설명하기 위해 제공될 뿐이다.

#### 발명의 내용

[0005]

본 명세서에 기술된 적어도 일부 실시예는, 정보 사용자 인터페이스 요소와 애플리케이션 사용자 인터페이스 요소를 연속 스트림 형태로 합성하는 기법에 관한 것이다. 이는, 정보의 세션이 제시되면, 해당 세션에 적용될 영화성(cinematicity)의 레벨을 조정함으로써 행해진다. 정보를 나타내는 사용자 인터페이스 요소를 제시함에 있어서 영화성의 레벨이 높을수록 보다 높은 정도의 움직임이 이용된다. 그에 반해서, 영화성의 레벨이 낮을수록 보다 낮은 정도의 움직임이 이용되거나, 또는, 정보를 나타내는 사용자 인터페이스 요소의 움직임이 없다.

[0006]

본 개요는 청구항의 청구대상의 핵심적인 특징이나 필수적인 특징들을 나타내는 것이 아니며, 청구항의 청구대상의 범위를 제한하는 데 이용되는 것도 아니다.

#### 도면의 간단한 설명

[0007]

상술한 장점 및 특징과, 다른 장점 및 특징을 얻을 수 있는 방식을 기술하기 위해, 첨부 도면들을 참조하여 다양한 실시예를 더욱 상세하게 설명한다. 이들 도면은 예시적인 실시예만을 나타내는 것으로, 발명의 범위를 한정하는 것이 아니며, 이하 첨부 도면을 이용해서, 추가적인 특징 및 세부사항을 갖고 실시예를 기술 및 설명한다.

도 1은 본 명세서에서 기술된 몇몇 실시예가 사용될 수 있는 컴퓨팅 시스템을 개략적으로 도시하는 도면,

도 2는 단일 데이터 소스와 단일 데이터 타깃을 연결하는 단일 링크가 존재하고, 해당 링크에 의해 나타내진 변환이 데이터 타깃에서의 값을 생성하기 위해 데이터 소스에서의 값을 입력으로서 이용하여 자동으로 수행되는, 단순한 변환 체인을 개략적으로 도시하는 도면,

도 3은 2개의 데이터 타깃에서의 출력 값을 생성하기 위해 3개의 데이터 소스로부터의 입력 값을 이용하여 변환이 수행되는 다른 단순한 예시적인 변환 체인을 개략적으로 도시하는 도면,

도 4는 도 2의 변환 체인 및 도 3의 변환 체인을 조합한 형태의 변환 체인을 도시하는 도면,

도 5는 본 명세서에서 기술된 원리가 작동할 수 있고, 복합 애플리케이션의 구성 변환 체인들과 연계되는 다수의 장치를 포함하는 한편, 입력 장치 및 출력 장치도 포함하는 예시적인 환경을 도시하는 도면,

도 6a 내지 도 6d는 도 5에서의 각각의 장치와 연계될 수 있는 예시적인 변환 체인을 각각 도시하는 도면(다른 변환 체인과 연결되지 않아서 데이터가 흘러가지 않게 되는 화살표는 "X"로 표시되고, 변환 체인 자체의 노드가 아닌 종속 요소는 점선 테두리로 표시됨),

도 7a는, 도 6a 및 도 6b의 변환 체인들의 연결을 나타내는 증대된 변환 체인을 도시하는 도면,

도 7b는, 도 6a 및 도 6c의 변환 체인들의 연결을 나타내는 증대된 변환 체인을 도시하는 도면,

도 7c는, 도 6b 및 도 6c의 변환 체인들의 연결을 나타내는 증대된 변환 체인을 도시하는 도면,

도 7d는, 도 6a 및 도 6d의 변환 체인들의 연결을 나타내는 증대된 변환 체인을 도시하는 도면,

도 8a는, 도 6a, 도 6b 및 도 6c의 변환 체인들의 연결을 나타내는 증대된 변환 체인을 도시하는 도면,

도 8b는, 도 6a, 도 6b 및 도 6d의 변환 체인들의 연결을 나타내는 증대된 변환 체인을 도시하는 도면,

도 8c는, 도 6a, 도 6c 및 도 6d의 변환 체인들의 연결을 나타내는 증대된 변환 체인을 도시하는 도면,

도 9는, 도 6a, 도 6b, 도 6c 및 도 6d의 변환 체인들의 연결을 나타내는 증대된 변환 체인을 도시하는 도면,

도 10은 복합 애플리케이션을 촉진하기 위한 준비 방법의 흐름도를 도시하는 도면,

도 11은, 도 10, 도 12, 도 13 및 도 14의 방법을 수행하는데 이용될 수 있으며, 변환 체인 클래스 정의들로 이루어진 라이브러리 및 장치 레지스트레이션 뿐만 아니라 프리젠테이션 서비스를 포함하는 시스템을 개략적으로 도시하는 도면,

도 12는 변환 체인 클래스의 2개의 인스턴스를 연결하는 방법의 흐름도를 도시하는 도면,

도 13은 대형 변환 체인으로부터 변환 체인을 분리하는 방법의 흐름도를 도시하는 도면,

도 14는 변환 체인의 하나 이상의 프리젠테이션 노드에서의 변경을 적절한 장치 상으로 렌더링하는 방법의 흐름도를 도시하는 도면,

도 15a 내지 도 15j는 본 명세서에서 사용자가 다수의 장치를 주문하게 되는, "장치 상황(device scenario)"이라고 하는 상황을 경험할 수 있는 다양한 사용자 인터페이스를 도시하는 도면,

도 15k 내지 도 15o는, 도 15a 내지 도 15j의 장치 상황에서 이용된 복합 애플리케이션의 변경시에 보여질 수 있는 다양한 사용자 인터페이스를 도시하는 도면,

도 16은 정보 사용자 인터페이스 요소와 애플리케이션 사용자 인터페이스 요소를 연속 스트림 형태로 합성하는 방법의 흐름도를 도시한다.

### **발명을 실시하기 위한 구체적인 내용**

[0008] 본 명세서에 기술된 적어도 일부 실시예는 정보 사용자 인터페이스 요소와 애플리케이션 사용자 인터페이스 요소를 연속 스트림 형태로 합성하는 기법에 관한 것이다. 이는, 정보의 세션이 제시되면, 해당 세션에 적용될 영화성의 레벨을 조정함으로써 행해진다. 정보를 나타내는 사용자 인터페이스 요소를 제시함에 있어서 영화성의 레벨이 높을수록 보다 높은 정도의 움직임이 이용된다. 그에 반해서, 영화성의 레벨이 낮을수록 보다 낮은 정도의 움직임이 이용되거나, 또는, 정보를 나타내는 사용자 인터페이스 요소의 움직임이 없다.

[0009] 컴퓨팅 시스템에 관한 설명은 먼저 도 1에 대하여 기술될 것이다. 이후, 나머지 도면에 대하여 복합 장치 애플리케이션 기술이 기술될 것이다.

[0010] 컴퓨팅 시스템은 현재, 점점 더 매우 다양한 형태를 취하고 있다. 컴퓨팅 시스템은, 예컨대, 핸드헬드 장치, 어플라이언스, 랙탑 컴퓨터, 데스크탑 컴퓨터, 메인프레임, 분산형 컴퓨팅 시스템, 또는 심지어 종래에는 컴퓨팅 시스템으로 간주되지 않았던 장치일 수 있다. 이 상세한 설명 및 청구범위에 있어서, "컴퓨팅 시스템"이라는 용어는 적어도 하나의 물리적 및 유형(tangible)의 프로세서, 및 프로세서에 의해 실행될 수 있는 컴퓨터 실행가능 명령어를 구비할 수 있는 물리적 및 유형의 메모리를 포함하는 임의의 장치 또는 시스템(또는 그 조합)

을 포함하는 것으로서 폭넓게 규정된다. 메모리는 임의의 형태를 취할 수 있으며, 컴퓨팅 시스템의 성질 및 형태에 의존할 수 있다. 컴퓨팅 시스템은 네트워크 환경에 걸쳐 분산될 수 있으며, 다수의 구성 컴퓨팅 시스템을 포함할 수 있다.

[0011] 도 1에 도시된 바와 같이, 그 가장 기본적인 구성에 있어서, 컴퓨팅 시스템(100)은 통상적으로 적어도 하나의 하드웨어 처리 유닛(102) 및 메모리(104)를 포함한다. 메모리(104)는 휘발성 메모리, 비휘발성 메모리, 또는 두 메모리의 몇 가지 조합일 수 있는 물리 시스템 메모리일 수 있다. "메모리"라는 용어는, 본 명세서에서는 물리적 저장 매체와 같은 비휘발성 대용량 저장 장치를 의미하는데 사용될 수도 있다. 컴퓨팅 시스템이 분산형이면, 처리, 메모리 및/또는 저장 성능도 마찬가지로 분산될 수 있다. 본 명세서에서 사용되는 바와 같은, "실행가능 모듈" 또는 "실행가능 컴포넌트"라는 용어는 컴퓨팅 시스템 상에서 실행될 수 있는 소프트웨어 객체, 라우팅, 또는 방법을 의미할 수 있다. 본 명세서에 기술된 서로 다른 컴포넌트, 모듈, 엔진, 및 서비스는 컴퓨팅 시스템 상에서 실행하는 객체 또는 프로세스(예컨대, 별도의 스레드(thread))로서 구현될 수 있다.

[0012] 후술하는 설명에 있어서, 실시예는 하나 이상의 컴퓨팅 장치에 의해 수행되는 동작들을 참조하여 기술된다. 상기와 같은 동작들이 소프트웨어로 구현되면, 동작을 수행하는 연관 컴퓨팅 시스템의 하나 이상의 프로세서는 컴퓨터 실행가능 명령어를 실행한 것에 응답하여 컴퓨팅 시스템의 작업을 관리한다. 예컨대, 상기와 같은 컴퓨터 실행가능 명령어는 컴퓨터 프로그램 제품을 형성하는 하나 이상의 컴퓨터 판독가능 매체 상에서 구체화될 수 있다. 상기와 같은 작업의 예시는 데이터의 조작을 수반한다. 컴퓨터 실행가능 명령어(및 조작된 데이터)는 컴퓨팅 시스템(100)의 메모리(104)에 저장될 수 있다. 컴퓨팅 시스템(100)은 또한, 컴퓨팅 시스템(100)이 예컨대, 네트워크(110)를 통해 다른 메시지 프로세서와 통신할 수 있게 하는 통신 채널(108)을 포함할 수 있다.

[0013] 컴퓨팅 시스템(100)은 또한, 잠재적으로 디스플레이, 스피커, 발광부, 액추에이터 등과 같은 출력 랜더링 컴포넌트를 포함할 수 있다. 컴퓨팅 시스템(100)은 또한, 키보드, 포인터 장치(예컨대, 마우스 또는 트랙 패드), 음성 인식 장치, 및 가능하게는 물리적 센서(예컨대, 온도계, GPS, 광 검출기, 나침반, 가속도계 등)와 같은 입력 컴포넌트를 포함할 수 있다.

[0014] 본 명세서에 기술된 실시예는, 아래에서 더 상세하게 논의되는 바와 같이, 예컨대 하나 이상의 프로세서 및 시스템 메모리와 같은 컴퓨터 하드웨어를 포함하는 특수 목적 컴퓨터 또는 범용 컴퓨터를 포함하거나 또는 이용할 수 있다. 본 명세서에 기술된 실시예는 또한, 컴퓨터 실행가능 명령어 및/또는 데이터 구조를 반송 또는 저장하기 위한 물리적 및 그 밖의 컴퓨터 판독가능 매체를 포함한다. 상기와 같은 컴퓨터 판독가능 매체는 범용 또는 특수 목적 컴퓨터 시스템에 의해 액세스될 수 있는 시판중인 임의의 매체일 수 있다. 컴퓨터 실행가능 명령어를 저장하는 컴퓨터 판독가능 매체는 물리적 저장 매체이다. 컴퓨터 실행가능 명령어를 반송하는 컴퓨터 판독가능 매체는 전송 매체이다. 따라서, 한정이 아닌 예시로서, 본 발명의 실시예는 적어도 2가지의 명백하게 다른 종류의 컴퓨터 판독가능 매체, 즉 컴퓨터 저장 매체 및 전송 매체를 포함할 수 있다.

[0015] 컴퓨터 저장 매체는, 원하는 프로그램 코드 수단을 컴퓨터 실행가능 명령어 또는 데이터 구조의 형태로 저장하는데 사용될 수 있으며 범용 또는 특수 목적 컴퓨터에 의해 액세스될 수 있는, RAM, ROM, EEPROM, CD-ROM 또는 그 밖의 광 디스크 저장부, 자기 디스크 저장부 또는 그 밖의 자기 저장 장치, 또는 임의의 다른 저장 매체를 포함한다.

[0016] "네트워크"는 컴퓨터 시스템들 및/또는 모듈들 및/또는 그 밖의 전자 장치들 사이에서 전자 데이터의 전송을 가능하게 하는 하나 이상의 데이터 링크로서 규정된다. 정보가 네트워크 또는 다른 통신 접속부(유선, 무선, 또는 유선 또는 무선의 조합)를 거쳐 컴퓨터에 전송 또는 제공될 때, 컴퓨터는 해당 접속부를 실질적으로 전송 매체로서 간주한다. 전송 매체는, 원하는 프로그램 코드 수단을 컴퓨터 실행가능 명령어 또는 데이터 구조의 형태로 반송하는데 이용될 수 있으며 범용 또는 특수 목적 컴퓨터에 의해 액세스될 수 있는 네트워크 및/또는 데이터 링크를 포함할 수 있다. 상기의 조합이 또한 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.

[0017] 또한, 다양한 컴퓨터 시스템 컴포넌트에 있어서, 컴퓨터 실행가능 명령어 또는 데이터 구조 형태의 프로그램 코드 수단은 전송 매체로부터 컴퓨터 저장 매체로(또는 반대로) 자동으로 전송될 수 있다. 예컨대, 네트워크 또는 데이터 링크를 통해 수신된 컴퓨터 실행가능 명령어 또는 데이터 구조는 네트워크 인터페이스 모듈(예컨대, "NIC") 내의 RAM에서 버퍼될 수 있으며, 이후 결국에는 컴퓨터 시스템 RAM 및/또는 컴퓨터 시스템의 딜 휘발성 인 컴퓨터 저장 매체에 전송될 수 있다. 따라서, 컴퓨터 저장 매체는 전송 매체를 또한(또는 심지어 주로) 이용하는 컴퓨터 시스템 컴포넌트에 포함될 수 있다는 점을 이해해야 한다.

[0018] 컴퓨터 실행가능 명령어는 예컨대, 프로세서에서의 실행시에, 범용 컴퓨터, 특수 목적 컴퓨터, 또는 특수 목적

처리 장치로 하여금 특정 기능을 또는 기능들의 그룹을 수행하게 하는 명령어 및 데이터를 포함한다. 컴퓨터 실행가능 명령어는 예컨대, 이진수일 수 있거나, 또는 심지어, 어셈블리 언어와 같은 중간 포맷 명령어와 같이, 프로세서에 의한 직접 실행 전에 약간의 번역(예컨대, 편집)을 겪는 명령어, 또는 심지어 소스 코드일 수 있다. 청구대상이 구조적 특징 및/또는 방법론적 동작에 특정된 표현으로 기술되어 있지만, 청구범위의 청구항들에 규정된 청구대상은 반드시 상술한 특징 또는 동작에 한정되는 것은 아님을 이해해야 한다. 오히려, 상술된 특징 및 동작은 청구항들을 구현하는 예시적인 형태로 개시된다.

[0019] 당업자라면, 본 발명이, 퍼스널 컴퓨터, 데스크탑 컴퓨터, 랩탑 컴퓨터, 메시지 프로세서, 핸드헬드 장치, 멀티 프로세서 시스템, 마이크로프로세서 기반 또는 프로그램 가능 가전제품, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 모바일 전화기, PDA, 페이저, 라우터, 스위치 등을 포함하는 많은 유형의 컴퓨터 시스템 구성을 갖춘 네트워크 컴퓨팅 환경에서 실행될 수 있음을 인식할 것이다. 본 발명은 또한, 네트워크를 통해 연결되는(유선 데이터 링크, 무선 데이터 링크, 또는 유선 및 무선 데이터 링크의 조합에 의함) 로컬 및 원격 컴퓨터 시스템 양자가 모두 태스크를 수행하는 분산형 시스템 환경에서 실행될 수 있다. 분산형 시스템 환경에 있어서, 프로그램 모듈은 로컬 및 원격 메모리 저장 장치 양자 모두에 위치될 수 있다.

[0020] 본 명세서에서 기술된 원리는 변환 체인을 이용해서 작동한다. 변환 체인은, 각각이 데이터 소스 또는 데이터 타깃을 나타낼 수 있는 상호접속된 노드들의 세트이다. 노드들 사이에는 링크들이 존재하고, 각각의 링크는 변환을 나타낸다. 임의의 특정한 링크에 대하여, 연계된 변환은 링크에 대한 입력단에 위치해 있는 하나 이상의 데이터 소스의 값의 복제를 수신하고, 링크의 출력단에 위치된 하나 이상의 데이터 타깃에 제공되는 결과 값을 생성한다. 임의의 특정한 변환에 대하여, 그 입력단에 있는 하나 이상의 데이터 소스의 값이 변경되면, 변환은 자동으로 재평가되고, 결국 잠재적으로 변환의 출력단에 있는 하나 이상의 데이터 타깃의 값이 변경된다.

[0021] 일 실시예에 있어서, 변환 체인이 얼마나 복잡한지에 관계없이, 변환은 하나 이상의 값을 입력으로서 수신하고 하나 이상의 결과 값을 출력으로서 제공할 수 있는 방정식, 규칙(rules), 제한(constraints), 시뮬레이션, 또는 임의의 다른 변환 유형을 표현하는 선언문(declarative statements)으로 구성될 수 있다. 변환 체인의 예시는 셀들 중 어느 하나가 데이터 소스 또는 데이터 타깃일 수 있는 스프레드시트 프로그램이다. 방정식(즉, 변환)은 임의의 셀과 연계될 수 있으며, 해당 셀이 방정식의 결과가 위치되는 데이터 타깃으로 되게 한다.

[0022] 단지 예시로서, 도 2는 단순한 변환 체인(200)을 도시하고, 해당 체인에는 단일의 링크(220)가 존재한다. 본 명세서 전반에서 사용된 도면 표기에 있어서, 링크는 화살표로 도시될 것이고, 화살표의 꼬리가 입력단을 나타내고, 화살표의 머리가 출력단을 나타낸다. 링크의 입력단에 다수의 데이터 소스가 존재하는 경우에는, 화살표는 다수의 꼬리로 나타내질 것이다. 화살표의 꼬리에서 데이터 소스의 값의 복제는 변환으로의 입력을 나타낸다. 변환의 결과 값에 의해 영향을 받는 다수의 데이터 타깃이 존재하는 경우에는, 화살표는 다수의 머리로 나타내질 것이다. 화살표의 머리에서 데이터 타깃의 값은 변환으로부터의 출력을 나타낸다.

[0023] 예를 들어, 도 2는 데이터 소스(201), 데이터 타깃(202), 및 단일의 링크(220)를 포함하는 단순한 변환 체인(200)을 도시한다. 링크(220)는 데이터 타깃(202)에서 값(212)을 생성하기 위해 데이터 소스(201)에서 값(211)의 복제에 대하여 수행된 변환을 나타낸다. 값(211)이 변경되면, 링크(220)에 의해 나타내진 변환이 잠재적으로 재평가되고, 결국 데이터 타깃(202)에서의 값(212)이 변경된다.

[0024] 도 3은 3개의 데이터 소스(301, 302, 및 303); 2개의 데이터 타깃(304 및 305), 및 단일의 링크(320)를 포함하는 단일의 예시적인 다른 변환 체인을 도시한다. 링크(320)는 데이터 타깃(304 및 305)에서의 값을 생성하기 위해, 데이터 소스(301, 302 및 303) 내의 값의 복제에 대하여 수행되는 변환을 나타낸다. 데이터 소스(301, 302 또는 303) 내의 값을 중 어느 하나가 변경되면, 변환 링크(320)가 잠재적으로 재평가되고, 결국 어느 하나 이상의 데이터 타깃(304 및 305) 내의 값이 변경된다.

[0025] 도 4는 다른 예시적인 변환 체인(400)을 도시하는 한편, 더욱 복잡한 변환 체인을 생성하기 위해, 하나의 링크에 대한 데이터 소스가 다른 링크에서의 데이터 타깃으로 될 수 있게 변환 체인들이 서로에 대하여 구축될 수 있는 원리를 도시한다. 예를 들어, 변환 체인(400)은 변환 체인(200)의 인스턴스(401) 및 변환 체인(300)의 인스턴스(402)를 포함한다. 이 경우에, 링크(220)의 데이터 타깃(202)은 링크(320)의 데이터 소스(301)이기도 하다. 데이터 소스(201)에 의해 값이 변경되면, 링크(220)에 의해 나타내진 변환이 잠재적으로 재평가되고, 결국 다음 링크(320)의 데이터 소스(301)와 마찬가지인 데이터 타깃(202)에서의 값이 변경되게 된다. 유사하게, 데이터 소스(301)의 값에 있어서의 변경은 변환 링크(320)가 잠재적으로 재평가되는 결과로 되어서, 결국 어느 하나 이상의 데이터 타깃(304 및 305) 내의 값이 변경되게 된다. 데이터 타깃(304 및 305)은 마찬가지로, 또 다른 링크의 데이터 소스를 나타낼 수도 있다. 그에 따라, 복잡한 변환 체인에 있어서, 값의 변경은 변환 체인

내의 변환의 적절한 자동 재평가를 통해 변환 체인에서의 다수의 노드를 통한 전파된 값의 변경을 야기할 수 있다.

[0026] 예시적인 변환 체인(400)이 단지 2개의 링크를 포함하고 있지만, 변환 체인은 매우 복잡할 수 있으며, 가산 가능한 노드들 및 해당 가산 가능한 노드들을 연결하는 관련 링크들을 수반할 수 있다. 본 명세서에 기술된 원리는 변환 체인의 복잡도에 관계없이 작동할 수 있다.

[0027] 도 5는 6개의 장치(501 내지 506)가 존재하는 환경(500)을 도시한다. 생략 부호(507)는 환경(500) 내에 존재하는 장치의 수에 있어서의 융통성을 나타낸다. 실제로, 매우 동적으로 환경(500)에 드나드는 장치가 있을 수 있다. 필수적인 것은 아니지만, 각각의 장치(507)는 도 1의 컴퓨팅 시스템(100)에 대하여 기술된 대로 구성될 수 있다.

[0028] 환경(500)은, 종종 그럴 수 있을지라도, 모든 장치가 동일 근거리 안에 위치되는 물리적 환경으로 될 필요는 없다. 대신에, 환경(500)은, 하나 이상의 사용자가 입력을 제공할 수 있게 하는 일련의 장치 및 다수의 사용자에게 출력이 제공될 수 있게 하는 다수의 장치가 있는 임의의 환경으로서 생각될 수 있다.

[0029] 일부 장치(예컨대, 장치(501 내지 504))는 협력 애플리케이션을 갖는 단순히 협력하는 장치들 이상의 무언가를 형성하는 것을 돋는다. 대신에, 장치들(501 내지 504) 각각은 복합 애플리케이션의 컴포넌트와 연관된다. 장치(501 내지 504) 중 어느 하나가 환경(500)을 벗어나면, 복합 애플리케이션은 더 작아지고, 그에 따라 결국 애플리케이션의 기능이 변경되게 된다. 한편, 애플리케이션의 컴포넌트를 갖는 장치(예컨대, 장치(501 내지 504))가 환경(500)에 진입하면, 복합 애플리케이션은 실제로 더 커지고, 그에 따라 실제로 복합 애플리케이션의 기능 및 해당 구조가 변경된다. 본 명세서에서 기술된 원리에 따라, 하나의 장치의 변환 체인은 다른 장치의 변환 체인과 연결되고, 결국 증대된 장치 세트를 더욱 효과적으로 이용할 수 있는 보다 큰 대형 변환 체인으로 된다.

[0030] 장치(505)는 환경(500)에서 이용될 수 있는 출력 장치를 나타내지만, 변환 체인을 복합 애플리케이션의 대형 변환 체인에 반드시 제공하는 것은 아니다. 예를 들어, 장치(505)는 대형 스크린 디스플레이일 수 있다. 장치(506)는 환경(500)에서 이용될 수 있는 입력 장치를 나타내지만, 변환 체인을 복합 애플리케이션의 대형 변환 체인에 반드시 제공하는 것은 아니다. 예를 들어, 장치(506)는 마이크일 수 있다. 장치(501 내지 504)의 존재는 각각의 연계된 변환 체인의 부분이 복합 애플리케이션의 대형 변환 체인에 제공되게 한다. 그러나, 장치(501 내지 504)도 역시 전체적으로 복합 애플리케이션에 의해 이용될 수 있는 입력 능력 및 출력 능력을 가질 수 있다. 환경(500)은 아래에서 추가로 기술되는 외부 시스템(510)을 선택적으로 포함할 수 있다.

[0031] 도 6a 내지 도 6d는 예시적인 변환 체인 인스턴스 또는 클래스(600A 내지 600D)를 도시한다. 인스턴스는 클래스와 동일한 구조를 갖게 되므로, 도시된 형태는 변환 인스턴스 뿐만 아니라 변환 클래스를 나타내는 것으로 간주될 수 있다. 그러나, 인스턴스는 변환 체인의 하나 이상의 노드 각각과 연계된 특정한 인스턴스 상태를 갖게 된다. 그에 따라, 요소(600A 내지 600D)는 변환 체인 클래스 또는 변환 체인 인스턴스라고도 할 수 있다. "변환 체인"이라는 용어는 일반적으로 변환 체인 클래스와 그 연계된 변환 체인 인스턴스를 나타내는데 이용될 것이다. 예시로서, 변환 체인 인스턴스(600A 내지 600D)는 각각의 장치(501 내지 504)와 연계될 수 있다.

[0032] 예시적인 변환 체인(600A 내지 600D)은 지나치게 복잡한 예시에 의해 본 명세서에 기술된 폭넓은 원리를 이해하기 어렵게 만드는 것을 피하기 위해 비교적 단순하다. 즉, 본 명세서에 기술된 원리는, 변환 체인이 얼마나 복잡한지에 관계없이, 또한 환경 내에서 복합 애플리케이션을 형성하는 변환 체인 및 연계된 장치의 수에 관계없이 적용된다.

[0033] 도 6a 내지 도 6d의 표기에 있어서, 변환 클래스(600N)(여기서, N은 범위가 A 내지 D임)에 속하는 노드는 접미사 N을 이용해서 나타내진다. 예를 들어, 도 6a에서, 변환 체인(600A)은 노드(601A, 602A, 603A, 및 604A)를 포함한다. 나머지 요소(601B, 601C 및 601D)는 접미사 "A"로 끝나지 않기 때문에, 변환 체인(600A) 내의 노드가 아니다. 대신에, 요소(601B, 601C 및 601D)는 다른 변환 체인과의 종속성을 나타낸다.

[0034] 도 6a 내지 도 6d, 도 7a 내지 도 7d, 도 8a 내지 도 8c, 및 도 9 전반에서, 변환 체인 자체의 노드보다는, 오히려 종속 요소(dependency element)인 해당 요소들을 강조하기 위해, 종속 요소는 점선 경계선으로 나타내진다. 데이터는, 변환 체인이 종속 요소에 의해 나타내진 노드를 포함하는 다른 변환 체인과 연결되지 않는 한, 노드로부터 종속 요소까지 흘러가지 않는다. 데이터가 특정한 변환을 따라 흘러갈 수 없다는 사실은 도면 전반에서 "X"로 표시된 링크에 의해 나타내진다.

[0035] 예를 들어, 변환 체인(600A)에서의 요소(601B)는 변환 체인(600B)에서의 노드(601B)와의 종속성을 나타낸다.

종속 요소(601B)는 경계선이 점선으로 되어 있고, 이 단계에서는, 변환 체인(600A)이 변환 체인(600B)과 연결되지 않기 때문에, 종속 요소(601B)로 이어지거나 또는 그로부터 나오는 모든 링크는 "X"로 표시된다. 변환 체인(600A)에서의 요소(601C)는 변환 체인(600C)에서의 노드(601C)와의 종속성을 나타낸다. 변환 체인(600A)에서의 요소(601D)는 변환 체인 클래스(600D)에서의 노드(601D)와의 종속성을 나타낸다.

- [0036] 변환 체인 인스턴스(600A)는, 단독으로 애플리케이션으로서 기능할 수 있다. 예컨대, 데이터 소스(601A)로부터의 값은 변환된 결과를 데이터 타깃(604A)의 값으로서 형성하는데 이용될 수 있다. 또한, 데이터 소스(601A 및 602A)로부터의 값들은 변환되어서, 결국 데이터 타깃(603A)의 값으로 될 수 있다. 변환 체인 인스턴스(600A)가 단독이면, 요소(601B, 601C 및 601D)로 이어지거나 또는 그로부터 나오는 변환은 평가되지 않는다.
- [0037] 변환 체인(600B)은 3개의 노드(601B, 602B 및 603B)를 포함한다. 그러나, 변환 체인(600B)은 상이한 변환 체인에서의 노드를 나타내는 종속 요소(601A, 602A, 601C 및 603C)를 또한 포함한다. 또한, 변환 체인 인스턴스(600B)는 단일 애플리케이션으로서 독립적으로 동작할 수 있다. 예컨대, 데이터 소스(601B)로부터의 값은 변환을 통해 제공되어서 데이터 타깃(602B)에 대한 결과 값을 생성할 수 있다. 데이터 소스(602B)로부터의 값은 변환을 통해 제공되어서 데이터 타깃(603B)에 대한 결과 값을 생성할 수 있다.
- [0038] 변환 체인 인스턴스(600A 및 600B)가 독립적으로 동작할 수 있는 하지만, 도 7a는 변환 체인(600B)과 연결된 변환 체인(600A)을 포함하는 연결 변환 체인(700A)을 도시한다. 각각의 변환 체인에서의 종속 요소는 이제 실제 노드로 대체되어 참조된다. 예컨대, 도 6a의 종속 요소(601B)는 이제 노드(601B)이며, 도 6b의 종속 요소(601A)는 이제 노드(601A)이다. 접미사 A 또는 B를 갖는 모든 노드가 변환 체인(700A) 내의 노드이며, 접미사 C 및 D를 갖는 해당 노드만이 종속 요소이다. 예컨대, 노드(601A, 602A, 603A, 604A, 601B, 602B 및 603B)는 증대된 변환 체인(700A) 내의 노드이고, 복합 애플리케이션의 기능은 개개의 변환 체인(600A 및 600B) 단독의 기능의 합보다 다소 양호해진다.
- [0039] 변환 체인(600C)은 3개의 노드(601C, 602C 및 603C)를 포함한다. 그러나, 변환 체인(600C)은 상이한 변환 체인에서의 노드를 나타내는 종속 요소(603A, 601B 및 603B)를 또한 포함한다. 또한, 변환 체인 인스턴스(600C)는 단일 애플리케이션으로서 독립적으로 동작할 수 있다. 예컨대, 데이터 소스(601C)로부터의 값은 변환을 통해 제공되어서 데이터 타깃(602C)에 대한 결과 값을 생성할 수 있다. 유사하게, 데이터 소스(601C)로부터의 값은 변환을 통해 제공되어서 데이터 타깃(603C)에 대한 결과 값을 생성할 수 있다.
- [0040] 변환 체인 인스턴스(600A 및 600C)가 독립적으로 동작할 수 있는 하지만, 도 7b는 변환 체인(600C)과 연결된 변환 체인(600A)을 포함하는 연결 변환 체인(700B)을 도시한다. 각각의 변환 체인에서의 종속 요소는, 해당 종속 요소가 어느 하나의 변환 체인(600A 또는 600C) 내의 노드를 나타내도록, 이제 실제 노드로 대체되어 참조된다. 이제, 접미사 A 또는 C를 갖는 모든 노드가 변환 체인 내의 노드이며, 접미사 B 또는 D를 갖는 해당 노드만이 종속 요소이다. 예컨대, 노드(601A, 602A, 603A, 604A, 601C, 602C 및 603C)는 증대된 변환 체인(700B) 내의 노드이다. 복합 애플리케이션의 기능은 개개의 변환 체인 인스턴스(600A 및 600C)의 기능의 합보다 양호해진다.
- [0041] 도 7c는 변환 체인 클래스(600C)와 연결된 변환 체인 클래스(600B)를 포함하는 연결 변환 체인(700C)을 도시한다. 각각의 변환 체인에서의 종속 요소는, 해당 종속 요소가 어느 하나의 변환 체인(600B 또는 600C) 내의 노드를 나타내도록, 실제 노드로 대체되어 참조된다. 이제, 접미사 B 또는 C를 갖는 모든 노드가 변환 체인 내의 노드이며, 접미사 A 또는 D를 갖는 해당 노드만이 종속 요소이다. 예를 들어, 노드(601B, 602B, 603B, 601C, 602C 및 603C)는 증대된 변환 체인(700C) 내의 노드이며, 복합 애플리케이션의 기능은 개개의 변환 체인 인스턴스(600B 및 600C)의 기능의 합보다 양호해진다.
- [0042] 도 8a는 연결되어 있는 변환 체인(600A, 600B 및 600C)을 포함하는 연결 변환 체인(800A)을 도시한다. 각각의 변환 체인에서의 종속 요소는, 해당 종속 요소가 어느 하나의 변환 체인(600A, 600B 또는 600C) 내의 노드를 나타내도록, 실제 노드로 대체되어 참조된다. 도시된 모든 노드는, 종속 요소(601D)를 제외하고는, 실제로 변환 체인 내의 노드라는 점에 유의한다. 복합 애플리케이션의 기능은 개개의 변환 체인(600A, 600B 및 600C)의 기능의 합보다 양호해진다.
- [0043] 변환 체인(600D)은 2개의 노드(601D 및 602D)를 포함한다. 그러나, 변환 체인(600D)은 상이한 변환 체인 클래스(600A)에서의 노드를 나타내는 단일의 종속 요소(603A)를 또한 포함한다. 또한, 변환 체인 클래스(600D)의 인스턴스는 단일의 애플리케이션으로서 독립적으로 동작할 수 있다. 예를 들어, 데이터 소스(601D)로부터의 값은 변환을 통해 제공되어서 데이터 타깃(602D)에 대한 결과 값을 생성할 수 있다.

- [0044] 변환 체인 인스턴스(600A 및 600D)가 독립적으로 동작할 수 있기는 하지만, 도 7D는 변환 체인(600D)과 연결된 변환 체인(600A)을 포함하는 연결 변환 체인(700D)을 도시한다. 각각의 변환 체인에서의 종속 요소는, 해당 종속 요소가 어느 하나의 변환 체인(600A 또는 600D) 내의 노드를 나타내도록, 실제 노드로 대체되어 참조된다. 이제, 접미사 A 또는 D를 갖는 모든 노드가 변환 체인 내의 노드이며, 접미사 B 또는 C를 갖는 해당 노드만이 종속 요소이다. 예를 들어, 노드(601A, 602A, 603A, 604A, 601D 및 602D)는 증대된 변환 체인(700D) 내의 노드이며, 복합 애플리케이션의 기능은 개개의 변환 체인(600A 및 600D)의 기능의 합보다 다소 양호해진다.
- [0045] 도 7a 내지 도 7d는 변환 체인(600A, 600B, 600C 및 600D) 중 2개를 수반하는 가능한 모든 치환 및 단지 2개의 변환 체인을 도시한다. 변환 체인(600B 및 600D)은, 어느 변환 체인도 다른 변환 체인에서의 노드를 나타내는 종속 요소를 갖지 않기 때문에, 2개의 변환 체인 조합으로 직접 연결되지 않는다. 또한, 변환(600C 및 600D)은, 어느 것도 다른 것에 대한 종속성 참조를 갖지 않기 때문에, 2개의 변환 체인 조합으로 직접 연결되지 않는다.
- [0046] 도 8a는 변환 체인(600A, 600B, 600C 및 600D) 중 3개로 이루어진 3가지의 가능한 조합 중 하나 및 단지 3개의 변환 체인을 도시한다. 특히, 도 8a는 변환 체인(600A, 600B 및 600C)을 조합한 증대된 변환 체인(800A)을 도시한다. 도 8b는 변환 체인(600A, 600B 및 600D)을 조합한 증대된 변환 체인(800B)을 도시한다(모든 노드는 종속 요소(601C 및 603C)를 제외하고는, 변환 체인의 일부임). 도 8c는 변환 체인(600A, 600C 및 600D)을 조합한 증대된 변환 체인(800C)을 도시한다(모든 노드는 종속 요소(601B 및 603B)를 제외하고는, 변환 체인의 일부임). 변환 체인(600D)이 변환 체인(600B)(또는 그 반대 또한 마찬가지)에 대한, 또는 변환 체인(600C)(또는 그 반대 또한 마찬가지)에 대한 종속성 참조를 포함하지 않기 때문에, 변환 체인(600B, 600C, 및 600D)의 조합은 없다는 점에 유의한다. 도 9는 조합된 모든 변환 체인(600A, 600B, 600C 및 600D)을 포함하는 조합 변환 체인(900)을 도시한다.
- [0047] 그에 따라, 환경에서 각각의 장치(501 내지 504)와 연계된 변환 체인(600A, 600B, 600C 및 600D)이 주어지면, 형성될 수 있는 8가지의 가능한 복합 애플리케이션이 존재한다(도 7a 내지 도 7d, 도 8a 내지 도 8c, 및 도 9의 변환 체인에 대응). 따라서, 다양한 장치의 변환 체인들이 환경에 대하여 연결 및 분리되기 때문에, 해당 변환 체인 자체가 변경되고, 그에 따라 복합 애플리케이션의 구조가 변경된다. 예를 들어, 데이터 소스(601A)의 값의 변경은, 데이터 소스(601A)가 단독으로 변환 체인(600A) 내에 있는지, 변환 체인(700A) 내에 있는지, 변환 체인(700B) 내에 있는지, 변환 체인(700D) 내에 있는지, 변환 체인(800A) 내에 있는지, 변환 체인(800B) 내에 있는지, 변환 체인(800C) 내에 있는지, 또는 변환 체인(900) 내에 있는지의 여부에 따라 해당 변경의 효과가 하나 이상의 변환을 통해 자동으로 전파되기 때문에 변환 체인에 매우 상이한 영향을 줄 수 있다.
- [0048] 예시로서, 장치(501)가 먼저 변환 체인(600A)이 이용되는 환경에 진입한다고 상정한다. 이후, 장치(502)는 변환 체인(600B)이 변환 체인(600A)에 연결되어서, 결국 변환 체인(700A)으로 되는 환경에 진입한다. 따라서, 동작하는 변환 체인은 변환 체인(600A)으로부터 변환 체인(700A)으로 변경된다. 이어서, 제 3 장치(503)는 변환 체인(600C)이 변환 체인(700A)에 연결되어서, 결국 변환 체인(800A)으로 되는 환경에 진입한다고 상정한다. 따라서, 동작하는 변환 체인은 변환 체인(700A)으로부터 변환 체인(800A)으로 변경된다. 이어서, 장치(500B)가 나간다고 상정한다. 그러면, 변환 체인(700B)이 동작하게 된다. 이어서, 변환 체인(600D)이 변환 체인(700B)에 연결되어서, 결국 동작하는 변환 체인이 되는 변환 체인(800C)으로 되는 환경에 장치(504)가 진입한다고 상정한다. 이어서, 장치(500C)가 나가고, 결국 변환 체인(700D)이 동작하게 된다고 상정한다. 이어서, 장치(501)가 나가고, 결국 변환 체인(600D)이 동작하게 된다고 상정한다. 마지막으로, 장치(504)가 나가고, 환경에는 동작하는 변환 체인은 남아 있지 않게 된다. 이러한 상황에서, 동작하는 애플리케이션은 구조가 다음과 같이 7차례 변경되었다(그에 따라 기능도 변경됨): 1) 변환 체인(600A)으로 시작, 2) 이어서, 변환 체인(700A)으로 이동, 3) 이어서, 변환 체인(800A)으로 이동, 4) 이어서, 변환 체인(700B)으로 이동, 5) 이어서, 변환 체인(800C)으로 이동, 6) 이어서, 변환 체인(700D)으로 이동, 7) 이어서, 변환 체인(600D)으로 완료.
- [0049] 도 10은 다수의 연결된 변환 체인 인스턴스에 의해 나타내진 복합 애플리케이션의 촉진을 준비하는 방법(1000)의 흐름도를 도시한다. 방법(1000)은 환경(500) 내에 있는 어느 하나의 장치(501 내지 507)에 의해 수행될 수 있다. 대안으로서 또는 추가적으로, 방법(1000)은 예컨대, 도 5의 외부 시스템(510)에 의하는 등과 같이, 장치(501 내지 507) 외부에서 수행될 수 있다. 예를 들어, 외부 시스템(510)은 클라우드 컴퓨팅 환경에서 제공되는 서비스일 수 있다. 방법(1000)이 어디에서 수행되는지에 관계없이, 도 11은 방법(1000)을 수행할 수 있는 시스템(1100)을 도시한다. 그에 따라, 시스템(1100)이 하나 이상의 장치(501 내지 507)에 포함되는지의 여부 또는 시스템(1100)이 도 5의 외부 시스템(510)인지의 여부, 또는 그 조합에 관계없이, 방법(1000)은, 이제 도 11의

시스템(1100)을 빈번하게 참조하여 기술될 것이다.

[0050] 방법(1000)은 시간 종속성을 갖지 않는 2개의 동작(1001 및 1002)을 포함한다. 예를 들어, 방법은 상이한 변환 체인 클래스들 사이의 종속성을 식별하는 단계를 포함한다(동작(1001)). 이 종속성은 기본적으로, 상이한 변환 체인 클래스들의 인스턴스들이 호환되는 것을 허용하기 위해(추가적인 인스턴스-기반 승인을 받음), 해당 클래스를 생성한 클래스 저자(author)에 의해 생성된 인가(authorizations)이다.

[0051] 예를 들어, 시스템(1100)은 로직 컴포넌트(1101), 및 6개의 변환 체인 클래스 정의(1111 내지 1116)를 포함하는 것으로 도시되는 변환 클래스 정의 라이브러리(1110)를 포함한다. 그러나, 생략 부호(1117)는 변환 체인 클래스 정의 라이브러리(1110)가 임의의 수의 변환 체인 클래스 정의를 포함할 수 있음을 나타낸다. 예시로서, 변환 체인 클래스 정의(1111, 1112, 1113 및 1114)는 제각기 도 6a 내지 도 6d의 변환 체인 클래스(600A, 600B, 600C 및 600D)를 규정한다고 상정한다. 각각의 변환 체인 클래스가 종속 요소를 규정하기 때문에, 시스템(1100)은 라이브러리(1110)를 이용해서 상이한 변환 체인 클래스들 사이의 종속성을 식별한다.

[0052] 예컨대, 변환 체인(600A)의 종속 요소(601B) 및 변환 체인(600B)에서의 종속 요소(601A)를 관찰함으로써, 로직 컴포넌트(1101)는 변환 체인 클래스(600A 및 600B)가 연결되었을 경우 또는 해당 클래스의 인스턴스들이 연결되었을 경우에 존재하게 되는 노드(601A 및 601B) 사이의 종속성을 식별할 수 있다. 로직 컴포넌트(1101)는 변환 체인(600A 및 600B) 양자의 저자가 이들 종속 요소에서 연결을 인가하는 것에 동의한다고 추론할 수 있다(실행 시에 사용자가 또한 인스턴스 레벨에서 인가한다고 추정).

[0053] 방법(1000)은 또한, 어느 하나의 변환 체인 클래스의 인스턴스와 연계되는 장치를 식별하는 단계를 포함한다(동작(1002)). 장치는, 어쩌면 장치 식별자 뿐만 아니라 잠재적으로 장치가 이용하려는 임의의 변환 체인 클래스로 시스템(1100)에 대하여 자체적으로 식별할 수 있다. 대안으로서, 시스템(1100)은 각각의 장치와 연계된 특정한 디폴트 변환 클래스를 구비할 수 있고, 그에 따라 단지 어쩌면 장치 식별자를 수신하게 된다. 장치가 이용을 허가받은 일부 변환 체인 클래스 및 장치가 이용을 허가받지 못한 다른 변환 체인 클래스가 있을 수 있다.

[0054] 예를 들어, 시스템(1100)은 또한, 다양한 변환 체인 클래스와 연계되는 장치의 레지스트레이션을 포함하는 장치 레지스트레이션(1120)을 포함한다. 예를 들어, 생략 부호(1126)가 장치 레지스트레이션(1120)이 임의의 수의 장치 레지스트레이션을 포함할 수 있음을 나타내고 있기는 하지만, 장치 레지스트레이션(1120)은 5개의 장치 레지스트레이션(1121 내지 1125)을 포함하고 있다. 또한, 장치 레지스트레이션(1120)은 장기간 지속될 수 있거나, 및/또는 어쩌면 단지 잠시동안만 유지될 수 있다. 어떻든 간에, 시스템(1100)은 다수의 장치를 식별하고, 해당 장치에 대하여 연계된 변환 체인을 찾는다. 단지 예시이지만, 어쩌면 도 5의 장치(501 내지 504)는 레지스트레이션(1121 내지 1124)과 연계된다. 이러한 연계에 의해, 시스템(1100)은 변환 체인 인스턴스(600A, 600B, 600C, 600D)(제각기, 변환 체인 클래스 정의(1111, 1112, 1113 및 1114)에 의해 규정됨)가 제각기 해당 장치(501 내지 504)와 연계된다는 것을 검출할 수 있다.

[0055] 일부 실시예에 있어서, 각각의 장치를 대신하여 시스템(1100) 자체에서 특정한 변환 체인 클래스의 인스턴스의 표현(representation)을 실행한다. 예를 들어, 변환 체인 클래스(600A)가 장치(501)와 연계된다고 식별되면, 시스템(1100)은 장치(501)에 대한 변환 체인 인스턴스(600A)의 표현을 동작시킬 수 있다. 그에 따라, 입력이 장치로부터 수신되면, 시스템(1100)은 변환 체인에서의 특정 노드에 대하여 변경을 식별하고, 변환 체인 전반에 걸쳐 체인에 전파한다. 이는 "핸드온(hands-on)" 해법이라고도 한다. 이 핸드온 해법은, 입력을 제공하고 있는 장치 자체보다 처리 능력이 우수한 장치 또는 시스템 또는 클라우드에 처리가 분담될 수 있다는 점에서 약간 유리하다.

[0056] 본 명세서에서 "핸드오프(hands-off)" 해법이라고도 하는 다른 해법에서는, 장치는 클래스의 인스턴스를 자체 실행함으로써 변환 클래스와 연계된다. 종속성이 다른 장치와 연계된 다른 변환 체인에 대하여 발생할 때, 해당 다른 장치도 핸드오프 해법에 관계되어 있다면, 데이터는 해당 장치로 직접적으로(예컨대, 피어 투 피어 네트워킹을 통해) 훌러갈 수 있다. 대안으로서, 장치는 데이터를 다른 장치로 간접적으로(예컨대, 외부 시스템 또는 클라우드를 통해) 훌러가게 할 수 있다. 외부 시스템(510)이 변환 체인 클래스의 인스턴스의 표현을 해당 다른 장치에 대하여 실행하고 있으면, 장치는 나머지 변환 체인에의 추가 전파를 위해 데이터를 외부 시스템(510)으로 대신 훌러가게 할 수 있다.

[0057] 도 12는 변환 클래스의 2개의 인스턴스를 연결하는 방법(1200)의 흐름도를 도시한다. 먼저, 변환 체인 클래스들이 연결될 수 있는 유형인지가 확인된다(동작(1201)). 예를 들어, 변환 체인 클래스들이 서로 참조하지 않는 경우에는, 어쩌면 각각의 클래스의 인스턴스들을 연결하기 위한 클래스-기반 인가가 없을지도 모른다. 예를 들

어, 변환 체인 클래스(600D)가 변환 체인 클래스(600B) 또는 변환 체인 클래스(600C)와 연결될 수 없는데, 그 이유는 변환 체인 클래스(600D)가 변환 체인 클래스(600B 또는 600C)에서의 요소를 참조하지 않기 때문이다. 그러나, 변환 체인 클래스(600D)는 변환 체인 클래스(600A)와 연결될 수 있는데, 그 이유는 이들이 서로에 대한 종속성 참조를 상호간에 포함하기 때문이다.

[0058] 그러나, 이 예시에 있어서, 변환 체인 클래스(600D)가 변환 체인 클래스(600B 및 600C)에 직접적으로 연결될 수는 없지만, 변환 체인 클래스(600D)는, 변환 체인(700A, 700B 및 800A)이 변환 체인(600B 및 600C) 중 하나 또는 둘 모두를 포함하더라도, 변환 체인(700A, 700B 및 800A)과 연결될 수는 있다.

[0059] 그러나, 대안적인 실시예에 있어서, 변환 체인 클래스의 저자는 다른 변환 체인 클래스에 연결하는 것에 대한 추가적인 제한을 지정할 수 있다. 예를 들어, 해당 변환 클래스 인스턴스가 이미 복합 변환 체인이거나 및/또는 복합 변환 체인이 특정한 구성 변환 체인을 구비하면, 저자는 특정 변환 클래스 인스턴스에 연결하는 것이 허가되지 않는다는 일반적 제한을 명시할 수 있다. 예를 들어, 2개의 변환 체인 인스턴스의 연결시에, 시스템(1100)은 복합 변환 체인을 구성하는데 이용되었던 변환 체인 식별자의 트랙을 그 지점까지는 유지할 수 있다. 해당 목록은 클래스-기반 인가에 대한 조건이 충족되어 있는지의 여부를 확인하는데 이용될 수 있다.

[0060] 저자는 또한, 단일 종속성 단위로 제한을 표현할 수도 있다. 예를 들어, 변환 체인 클래스(600A)의 종속 요소(601B)에 있어서, 저자는 종속 요소가 연결되는 변환 체인이 상대측에 의해 저작된 식별된 변환 체인 클래스를 포함하지 않는 경우에만 해당 종속 요소에 대하여 연결이 인가된다고 표현할 수 있다. 저자는 또한, 종속성을 잊게 되는 변환 자체(예컨대, 노드(601A)와 종속 요소(601B) 사이)에 제한 또는 조건을 기입함으로써, 변환 체인에서 다른 연결 변환 체인으로 흘러가는 데이터를 제어할 수도 있다.

[0061] 그러나, 변환 체인 클래스들이 호환될 수 있다고 해도, 이는, 사용자가 해당 변환 체인 클래스의 특정 인스턴스를 다른 변환 체인 클래스의 다른 인스턴스와 연결하기를 바란다는 것을 의미하는 것은 아니다. 결국, 데이터 자체(예컨대, 인스턴스 상태)가 사용자에게 민감한 것일 수 있다. 그에 따라, 방법(1200)은 또한, 상이한 변환 체인 클래스들의 인스턴스들이 연결되어야 한다고 결정하는 단계를 포함한다(동작(1202)).

[0062] 상이한 변환 체인 클래스들의 2개의 인스턴스를 연결하도록 인가하는 연결 기준은 하기의 것들을 하나 이상 포함할 수 있다: 즉, 사용자가 미팅 참석자 목록에 있는지의 여부, 다양한 장치들의 사용자들의 관계(예컨대, 가족, 소셜 네트워크 친구 등), 장치간의 통신 능력(예컨대, 근거리), 각각의 장치의 근접도(예컨대, 동일 회의실 내), 사용자의 요청 등. 예를 들어, 연결 기준은, 예컨대 인스턴스의 연관 사용자들이 같은 팀에 속해 있는지와 같은 일부 비즈니스 기준을 포함할 수 있다. 다른 예시로서, 어떤 장치는 소매점 또는 호텔에 있는 키오스크(kiosk)일 수 있고, 이 경우, 고객은 키오스크를 이용하고, 점원 또는 관리인은 자동으로 자신의 장치를 이용해서 그들의 변환 체인을 키오스크의 변환 체인과 연결하게 되고, 그에 따라 복합 애플리케이션을 이용하는 고객과 상호작용할 수 있다. 연결 기준에는 조건이 적용될 수 있다. 예를 들어, 관리인이 주위에 없는 경우(어쩌면, 고객의 애플리케이션과 연결하도록 짹이 되는 애플리케이션을 적극적으로 이용하지 않는 관리인 또는 네트워크를 벗어나 있는 관리인에 의해 검출될 수 있음) 벨홉(bellhop) 장치가 고객의 애플리케이션에 연결될 수 있다.

[0063] 장치와 연계된 변환 체인 클래스의 인스턴스가 해당 클래스에 의해 규정된 디폴트 상태에서 시작되어야 하는 것이 아니면, 핸드온 해법에 있어서 장치가 인스턴스 데이터(예컨대, 장치와 연계된 변환 체인의 노드들 각각의 값)를 시스템(1100)에 업로드할 수 있다(동작(1203)). 핸드오프 해법에 있어서는, 아마도, 단지 경계 노드의 인스턴스 데이터만이 시스템(1100)에 업로드된다.

[0064] 이후, 변환 체인 인스턴스는 나머지 변환 체인과 연결된다(동작(1204)). 예를 들어, 이는 타깃 변환 체인 인스턴스에서의 노드에 대응하는 종속 요소에 대한 소스 변환 체인 인스턴스를 검색함으로써 달성될 수 있다. 이러한 종속 요소가 발견되면, 해당 종속 요소는 타깃 변환 체인 인스턴스에서 실제 식별된 노드로 대체되고, 그에 따라 소스 변환 체인 인스턴스와 타깃 변환 체인 인스턴스 사이에 실제 링크가 생성된다. 이는 상기와 같이 타깃 변환 체인 인스턴스에서의 노드를 식별하는 소스 변환 체인 인스턴스에서의 종속 요소를 발견한 전부에 대하여 반복된다. 타깃 변환 체인 인스턴스에서의 노드에 대응하지 않는 소스 변환 체인 인스턴스에서 발견된 종속 요소가 있으면, 해당 종속 요소는 병합된 변환 체인 인스턴스에서 종속 체인 요소를 유지한다.

[0065] 도 6a 내지 도 6d의 예시에 있어서는, 어떠한 변환 체인 클래스(600A 내지 600D)도 동일 클래스를 나타내는 종속 요소를 포함하지 않는다는 점에 유의한다. 그러나, 본 명세서에 기술된 원리는 특정 변환 체인 클래스의 인스턴스가 동일한 특정 변환 체인 클래스의 다른 인스턴스에서의 다른 노드를 나타내는 종속 요소를 가질 수 있

는 경우에도 작동할 수 있다. 또한, 특정 노드에 의한 연결이 허가되면, 하나 이상의 상이한 변환 체인 클래스로부터의 다수의 노드가 특정 노드에 연결될 수 있다.

[0066] 연결시에는, 변환 체인이 증대되고, 데이터는, 변환 체인이 마치 처음부터 원래 증대된 형태로 저작되었던 것처럼, 증대된 변환 체인 내를 흐른다(동작(1205)). 예를 들어, 변환 체인 클래스(600A)의 인스턴스가 변환 체인 클래스(600B)의 인스턴스에 연결되면, 그 결과는 변환 체인 클래스(700A)의 단일 인스턴스가 된다. 변환 체인 클래스(600A)의 인스턴스가 변환 체인 클래스(600C)의 인스턴스에 연결되면, 그 결과는 변환 체인 클래스(700B)의 단일 인스턴스가 된다. 변환 체인 클래스(600B)의 인스턴스가 변환 체인 클래스(600D)의 인스턴스에 연결되면, 그 결과는 변환 체인 클래스(700D)의 단일 인스턴스가 된다.

[0067] 변환 체인 클래스(800A)의 인스턴스는 변환 체인 클래스(700A 및 600C)의 인스턴스들을 연결하거나 또는 변환 체인 클래스(600A 및 700C)의 인스턴스들을 연결함으로써 형성될 수 있다. 변환 체인 클래스(800B)의 인스턴스는 변환 체인 클래스(700A 및 600D)의 인스턴스들을 연결함으로써 형성될 수 있다. 변환 체인 클래스(800C)의 인스턴스는 변환 체인 클래스(700B 및 600D)의 인스턴스들을 연결함으로써 형성될 수 있다. 이 연결 작업은, 주어진 환경에서 협업이 가능한 다수의 장치가 있는 상황에서는, 수차례 수행됨으로써 매우 큰 변환 체인을 생성할 수 있다.

[0068] 그에 따라, 연결시에(동작(1204)), 데이터는 심지어 구성 변환 체인들 사이의 경계로 이용되는 것을 가로질러 증대된 변환 체인 내에서 자유롭게 흐를 수 있다(동작(1205)). 예를 들어, 도 7a의 변환 체인(700A)에서, 데이터가 (적절한 변환을 통해) 노드(601A)로부터 노드(604A)로 흘러갈 수 있듯이, 데이터는 정확하게 (적절한 변환을 통해) 노드(601A)로부터 노드(601B)로 (애플리케이션 프로그램 인터페이스의 사용자 없이, 또한 기능 호출 없이) 자유롭게 흘러갈 수 있다.

[0069] 그에 따라, 시스템은 비교적 자동화되고 편리한 방식으로 매우 다양한 장치들과 연계된 변환들을 연결해서 다수의 장치에 걸쳐 공유되는 복합 애플리케이션의 변환 체인을 증대하는데 이용될 수 있다. 이는 매우 다양한 상황을 가능하게 한다.

[0070] 예를 들어, 특정 회의실에서의 미팅에 도착하는 개개인을 고려한다. 제 1 사용자가 랩탑 컴퓨터와 스마트폰을 가지고 회의실에 들어오는 경우에, 제 1 사용자의 스마트폰과 랩탑에 걸쳐 단일의 애플리케이션이 유효하게 실행되도록, 스마트폰과 연계된 변환 체인 인스턴스는 랩탑의 변환 체인 인스턴스와 연결된다. 잠시 후에, 제 2 사용자가 회의실에 들어온다. 제 2 사용자도 물론 미팅에 초대받은 사용자이므로, 해당 사용자의 스마트폰 변환 체인은 기존의 복합 변환 체인에 연결되어서 변환 체인을 더 증대시킨다. 이후, 텔레비전 스크린이 자동으로 켜져서 미팅과 연계된 일부 데이터를 출력한다. 제 3 사용자가 태블릿 컴퓨터를 가지고 들어오고, 마찬가지로 미팅에 초대받은 사용자이다. 따라서, 태블릿 컴퓨터와 연계된 변환 체인이 복합 변환 체인에 연결되어서 변환 체인을 더 증대시킨다. 따라서, 더 많은 장치가 변환 체인들을 연결하기에 적합한 환경에 합류할수록, 모든 장치에 걸쳐 실행되는 애플리케이션이 사실상 증대됨으로써 애플리케이션 자체의 기능이 변경된다. 다수의 사용자가 장치들 중 적어도 하나 또는 일부를 각각 관찰하도록 하는 환경을 다수의 사용자가 각각 공유할 수 있다.

[0071] 도 13은 변환 체인을 분리하는 방법(1300)의 흐름도를 도시한다. 장치가 환경에 들어올 때 변환 체인들을 연결하는 방법이 있는 것과 마찬가지로, 장치가 환경을 나갈 때 변환 체인들을 분리하는 방법이 있다. 먼저, 시스템은 변환 체인 클래스의 인스턴스가 분리되어야 한다고 결정한다(동작(1301)). 응답으로서, 해당 인스턴스에 대한 데이터의 흐름이 중단된다(동작(1302)). 인스턴스가 분리되어야 한다고 결정하는데 이용되는 분리 기준은 임의의 기준을 포함할 수 있지만, 예시로서는, 해당 인스턴스와 연계된 장치의, 환경 내의 다른 장치와의 근접도, 미팅 종료 상태, 해당 인스턴스와 연계된 장치와 환경 내의 다른 장치 사이의 통신 능력, 사용자의 요청 등을 포함할 수 있다. 따라서, 변환 체인은 장치들이 환경에 드나들에 따라 동적으로 변경될 수 있다.

[0072] 일 실시예에 있어서, 시스템(1100)은 프리젠테이션 서비스(1150)를 포함한다. 실행 중인 변환 체인 인스턴스의 하나 이상의 노드는 하나 이상의 주변 장치에 대한 렌더링을 구동할 수 있다. 유사하게, 하나 이상의 장치는 변환 체인 인스턴스의 하나 이상의 노드에 입력을 제공할 수 있다. 예를 들어, 도 5의 환경(500)에서 변환 체인(900)의 인스턴스가 실행중이고, 또한 장치(501 내지 504)가 각각의 변환 체인 클래스(600A 내지 600D)의 인스턴스와 연계됨으로써, 결국 증대된 변환 클래스(900)의 인스턴스로 된다고 상정한다. 도 9를 참조하면, 노드(601A 및 601C)는 입력 노드이고, 노드(604A, 603B, 602C 및 602D)는 출력 노드라고 상정한다. 프리젠테이션 서비스(1150)는 어떤 장치가 노드(601A 및 601C)에 입력을 제공할 수 있을지를 선택할 수 있고, 만일 그렇다면,

어떤 변환이 수행되어야 할지를 선택할 수 있다. 일 예시로서, 장치들은 각각의 부분 내의 입력 노드가 각각의 장치에 의해 제공되도록 그들의 변환 체인의 각각의 부분과 연계된다. 따라서, 기본적으로, 원래 변환 체인(600A)의 일부인 입력 노드(601A)는 장치(501)로부터 입력을 제공받을 수 있다(잠재적으로, 일부 변환 이후). 또한, 기본적으로, 원래 변환 체인(600C)의 일부인 입력 노드(601C)는 장치(503)로부터 입력을 제공받을 수 있다(잠재적으로, 일부 변환 이후). 프리젠테이션 서비스(1150)는 또한, 어떤 장치가 노드(604A, 603B, 602C 및 602D)로부터 출력을 렌더링해야 할지를 선택할 수 있고, (만일 그렇다면) 어떤 변환이 수행되어야 할지를 선택할 수 있다.

[0073] 도 14는 다중 장치 환경에서 변환 체인의 출력을 렌더링하는 방법(1400)의 흐름도를 도시한다. 방법(1400)은 도 11의 시스템(1100)에 의해 수행될 수 있거나, 또는 어쩌면 도 11의 프리젠테이션 서비스(1150)에 의해 수행될 수 있다. 로직 컴포넌트(1101)는 변환 체인 클래스의 인스턴스의 특정 노드(프리젠테이션 노드)가 변경되었는지 판정한다(동작(1401)). 예를 들어, 본 예시에 있어서, 노드(604A, 603B, 602C 및 602D)는 각각 출력 노드, 또는 "프리젠테이션 노드"이다.

[0074] 로직 컴포넌트는 또한, 적어도 해당 변경된 출력의 버전이 렌더링될 수 있는 다수의 후보 장치의 특성을 식별한다(동작(1402)). 예를 들어, 로직 컴포넌트(1101)는 각 장치의 렌더링 능력을 결정하기 위해 장치 레지스트리(1120)를 참조할 수 있거나, 및/또는 장치에게 직접 문의할 수 있다. 각 장치의 특성은 또한, 장치에 대한 각 사용자의 병렬성(juxtaposition)을 포함할 수도 있다.

[0075] 이후, 시스템은 식별된 특성에 기초하여 적어도 렌더링의 버전을 렌더링할 후보 장치들 중 적어도 하나를 선택한다(동작(1403)). 시스템은 출력의 가용성을 최대화하기 위해 선택을 할 수 있다. 예를 들어, 출력이 모든 사용자에 대하여 의도된 것이고, 모든 사용자가 물리적으로 존재하면, 출력은 대형 스크린에 렌더링될 수 있다. 출력이 상호적이고, 또한 특정 장치로부터의 입력을 요구하면, 해당 장치가 출력을 위해 선택될 수 있다.

[0076] 시스템은 장치에의 실제 출력 전에 임의의 변환이 렌더링에 적용되어야 할지를 판정한다(판정 블록(1404)). 변환은 선택된 장치의 적합성을 고려하고, 이 경우에, 변환은 렌더링에 대하여 선택된 장치에 따라 다를 수 있다. 변환은 렌더링의 유형을 바꾸는 것(예컨대, 텍스트를 음성으로, 음성을 텍스트로, 영상을 텍스트로, 텍스트를 영상으로 등)을 예로서 포함한다. 변환은 출력의 영화화(cinematization)를 포함할 수도 있다. 예를 들어, 영상은 입력으로부터 생성될 수 있는데, 해당 입력에서는, 배경이 제공되고, 어쩌면 배경이 적절히 변경되고, 또한 움직이는 요소가 영상의 안팎으로 이동된다.

[0077] 변환이 수행되어야 한다면(판정 블록(1404)에서 "예"), 해당 변환이 수행된다(동작(1405)). 마지막으로, 선택된 장치 상에서 렌더링이 촉진된다(동작(1406)). 그에 따라, 복합 변환 체인은 입력을 임의의 장치로부터 임의의 형태로 취하고, 필요에 따라 해당 입력을 복합 변환 체인의 입력 노드에 의해 인식되는 형태로 변환할 수 있다. 따라서, 이 입력은 변환 체인 전반에 걸쳐 전파된다. 하나 이상의 프리젠테이션 노드의 값이 변경되면, 출력을 렌더링하기 위해 적절한 장치가 선택될 수 있다. 그에 따라, 해당 장치가 프리젠테이션 노드를 포함하는 변환 체인의 부분과 연계되었는지에 관계없이, 렌더링에 가장 적합한 장치가 선택될 수 있다. 예를 들어, 프리젠테이션 노드(604A)의 값의 변경은 변환 체인 인스턴스(600A)에 대응하는 장치(501)에 관하여 전적으로 또는 심지어 전혀 렌더링될 필요가 없다. 일부 실시예에 있어서, 렌더링은 정보를 몇몇 유형의 움직임을 포함하는 몇 가지 물리적 작동으로 전환하는 것을 수반할 수 있다. 예를 들어, 렌더링은 문의 개방 또는 폐쇄, 밸브의 개방 또는 폐쇄, 문의 잠금 또는 잠금해제, 텔레비전의 켜기 또는 끄기 등을 포함할 수 있다.

[0078] 상세한 상황(이하, "장치 상황"이라고 함)은 이하에서 도 15a 내지 도 15m("도 15"라고 통칭함)에 대하여 기술될 것이다. 이 상황은 본 명세서에 기술된 폭넓은 원리에 의해 가능해지는 무한한 종류의 다양한 상황 중 단지 하나의 상황으로서 제공된다. 이 상황에서, 사용자는 다양한 컴퓨팅 장치를 주문하고 있다. 상기 상황은 도 15a에서 2개의 장치(1501 및 1052)만이 보이는 것으로 시작하지만, 해당 상황은 4개의 장치(1501, 1502, 1503 및 1504)를 수반한다. 4개의 장치(1501 내지 1504)는 각각 복합 애플리케이션으로의 입력의 제공 및 그로부터의 출력의 수신에 참여한다.

[0079] 도 15a는 랩탑 컴퓨터인 2개의 장치(1501 및 1502)를 도입한다. 이들 2개의 장치(1501 및 1502)는 초기에는 단지 운영 환경에 있는 장치일 뿐이고, 그에 따라 그들 각각의 변환 체인이 연결되어 있는 것으로 상정한다. 장치(1501)와 연계된 변환 체인은 제조자(Manufacturer)의 내비게이션 및 선택을 허용한다. 장치(1502)와 연계된 변환 체인은 제조자에 의해 제공된 장치 목록의 내비게이션을 허용하는 한편, 해당 목록에 있는 임의의 장치의 카운트를 사용자가 변경할 수 있게 한다.

- [0080] 구체적으로, 장치(1501)는 사용자가 다수의 제조자에 걸쳐 가로로 스크롤해서 제조자를 선택할 수 있게 한다. 도 15a에서, 장치(1501)는 상기와 같은 제조자가 "Vendor(벤더) 7", "Vendor 8" 및 "Vendor 9"로 레이블링한 단지 3개만을 보여주고, 사용자가 "Vendor 8"을 선택한 상태이다. 한편, 장치(1502)는 장치(1501)에서 선택된 벤더에 의해 제공되는 가로로 스크롤 가능한 장치 목록을 보여준다. 도 15a에서, Vendor 8이 장치(1501) 상에서 선택되기 때문에, 장치(1502)는 Vendor 8에 의해 제공된 장치 목록을 보여준다. 장치 목록에 있는 각각의 아이템은 주문된 장치의 사진(1505), 장치의 이름(1506), 장치의 단가(1507), 및 장치의 카운트(1508)를 포함한다. 각각의 장치에 대한 카운트는 상응하는 스크롤 제어부를 이용해서 제어될 수 있다.
- [0081] 도 15b는, 사용자가 제조자의 목록을 좌측으로 스크롤했고, 또한 "Vendor 4"를 선택한, 도 15a의 상태로부터 진전된 상황의 상태를 보여준다. 그에 따라, 장치(1502)로부터의 출력은 Vendor 4에 의해 제조된 장치의 목록을 디스플레이하도록 자동으로 변경된다. 그렇게 하기 위해, 데이터는 장치(1501)와 연계된 변환 체인으로부터 장치(1502)와 연계된 변환 체인으로 (애플리케이션 프로그램 인터페이스를 이용하는 일 없이) 자동으로 흘러갔다.
- [0082] 도 15c는, 사용자가 제조자의 목록을 좌측으로 스크롤했고, 또한 "Vendor 2"를 선택한, 도 15b의 상태로부터 진전된 상황의 상태를 보여준다. 그에 따라, 장치(1502)로부터의 출력은 Vendor 2에 의해 제조된 장치의 목록을 디스플레이하도록 자동으로 변경된다. 그렇게 하기 위해, 데이터는 다시 장치(1501)와 연계된 변환 체인으로부터 장치(1502)와 연계된 변환 체인으로 자동으로 흘러갔다.
- [0083] 도 15d는, 사용자가 Vendor 2에 의해 제공된 2개의 장치에 대한 주문 카운트를 변경하기 위해 장치(1502) 상에 디스플레이된 스크롤 제어부(1511 및 1512)를 이용한, 도 15c의 상태로부터 진전된 상황의 상태를 보여준다. 도 15e는, 사용자가 장치(1502)를 이용해서 Vendor 2에 의해 제공된 장치 목록을 통해 우측으로 스크롤하고, 또한 Vendor 2에 의해 제공된 2개의 추가적인 장치에 대한 주문 카운트를 변경하기 위해 장치(1502) 상에 디스플레이된 스크롤 제어부(1513 및 1514)를 이용한, 도 15d의 상태로부터 진전된 상황의 상태를 보여준다. 그에 따라, 이 단계에서는, 사용자가 Vendor 2에 의해 제공된 4개의 장치에 대하여 주문 카운트를 입력했다.
- [0084] 도 15f는, 해당 환경으로 제 3 장치(1503)가 도입된, 도 15e의 상태로부터 진전된 상황의 상태를 보여준다. 예를 들어, 제 3 장치(1503)는 최대 단가 및 전체 주문에 대한 총 비용을 디스플레이하는 변환 체인과 연계될 수 있다. 해당 환경에 장치(1503)가 도입되면, 변환 체인의 해당 부분이 기존의 복합 변환 체인과 연결되고, 그에 따라 상기와 같은 비용 데이터를 장치(1503)에 출력하도록 복합 애플리케이션의 기능이 변경된다. 변환 체인 인스턴스가 연결되면, 카운트 및 단가를 나타내는 데이터가 변환 체인의 해당 부분으로 흘러가서, 장치(1503)에 즉시 채워지게 된다. 흐름은 장치(1503)가 최대 단가를 나타내는 것과는 반대 방향으로도 발생하며, 그에 따라 장치(1502)에서 보여진 장치 목록이 최대 단가보다 낮은 임의의 장치로 제한된다는 점에 유의한다. 이 경우에, 앞서 열거된 장치는 모두 최대 단가를 하회하고 있었기 때문에, 장치(1502) 상에서의 시각화(visualization)에 있어서는 변화가 없다.
- [0085] 도 15g는, 사용자가 Vendor 2에 의해 제공된 또 다른 장치에 대한 카운트를 입력하기 위해 장치(1502) 상에서 스크롤 제어부(1515)를 이용한, 도 15f의 상태로부터 진전된 상황의 상태를 보여준다. 카운트 데이터 및 단가 데이터는 장치(1503)에 대응하는 변환 체인의 부분으로 흘러가서, 결국 장치(1503) 상에 디스플레이된 총 비용이 변경된다.
- [0086] 도 15h는, 사용자가 장치(1503) 상의 스크롤 제어부(1516)를 이용해서 최대 단가를 \$987로 하향 조정한, 도 15g의 상태로부터 진전된 상황의 상태를 보여준다. 해당 최대 단가 변경은 장치(1503)와 연계된 변환 체인 부분으로부터 장치(1502)와 연계된 변환 체인 부분으로 흘러가고, Vendor 2에 의해 제공된 몇 개의 장치가 목록에서 빠지게 된다. 사용자가 금액을 다시 상향시키면, 해당 빠진 품목들이 자동으로 다시 목록에 추가될 것이고, 아마도 장치(1502)가 도 15g에 도시된 상태로 되돌려질 것이다.
- [0087] 도 15i는, 해당 환경에 제 4 장치(1504)가 도입된, 도 15g의 상태로부터 진전된 상황의 상태를 보여준다. 예를 들어, 제 4 장치(1504)는 장치의 최대 중량을 디스플레이하는 변환 체인과 연계될 수 있다. 해당 환경에 장치(1504)가 도입되면, 변환 체인의 해당 부분이 기존의 복합 변환 체인과 연결되고, 그에 따라 상기와 같은 비용 데이터를 출력하도록 복합 애플리케이션의 기능이 변경된다. 변환 체인 인스턴스가 연결되면, 최대 중량을 나타내는 데이터는 장치(1504)와 연계된 변환 체인의 부분으로부터 장치(1502)와 연계된 변환 체인의 부분으로 흘러간다. 장치(1502)는 최대 중량보다 무거운 각각의 장치와 연계된 중량초과 경고(1518A, 1518B, 1518C 및 1518D)를 디스플레이하는 것으로 응답한다. 현재, 도 15i에서, 중량은 단지 606 그램이기 때문에, 모든 장치에 상기와 같은 경고가 나열된다.

- [0088] 도 15j는, 사용자가 최대 중량을 2 킬로그램 이상으로 증가시키기 위해 장치(1504) 상의 스크롤 제어부(1516)를 이용한, 도 15i의 상태로부터 진전된 상황의 상태를 보여준다. 최대 중량 변경은 장치(1504)와 연계된 변환 체인의 부분으로부터 장치(1502)에 대응하는 변환 체인의 부분으로 흘러가고, 결국 중량초과 경고(1518C 및 1518D)가 제거된다(중량초과 경고(1518A 및 1518B)는 유지됨).
- [0089] 이것으로 사용자 관점에서의 상황이 종료된다. 도 15k 내지 도 15o는, 저작시에, 저자가 변환을 선언적으로 변경해서 복합 애플리케이션의 기능을 변경할 수 있음을 도시한다. 도 15k에서, 사용자 인터페이스 요소(1520)는 슬라이더(1516)에 의해 표시된 값보다 적은 제조자 권장 소매가를 갖는 해당 장치들을 걸러내는 선언형 변환(declarative transformation)을 포함하는 것으로 도시된다. 도 15l은 이제는 이 제한이 제거되고, 그에 따라 디스플레이된 장치가 이제는 슬라이더(1516)의 값에 의해 걸러내지지 않음을 도시한다.
- [0090] 도 15m에서, 사용자 인터페이스 요소(1521)는 제조자 권장 소매가에 대하여 사용되는 강조 컬러를 규정하는 것으로 도시된다. 도 15n은 강조 컬러가 이제는, 제조자 권장 소매가가 슬라이더 제어부(1516)에 의해 표시된 값을 상회하는지의 여부에 의존함을 도시한다. 상회한다면, 강조 컬러는 예를 들어, 녹색으로 변경된다. 도 15o는, 이 상황에서 슬라이더 제어부(1516)가 증가됨에 따라, 제조자 권장 소매가가 슬라이더 제어부(1516)의 값보다 큰 경우에 장치가 목록에서 제거되는 상태로부터, 제조자 권장 소매가가 (슬라이더(1516)의 값에 비해) 제조자 권장 소매가를 강조 표시하는 상태로, 복합 애플리케이션의 거동이 변경했음을 도시한다.
- [0091] 상술한 바와 같이, 프리젠테이션 서비스(1150)가 프리젠테이션 노드로부터의 출력에 대하여 행할 수 있는 한 가지 변환은 해당 출력을 영화화하는 것이다. 영화화는 하나 이상의 영화적 기법의 이용을 의미한다. 영화성의 레벨은, 각각의 영화적 기법이 이용되는 정도와 결합된, 이용되는 영화적 기법들의 수를 의미한다.
- [0092] 영화적 기법은 사용자 인터페이스 상에서의 사용자 인터페이스 요소의 움직임의 정도를 조정하는 것을 포함한다. 사용자 인터페이스 내로의 및 사용자 인터페이스 회로의 사용자 인터페이스 요소의 움직임은 사용자가 사용자 인터페이스를 표준 사용자 인터페이스보다 더 영화같이 취급하게 하고, 그에 따라 사용자가 사용자 인터페이스의 다수의 양태를 가볍게 이해할 수 있게 된다. 사용자 인터페이스 요소의 움직임이 적거나 또는 심지어 정지해 있으면, 사용자가 해당 요소와 상호작용해서 더욱 명확하게 초점을 맞추는 것이 허용된다. 따라서, 사용자가 사용자 인터페이스 요소와 상호작용하겠다는 의사를 표현하면, 해당 상호작용을 허용하기 위해 사용자 인터페이스 요소의 일반적인 움직임이 감소된다.
- [0093] 다른 영화적 기법은 제시되어 있는 토픽에 관련된 배경을 이용하는 것이다. 예를 들어, 플레이에 있어서, 해당 플레이의 설정을 전달하기 위해 배경이 이용된다. 토픽이 변경되거나, 또는 이전의 배경이 얼마간 존재하고 있는 경우에는, 배경이 주기적으로 변경될 수 있다. 흥미로운 배경의 사용자는 사용자가 사용자 인터페이스를 전체로서 다수의 레이어(layer)로 보게 한다. 인간은 선사 시대 아래로 다수의 레이어로 된 장면을 이해해 오고 있다. 예를 들어, 우리가 수평선을 바라보면, 우리는 그것이 가까이 있는 들판인지 또는 멀리 떨어진 산봉우리인지, 우리가 대상물을 똑바로 쳐다 보고 있는지, 또는 대상물이 약간 우리의 주변 시야에 있는지와 같은 다양한 양태에 초점을 맞출 수 있다.
- [0094] 또 다른 영화적 기법은 사용자 인터페이스의 특정한 양태에 강조를 제공하기 위해 조명 및 강조하기를 이용하는 것이다. 점프 컷(jump cuts)을 이용하는 것은, 다른 방식으로 정보를 제시하기 위해 사용자 인터페이스의 다수의 레이어가 변경되는 영화적 기법이다. 영화적 기법은 논의의 토픽에 어울리는 합성된 사용자 인터페이스 요소(예컨대, 배경, 또는 전경에 있는 다른 사용자 인터페이스 요소)를 이용하는 것일 수 있다. 다른 영화적 기법은 제공되어 있는 정보의 분위기 또는 토픽에 적합한 배경 음악을 이용하는 것이다. 빨리감기 또는 되감기 특징을 제공하는 것도 역시 디지털 비디오 레코더를 모방한 영화적 기법이다. 설정(2차원 또는 3차원 여부)에 대한 관점의 패닝(panning) 또는 변경도 역시 영화적 기법이다. 영화성은 특정한 사용자 인터페이스 요소에 초점을 맞추는 것을 허용함으로써(예컨대, 배경을 흐리게 하거나 또는 제거하는 것에 의해, 또는 배경을 정지시키는 것에 의해) 감소될 수 있다.
- [0095] 하나의 디스플레이가 변환 체인의 출력을 영화적인 방식으로 디스플레이하고 있을 수 있지만, 다른 사용자는 사용자 자신의 개인 장치로 정보와의 더 많은 상호작용을 수행하게 될 수 있고, 그에 따라 동일한 세션 정보의 코어 스트림에 영화성이 덜 적용된다는 점에 유의한다. 그에 따라, 일반적으로는 사용자들이 세션 정보의 전체적인 영화적 뷰에 함께할 수 있지만, 개개인들은 원할 때 정보와 상호작용해서 자신의 경험을 세션 정보와 커스터마이징할 수 있다.
- [0096] 도 16은 정보 사용자 인터페이스 요소와 애플리케이션 사용자 인터페이스 요소를 연속 스트림 형태로 합성하는

방법(1600)의 흐름도를 도시한다. 방법(1600)은, 예를 들어, 세션 정보가 렌더링되어야 할 때 프리젠테이션 서비스(1150)에 의해 수행될 수 있다. 프리젠테이션 서비스는 세션 정보를 수신하고(동작(1601)), 해당 세션 정보에 적용할 영화성의 레벨을 결정한다(동작(1602)). 이후, 프리젠테이션은 영화성을 적용하고(동작(1603)), 영화적인 출력이 렌더링되게 한다(동작(1604)). 그와 같이 행함에 있어서, 프리젠테이션 서비스는 세션이 제시되고 방법(1600)이 계속적으로 수행됨에 따라 세션 정보에 적용될 영화성의 레벨을 조정할 수 있다.

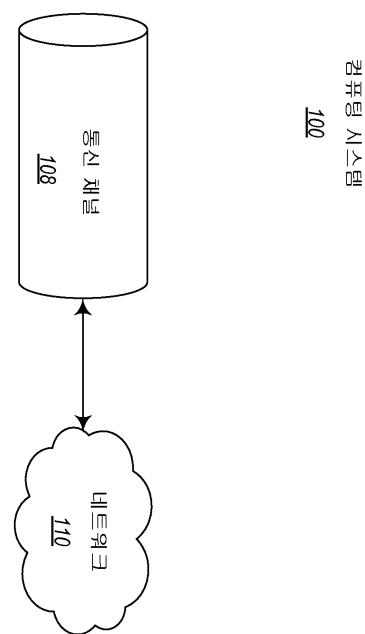
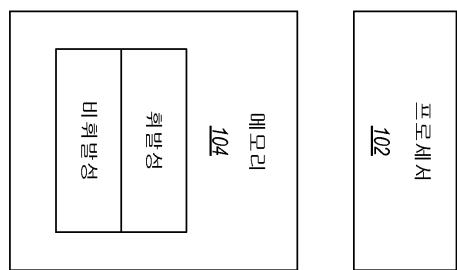
[0097] 사용자는 (예컨대, 렌더링에서 보여지는 객체에 관한 정보를 주입하거나, 스트림의 속도 또는 템포를 변경하거나("xxx가 발견/도달될 때까지... 서둘러 이 부분을 끝내라"), 또는 유사한 클립/프레임/패턴/어커런스(occurrences)를 찾기 위해) 영화화된 렌더링과 상호작용할 수 있다. 그래서, 영화화의 일부는 사용자가 상호작용할 수 있는 핫스팟을 생성하는 것, 사용자와의 디아일로그를 생성하는 것, 또는 사용자가 상호작용할 수 있는 시기 및 방법에 대한 임의의 시각적 또는 청각적 큐(cue)를 제공하는 것을 포함할 수 있다.

[0098] 그에 따라, 본 명세서에 기술된 원리는 다수의 장치 및 다수의 사람들이 존재하는 환경에서 영화적 기법의 이용을 제공한다. 영화성의 레벨은 장치마다 다를 수 있으며, 사용자가 정보와의 상호작용에 초점을 맞출 때 경시적으로 변경될 수 있다. 한편, 보다 높은 영화성의 레벨을 이용하면, 사용자가 다량의 데이터에 대한 종합적인 평가를 할 수 있게 된다.

[0099] 본 발명은 그 정신 또는 기본 특성으로부터 벗어남 없이 다른 특정한 형태로 구현될 수 있다. 기술된 실시예는 모든 점에 있어서 제한이 아닌 예시로서만 고려되어야 한다. 따라서, 발명의 범위는 전술한 상세한 설명이 아닌 첨부된 청구항에 의해서 나타내진다. 청구항의 등가물의 의미 및 범위 내의 모든 변경은 그 범위 내에 들어간다.

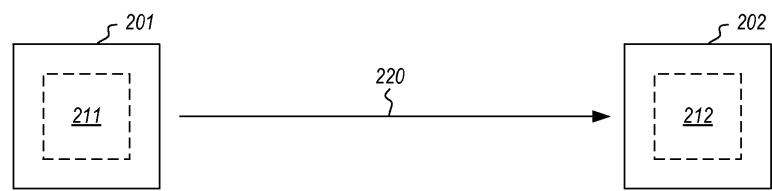
도면

도면1



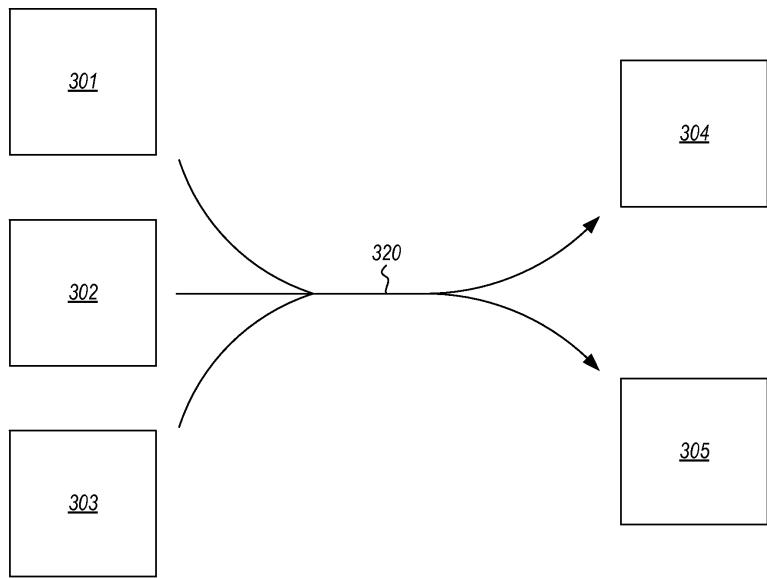
도면2

200

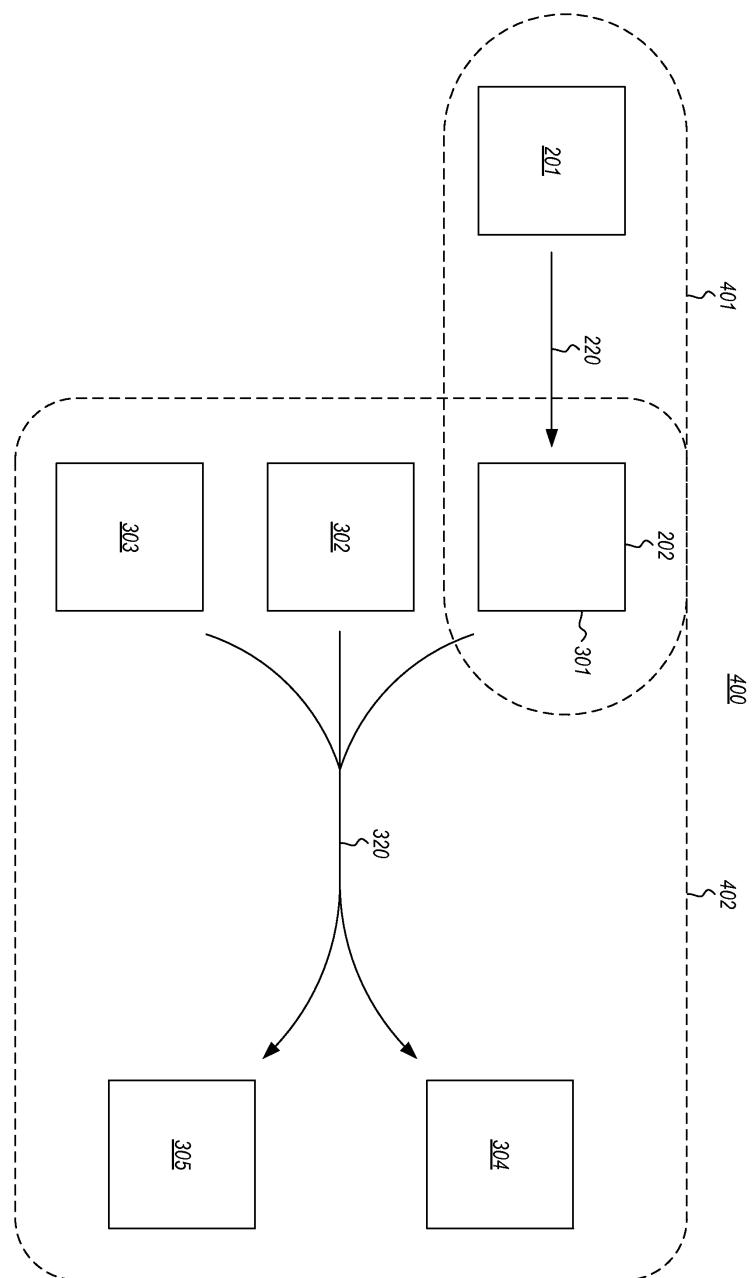


도면3

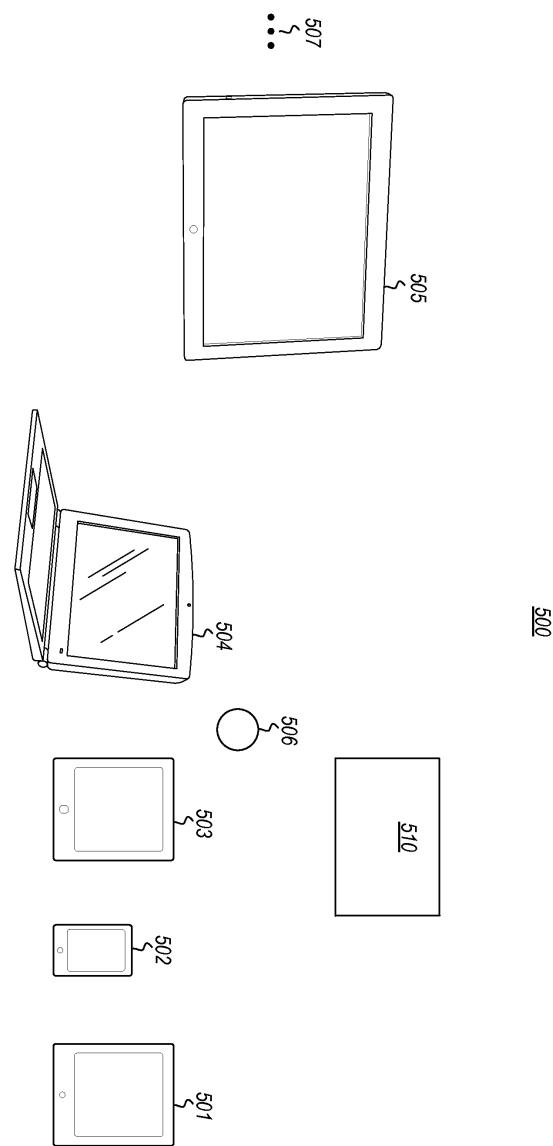
300



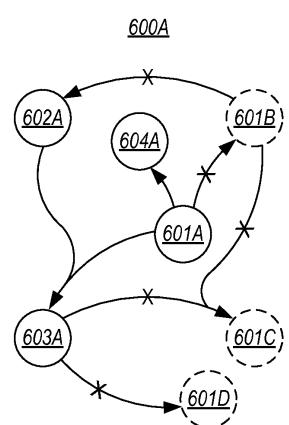
도면4



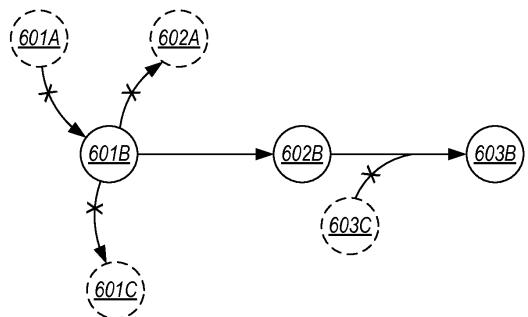
도면5



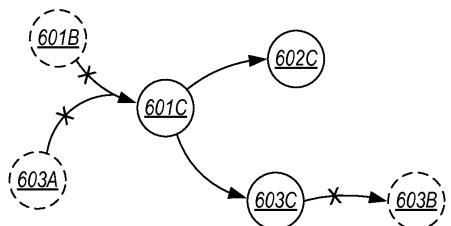
도면6a



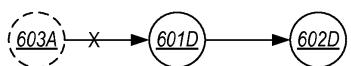
## 도면6b

600B

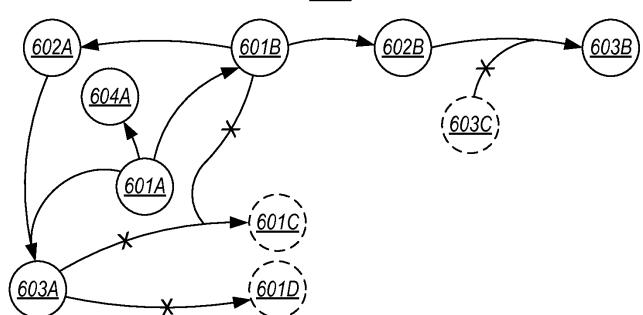
## 도면6c

600C

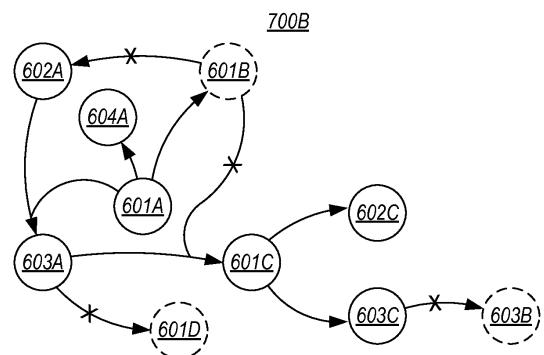
## 도면6d

600D

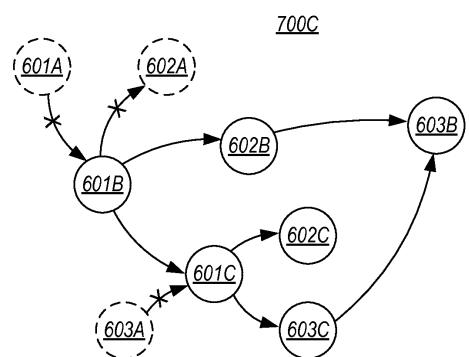
## 도면7a

700A

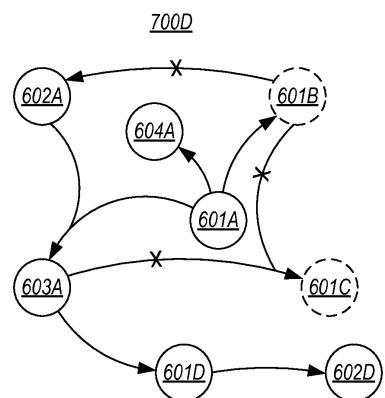
도면7b



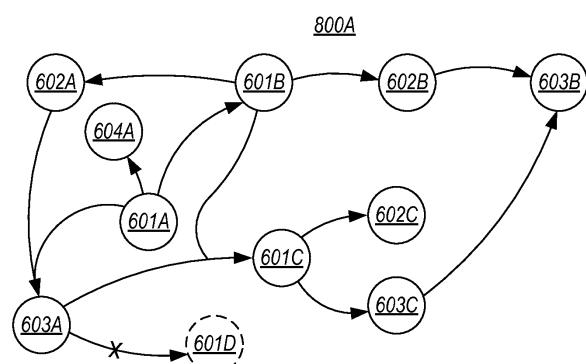
도면7c



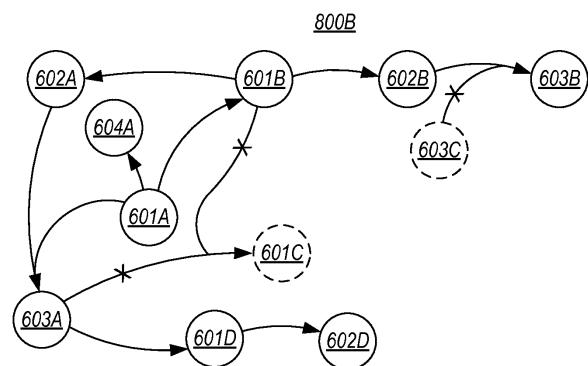
도면7d



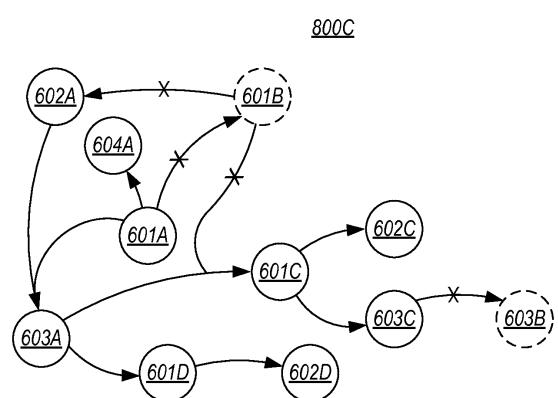
도면8a



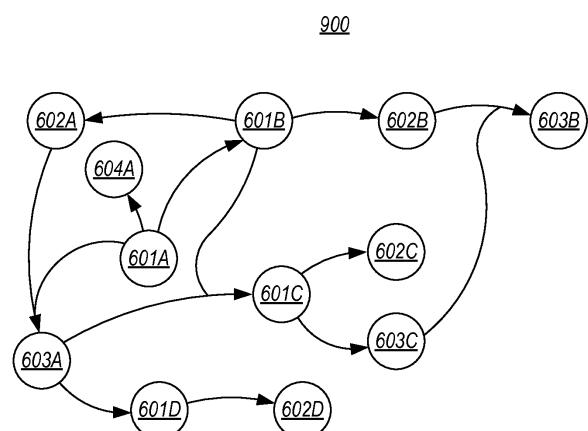
## 도면8b



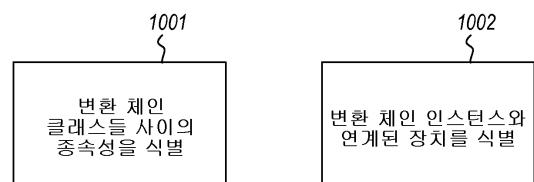
## 도면8c

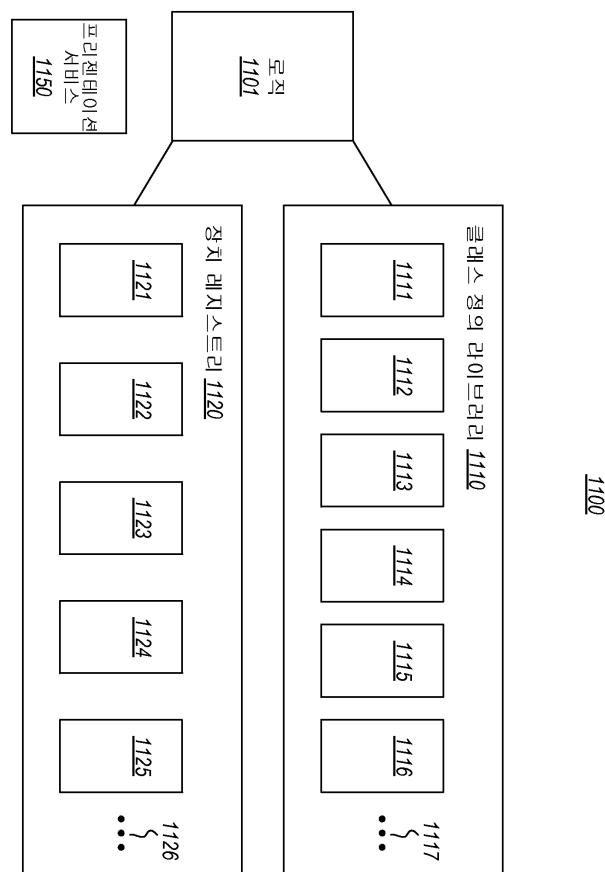
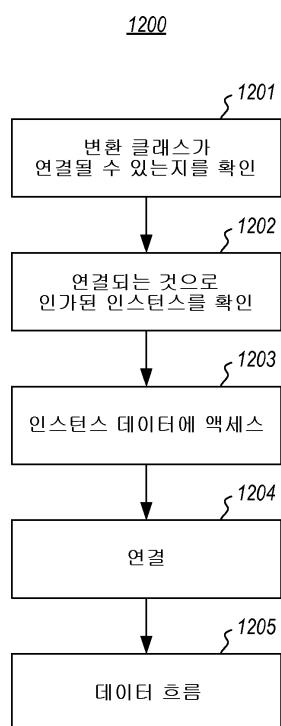


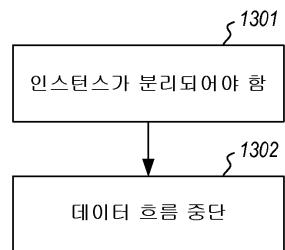
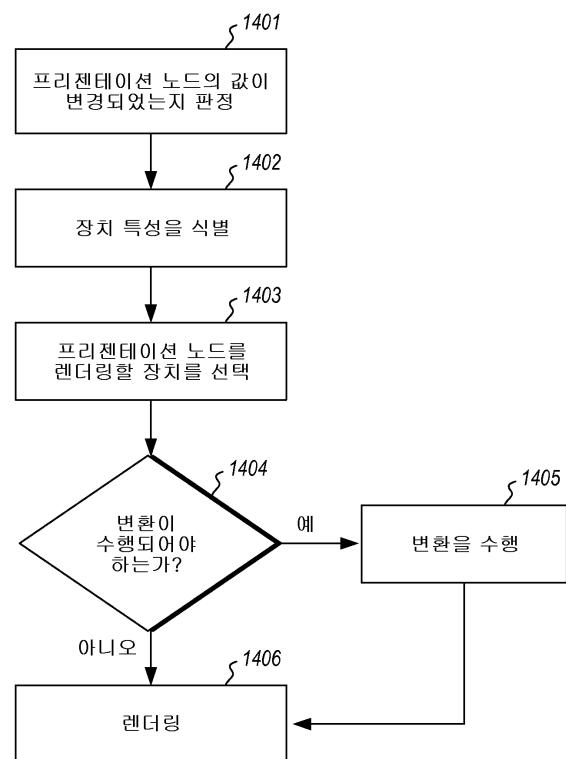
## 도면9



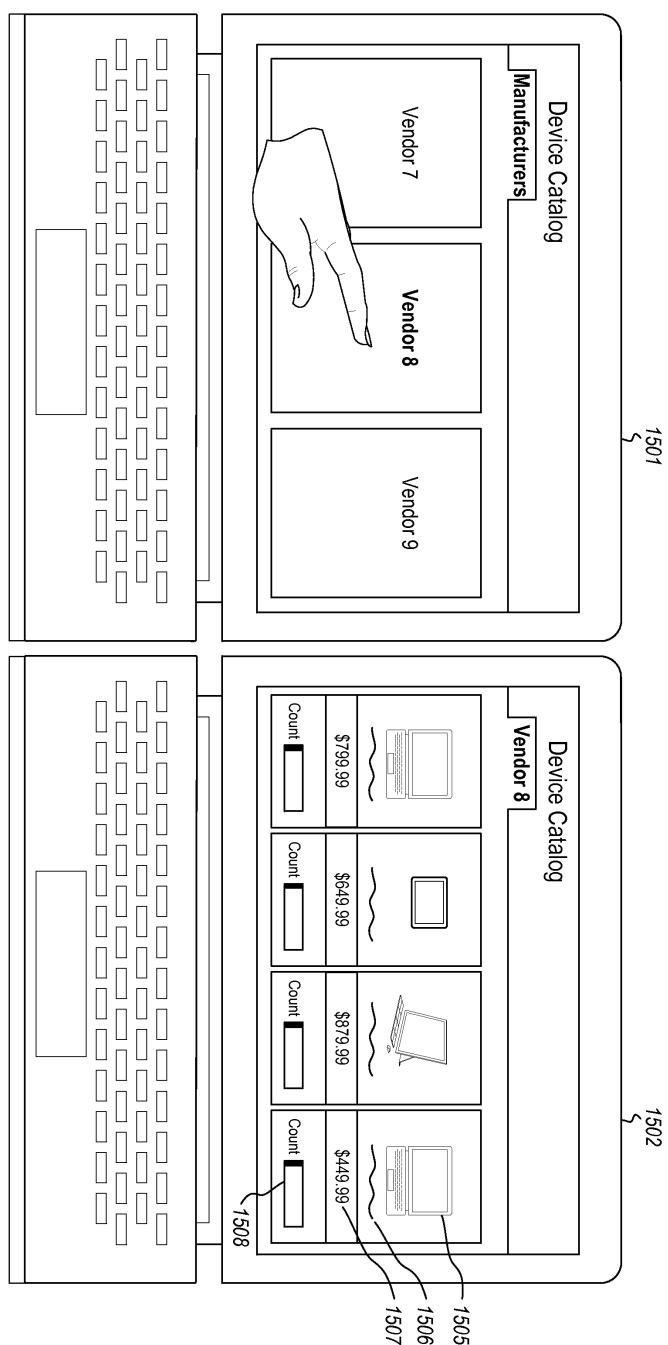
## 도면10

1000

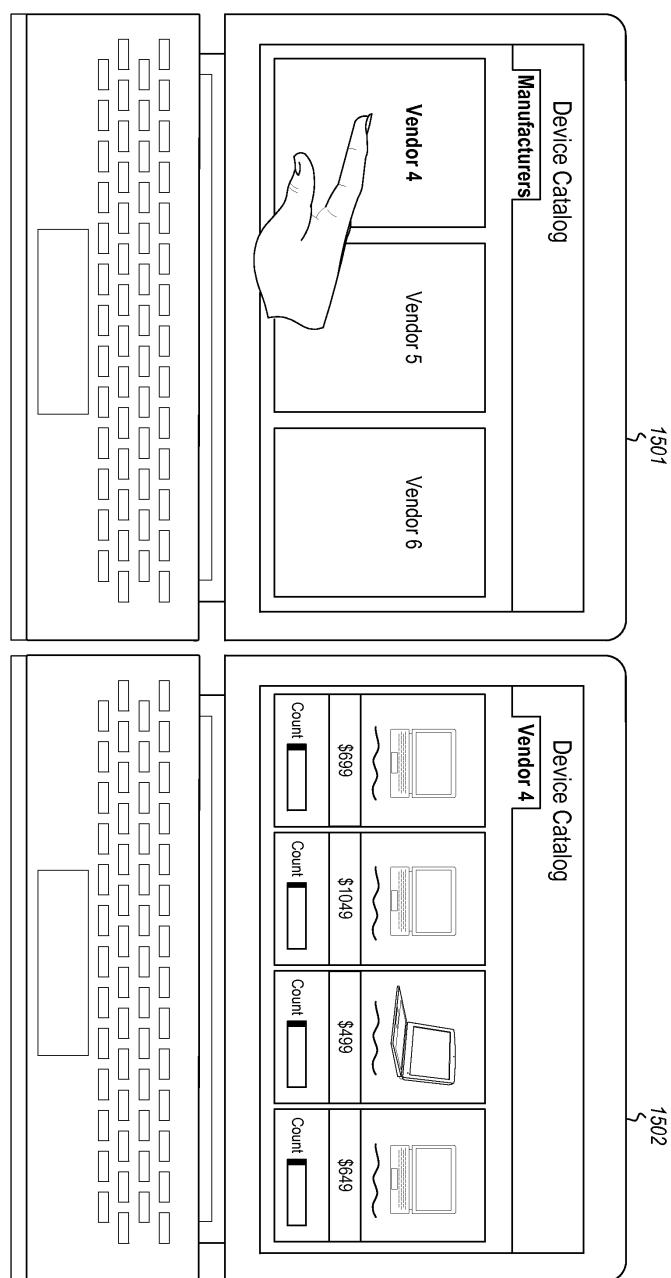
**도면11**1100**도면12**

**도면13**1300**도면14**1400

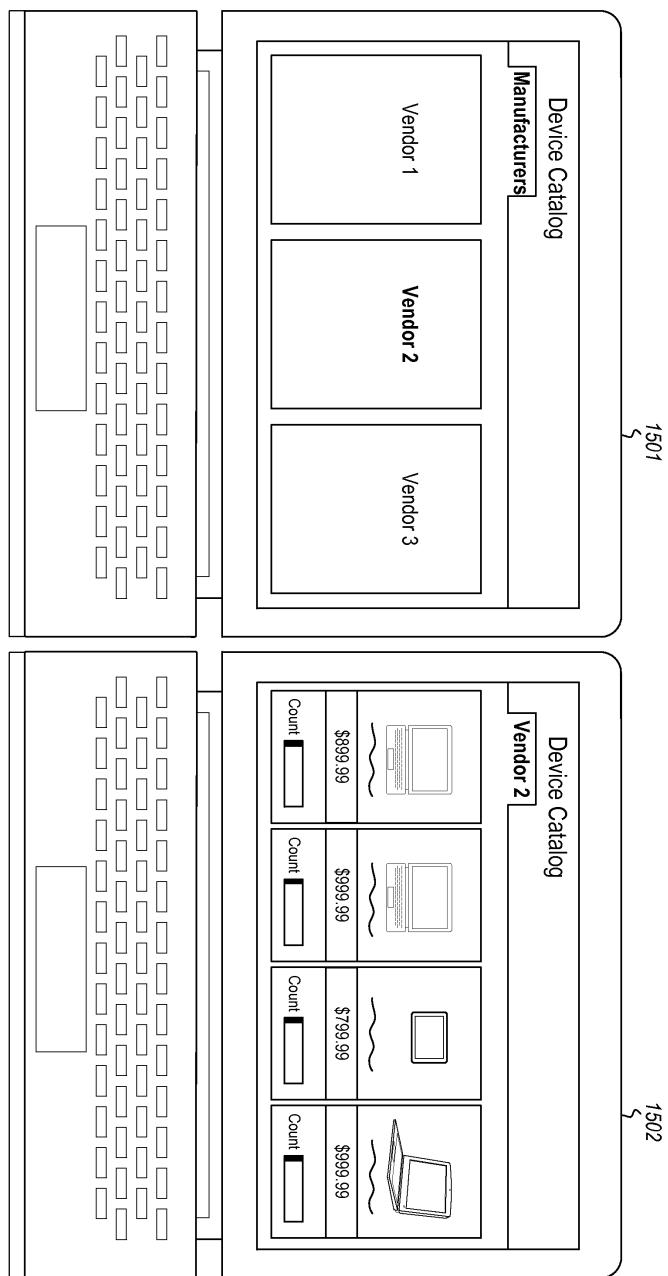
도면15a



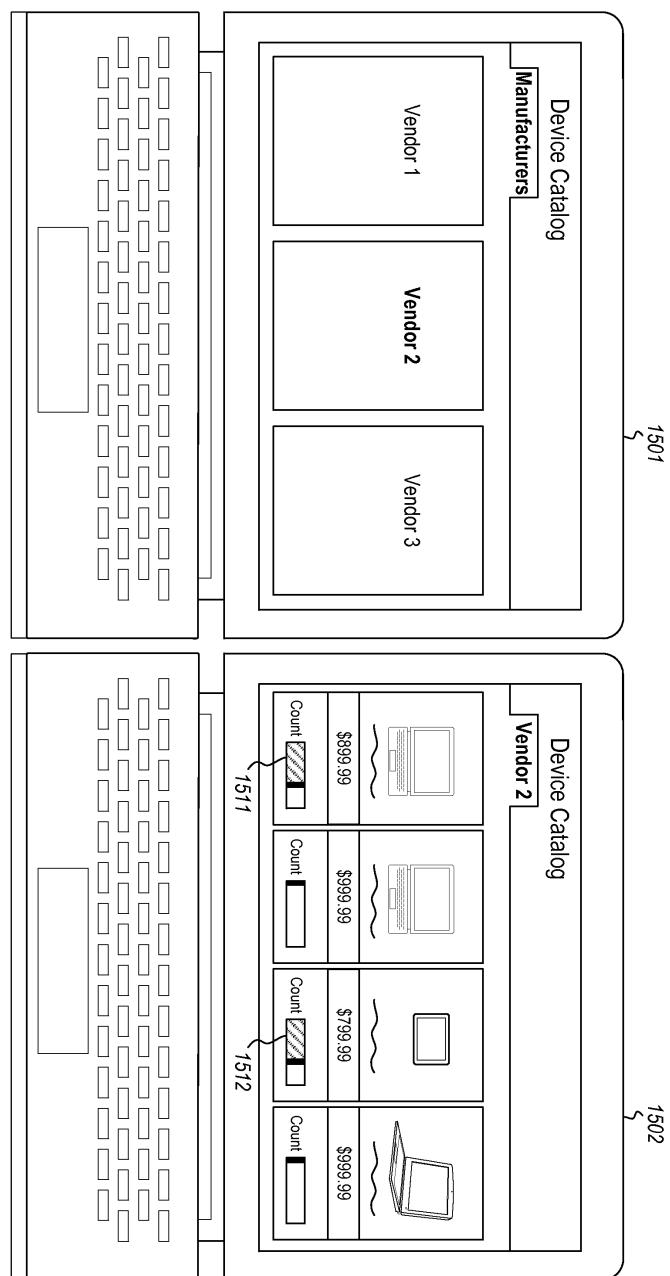
도면15b



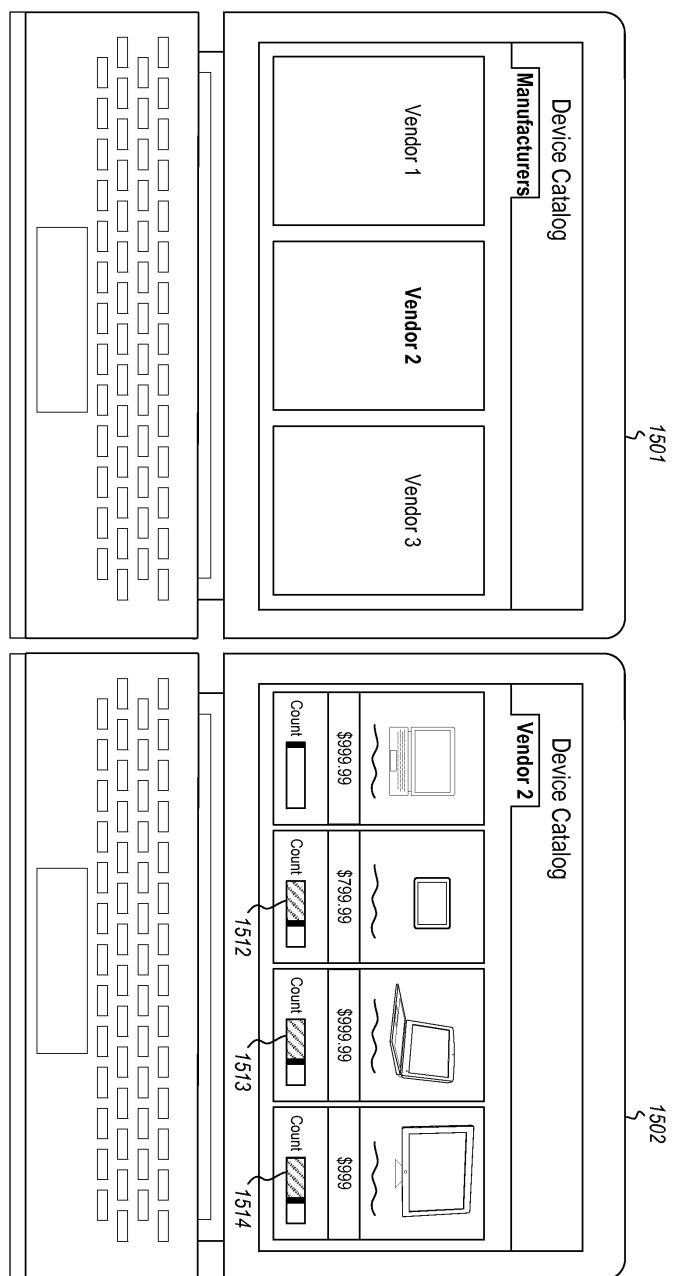
도면15c



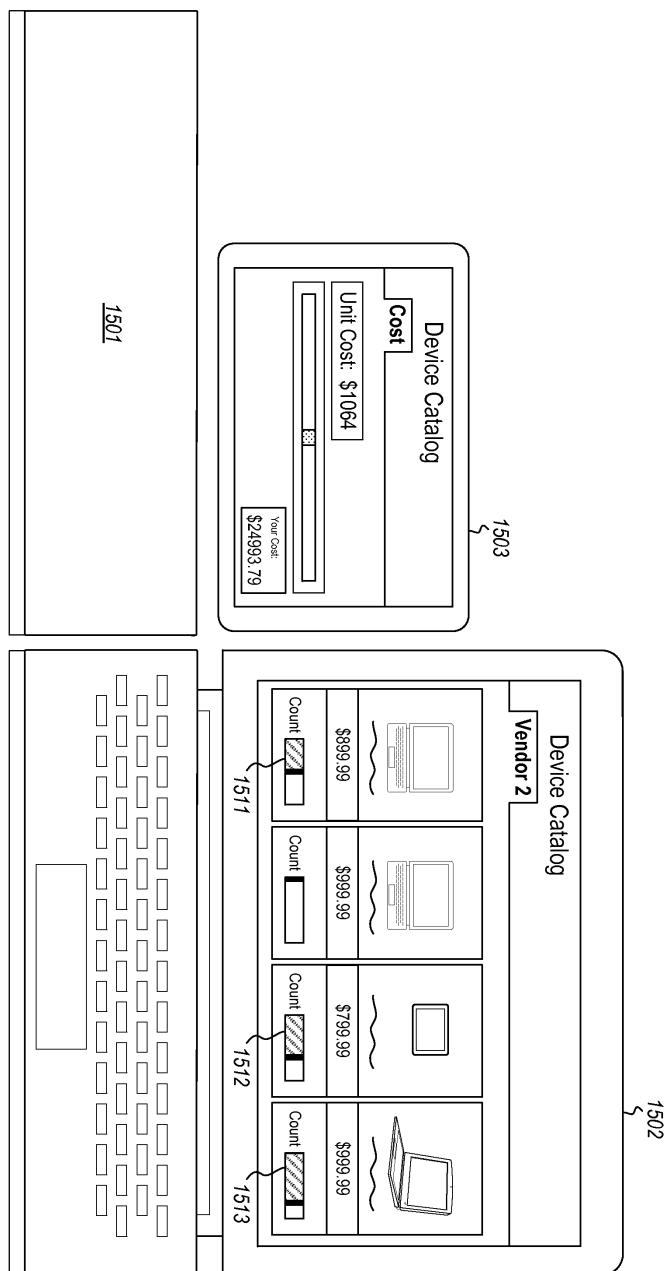
도면15d



도면15e

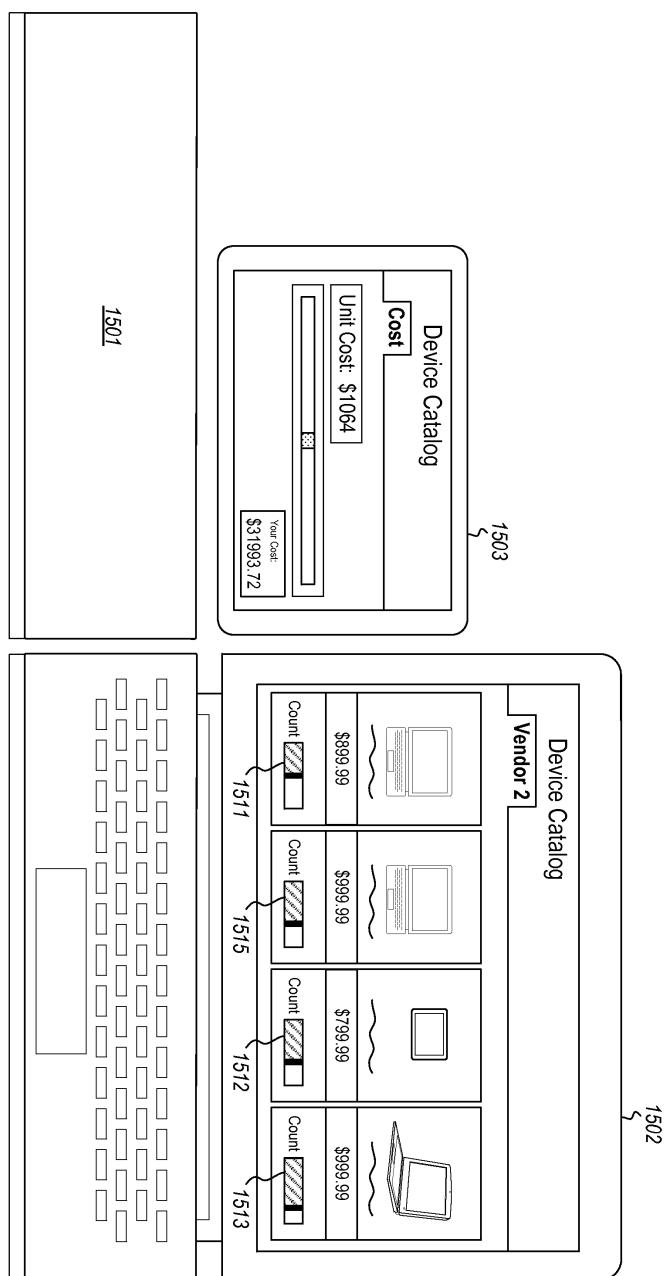


도면15f

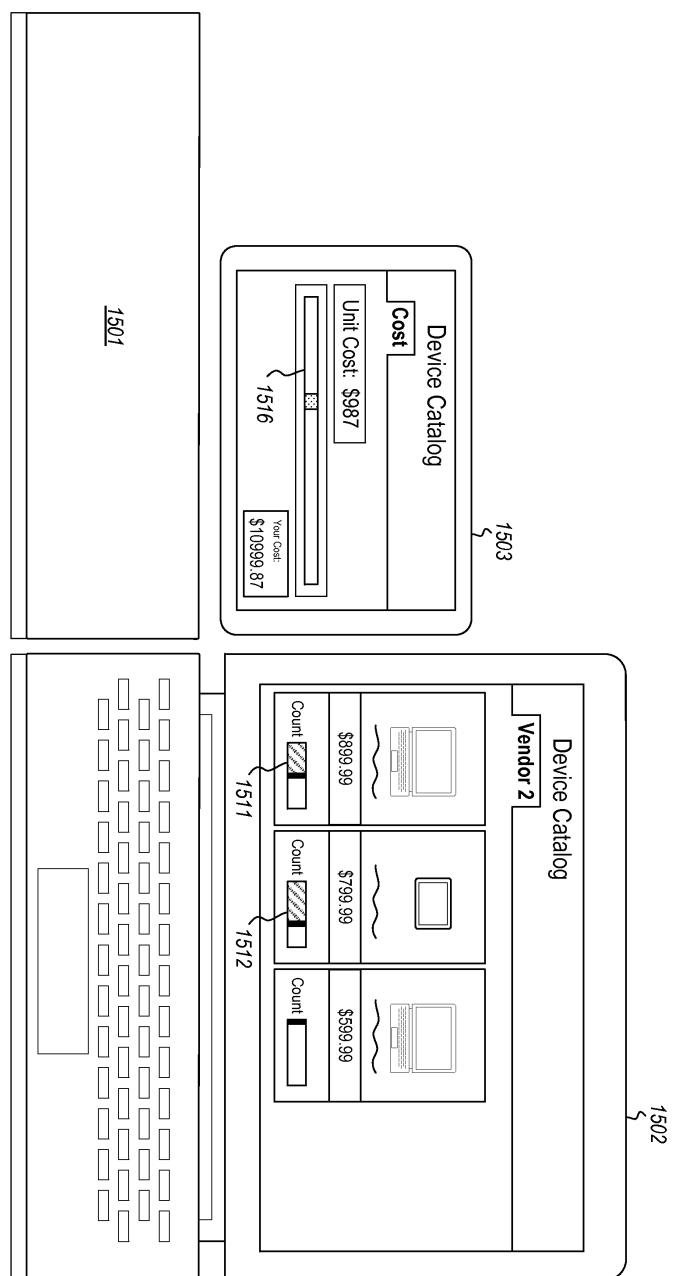


1501

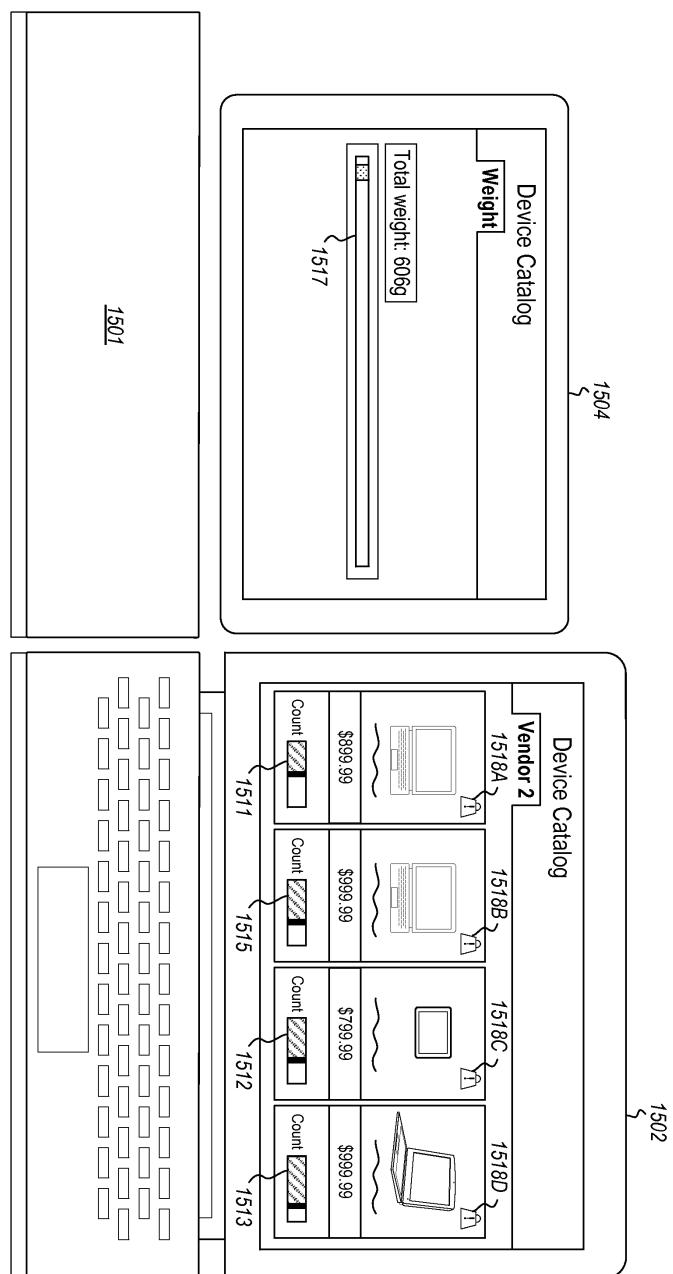
도면15g



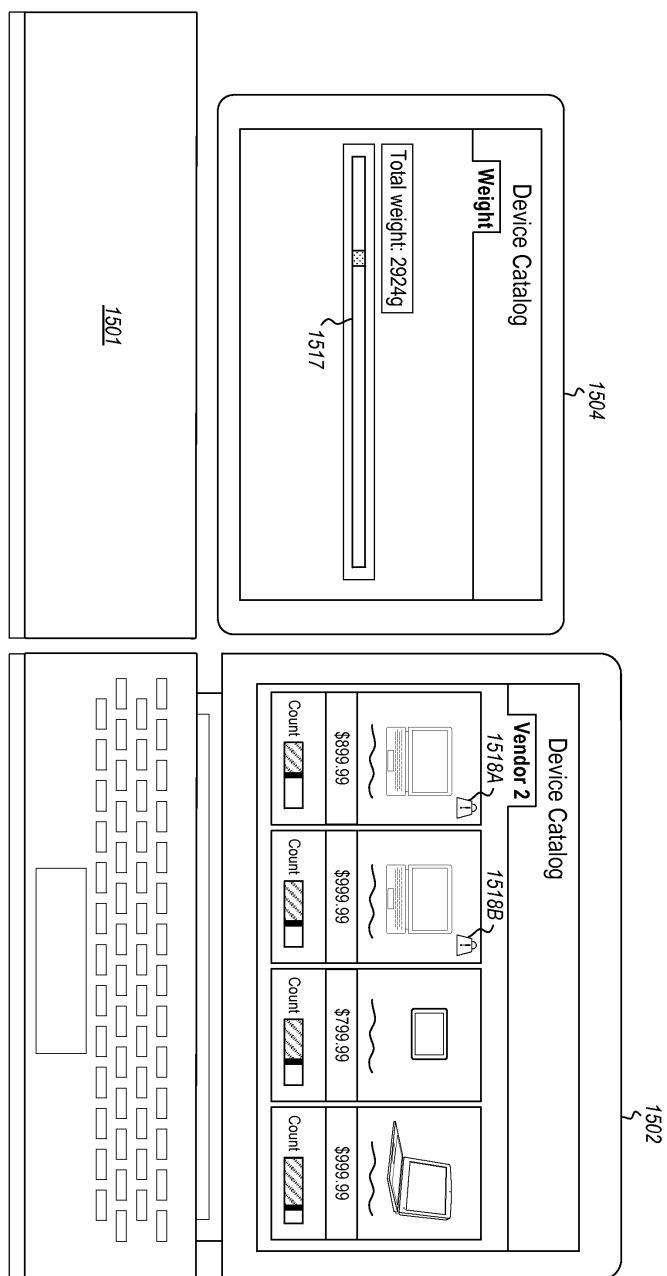
도면15h



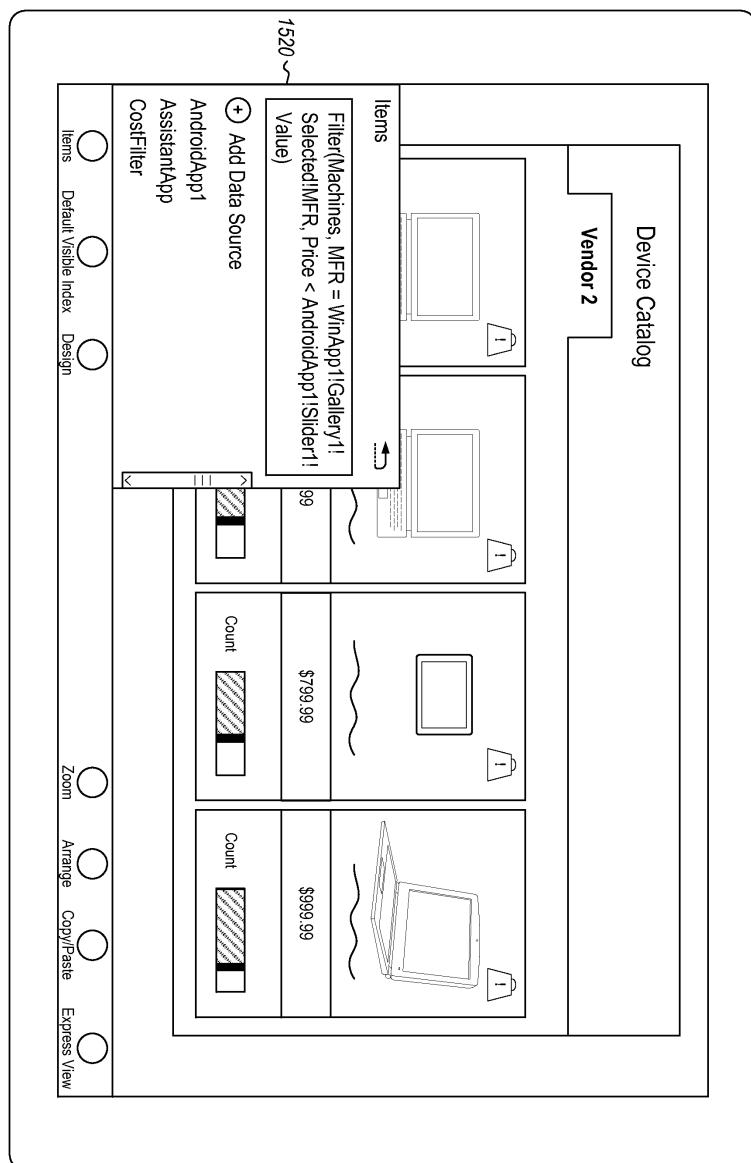
도면15i



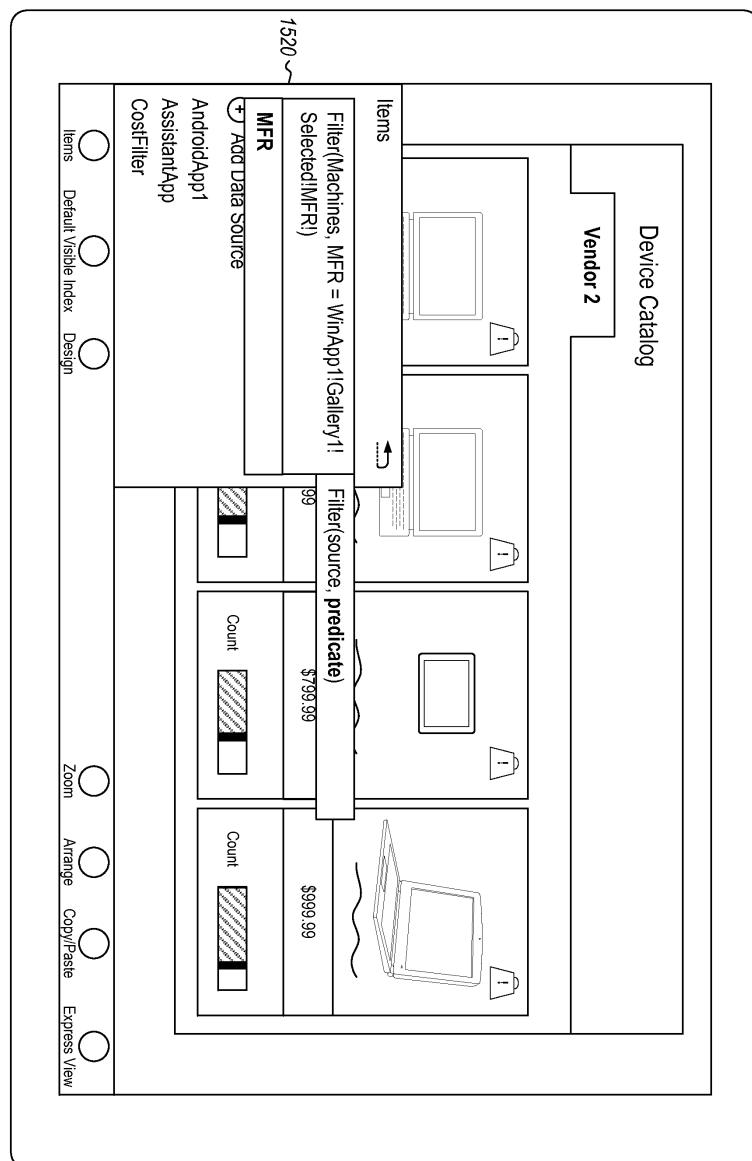
도면15j



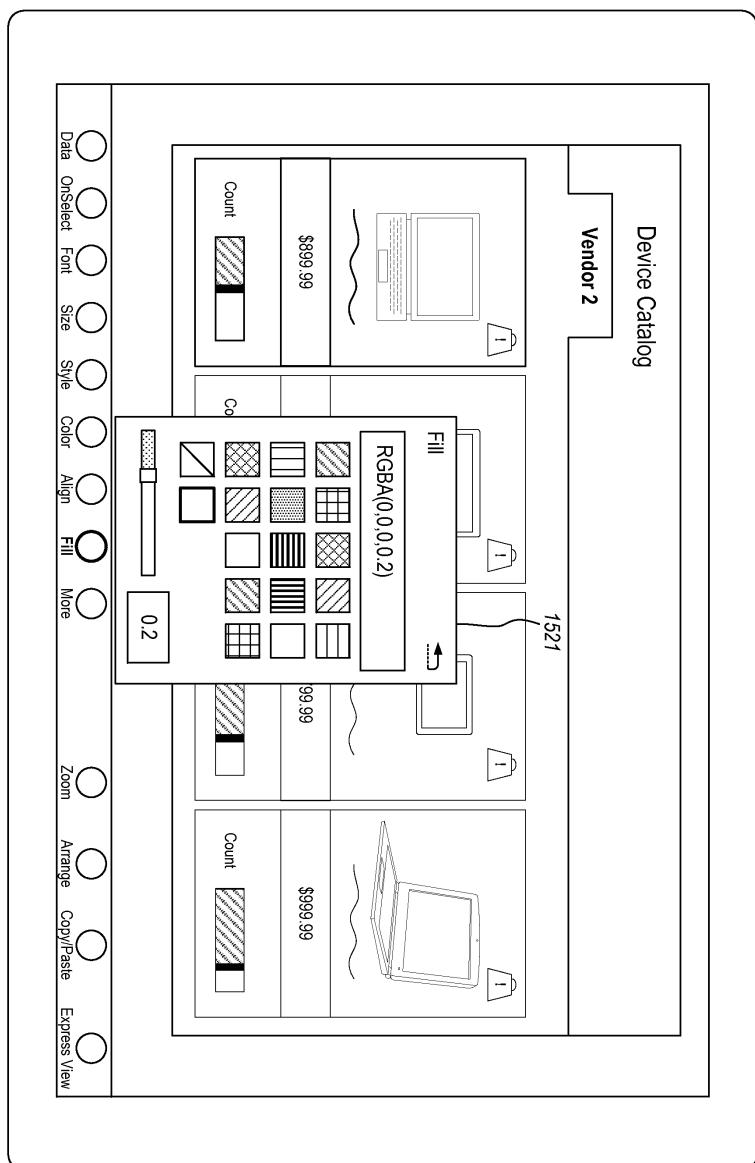
도면15k



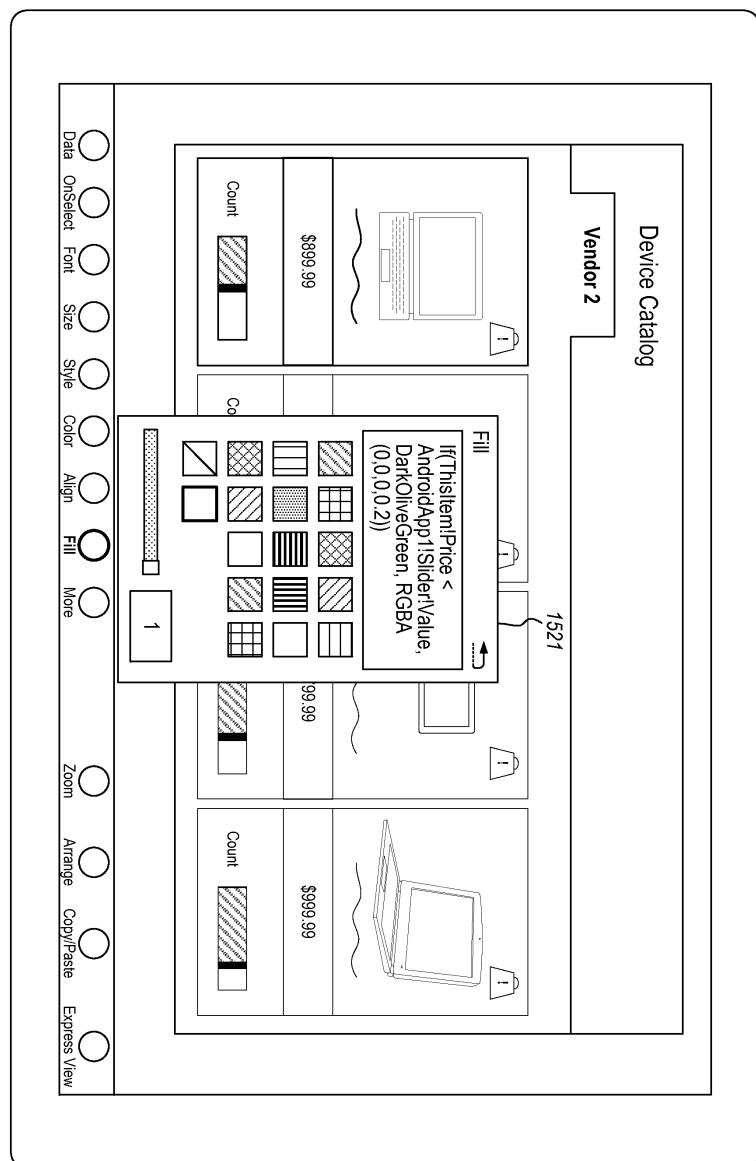
도면151



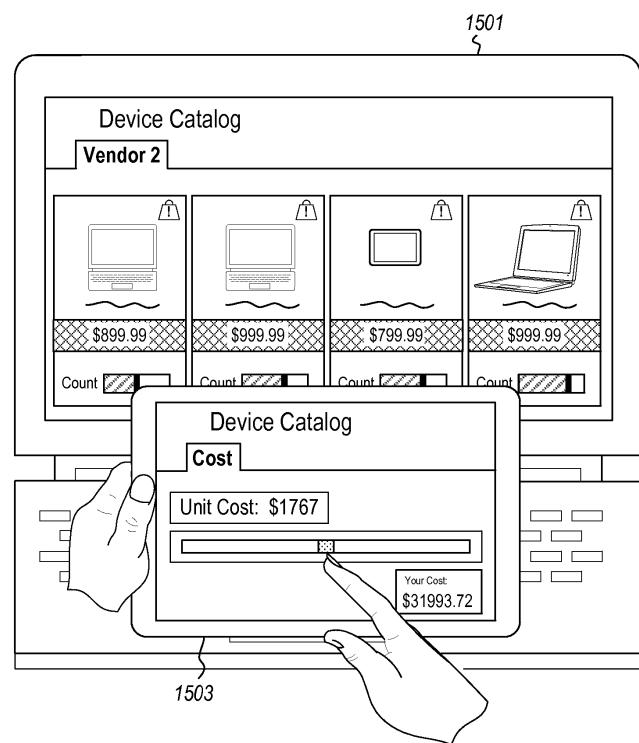
도면15m



도면15n



## 도면15o



## 도면16

1600