(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0017050
(43) 공개일자 2016년02월15일

(51) 국제특허분류(Int. Cl.)
G06Q 10/10 (2012.01) *H04L 12/58* (2006.01)
(52) CPC특허분류(Coo. Cl.)
G06Q 10/101 (2013.01)
G06Q 10/103 (2013.01)
(21) 출원번호 10-2016-7000174
(22) 출원일자(국제) 2014년02월25일
심사청구일자 없음
(85) 번역문제출일자 2016년01월05일
(86) 국제출원번호 PCT/US2014/018375
(87) 국제공개번호 WO 2014/197015
국제공개일자 2014년12월11일
(30) 우선권주장
61/832,106 2013년06월06일 미국(US)
14/090,830 2013년11월26일 미국(US)

(71) 출원인
하워쓰, 인크.
미국 49423 미시건 홀랜드 원 하워쓰 센터
(72) 발명자
폴레이 데이브 엠.
미국 워싱턴 98122 시애틀 수트 202 이. 파인 900
피어슨 아담
미국 워싱턴 98122 시애틀 수트 202 이. 파인 900
엔트레킨 데미안
미국 미시간 49423 홀란드 원 하워쓰 센터
(74) 대리인
박장원

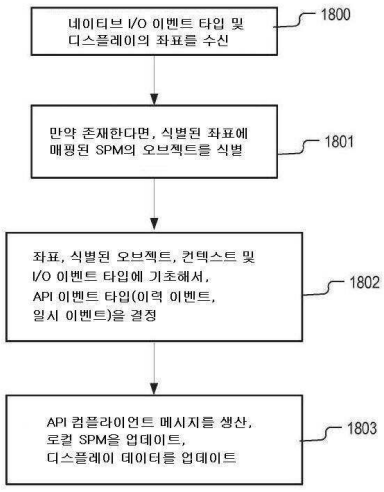
전체 청구항 수 : 총 75 항

(54) 발명의 명칭 **공간 이벤트 맵을 포함하는 협업 시스템**

(57) 요 약

공간 이벤트 맵 시스템은 작업공간에 이벤트들을 위치시키는 공간 이벤트 맵을 유지하는 서버측 데이터 프로세서를 포함한다. 공간 이벤트 맵은 이벤트들의 로그, 상기 작업공간의 이벤트의 그래픽 타겟의 위치 및 시간을 갖는 입력들을 포함한다. 상기 시스템은 이벤트, 작업공간의 이벤트의 그래픽 타겟의 위치, 및 시간을 포함하는 메시지들을 클라이언트측 네트워크 노드들에 송신하고, 그래픽 타겟을 생성 또는 수정하는 이벤트들을 식별하는 메시지들을 수신하고, 대응하는 입력들을 상기 이벤트들의 로그에 추가기 위한 로직을 포함한다. 상기 이벤트들은 다른 클라이언트측 네트워크 노드들에 송신되고 대응하는 이력 이벤트들에 대한 로그에 추가되는 이력 이벤트들과, 상기 로그에 대응하는 입력들을 추가함이 없이 다른 클라이언트측 네트워크 노드들에 송신되는 일시 이벤트들을 포함한다.

대 표 도 - 도18



(52) CPC특허분류(Coo. Cl.)
H04L 51/04 (2013.01)

특허청구의 범위

청구항 1

작업공간(workspace)에 이벤트들을 위치시키는(locate) 공간 이벤트 맵(spatial event map)을 유지하고 협업 디스플레이들(collaborative displays)에 대한 통신 자원들을 제공하는 데이터 프로세서를 포함하는 공간 이벤트 맵 시스템.

청구항 2

제 1항에 있어서,

상기 공간 이벤트 맵은 이벤트들의 로그, 상기 작업공간의 이벤트의 그래픽 타겟의 위치 및 시간을 갖는 상기 로그의 입력들을 포함하는 것을 특징으로 하는 시스템.

청구항 3

제 2항에 있어서,

이벤트, 작업 공간의 이벤트의 그래픽 타겟의 위치 및 시간을 포함하는 메시지들을 클라이언트측 네트워크 노드들에 송신하기 위한 로직을 포함하는 것을 특징으로 하는 시스템.

청구항 4

제 2항에 있어서,

작업공간의 위치 및 시간을 갖는 그래픽 타겟을 생성 또는 수정하는 이벤트들을 식별하는 메시지들을 수신하고 이벤트들의 로그에 대한 대응하는 입력을 추가하기 위한 로직을 포함하는 것을 특징으로 하는 시스템.

청구항 5

제 2항에 있어서,

작업공간의 위치 및 시간을 갖는 그래픽 타겟을 생성 또는 수정하는 이벤트들을 식별하는 클라이언트측 네트워크 노드로부터의 메시지들을 수신하기 위한 로직을 포함하고, 상기 이벤트들은 이력 이벤트들(history events) 및 일시 이벤트들(ephemeral events)을 포함하는 것을 특징으로 하는 시스템.

청구항 6

제 5항에 있어서,

로그에 대응하는 입력들을 추가함이 없이 다른 클라이언트측 네트워크 노드들에 일시 이벤트들을 송신하고, 다른 클라이언트측 네트워크 노드들에 이력 이벤트들을 송신하고, 대응하는 이력 이벤트들에 대한 로그에 입력들을 추가하기 위한 로직을 포함하는 것을 특징으로 하는 시스템.

청구항 7

제 1항에 있어서,

상기 작업공간은 경계가 정해지지 않은 가상 영역을 포괄하는 것을 특징으로 하는 시스템.

청구항 8

제 1항에 있어서,

상기 작업공간은 실제적으로 한정되지 않은(practically unlimited) 가상 영역을 포괄하는 것을 특징으로 하는 시스템.

청구항 9

시스템으로서,

서버측 네트워크 노드를 포함하고, 상기 서버측 네트워크 노드는 작업공간에 위치들을 갖는 그래픽 타겟들에 관한 이벤트들의 로그와, 상기 이벤트의 그래픽 타겟의 작업공간의 위치와, 이벤트의 시간 및 상기 이벤트의 그래픽 타겟의 타겟 식별자를 포함하는 상기 로그의 입력들을 저장하는 메모리를 포함하고, 상기 네트워크 노드는:

복수 개의 활성 클라이언트측 네트워크 노드들에 링크들을 설정하고, 상기 작업공간에 위치들을 갖는 그래픽 타겟들의 수정 및 생성에 관한 이벤트들을 식별하는 메시지들을 수신하고, 상기 메시지들에 응답하여 상기 로그에 이벤트들을 추가하고, 다른 활성 클라이언트측 네트워크 노드들에 특정한 클라이언트측 네트워크 노드로부터 수신된 메시지들에서 식별된 이벤트들에 관한 메시지들을 배포하기 위한 로직을 포함하는 것을 특징으로 하는 시스템.

청구항 10

제 9항에 있어서,

상기 로직은:

클라이언트측 네트워크 노드들에 상기 로그의 부분들을 전달하는 메시지를 송신하고, 상기 작업공간에 위치들을 갖는 그래픽 타겟들에 관한 이벤트들을 식별하는 데이터를 전달하는 클라이언트측 네트워크 노드들로부터 메시지들을 수신하기 위한 애플리케이션 프로그램 인터페이스를 포함하는 것을 특징으로 하는 시스템.

청구항 11

제 9항에 있어서,

상기 애플리케이션 프로그램 인터페이스는 하나의 클라이언트측 네트워크 노드로부터 다른 클라이언트측 네트워크 노드에 수신된 이벤트들을 배포하기 위한 프로세스를 포함하는 것을 특징으로 하는 시스템.

청구항 12

제 9항에 있어서,

상기 이벤트들은 상기 로그에 저장되고 다른 클라이언트측 네트워크 노드들에 배포될 이벤트의 제1 클래스 및 다른 클라이언트측 네트워크 노드들에 배포되지만 로그에 저장되지는 않을 이벤트의 제2 클래스를 포함하는 것을 특징으로 하는 시스템.

청구항 13

제 9항에 있어서,

상기 그래픽 타겟들은 텍스트 및 스트로크들(strokes)을 포함하는 것을 특징으로 하는 시스템.

청구항 14

제 9항에 있어서,

상기 로그의 입력들은 디스플레이에 그래픽 타겟을 렌더링하기 위해 사용되는 그래픽 구성물들(graphical constructs)을 식별하는 파라미터(예: url 또는 실제 파일)를 포함하는 것을 특징으로 하는 시스템.

청구항 15

제 9항에 있어서,

상기 로그의 입력들은 상기 이벤트와 관련된 사용자를 식별하는 파라미터를 포함하는 것을 특징으로 하는 시스템.

청구항 16

제 9항에 있어서,

상기 작업공간은 경계가 정해지지 않은 가상 영역을 포괄하는 것을 특징으로 하는 시스템.

청구항 17

제 9항에 있어서,

상기 작업공간은 실제적으로 한정되지 않은 가상 영역을 포괄하는 것을 특징으로 하는 시스템.

청구항 18

시스템으로서,

클라이언트측 네트워크 노드를 포함하고, 상기 클라이언트측 네트워크 노드는 물리적 디스플레이 공간, 사용자 입력 디바이스, 프로세서 및 통신 포트를 포함하며, 상기 클라이언트측 네트워크 노드는:

서버측 네트워크 노드와 링크를 설정하고;

상기 서버측 네트워크 노드로부터 작업공간의 위치들을 갖는 그래픽 타겟들에 관한 이벤트들의 공간 이벤트 로그와, 이벤트의 그래픽 타겟의 작업공간의 위치, 상기 이벤트의 시간 및 상기 그래픽 타겟의 타겟 식별자를 포함하는 상기 로그의 입력들 중 적어도 일부를 검색하고;

상기 물리적 디스플레이 공간의 디스플레이 가능한 영역을 상기 작업공간 내의 매핑된 영역에 매핑하고, 상기 매핑된 영역 내에 상기 공간 이벤트 로그의 입력들을 식별하고, 상기 디스플레이 가능한 영역에 식별된 입력들에 의해 식별된 그래픽 타겟들을 렌더링하며;

상기 디스플레이 가능한 영역 내에 디스플레이되는 그래픽 타겟들의 수정 및 생성에 관한 이벤트들을 생성하는 사용자 입력 디바이스로부터 입력 데이터를 승인하고, 상기 서버측 네트워크 노드에 상기 이벤트들에 기초한 메시지들을 송신하도록 로직으로 구성되는 것을 특징으로 하는 시스템.

청구항 19

제 18항에 있어서,

상기 로직은 상기 서버측 네트워크 노드와 통신하기 위한 프로세스를 포함하는 애플리케이션 인터페이스를 포함하는 것을 특징으로 하는 시스템.

청구항 20

제 18항에 있어서,

상기 이벤트들은 다른 클라이언트측 네트워크 노드들 사이에 배포되고 상기 서버측 네트워크 노드의 공간 이벤트 로그에 추가될 이력 이벤트들로 지정된 이벤트의 제1 클래스와, 다른 클라이언트측 네트워크 노드들 사이에 배포되지만 상기 서버측 네트워크 노드의 공간 이벤트 로그에 추가되지는 않을 일시 이벤트들로서 지정된 이벤트의 제2 클래스를 포함하는 것을 특징으로 하는 시스템.

청구항 21

제 18항에 있어서,

상기 그래픽 타겟들은 텍스트 및 스트로크들을 포함하는 것을 특징으로 하는 시스템.

청구항 22

제 18항에 있어서,

상기 로그의 입력들은 디스플레이에 상기 그래픽 타겟을 렌더링하기 위해 사용되는 그래픽 구성물들을 식별하는 파라미터를 포함하는 것을 특징으로 하는 시스템.

청구항 23

제 18항에 있어서,

상기 로그의 입력들은 상기 이벤트와 관련된 사용자를 식별하는 파라미터를 포함하는 것을 특징으로 하는 시스템.

청구항 24

제 18항에 있어서,

상기 로직은 상기 작업공간에 상대적인 디스플레이가능한 영역을 패닝 및 주밍(panning and zooming)하기 위한 절차들을 포함하는 것을 특징으로 하는 시스템.

청구항 25

제 18항에 있어서,

상기 작업공간은 경계가 정해지지 않은 가상 영역을 포괄하는 것을 특징으로 하는 시스템.

청구항 26

제 18항에 있어서,

상기 작업공간은 실제적으로 한정되지 않은 가상 영역을 포괄하는 것을 특징으로 하는 시스템.

청구항 27

협업 작업공간을 동작시키기 위한 방법으로서,

작업공간에 이벤트들을 위치시키는 공간 이벤트 맵을 네트워크 노드에 유지하는 단계와; 그리고

상기 공간 이벤트 맵에 액세스를 제공하기 위해서 다른 네트워크 노드들과 통신하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 28

제 27항에 있어서,

상기 공간 이벤트 맵은 이벤트들의 로그, 상기 작업공간의 이벤트의 그래픽 타겟의 위치 및 시간을 갖는 상기 로그의 입력들을 포함하는 것을 특징으로 하는 방법.

청구항 29

제 28항에 있어서,

이벤트, 상기 작업공간의 이벤트의 그래픽 타겟의 위치 및 시간을 식별하는 메시지들을 클라이언트측 네트워크 노드들에 송신하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 30

제 28항에 있어서,

상기 작업공간의 위치 및 시간을 갖는 그래픽 타겟을 생성 또는 수정하는 이벤트들을 식별하는 메시지들을 수신하는 단계와, 그리고 상기 이벤트들의 로그에 대응하는 입력들을 추가하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 31

제 28항에 있어서,

상기 작업공간의 위치 및 시간을 갖는 그래픽 타겟을 생성 또는 수정하는 이벤트들을 식별하는 클라이언트측 네트워크 노드로부터의 메시지들을 수신하는 단계를 포함하며, 상기 이벤트들은 이력 이벤트들 및 일시 이벤트들을 포함하는 것을 특징으로 하는 방법.

청구항 32

제 31항에 있어서,

상기 로그에 대응하는 입력들을 추가함이 없이 다른 클라이언트측 네트워크 노드들에 일시 이벤트들을 식별하는 메시지들을 송신하고, 다른 클라이언트측 네트워크 노드들에 상기 이력 이벤트들을 식별하는 메시지들을 송신하고, 그리고 상기 대응하는 이력 이벤트들에 대해 상기 로그에 입력들을 추가하는 단계를 포함하는 것을 특징으

로 하는 방법.

청구항 33

제 27항에 있어서,

상기 작업공간은 경계가 정해지지 않은 가상 영역을 포괄하는 것을 특징으로 하는 방법.

청구항 34

제 27항에 있어서,

상기 작업공간은 실제적으로 한정되지 않은 가상 영역을 포괄하는 것을 특징으로 하는 방법.

청구항 35

방법으로서,

서버측 네트워크 노드에, 작업공간의 위치들을 갖는 그래픽 타겟들에 관한 이벤트들의 로그 및 상기 이벤트의 그래픽 타겟의 작업공간의 위치, 상기 이벤트의 시간 및 상기 이벤트의 그래픽 타겟의 타겟 식별자를 포함하는 상기 로그의 입력들을 저장하는 단계와;

복수 개의 활성 클라이언트측 네트워크 노드들에 링크들을 설정하는 단계와;

상기 설정된 링크들에 상기 작업공간의 위치들을 갖는 그래픽 타겟들의 수정 및 생성에 관한 이벤트들을 식별하는 메시지들을 수신하는 단계와, 상기 메시지들에 응답하여 상기 로그에 이벤트들을 추가하는 단계와, 그리고 특정한 클라이언트측 네트워크 노드로부터 다른 활성 클라이언트측 네트워크 노드에 수신된 메시지들에서 식별된 이벤트들에 관한 설정된 링크들에 메시지들을 배포하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 36

제 35항에 있어서,

애플리케이션 프로그램 인터페이스를 실행하는 단계를 포함하며, 상기 애플리케이션 프로그램 인터페이스에 의해서, 클라이언트측 네트워크 노드들에 상기 로그의 일부들을 전달하는 메시지들을 송신하고, 상기 작업공간의 위치들을 갖는 그래픽 타겟들에 관한 이벤트들을 식별하는 데이터를 전달하는 클라이언트측 네트워크 노드들로부터 메시지들을 수신하는 것을 특징으로 하는 방법.

청구항 37

제 35항에 있어서,

상기 이벤트들은 상기 로그에 저장되고 다른 클라이언트측 네트워크 노드들에 배포될 이벤트의 제1 클래스와, 다른 클라이언트측 네트워크 노드들에 배포되지만 상기 로그에 저장되지는 않을 이벤트의 제2 클래스를 포함하는 것을 특징으로 하는 방법.

청구항 38

제 35항에 있어서,

상기 그래픽 타겟들은 텍스트 및 스트로크들을 포함하는 것을 특징으로 하는 방법.

청구항 39

제 35항에 있어서,

상기 로그의 입력들은 디스플레이에 그래픽 타겟을 렌더링하기 위해 사용되는 그래픽 구성물들을 식별하는 파라미터를 포함하는 것을 특징으로 하는 방법.

청구항 40

제 35항에 있어서,

상기 로그의 입력들은 상기 이벤트와 관련된 사용자를 식별하는 파라미터를 포함하는 것을 특징으로 하는 방법.

청구항 41

제 35항에 있어서,

상기 작업공간은 경계가 정해지지 않은 가상 영역을 포괄하는 것을 특징으로 하는 방법.

청구항 42

제 35항에 있어서,

상기 작업공간은 실제적으로 한정되지 않은 가상 영역을 포괄하는 것을 특징으로 하는 방법.

청구항 43

방법으로서,

클라이언트측 네트워크 노드로부터 서버측 네트워크 노드에 링크를 설정하는 단계와;

작업공간의 위치들을 갖는 그래픽 타겟들에 관한 이벤트들의 공간 이벤트 로그의 적어도 일부를 상기 서버측 네트워크 노드로부터 클라이언트측 네트워크 노드에서 검색하는 단계와, 상기 로그의 입력들은 이벤트의 그래픽 타겟의 작업공간의 위치, 상기 이벤트의 시간, 상기 그래픽 타겟의 타겟 식별자를 포함하며;

상기 클라이언트측 네트워크 노드의 물리적 디스플레이 공간의 디스플레이 가능한 영역을 상기 작업공간 내의 매핑된 영역과 매핑하는 단계와, 상기 매핑하는 단계는 매핑된 영역 내의 공간 이벤트 로그의 입력들을 식별하고, 상기 식별된 입력들에 의해 식별된 그래픽 타겟들을 디스플레이 가능한 영역에 렌더링하기 위함이며;

상기 디스플레이 가능한 영역 내에 디스플레이되는 그래픽 타겟들의 수정 및 생성에 관한 이벤트들을 생성하는 클라이언트측 네트워크 노드에서의 사용자 입력 디바이스로부터의 입력 데이터를 승인하는 단계와; 그리고

상기 설정된 링크의 이벤트들에 기초해서 메시지들을 상기 서버측 네트워크 노드에 송신하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 44

제 43항에 있어서,

상기 서버측 네트워크 노드에서 실행되는 애플리케이션 프로그램 인터페이스에 따르는(compliant) 메시지들을 송신하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 45

제 43항에 있어서,

상기 이벤트들은 다른 클라이언트측 네트워크 노드들 사이에 배포되고 상기 서버측 네트워크 노드의 공간 이벤트 로그에 추가될 이력 이벤트들로서 지정된 이벤트의 제1 클래스와, 다른 클라이언트측 네트워크 노드들 사이에 배포되지만 상기 서버측 네트워크 노드의 공간 이벤트 로그에 추가되지는 않을 일시 이벤트들로서 지정된 이벤트의 제2 클래스를 포함하는 것을 특징으로 하는 방법.

청구항 46

제 43항에 있어서,

상기 그래픽 타겟들은 텍스트 및 스트로크들을 포함하는 것을 특징으로 하는 방법.

청구항 47

제 43항에 있어서,

상기 로그의 입력들은 디스플레이에 그래픽 타겟을 렌더링하기 위해 사용되는 그래픽 구성물들을 식별하는 파라미터를 포함하는 것을 특징으로 하는 방법.

청구항 48

제 43항에 있어서,

상기 로그의 입력들은 상기 이벤트와 관련된 사용자를 식별하는 파라미터를 포함하는 것을 특징으로 하는 방법.

청구항 49

제 43항에 있어서,

상기 클라이언트측 네트워크 노드에서 상기 작업공간에 상대적인 디스플레이 가능한 영역을 패닝 및 주밍하기 위한 절차들을 실행하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 50

제 43항에 있어서,

상기 작업공간은 무한의 가상 영역을 포괄하는 것을 특징으로 하는 방법.

청구항 51

제 43항에 있어서,

상기 작업공간은 실제적으로 한정되지 않은 가상 영역을 포괄하는 것을 특징으로 하는 방법.

청구항 52

작업공간에 이벤트들을 위치시키는 공간 이벤트 맵을 포함하는 기계 판독가능 데이터 구조를 수록하는 비일시적인 컴퓨터 판독가능 매체를 포함하는 물로서, 상기 공간 이벤트 맵은 이벤트들의 로그, 상기 이벤트의 그래픽 타겟의 위치 및 시간을 갖는 상기 로그의 입력들을 포함하는 것을 특징으로 하는 물(product).

청구항 53

제 52항에 있어서,

상기 물은 이벤트, 상기 작업공간의 이벤트의 그래픽 타겟의 위치 및 시간을 식별하는 메시지들을 클라이언트측 네트워크 노드들에 송신하기 위한 실행가능한 명령어들을 수록하는 것을 특징으로 하는 물.

청구항 54

제 52항에 있어서,

상기 물은 상기 작업공간의 위치 및 시간을 갖는 그래픽 타겟을 생성 또는 수정하는 이벤트들을 식별하는 메시지들을 수신하고, 상기 이벤트들의 로그에 대응하는 입력들을 추가하기 위한 실행가능한 명령어들을 수록하는 것을 특징으로 하는 물.

청구항 55

제 52항에 있어서,

상기 물은 상기 작업공간의 위치 및 시간을 갖는 그래픽 타겟을 생성 또는 수정하는 이벤트들을 식별하는 클라이언트측 네트워크 노드로부터의 메시지들을 수신하기 위한 실행가능한 명령어들을 수록하며, 상기 이벤트들은 이력 이벤트들 및 일시 이벤트들을 포함하는 것을 특징으로 하는 물.

청구항 56

제 55항에 있어서,

상기 물은 상기 로그에 대응하는 입력들을 추가함이 없이 다른 클라이언트측 네트워크 노드들에 일시 이벤트들을 식별하는 메시지들을 송신하고, 상기 다른 클라이언트측 네트워크 노드들에 이력 이벤트들을 식별하는 메시지들을 송신하고, 그리고 대응하는 이력 이벤트들에 대한 로그에 입력들을 추가하기 위한 실행가능한 명령어들을 수록하는 것을 특징으로 하는 물.

청구항 57

제 52항에 있어서,

상기 작업공간은 무한의 가상 영역을 포괄하는 것을 특징으로 하는 물.

청구항 58

제 52항에 있어서,

상기 작업공간은 실제적으로 한정되지 않은 가상 영역을 포괄하는 것을 특징으로 하는 물.

청구항 59

물로서,

비일시적 컴퓨터 판독가능 메모리와;

작업공간에 위치들을 갖는 그래픽 타겟들에 관한 이벤트들의 로그와, 상기 이벤트의 그래픽 타겟의 작업공간의 위치, 상기 이벤트의 시간 및 상기 이벤트의 그래픽 타겟의 타겟 식별자를 포함하는 상기 로그의 입력들을 포함하는 상기 메모리에 수록된 기계 판독가능 데이터 구조와;

복수 개의 활성 클라이언트측 네트워크 노드들에 링크들을 설정하기 위한 실행가능한 명령어들과; 그리고

상기 작업공간에 위치들을 갖는 그래픽 타겟들의 수정 및 생성에 관한 이벤트들을 식별하는 설정된 링크들에 메시지들을 수신하고, 상기 메시지들에 응답하여 상기 로그에 이벤트들을 추가하고, 그리고 특정한 클라이언트측 네트워크 노드로부터 다른 활성 클라이언트측 네트워크 노드들에 메시지들에서 식별된 이벤트들에 관한 설정된 링크들에 대한 메시지들을 배포하기 위한 실행가능한 명령어들을 포함하는 것을 특징으로 하는 물.

청구항 60

제 59항에 있어서,

애플리케이션 프로그램 인터페이스를 실행하는 것을 포함하고, 상기 애플리케이션 프로그램 인터페이스에 의해 상기 로그의 부분을 클라이언트측 네트워크 노드들에 전달하는 메시지들을 송신하고, 클라이언트측 네트워크 노드들로부터 상기 작업공간의 위치들을 갖는 그래픽 타겟들에 관한 이벤트들을 식별하는 데이터를 전달하는 메시지들을 수신하는 것을 특징으로 하는 물.

청구항 61

제 59항에 있어서,

상기 이벤트들은 상기 로그에 저장되고 다른 클라이언트측 네트워크 노드들에 배포될 이벤트의 제1 클래스와, 다른 클라이언트측 네트워크 노드들에 배포되지만 상기 로그에 저장되지는 않을 이벤트의 제2 클래스를 포함하는 것을 특징으로 하는 물.

청구항 62

제 59항에 있어서,

상기 그래픽 타겟들은 텍스트 및 스트로크들을 포함하는 것을 특징으로 하는 물.

청구항 63

제 59항에 있어서,

상기 로그의 입력들은 디스플레이의 그래픽 타겟을 렌더링하기 위해 사용되는 그래픽 구성물들을 식별하는 파라미터를 포함하는 것을 특징으로 하는 물.

청구항 64

제 59항에 있어서,

상기 로그의 입력들은 상기 이벤트와 관련된 사용자를 식별하는 파라미터를 포함하는 것을 특징으로 하는 물.

청구항 65

제 59항에 있어서,

상기 작업공간은 무한의 가상 영역을 포괄하는 것을 특징으로 하는 물.

청구항 66

제 59항에 있어서,

상기 작업공간은 실제적으로 한정되지 않은 가상 영역을 포괄하는 것을 특징으로 하는 물.

청구항 67

물로서,

비일시적 컴퓨터 판독가능 메모리와;

클라이언트측 네트워크 노드로부터 서버측 네트워크 노드에 링크를 설정하기 위한 실행가능한 명령어들과;

작업공간의 위치들을 갖는 그래픽 타겟들에 관한 이벤트들의 공간 이벤트 로그와, 이벤트의 그래픽 타겟의 작업공간의 위치, 상기 이벤트의 시간 및 상기 그래픽 타겟의 타겟 식별자를 포함하는 상기 로그의 입력들 중 적어도 일부를 상기 클라이언트측 네트워크 노드로부터 상기 서버측 네트워크 노드에서 검색하기 위한 실행가능한 명령어들과;

상기 클라이언트측 네트워크 노드의 물리적 디스플레이 공간의 디스플레이 가능한 영역을 상기 작업공간의 매핑된 영역과 매핑하기 위한 실행가능한 명령어들과, 이러한 매핑은 매핑된 영역 내의 공간 이벤트 로그의 입력들을 식별하고, 식별된 입력들에 의해 식별된 그래픽 타겟들을 상기 디스플레이 가능한 영역에 렌더링하기 위함이며;

사용자 입력 디바이스로부터의 입력 데이터를 클라이언트측 네트워크 노드에서 승인하기 위한 실행가능한 명령어들과, 상기 클라이언트측 네트워크 노드는 상기 디스플레이 가능한 영역 내에 디스플레이되는 그래픽 타겟들의 수정 및 생성에 관한 이벤트들을 생성하며; 그리고

상기 설정된 링크의 이벤트들에 기초한 메시지들을 상기 서버측 네트워크 노드에 송신하기 위한 실행가능한 명령어들을 포함하는 것을 특징으로 하는 물.

청구항 68

제 67항에 있어서,

상기 서버측 네트워크 노드에서 실행되는 애플리케이션 프로그램 인터페이스를 따르는 메시지들을 송신하기 위한 실행가능한 명령어들을 포함하는 것을 특징으로 하는 물.

청구항 69

제 67항에 있어서,

상기 이벤트들은 다른 클라이언트측 네트워크 노드들 사이에 배포되고 상기 서버측 네트워크 노드의 공간 이벤트 로그에 추가될 이력 이벤트들로서 지정된 이벤트의 제1 클래스와, 다른 클라이언트측 네트워크 노드들 사이에 배포되지만 상기 서버측 네트워크 노드의 공간 이벤트 로그에 추가되지는 않을 일시 이벤트들로서 지정된 이벤트의 제2 클래스를 포함하는 것을 특징으로 하는 물.

청구항 70

제 67항에 있어서,

상기 그래픽 타겟들은 텍스트 및 스트로크들을 포함하는 것을 특징으로 하는 물.

청구항 71

제 67항에 있어서,

상기 로그의 입력들은 상기 그래픽 타겟을 디스플레이에 렌더링하기 위해 사용되는 그래픽 구성물들을 식별하는 파라미터를 포함하는 것을 특징으로 하는 물.

청구항 72

제 67항에 있어서,

상기 로그의 입력들은 상기 이벤트와 관련된 사용자를 식별하는 파라미터를 포함하는 것을 특징으로 하는 물.

청구항 73

제 67항에 있어서,

상기 클라이언트측 네트워크 노드에서 상기 작업공간에 상대적인 디스플레이 가능한 영역을 패닝 및 주밍하기 위한 절차들을 실행하기 위한 실행가능한 명령어들을 포함하는 것을 특징으로 하는 물.

청구항 74

제 67항에 있어서,

상기 작업공간은 무한의 가상 영역을 포괄하는 것을 특징으로 하는 물.

청구항 75

제 67항에 있어서,

상기 작업공간은 실제적으로 한정되지 않은 가상 영역을 포괄하는 것을 특징으로 하는 물.

명 세 서

기 술 분 야

[0001] 본 발명은 디지털 협업(digital collaboration)용 장치들, 방법들 및 시스템들에 관한 것이고, 더 특별하게는 다수의 동시 사용자들이 글로벌 작업공간 데이터(global workspace data)에 액세스(access)를 갖는 것을 용이하게 하는 디지털 디스플레이 시스템들에 관한 것이다.

배 경 기 술

[0002] 디지털 디스플레이들은 인터렉티브 표시들(interactive presentations) 및 화이트보드들과 유사한 방식의 기타 목적들을 위해 종종 사용된다. 일부 디스플레이들은 네트워크화되고 협업을 위해 사용될 수 있어서, 하나의 디스플레이 상의 디스플레이 이미지에 대한 수정들(modifications)은 다른 디스플레이에 복제(replicate)된다. 큰 스케일의 디스플레이들은 2명 이상의 사용자들이 동일한 표면 상에 동시에 표시하거나 주석을 다는 기회를 제공한다. 하지만, 다수의 사용자들의 협업에서 문제들이 발생할 수 있고, 일부 상황들에서 단일 디스플레이의 다수의 사용자들의 사용은 이들의 표현의 유연성(flexibility)을 제한할 수 있다.

[0003] 또한, 디지털 디스플레이들은 큰 인터렉션 표면을 제공하도록 구성되는 단일 공간(single room)의 큰 디스플레이 스크린들 또는 스크린들의 어레이들을 포함한다. 그러므로, 큰 디지털 디스플레이들이 서로 다른 협업들에 대해 서로 다른 때에 많은 사용자들에 의해 공유되는 것이 예상된다. 협업을 위한 작업공간 데이터가 인가된 사용자들(authorized users)에 한정된 액세스를 갖는 비밀이지만, 상기 사용자들이 인터렉션하는 디지털 디스플레이들이 많은 장소들에 분산되어 있고 필연적으로 단일 사용자의 배타 제어(exclusive control)하에 있지는 않은 경우에, 협업에 대한 액세스의 보안에 문제가 발생한다.

[0004] 이에 더하여, 시스템의 분산성(distributed nature)은 동시에 그리고 어떤 다른 사용자도 작업공간 데이터를 관찰(observe)하고 있지 않을 때 동일한 작업공간 데이터와 인터렉션(interaction)하고 이를 변경할 수 있는 서로 다른 장소들의 다수의 사용자들의 가능성을 낳는다. 이는 다수의 위치들의 동시성(concurrency) 및 작업공간 데이터의 현재 상태에 관한 정보를 공유하는 것에 문제를 일으킨다.

[0005] 그러므로, 각각의 사용자가 아이디어들의 실시간 교환으로 자신의 아이디어들을 표현할 최대의 자유를 가지는 반면에 협업의 비밀성(confidential nature)을 보호하기 위해 적합한 보안을 제공하는 방식으로 디스플레이들의

분산된 네트워크에 작업공간 데이터를 다수의 사용자들이 공유하도록 하기 위한 방식들을 찾는 것이 바람직할 것이다. 그러므로, 상기 문제에 대한 로버스트(robust)한 해결책들을 만들 기회가 발생한다. 더 나은 아이디어들, 협업 및 결과들이 달성될 것이다.

발명의 내용

[0006] 공간 이벤트 맵(spatial event map)에 기초해서 이미지들을 디스플레이하기 위해 사용되는 많은 분산된 디지털 디스플레이들을 갖는 공간 이벤트 맵을 구현하는 협업 시스템이 설명된다. 공간 이벤트 맵들은 단일 위치에 단일 디스플레이를 갖는 시스템들로도 배치될 수 있다.

[0007] 다수의 디바이스들 및 위치들을 통해 액세스 가능한 세션 각각에 대한 실질적으로 한정되지 않은 양의 2D 및 3D 작업공간을 지원하는 시스템이 개시된다.

[0008] 작업공간에 이벤트들을 위치시키는(locate) 입력들을 포함하는 공간 이벤트 맵에 기초해서 협업 시스템이 설명된다. 공간 이벤트 맵은 이벤트들의 로그(log)를 포함하고, 상기 로그의 입력들은 작업공간의 이벤트의 그래픽 타겟(graphical target)의 위치 및 시간을 갖는다. 또한, 로그의 입력들은 그래픽 타겟을 디스플레이에 렌더링하기 위해 사용되는 그래픽 구성물들(graphical constructs)을 식별하는 파라미터(예를 들어, url 또는 실제 파일)를 포함한다. 공간 이벤트 맵이 인가된 클라이언트들에 대해 액세스 가능해지고, 클라이언트들은 로컬 디스플레이 영역들을 렌더링하고, 공간 이벤트 맵에 추가되고 다른 클라이언트들과 공유될 수 있는 이벤트들을 생성하기 위해 공간 이벤트 맵을 활용하는 협업 시스템을 형성하기 위해 인터렉션하는 서버측 네트워크 노드들 및 클라이언트측 네트워크 노드들이 설명된다.

[0009] 특정 협업 세션과 관련된 작업공간이 특정 경계(boundary) 없이 기준 프레임(frame of reference)을 제공하는 시간 및 가상 협업 공간에 이벤트들을 위치시키는 경계가 정해지지 않은(unbounded) 가상 영역으로서 표현된다. 상기 작업공간은 클라이언트측 네트워크 노드가 가상 영역의 경계들을 넘어서 네비게이팅(navigating)할 가능성(likelihood)은 무시할 수 있을 만큼 큰 사이즈를 가지는 점에서 실제적으로 한정되지 않은(practically unlimited)인 가상 영역을 포괄한다. 예를 들어, 1,000,000 픽셀 곱하기 1,000,000 픽셀을 포함하는 물리적 디스플레이 공간에 매핑(mapping)되는 가상 영역을 포괄하는 사이즈는 일부 설정들에서 실제적으로 한정되지 않은 것으로 고려된다. 일부 예시들에서, 상기 작업공간은 상기 작업공간의 사이즈가 가상 공간 내의 위치들을 식별하기 위해 사용되는 어드레싱 체계(addressing scheme)의 범위에 의해 한정될 뿐이라는 점에서 본질적으로 "무한(infinite)"이다. 또한, 상기 시스템은 다수의 작업공간들을 포함할 수 있고, 작업공간 각각은 단일 사용자에 의해서 또는 사용자 그룹에 의해서 액세스를 위해 개별적으로 구성될 수 있다.

[0010] 협업 시스템은 서버측 네트워크 노드들 및 클라이언트측 네트워크 노드들이 협업 이벤트들에 관해 통신할 수 있도록 애플리케이션 프로그램 인터페이스(API)에 따라 구성될 수 있다. 작업공간의 위치 및 시간을 갖는 그래픽 타겟을 생성 또는 수정하는 이벤트들을 식별하는 메시지들이 정의될 수 있다. 이벤트들은 이력 이벤트들(history events) 및 일시 이벤트들(ephemeral events)로서 분류되고, 이력 이벤트들은 공간 이벤트 맵에 저장되고, 일시 이벤트들은 공간 이벤트 맵에 영구적으로 저장되지는 않지만 협업 세션의 다른 클라이언트들 사이에 배포된다.

[0011] 공간 이벤트 맵을 활용하는 작업공간의 참여를 제공하는 동작들의 세트(set) 및 동작들에 대한 파라미터들을 포함하는 애플리케이션 프로그램 인터페이스(API)가 제공된다. 동작들의 세트는 하드웨어에 의해 보조되고(hardware-assisted), 파라미터들을 사용하고 API의 동작들을 수행하도록 구성되는 소프트웨어 구현 기능들을 이용하여 데이터 프로세서들에서 구현될 수 있다.

[0012] 위의 발명의 내용은 본 명세서에서 설명되는 협업 시스템의 일부 양상들의 기초적인 이해를 제공하기 위해서 제공된다. 이 발명의 내용은 본 발명의 핵심 또는 중요한 요소들(key or critical elements)을 식별하거나 본 발명의 범위를 묘사(delineate)하려고 의도되지 않았다.

도면의 간단한 설명

[0013] 본 발명은 본 발명의 특정 실시예들에 관해 설명될 것이고, 척도대로 그려지지 않은 도면들에 대한 참조가 이뤄질 것이다.

도 1a 및 1b(집합적으로 도 1)은 디지털 디스플레이 협업 환경의 예시 양상들을 설명한다.

도 2는 인가된 사용자들에 의한 사용을 위해 작업공간 데이터가 전달(deliver)되는 지리적으로 분산된 복수 개

의 디스플레이 벽들(display walls)을 포함하는 협업 시스템을 설명한다.

도 3 및 4는 도 1의 디스플레이상의 드로잉 영역 행동(drawing region behavior)의 양상들을 설명한다.

도 5a 내지 5e(집합적으로 도 5)는 작업공간에 대한 작업공간 데이터의 부분들에 대한 데이터 구조들의 단순화된 도면이다.

도 6은 본 명세서에서 설명되는 시스템의 일 예시의 작업공간의 행동자들(actors)의 기능적 특징들을 설명한다.

도 7은 연합된 디스플레이들(federated displays)을 이용하여 구현되는 디지털 디스플레이의 도면이다.

도 8은 예를 들어, 클라이언트 디바이스 컴퓨터 시스템(도 1b)과 같은 컴퓨터 시스템(110)의 단순화된 블럭도이다.

도 9는 클라이언트 디바이스 컴퓨터 시스템(110)(도 1b)에 액세스 가능하게 저장된 데이터베이스의 개략도이다.

도 10은 협업 시스템을 위해 사용되는 사용자 로그인 시퀀스의 서버측 로직의 양상들을 설명하는 흐름도이다.

도 11은 협업 시스템을 위해 사용되는 사용자 로그인 시퀀스의 클라이언트측 로직의 양상들을 설명하는 흐름도이다.

도 12는 협업 시스템을 위해 사용되는 벽의 디스플레이 클라이언트에 대한 서버측 로직의 양상들을 설명하는 흐름도이다.

도 13은 협업 시스템의 분산된 디스플레이 벽들의 활용을 관리하는 서버측 로직의 양상들을 설명하는 흐름도이다.

도 14는 협업 시스템의 디스플레이로서 활용되는 연합된 디스플레이에 대한 클라이언트측 로직의 양상들을 설명하는 흐름도이다.

도 15는 디스플레이들이 폭넓게 분산된, 분산된 디스플레이 협업을 지원하는 시스템을 도 1b의 스타일로 설명한다.

도 16은 클라이언트측 네트워크 노드 및 디스플레이의 단순화된 기능적 블럭도이다.

도 17은 도 16의 클라이언트측 네트워크 노드와 같은 클라이언트측 네트워크 노드의 동작을 설명하는 흐름도이다.

도 18은 도 16의 클라이언트측 네트워크 노드와 같은 클라이언트측 네트워크 노드의 로직에 의해 실행되는 절차를 설명하는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0014] 다음의 설명은 어떤 통상의 기술자로 하여금 본 발명을 생산 및 사용하도록 하기 위해 제시되고, 특정한 응용 및 그 요건들(requirements)의 컨텍스트에서 제공된다. 개시된 실시예들에 대한 다양한 수정들은 통상의 기술자들에게 쉽게 명백할 것이고, 본 명세서에 정의된 일반적인 원칙들은 본 발명의 사상 및 범위로부터 벗어남이 없이 다른 실시예들 및 응용들에 적용될 수 있다. 그러므로, 본 발명은 도시된 실시예들에 한정되도록 의도되지 않지만, 본 명세서에서 개시된 원칙들 및 피처들(features)과 일관되는 가장 넓은 범위에 따르도록 의도된다.

[0015] "한정되지 않은 작업공간(unlimited workspace)" 문제는 사람들 및 디바이스들이 어떻게 작업공간과 인터렉션하는지를 시간이 지남에 따라 추적할 필요성(need)을 포함한다. 이 핵심 문제를 해결하기 위해서, 우리는 우리가 공간 이벤트 맵이라고 부르는 것을 생성했다. 상기 공간 이벤트 맵은 작업공간의 오브젝트들 및 이벤트들을 정의하기 위해 필요한 정보를 포함한다. 공간, 이벤트들, 공간의 이벤트들의 맵들, 다수의 동시 사용자들을 포함하는 다수의 사용자들에 의한 상기 공간에 대한 액세스의 관점으로부터 상기 기술을 고려하는 것이 유용하다.

[0016] 공간: 소정의 협력 세션에 대한 한정되지 않은의 양의 공간 정보를 지원하기 위해서, 우리는 작업공간으로 칭해지는 가상 공간을 조직하는 방식을 제공하고, 상기 작업공간은, 예를 들어 새로운 컨텐츠가 상기 공간에 추가될 수 있고, 컨텐츠가 상기 공간에 배열 및 재배열될 수 있고, 사용자가 상기 공간의 한 부분으로부터 다른 부분으로 네비게이팅(navigating)할 수 있고, 사용자가 필요할 때 상기 공간에 필요한 것들을 쉽게 찾을 수 있는 방식으로, 예를 들어 차원들 중 하나 또는 둘 모두에서 본질적으로 한정되지 않은 범위를 갖는 2차원 직교평면에 의해 특징화된다.

[0017] 이벤트들: 작업 공간과의 인터렉션들은 이벤트들로서 핸들링된다. 유형의 사용자 인터페이스 디바이스들을 통한 사람들, 그리고 시스템들은 작업 공간과 인터렉션한다. 이벤트들은 물리적인 디스플레이에서 디스플레이될 타겟 그래픽 오브젝트(target graphical object), 작업 공간 내의 생성, 수정 및 이동과 같은 동작, 타겟 그래픽 오브젝트 및 이들과 관련된 메타데이터의 삭제를 정의하거나 포인팅하는 데이터를 갖는다. 메타데이터는 창작자(originator), 날짜, 시간, 작업 공간에서의 위치, 이벤트 타입, 보안 정보 및 기타 메타데이터와 같은 정보를 포함한다.

[0018] 작업 공간의 이벤트들을 추적하는 것은 상기 시스템으로 하여금 작업 공간의 현재 상태에서 작업공간의 공간 이벤트들을 제시할 뿐만 아니라, 작업 공간을 다수의 디스플레이들 상의 다수의 사용자들과 공유하고, 컨텐츠에 관한 관련 외부 정보를 공유하고, 상기 공간 이벤트가 시간에 따라 어떻게 진화(evlove)하는지를 이해하는 것을 인에이블한다. 또한, 공간 이벤트 맵은 필요한 데이터의 양에 관해서 적정한 사이즈(reasonable size)를 갖는 반면에, 또한 경계가 정해지지 않은 작업 공간을 정의한다.

[0019] 상기 시스템에는 몇몇의 서로 다른 종류들의 이벤트들이 존재한다. 이벤트들은 영구적으로 저장되거나, 작업 공간의 유용한 수명(useful life) 동안 작업 공간을 유지하기 위해서 시스템에 의해 요구되는 길이의 시간 동안 저장되는 이력 이벤트들로도 언급되는 지속 이벤트들(persistent events)로서 분류된다. 이벤트들은 단기간 동안만 유용하거나 관심 있고 세션에 개입된(involved) 다른 클라이언트들 간에 실시간으로 공유되는 일시 이벤트들로서 분류된다. 지속 이벤트들은 되돌리기/플레이백 이벤트 스트림(undo/playback event stream)에 저장된 이력 이벤트들을 포함하고, 이벤트 스트림은 세션의 공간 이벤트 맵과 동일하거나 상기 공간 이벤트 맵으로부터 도출된다. 일시 이벤트들은 상기 시스템에 대한 되돌리기/플레이백 이벤트 스트림에 저장되지 않는 이벤트들을 포함한다. 공간 이벤트 맵, 또는 맵들은 상기 시스템의 작업 공간상의 지속 이벤트들과 일시 이벤트들 모두의 일부 실시예들에서 작업 공간의 시간들 및 위치들을 추적하기 위한 협업 시스템에 의해 사용될 수 있다.

[0020] 맵들: 작업 공간의 이벤트들의 맵은 이산적인 공간 이벤트들(discrete spatial events)의 총합을 포함한다. 작업 공간에 대한 지속 공간 이벤트들이 이용가능할 때, 그 작업 공간은 특정 사이즈의 디스플레이 가능한 영역을 갖고 디스플레이 가능한 영역에 디스플레이될 작업 공간의 위치 또는 영역을 식별하는 디스플레이 또는 스크린에 "매핑될(mapped)" 수 있다.

[0021] 멀티-사용자 액세스(Multi-User Access): 하나의 핵심 특징은 동시에 작업 공간에서 일하고 있는 모든 사용자들, 또는 다수의 사용자들이 거의 실시간 방식으로 다른 사용자들과의 인터렉션들을 볼 수 있어야 한다는 점이다. 공간 이벤트 맵은 지속 이벤트들 및 일시 이벤트들 둘 모두를 포함하여, 이들 각각의 디스플레이 가능한 영역들 내에서, 어떤 주어진 작업 공간상의 모든 사용자들에 대해서, 사용자들이 거의 실시간 이벤트들을 경험하기 위해서 서로 다른 물리적 위치들에서의 디스플레이들을 갖도록 한다.

[0022] 도 1a는 디지털 디스플레이 협업 환경의 예시적인 양상들을 설명한다. 상기 예시에서, 복수의 사용자들(101a-h)(집합적으로 101)은 도 1a에서 일반적으로 103a-d(집합적으로 103)로서 지정되는 복잡한 이미지들, 음악, 비디오, 문서들, 및/또는 기타 매체의 생성에 있어서 서로 협업하기를 원할 수 있다. 상기 설명된 예시의 사용자들은 서로 협업하기 위해서, 예를 들어, 태블릿(102a), 퍼스널 컴퓨터(PC)(102b) 및 다수의 넓은 포맷의 디스플레이들(102c, 102d, 102e)(집합적으로 디바이스들(102)) 등의 전자 네트워크 노드들로서 구성되는 다양한 디바이스들을 이용한다. 본 명세서에서 "벽(wall)"로서 종종 언급되는 넓은 포맷의 디스플레이(102c)는 하나 이상의 사용자들(예를 들어, 사용자들(101c 및 101d), 사용자들(101e 및 101f) 및 사용자들(101g 및 101h))을 수용한다. 클라이언트측 네트워크 노드들로 언급되는 사용자 디바이스들은 작업 공간의 이벤트들을 디스플레이하기 위해 디스플레이 가능한 영역이 할당(allocate)되는 디스플레이들을 갖는다. 주어진 사용자에 대한 디스플레이 가능한 영역은 작업 공간의 가상적으로 경계가 정해지지 않은 범위와 비교해서 한정된 영역 또는 범위를 각각 갖도록 디스플레이의 전체 스크린, 스크린의 서브세트(subset), 스크린에 디스플레이될 윈도우 등을 포함한다.

[0023] 도 1b는 도 1a와 동일한 환경을 설명한다. 도 1b에 도시된 바와 같이, 종종 본 명세서에서 "벽들"로서 언급되는 넓은 포맷의 디스플레이들(102c, 102d 및 102e)은 클라이언트 측의 물리적 네트워크 노드들(10) 각각에 의해 제어되며, 이들은 또한 하나 이상의 작업 공간들에 대한 공간 이벤트 스택을 저장하는 액세스 가능한 데이터베이스(106)를 갖는 서버측 물리적 네트워크 노드로서 구성되는 중앙 협업 서버(105)와 네트워크 통신 중이다. 본 명세서에서 사용되는 바와 같이, 물리적 네트워크 노드는 네트워크에 부여(attach)되는 활성 전자 디바이스이고, 통신 채널을 통해서 정보를 송신, 수신 및 포워딩할 수 있다. 네트워크 노드들로서 배치되는 전자 디바이스들의 예시들은 모든 다양한 컴퓨터들, 워크 스테이션들, 랩탑 컴퓨터들, 핸드헬드 컴퓨터들 및 스마트폰들을 포함한다. 본 명세서에서 사용되는 바와 같이, 용어 "데이터베이스"는 구조의 어떤 단일성(unity)을

반드시 의미하는 것은 아니다. 예를 들어, 2개 이상의 별개의 데이터베이스들이 함께 고려될 때, 본 명세서에서 그 용어가 사용되기 때문에 여전히 "데이터베이스"를 구성한다.

[0024] 협업 서버(105)에서 실행되는 애플리케이션은 Apache 또는 nginx와 같은 웹 서버 소프트웨어를 이용해서 호스팅된다. 상기 애플리케이션은, 예를 들어, LINUX와 같은 운영 체제들을 실행하는 가상 머신들 상에서 호스팅된다. 상기 서버(105)는 경험적으로(heuristically) 도 1b에 단일 컴퓨터로서 설명된다. 하지만, 서버 아키텍처는 다수의 컴퓨터들의 시스템들을 수반하고, 큰 스케일의 클라우드에 기초한 서비스들(large-scale cloud-based services)에 대해 통상적인 바와 같이, 시스템 각각은 서버 애플리케이션들을 실행한다. 서버 아키텍처는 협업 세션에서 클라이언트 각각에 대해 2개 이상의 채널을 포함하는 다양한 타입들의 통신 채널들에 대해 구성되는 통신 모듈을 포함한다. 예를 들어, 네트워크를 통한 거의 실시간의 업데이트들을 통해, 클라이언트 소프트웨어는 예를 들어 웹 소켓 프로토콜에 기초해서 메시지에 기초한 채널을 이용하는 것을 통해 서버 통신 모듈과 통신한다. 파일 업로드들뿐만 아니라 초기 큰 볼륨의 작업 공간 데이터를 수신하기 위해서, 클라이언트 소프트웨어는 HTTP를 통해 서버 통신 모듈과 통신한다. 서버는 예를 들어, Ruby-on-Rails에 의해 제공되는 JavaScript로 기록된 프론트-엔드 프로그램을 실행하고, 예를 들어, Oauth에 기초한 인증/인가(authentication/authorization)를 지원하고, 그리고 다수의 분산된 클라이언트들 간의 조정(coordination)을 지원한다. 서버 통신 모듈은 웹 소켓 애플리케이션과 같은, 작업 공간 데이터에 사용자 동작들을 기록하고, 그리고 응용 가능하다면 다른 클라이언트들에 사용자 동작들을 릴레잉(relaying)하는 기능들을 수행하는, 메시지에 기초한 통신 프로토콜 스택을 포함한다. 이 시스템은 예를 들어, node.JS 플랫폼상에서 실행되거나, 높은 로드의 소켓 애플리케이션들(high-load socket applications)을 핸들링하도록 설계된 다른 서버 기술들 상에서 실행된다.

[0025] 데이터베이스(106)는, 예를 들어, 디스플레이 캔버스 상에 디스플레이 가능한 오브젝트들에 관한 이벤트들을 포함하거나 식별하는 세션 각각의 공간 이벤트 맵에 대한 작업 공간 데이터 세트들의 디지털 표현을 저장한다. 작업 공간 데이터 세트는 공간 이벤트 스택의 형태로 구현되고, 적어도 지속 공간 이벤트들이 되돌리기 동작 동안 선입후출(first-in-last-out) 패턴으로 스택에 추가(푸시(push))되고 스택으로부터 제거(팝(pop))되도록 관리된다. 서로 다른 작업 공간들에 대한 작업 공간 데이터 세트들이 존재한다. 주어진 작업 공간에 대한 데이터 세트는 데이터베이스로 구성되거나, 상기 작업 공간에 링크된 기계 판독가능 문서로서 구성된다. 상기 작업 공간은 경계가 정해지지 않거나 가상적으로 경계가 정해지지 않은 치수들을 갖는다. 상기 작업 공간은 디스플레이 벽 상의 디스플레이 영역에 디스플레이 클라이언트에 의해 디스플레이 가능한 오브젝트들을 식별하는 이벤트 데이터 구조들을 포함하고, 작업 공간의 시간 및 위치를 이벤트 데이터 구조들에 의해 식별되는 오브젝트들과 관련시킨다. 디바이스(102) 각각은 전체 작업 공간의 일부만을 디스플레이한다. 디스플레이 벽은 오브젝트들을 디스플레이하기 위한 디스플레이 영역을 갖고, 상기 디스플레이 영역은 작업 공간의 중앙에 있는 지역(region)에 대응하는 작업 공간의 대응 영역에 매핑되거나, 그렇지 않으면 작업 공간의 사용자 위치와 함께 위치된다. 작업 공간의 대응 영역에 대한 디스플레이 영역의 매핑은 디스플레이에 렌더링될 디스플레이 영역 내에 있는 작업 공간 데이터의 오브젝트들을 식별하며, 그리고 디스플레이 상의 디스플레이 영역의 위치들에서의 사용자 터치 입력들을 링크하기 위한 오브젝트들을 식별하기 위해서 디스플레이 클라이언트에 의해 이용 가능하다.

[0026] 서버(105) 및 데이터베이스(106)는 작업 공간에 위치들을 갖는 그래픽 타겟들에 관한 이벤트들의 로그, 이벤트의 그래픽 타겟의 작업 공간의 위치를 포함하는 로그의 입력들, 이벤트의 시간, 그리고 이벤트의 그래픽 타겟의 타겟 식별자를 저장하는 메모리를 포함하는 서버측 네트워크 노드를 구성한다. 상기 서버는 복수 개의 활성 클라이언트측 네트워크 노드들에 링크들을 설정하고, 작업 공간에 위치들을 갖는 그래픽 타겟들의 변형 및 생성에 관한 이벤트들을 식별하는 메시지들을 수신하고, 상기 메시지들에 응답하여 상기 로그에 이벤트들을 추가하고, 그리고 특정한 클라이언트측 네트워크 노드로부터 다른 활성 클라이언트측 네트워크 노드들에 수신된 메시지들에 식별된 이벤트들에 관한 메시지들을 배포하기 위한 로직을 포함한다.

[0027] 상기 서버(105)의 로직은 로그의 부분들을 전달(carry)하는 메시지들을 송신하고, 작업 공간에 위치들을 갖는 그래픽 타겟들에 관한 이벤트들을 식별하는 데이터를 전달하는 클라이언트측 네트워크 노드들로부터 메시지들을 수신하는 절차들 및 파라미터들의 특정 세트를 포함하는 애플리케이션 프로그램 인터페이스를 포함한다.

[0028] 또한 상기 서버(105)의 로직은 하나의 클라이언트측 네트워크 노드로부터 다른 클라이언트측 네트워크 노드들에 수신된 이벤트들을 배포하기 위한 프로세스를 포함하는 애플리케이션 인터페이스를 포함한다.

[0029] API에 따르는 이벤트들은 로그에 저장되고 다른 클라이언트측 네트워크 노드들에 배포될 이벤트의 제1 클래스(이력 이벤트)와, 그리고 다른 클라이언트측 네트워크 노드들에 배포되지만 로그에 저장되지는 않을 이벤트의

제2 클래스(일시 이벤트)를 포함한다.

[0030] 상기 서버(105)는 복수 개의 작업 공간들에 대한 작업 공간 데이터 세트들을 저장하며, 그리고 상기 세션에 참여하는 디스플레이 클라이언트들에 상기 작업 공간 데이터를 제공한다. 상기 작업 공간 데이터는 그 후에 디스플레이에 디스플레이할 이미지들을 결정하고, 상기 디스플레이 표면상의 위치들에 대해 인터렉션을 위한 오브젝트를 할당하기 위해서, 디스플레이 클라이언트 소프트웨어를 포함하는 적절한 소프트웨어(112)를 갖는 컴퓨터 시스템(110)에 의해 사용된다. 상기 서버(105)는 서로 다른 협업 세션들에 대해서 다수의 작업 공간들을 저장 및 유지한다. 작업 공간 각각은 사용자들의 그룹과 관련되고, 상기 그룹의 인가된 사용자들에 의해서만 액세스하도록 구성된다.

[0031] 일부 대안들에서, 상기 서버(105)는 그 디바이스 상에서 뷰잉 가능한(viewable) 캔버스의 부분을 표시하는 각 디바이스(102)에 대한 "뷰포트(viewport)"를 추적하고, 각 디바이스(102)에 상기 뷰포트를 렌더링하기 위해 필요한 데이터를 제공한다.

[0032] 오브젝트들을 드로잉(drawing)하는 것, 사용자 입력들을 핸들링하는 것, 그리고 서버와 통신하는 것을 렌더링하는 역할의 클라이언트 디바이스 상에서 실행되는 애플리케이션 소프트웨어들은 HTML5 또는 기타 마크업에 기초한 절차들에 기초하고, 브라우저 환경에서 실행될 수 있다. 이는 많은 서로 다른 클라이언트 운영 체제 환경들의 쉬운 지원을 가능하게 한다.

[0033] 데이터베이스(106)에 저장된 사용자 인터페이스 데이터는 이미지 비트맵들, 비디오 오브젝트들, 멀티-페이지 문서들, 스케일링 가능한 벡터 그래픽들 등과 같은 그래픽 구성물들을 포함하는 다양한 타입들의 오브젝트들을 포함한다. 상기 디바이스들(102)은 각각 네트워크(104)를 통해 협업 서버(105)와 통신한다. 상기 네트워크(104)는 LAN들, WAN들, 라우터들, 스위치들, WiFi 컴포넌트들, 셀룰러 컴포넌트들, 유선 및 광학 컴포넌트들 및 인터넷과 같은 모든 형태의 네트워킹 컴포넌트들을 포함한다. 한 시나리오에서 2명 이상의 사용자들(101)은 동일 공간에 위치하고, 이들의 디바이스들(102)은 WiFi를 통해 협업 서버(105)와 통신한다. 다른 시나리오에서 2명 이상의 사용자들(101)은 수천 마일만큼 서로로부터 떨어져 있고 이들의 디바이스들(102)은 인터넷을 통해 협업 서버(105)와 통신한다. 벽들(102c, 102d 및 102e)은 이미지들을 디스플레이할 뿐만 아니라 스타일러스 또는 하나 이상의 손가락들과 같은 몸의 일부 둘 중 하나를 이용해서 디스플레이 표면들을 터치함으로써 제공되는 사용자 제스처들을 센싱할 수도 있는 멀티-터치 디바이스들일 수 있다. 일부 실시예들에서, 벽(예를 들어, 102c)은 하나 이상의 손가락들(또는 예를 들어, 손 전체)에 의한 터치와 스타일러스에 의한 터치를 구별할 수 있다. 일 실시예에서, 벽은 적외선 빛을 방출(emitting)하고 수신된 빛을 검출함으로써 터치를 센싱하고, 사용자의 손가락으로부터 반사된 빛은 상기 벽이 주변의 수신된 빛으로부터 구별하는 특징을 갖는다. 스타일러스는 상기 벽이 주변의 빛과 사용자의 손가락으로부터 반사된 빛 모두로부터 구별하는 방식으로 고유한 적외선 빛을 방출한다. 벽(102c)은 예를 들어, 수직 및 수평 둘 모두로 타일된(tiled) 핀란드, 헬싱키의 MultiTouch Ltd에 의해 제작된 모델 번호 MT553UTBL MultiTaction Cells의 어레이일 수 있다. 다양한 표현 수단을 제공하기 위해서, 상기 벽(102c)은 상기 벽(102c)이 "상태(state)"를 유지하는 방식으로 동작된다. 즉, 상기 벽(102c)은 (다른 것들 중에서도 특히) 입력의 순서에 따라 서로 다르게 주어진 입력에 반응한다. 예를 들어, 툴바를 이용해서, 사용자는 다수의 이용 가능한 브러시 스타일들 및 색상들 중 어느 것을 선택할 수 있다. 일단 선택되면, 상기 벽은 상기 스타일러스에 의한 후속의 스트로크들(strokes)이 선택된 브러시 스타일 및 색상을 이용해서 라인을 드로잉할 상태에 있게 된다.

[0034] 설명적인 실시예에서, 디스플레이 어레이는 전체 높이로 대략 6 피트 및 폭으로 30 피트의 디스플레이 가능한 영역을 갖고, 이는 다수의 사용자들이 상기 벽의 서로 다른 부분들에 서서 상기 벽을 동시에 조작하기에 충분히 넓은 것이다. 이 실시예에서의 상기 벽이 서로 다른 사용자들의 손가락들, 또는 서로 다른 사용자들에 의해 동작하는 스타일러스들을 구별하지 않기 때문에, 상기 벽 상에서의 표현의 유연성은, 하지만, 멀티-사용자 시나리오에서 제한될 수 있다. 그러므로 만약 한 사용자가 하나의 원하는 상태로 상기 벽을 놓는다면, 상기 벽은 제2 사용자의 입력이 다르게 처리될 것임을 인식할 방법을 갖지 않기 때문에, 제2 사용자는 동일한 상태를 사용하도록 제한된다.

[0035] 이 제한을 회피하기 위해서, 클라이언트측 네트워크 노드는 상기 벽(102c) 상에 "드로잉 영역들(drawing regions)"을 정의할 수 있다. 본 명세서에서 사용되는 바와 같이, 드로잉 영역은 상기 벽의 상태의 적어도 하나의 양상이 상기 벽 상의 다른 영역들과 독립적으로 변경될 수 있는 영역이다. 본 실시예에서, 드로잉 영역들 간에 서로 다를 수 있는 상태의 양상들은 스타일러스를 이용해서 벽 상에 드로잉되는 라인의 속성들을 포함한다. 손가락 터치 행동들에 대한 상기 시스템의 응답과 같은 상태의 다른 양상들은 드로잉 영역들에 의해 영향받지

않을 것이다.

[0036] 도 2는 지리적으로 분산되고 디스플레이 클라이언트들이 위치된 다수의 시설들(예를 들어, 시설 1 및 시설 2)에 링크될 수 있는 공유되는 서버(105)를 포함하는 분산된 협업 시스템을 설명한다. 예를 들어, 시설 1은 New York 시에 위치하는 반면, 시설 2는 Los Angeles에 위치한다. 협업 시스템에서 이용 가능한 디스플레이 클라이언트들이 위치하는 많은 다른 물리적 위치들이 존재할 수 있다. 이 예시에서, 시설 1은 제1 공간(151), 제2 공간(152) 및 제3 공간(153)을 포함한다. 시설 2는 제1 공간(161), 제2 공간(162) 및 제3 공간(163)을 포함한다. 시설 1의 제1 공간(151)은 복수 개의 디스플레이들을 이용해서 구현되는 넓은 포맷의 디스플레이를 포함한다. 시설 1의 제2 공간(152)은 중간 포맷의 디스플레이인 단일 스크린을 포함한다. 시설 1의 제3 공간(153)은 개인 컴퓨터 또는 랩탑이 선택된 작업 공간에서 인터렉션하는 세션에 대한 디스플레이 클라이언트로서 활용되는 사설 오피스(private office) 또는 기타 공간일 수 있다. 시설 2는 이 설명에서 시설 1과 유사하고, 제1 공간(161), 제2 공간(162) 및 제3 공간(163)을 포함한다. 시설 2의 제1 공간(161)은 복수 개의 디스플레이들을 이용해서 구현되는 넓은 포맷의 디스플레이를 포함한다. 시설 2의 제2 공간(162)은 중간 포맷의 디스플레이인 단일 스크린을 포함한다. 시설 2의 제3 공간(163)은 퍼스널 컴퓨터, 랩탑, 모바일 패드 또는 모바일 폰이 세션에 대한 디스플레이 클라이언트로서 활용될 수 있는 사설 오피스 또는 기타 공간일 수 있다.

[0037] 도 2는 원격적으로 위치하는 넓은 포맷 또는 중간 포맷의 디스플레이들(또는 벽들)에 따르는 분산된 협업 시스템들에 관해 발생하는 문제를 설명한다. 넓은 포맷 및 중간 포맷의 디스플레이는 통상적으로 개인 사용자의 배타적 제어(exclusive control)하에 있지 않다. 그러므로, 협업 서버(105)는 어떤 주어진 시간에 디스플레이들에 액세스를 갖는 사람에 관한 정보를 전혀 가지고 있지 않을 수 있다.

[0038] 도 3은 벽(102)을 설명한다. 이 예시에서의 벽은 6피트만큼 높고 30피트만큼 넓다. 상기 벽은 초기에 기본 배경색상 또는 이미지이고, 벽 전체에서 기본 드로잉 상태(default drawing state)를 갖는다. 또한, 작업 공간 데이터는 상기 벽의 디스플레이 영역의 물리적 위치들에 매핑되는 작업 공간의 위치들을 갖는, 집합적으로 오브젝트들(301)인 복수 개의 오브젝트들(301a 내지 301h)을 정의할 수 있다. 상기 오브젝트들(301)은 상기 디스플레이 영역에 렌더링되는 예를 들어, 텍스트, 이미지들 또는 드로잉을 포함하는 카드들을 포함한다. 또한, 상기 오브젝트들(301)은 사용자가 상기 카드의 컨텐츠, 또는 터치 스크린과 같은 디스플레이에서 사용자 인터페이스 도구들을 이용해서 카드에 링크되는 기능들과 인터렉션하도록 하는 피처들을 포함한다. 도 3에서도 설명된 바와 같이, 드로잉 오버레이 오브젝트(drawing overlay object)(302)는 상기 벽의 디스플레이 영역에 디스플레이될 수 있다.

[0039] 드로잉 상태는 상기 영역에 디스플레이된 오브젝트들(301 및 302)과 독립적인 영역의 피처이고, 도 3의 실시예에서 브러시 타입, 브러시 크기 및 색상과 같은 라인 겉모습 속성들(line appearance properties)을 포함하는 라인 드로잉 속성들에 의해 정의될 수 있다. 예시의 목적을 위해서, 상기 시스템은 사용자(101c)가 스타일러스 또는 하나 이상의 손가락들(때때로 본 명세서에서 집합적으로 기록 도구(writing implement)로서 언급됨) 둘 중 하나를 이용해서 상기 벽을 터치할 때, 근처에 툴바(210)가 나타나며 드로잉 영역(212)이 정의된다. 터치 포인트를 터치하는 것은 본 명세서에서 "사용자 입력을 개방(opening user input)"하는 것으로 때때로 언급되는 것 중 하나의 실시예이고, 다른 실시예들은 독자에게 명백할 것이다. 새롭게 정의된 드로잉 영역의 초기 드로잉 상태는 (브러시 타입=잉크, 두께=5㎜, 색상=백색과 같은)미리 정의된 기본 상태이며, 이는 다양한 실시예에서 상기 벽의 나머지의 기본 상태와 매칭되거나 매칭되지 않을 수 있다. 도 2의 실시예에서 드로잉 영역에 대해 설정된 드로잉 속성들은 드로잉 영역 전체에 적용될 수 있다. 라인 드로잉은 애플리케이션 프로그램이 벽(102c)의 어떤 특정한 영역의 소유권을 갖고 있는지 여부와 무관하게, 상기 컴퓨터 시스템(110)에서 실행 중인 어떤 상기 애플리케이션 프로그램의 논리적으로 위의 레이어(layer)에 있는 상기 벽 상에서 동작한다.

[0040] 도 3의 실시예에서, 비록 다른 실시예들에서 드로잉 영역들이 보다 짧고 그리고/또는 비-직사각형 모양들을 가질 수 있지만, 드로잉 영역들은 언제나 상기 벽의 전체 수직 범위를 채운다. 또한 도 3의 실시예에서 드로잉 영역들은 좌측 및 우측 경계(214 및 216)를 이용해서 알아볼 수 있을 정도로 경계가 정해지며(perceptibly demarcated), 다른 실시예에서, 배경 음영법(background shading)과 같은 다른 수단이 상기 영역의 경계를 정하기 위해서 이용될 수 있다. 또 다른 실시예에서 상기 영역 경계들은 상기 사용자가 알아볼 수 없다. 좌측 및 우측에 충분한 공간을 가정할 때, 클라이언트측 컴퓨터 시스템(110)은 사용자의 터치 포인트를 중심으로 하는 위치에 드로잉 영역을 스포닝(spawning)할 수 있다. 드로잉 영역들은 최소 폭 Wmin 및 이상적인 폭 Wideal을 갖는다. 최소 폭은 바람직하게는 상당히 제한받지 않는 표현(reasonably unfettered expression)을 가능하게 하도록 하는 가장 작은 폭인 것으로 선택되며, 도 3의 실시예에서 4피트이다. 이상적인 폭은 바람직하게는 평균 사용자의 팔이 수평으로 뻗어질 수 있는 가장 넓은 범위와 대략적으로 동일한 것으로 선택되며, 도 3의 실시예에서 6

피트이다.

[0041] 만약 사용자의 터치 포인트의 일 측에 충분한 공간이 있다면, 컴퓨터 시스템(110)은 초기 영역 폭을 Wideal로 설정한다. 이는 도 3에 설명되는 시나리오이다. 만약 사용자의 터치 포인트가 새로운 드로잉 영역이 상기 터치 포인트를 중심으로 하기에는 벽 에지(wall edge)에 지나치게 가깝다면, 컴퓨터 시스템(110)은 상기 새로운 드로잉 영역을 상기 벽 에지에 인접하게 할 것이다. 새로운 드로잉 영역은 충분한 공간이 이용 가능하다고 가정하면 여전히 폭 Wideal을 가질 것이며, 그래서 새로운 드로잉 영역은 사용자의 터치 포인트를 중심으로 하지 않을 것이다. 반면에, 만약 사용자의 터치 포인트가 터치 포인트를 중심으로 하는 드로잉 영역을 생성하도록 상기 벽 에지로부터 충분히 멀지만, 새로운 드로잉 영역이 벽 에지로부터 Wmin보다 가깝다면, 벽 에지와 새로운 드로잉 영역 간의 갭 공간은 사용 불가능한 것(unusable)으로 고려된다. 이 경우에 컴퓨터 시스템(110)은 사용 불가능한 공간을 채우기 위해서 새로운 드로잉 영역을 연장할 것이다.

[0042] 도 4는 예시의 목적을 위해서 2명의 사용자들(101c 및 101d)이 상기 벽을 동시에 사용하는 시나리오를 설명한다. 초기에, 사용자(101c)는 터치 포인트(516)에서 상기 벽(102c)을 터치하고, 그에 응답하여 상기 컴퓨터 시스템(110)은 툴바(510)와 함께 드로잉 영역(512)을 스포닝한다. 선택적으로, 사용자(101c)는 그 후에 영역(512) 내의 라인 겉모습 속성들을 변경하기 위해서 툴바(510)상의 제어들을 터치한다. 다음으로, 제2 사용자(101d)는 상기 벽(102c) 배경 내에 있는(즉, 모든 기존에 존재하는 드로잉 영역들 외부에 있는) 터치 포인트(518)에서 상기 벽(102c)을 터치한다. 그 후에 제2 드로잉 영역(514)은 툴바(520)와 함께 컴퓨터 시스템(110)에 의해 스포닝된다. 만약 사용자(101d)가 영역(514) 내에 이때 라인을 드로잉하면, 컴퓨터 시스템(110)은 드로잉 영역(512)에 대해 사용자(101c)에 의해 이전에 설정된 라인 속성들보다는 기본 라인 속성들을 이용해서 상기 라인을 페인팅할 것이다. 사용자(101d)는 그 후에 영역(514) 내의 라인 겉모습 속성들을 변경하기 위해서 툴바(520) 상의 제어들을 선택적으로 터치한다. 영역(514)에 드로잉된 후속의 라인들은 새로운 라인 겉모습 속성들을 채택할 것이다. 영역(512)의 라인 겉모습 속성들은 변경되지 않은 채로 유지될 것이다.

[0043] 드로잉 영역들은 또한 스타일러스의 이동을 자동적으로 추적하도록 만들어질 수 있다. 비록 다수의 가능한 추적 알고리즘들이 독자에게 분명할 것이지만, 이러한 최소 규칙들을 따르는 한 추적 알고리즘이 선호된다: (1) 스타일러스가 영역의 중심에 상대적으로 가까이 유지되는 한 영역은 움직이지 않으며; 그리고 (2) 스타일러스가 영역 경계에 접근할수록, 상기 영역은 상기 경계가 스타일러스의 앞에서 유지되도록 이동한다.

[0044] 드로잉 영역들은 로컬 디스플레이 벽에 영향을 주지만 글로벌 작업공간 데이터에는 영향을 주지 않는 사용자 인터렉션의 일 예시를 제공한다. 이 예시에서 설명된 바와 같이, 오브젝트들(301, 302)의 위치들은 드로잉 영역들, 툴바들 및 상기 영역들 내의 드로잉 오버레이들(drawing overlays)의 할당에 의해 영향받지 않는다. 물론 다른 타입들의 사용자 인터페이스 인터렉션들에서, 오브젝트들(301, 302)의 위치들은 이동될 수 있고, 그러한 이동들은 글로벌 작업공간 데이터의 오브젝트들에 관한 이벤트들일 수 있다.

[0045] 로컬 벽과의 인터렉션에 기초한 사용자 입력의 해석에 관한 다양한 행동들은 위의 참조에 의해 통합된, 함께 계류중인 2013년 2월 4일에 출원된 디지털 화이트보드에 대한 영역 다이나믹스라는 제목의 미국 출원번호 제13/758,984호에서 설명된다. 이러한 행동들은 일부 실시예들에서 협업 서버에서 유지되는 공유된 작업공간 데이터에 적은 영향이 있거나 전혀 영향이 없는 로컬 컴퓨터 시스템들(110)에 의해 실행될 수 있는 벽의 사용자 입력 및 이미지 데이터의 로컬 프로세싱을 설명한다.

[0046] 도 5a 내지 5e는 협업 서버(105)의 데이터베이스에 의해 유지되는 작업공간 데이터의 일부일 수 있는 데이터 구조들은 나타낸다. 도 5a에서, 이벤트 데이터 구조가 설명된다. 이벤트는 작업공간 데이터를 변경할 수 있는 작업공간 데이터와의 인터렉션이다. 그러므로, 이벤트는 대응하는 이벤트에 대해 하나 이상을 포함할 수 있는 이벤트 식별자, 타임스탬프, 세션 식별자, 이벤트 타입 파라미터, 클라이언트 식별자 및 작업공간의 위치들의 어레이를 포함한다. 예를 들어, 단일 오브젝트에 영향을 주는 경쟁 이벤트들에 대한 경합 상황(race condition)의 확률을 최소화하기 위해서, 타임스탬프는 수 밀리세컨드의 해상도 또는 더 정밀한 해상도를 갖는 것이 바람직하다.

[0047] 또한, 이벤트 데이터 구조는 클라이언트 디스플레이의 터치스크린 상의 스트로크가 링크된 작업공간 데이터의 오브젝트를 식별하는 UI 타겟을 포함한다. 이벤트들은 예를 들어 스트로크의 디스플레이 파라미터들을 표시하는 스타일 이벤트들을 포함한다. 이벤트들은 텍스트 오브젝트의 작업공간에서의 입력, 수정 또는 이동을 표시하는 텍스트 타입 이벤트를 포함한다. 상기 이벤트들은 카드 타입 오브젝트의 작업공간에서의 생성, 수정 또는 이동을 표시하는 카드 타입 이벤트를 포함한다. 상기 이벤트들은 스트로크에 대한 위치 어레이 및 예를 들어 색상들 및 라인 폭들과 같은 스트로크에 대한 디스플레이 파라미터들을 식별하는 스트로크 타입 이벤트를 포함한다.

[0048] 이벤트들은 지속, 이력 이벤트들 및 일시 이벤트들로서 분류된다. 작업공간 데이터에 추가하기 위한 이벤트들의 프로세싱 및 사용자들 간의 공유는 상기 이벤트의 분류에 의존할 수 있다. 이 분류는 이벤트 타입 파라미터에 내재하거나, 상기 분류를 표시하기 위해서 추가적인 플래그(flag) 또는 필드가 이벤트 데이터 구조에서 사용될 수 있다.

[0049] 공간 이벤트 맵은 이력 이벤트들에 대한 입력들을 갖는 이벤트들의 로그를 포함하고, 입력 각각은 도 5a에 설명된 것과 같은 구조를 포함한다. 서버측 네트워크 노드는 클라이언트측 네트워크 노드들로부터 일시 및 이력 이벤트들을 전달하는 메시지들을 수신하고, 상기 로그의 대응하는 입력들을 추가함이 없이 다른 클라이언트측 네트워크 노드들에 일시 이벤트들을 송신하며, 그리고 상기 로그에 대응하는 입력들을 추가하면서 다른 클라이언트측 네트워크 노드들에 이력 이벤트들을 송신하기 위한 로직을 포함한다.

[0050] 도 5b는 카드 데이터 구조를 설명한다. 카드 데이터 구조는 세션 식별자, 카드 타입 식별자, 어레이 식별자, 클라이언트 식별자, 카드들의 치수들, 카드와 관련된 파일의 타입 및 작업공간 내의 세션 위치를 포함하는 작업공간 데이터의 오브젝트에 대한 현재상태 정보를 식별하는 속성들의 캐시(cache)를 제공한다.

[0051] 도 5c는 청크(chunk)라고 불리는 캐치 가능한 세트(catchable set)로 다수의 이벤트들 및 오브젝트들을 통합하는 데이터 구조를 설명한다. 상기 데이터 구조는 세션 ID, 상기 청크에 포함된 이벤트들의 식별자 및 상기 청크가 생성되었던 타임스탬프를 포함한다.

[0052] 도 5d는 선택된 작업 공간의 세션에 참여하는 사용자에 대한 링크들을 위한 데이터 구조를 설명한다. 이 데이터 구조는 액세스 토큰, 세션 디스플레이 클라이언트에 대한 클라이언트 식별자, 디스플레이 클라이언트에 링크된 사용자 식별자, 사용자가 세션에 액세스했던 마지막 시간을 표시하는 파라미터 및 만료 시간(expiration time)과 상기 세션에 관한 다양한 정보를 전달하기 위한 쿠키를 포함한다. 예를 들어, 이 정보는 사용자에 대한 작업 공간 내의 현재 위치를 유지할 수 있고, 상기 현재 위치는 로그인이 관련된 디스플레이 클라이언트에 디스플레이할 작업 공간 데이터를 결정하기 위해서 사용자가 로그인할 때마다 사용될 수 있다.

[0053] 도 5e는 각각 디스플레이 클라이언트를 갖는 연합 디스플레이들에 의해 구현되는 넓은 포맷의 디스플레이들과 함께 사용되는 디스플레이 어레이 데이터 구조를 설명한다. 그러한 연합 디스플레이들의 디스플레이 클라이언트들은 단일 디스플레이로서 동작하도록 협력(cooperate)한다. 작업공간 데이터는 어레이 ID에 의해 디스플레이들의 어레이를 식별하고, 각 디스플레이의 세션 위치를 식별하는 디스플레이 어레이 데이터 구조를 유지한다. 각 세션 위치는 연합 디스플레이들의 영역 내의 x-오프셋 및 y-오프셋, 세션 식별자 및 깊이(depth)를 포함한다.

[0054] 시스템은 클라이언트측 네트워크 노드들과의 통신들을 암호화(encrypt)하고, 공간 이벤트 맵들이 저장된 데이터베이스를 암호화한다. 또한, 클라이언트측 네트워크 노드들에서, 클라이언트측 컴퓨터들에 대한 액세스를 얻은 침입자들(intruders)에 의한 상기 데이터에 대한 비인가된 액세스(unauthorized access)를 방지하기 위해서, 일부 실시예들에서 공간 이벤트 맵의 캐시된 복사본들(cached copies)이 암호화된다.

[0055] 도 6은 작업공간에 대한 작업공간 데이터를 생성, 수정, 배포 및 디스플레이하기 위해 사용되는 분산된 협업 시스템에 대한 기능적 아키텍처를 표현하는 도면이다. 기본 구성은 서버(105)와 같은 서버에 의해 실행되는 이벤트 데이터를 관리하는 협업 서비스(601), 피어 네트워크 노드(peer network node)와 같은 서버(105)와 같은 서버에 의해 실행되거나 상기 서버에 액세스 가능한 기타 컴퓨터 시스템들에 위치한 포털 서비스(portal service)(602), 그리고 사용자 인터렉션이 활성화되는 클라이언트측 네트워크 노드에 위치한 디스플레이 클라이언트(603)를 포함한다. 디스플레이 클라이언트(603)는 협업 서비스(601) 및 포탈(602)과 통신한다. 디스플레이 클라이언트(603)와 협업 서버(601) 간의 통신 채널(613)은 세션 이력의 다운로드 및 세션 이벤트들의 실시간 업데이트를 관리한다. 또한, 이 채널(613)을 통해서, 디스플레이 클라이언트(603)는 협업 서비스(601)에 이벤트들과 관련된 이미지들을 업로드한다. 디스플레이 클라이언트(603)는 통신 채널(623)을 통해서 포털(602)과 통신한다. 포털(602)은 작업공간 데이터, 세션 관리 및 사용자 관리에 대한 홈페이지를 관리한다. 이 포털은 통신 채널(613)의 대안으로서 또는 통신 채널(613)과 병렬적으로, 사용자 로그인, 인증들 및 이미지 파일들을 전달하도록 활용될 수 있다. 협업 서비스(601) 및 포털(602)은 채널(612)을 통해 통신한다. 협업 서비스(601) 및 포털(602)은 인증 및 인가 프로토콜들을 관리하고, 세션 관리 및 작업공간 데이터 관리를 조정한다.

[0056] 디스플레이 클라이언트(603)는 액세스 가능한 메모리에 저장된 컴퓨터 프로그램들을 갖는 물리적 또는 가상의 컴퓨터 시스템을 포함하는 클라이언트측 네트워크 노드의 일부일 수 있고, 상기 컴퓨터 프로그램들은 HTML5 클라이언트, 디스플레이 어레이 구현들을 위한 벽 어레이 조정 로직, 작업공간 데이터 파싱, 검색 및 렌더링 로직 및 서버와 디스플레이 벽에서 작업공간 데이터와의 실시간 인터렉션을 관리하는 세션 이벤트 애플리케이션을 지

원하는 로직을 제공한다.

[0057] 포털(602)은 액세스 가능한 메모리에 저장된 컴퓨터 프로그램들을 갖는 물리적 또는 가상의 컴퓨터 시스템을 포함하는 서버측 네트워크 노드의 일부일 수 있고, 상기 컴퓨터 프로그램들은 협업 서버에 대한 사용자 액세스를 지원하는 로직을 제공한다. 상기 로직은 로그인 자원들을 갖는 웹페이지, 사용자 계정들 및 세션 예상을 관리하기 위한 로직, OAuth에 기초한 서비스들과 같은 인가 서비스들을 제공하는 로직 및 계정 데이터와 같은 사용자들에 대한 초기 입력 포인트들을 제공한다.

[0058] 협업 서비스(601)는 세션 이벤트 데이터를 포함하는 서버측 네트워크 노드의 일부일 수 있고, 상기 세션 이벤트 데이터를 관리하고, 클라이언트들 간의 업데이트된 이벤트들을 조정하고, 클라이언트들에게 캐치 가능한 이력 및 이미지들을 전달하며, 작업공간 데이터에 저장된 데이터베이스에 대한 액세스를 제어한다.

[0059] 도 7은 연합 제어를 갖는 복수 개의 디스플레이들(701-704)에 의해 구현되는 디스플레이에 기초해서 디스플레이 벽의 구현을 위한 선택적 기술을 설명한다. 이 예시에서, 디스플레이(701-704) 각각은 대응하는 디스플레이 클라이언트(711-714)와 관련된다. 디스플레이 클라이언트 각각은 디스플레이 영역에 작업공간으로부터의 오브젝트들을 렌더링하기 위해 사용되는 브라우저를 실행할 수 있고, 상기 디스플레이 영역은 복수 개의 디스플레이들 각각에 대응하는 복수 개의 디스플레이 영역의 서브세트들을 갖는다. 디스플레이 클라이언트 각각은, 예를 들어, 오프셋 파라미터(예를 들어, 디스플레이(701)에 대해 0, 0; 디스플레이(702)에 대해 0,1; 디스플레이(703)에 대해 1,0; 디스플레이(704)에 대해 1,1)를 저장함으로써, 세션에 대한 디스플레이 영역의 서브세트의 디스플레이를 관리하도록 구성된다.

[0060] 디스플레이 클라이언트들(711-714) 각각은 협업 서버(105)와의 통신 채널(721-724)을 유지할 수 있고, 이는 또한 작업공간 데이터베이스(106)에 결합된다. 협업 서버(105) 및/또는 클라이언트는 인가된 사용자 각각에 대한 작업공간 내의 사용자 위치를 유지한다. 인가된 사용자가 로그인하고 디스플레이 캔버스로서 도 7에 도시된 것과 같은 디스플레이 어레이를 선택했을 때, 협업 서버는 디스플레이 클라이언트들(711-714) 각각을 상기 세션 및 사용자와 관련된 그룹에 링크할 수 있다. 상기 협업 서버는 그 후에 상기 그룹의 디스플레이 클라이언트들 각각에 대해 디스플레이 가능한 영역 또는 캔버스에 작업공간 내의 현재 사용자 위치를 다운로드한다. 상기 그룹의 디스플레이 클라이언트들은 오프셋 파라미터에 의해 표시된 작업공간의 서브세트에 매핑하기 위한 세션 위치들을 식별하기 위해서 이들의 오프셋 파라미터들을 독립적으로 적용한다. 한 대안에서, 협업 서버는 어레이 특징들에 따른 오프셋으로서 현재 사용자 위치를 각 클라이언트에 전달함으로써 디스플레이 클라이언트들(711-714) 각각과 통신하는 오프셋 계산을 관리한다.

[0061] 복수 개의 디스플레이들 중의 단일 디스플레이의 조정을 지원하기 위해서, 디스플레이 클라이언트들(711-714) 각각은 또한 디스플레이 영역의 관리에 대해 로컬이고 글로벌 작업공간 데이터에 영향을 주지 않는 이벤트들을 갖는 다른 디스플레이 클라이언트들 각각과 통신한다. 대안적으로, 디스플레이 클라이언트(711-714)는 협업 서버(105)와 단독으로 통신하고, 협업 서버(105)는 그 후에 로컬 이벤트들을 상기 세션과 관련된 디스플레이 클라이언트들의 그룹에 다시 다이렉팅(directing)하고, 글로벌 이벤트들을 상기 작업공간을 갖는 활성 세션들의 디스플레이 클라이언트들 모두 및 작업공간 데이터를 저장하는 데이터베이스에 다이렉팅한다.

[0062] 연합 디스플레이들로 구성되는 단일 디스플레이의 디스플레이 클라이언트들은 대응하는 디스플레이들에 결합된 개별 컴퓨터 시스템들로 구현되거나, 대응하는 디스플레이들에 결합된 가상 머신들을 갖는 단일 컴퓨터 시스템을 이용해서 구현된다.

[0063] 또한, 단일 디스플레이 구동기는 단일 디스플레이 벽으로서 배열된 물리적 디스플레이들의 모음의 전체 표면을 제어하도록 구성된다.

[0064] 공간 이벤트 맵 시스템은 어떤 수의 물리적 및 가상 머신들을 포함하는 클라이언트측 및 서버측 자원들에 의해 조정되어 실행되는 API를 포함한다. API의 일 예시는 아래에 설명된다. API는 다양한 방식들로 정의될 수 있고, 한편 서버측 네트워크 노드 또는 노드들의 공간 이벤트 맵의 유지를 지원하고 공간 이벤트 맵의 하나 또는 복수 개의 활성 클라이언트측 네트워크 노드들과의 공유를 지원하는 요소들을 포함한다. 이 예시에서, API는 2개의 서버들에 의해 관리되는 프로세스들로 나뉜다.

[0065] 소켓 요청 서버(웹소켓들) - 일단 연결된 관련 데이터(새로운 스트로크들, 카드들, 클라이언트들 등)를 갖는 클라이언트들을 업데이트하기 위해 사용된다. 또한 초기 연결 핸드셰이크(initial connection handshake)를 핸들링한다.

[0066] 서비스 요청 서버(HTTP/REST) - 캐시 가능한 응답들에 대해서뿐만 아니라 데이터(예를 들어, 이미지들 및 카드

들)를 포스팅하기 위해 사용된다.

[0067] 클라이언트측 네트워크 노드들은 API에 따라 구성되고, 대응하는 소켓 요청 클라이언트들 및 서비스 요청 클라이언트들을 포함한다.

[0068] 소켓 요청들:

[0069] 소켓 서버는 웹소켓들을 통해 연결들을 유지하는 네트워크 프로토콜을 실행한다. API에서 사용되는 메시지들은 웹소켓 프로토콜 내에 압축될(encapsulated) 수 있다. 메시지들은 개별의 UTP-8에 의해 인코딩된 JSON 어레이들일 수 있다.

[0070] 클라이언트측 네트워크 노드들에서의 협업 세션의 공간 이벤트 맵의 전부 또는 일부를 포함하는 이력의 초기 로딩은 캐싱(caching)을 지원하기 위한 웹소켓들보다는, 서비스 요청 서버를 통한 HTTP 요청들을 이용해서 수행된다.

[0071] 소켓 연결

[0072] http://localhost:4545/<sessionId>/socket?device=<device>

[0073] sessionId - - (string) 참여할 세션의 ID

[0074] device - - (string) 벽 또는 데스크탑과 같은 디바이스 타입

[0075] 메시지 구조

[0076] 각각의 메시지의 제1 요소는 상기 메시지가 발신된(originated) 클라이언트를 특정하는 송신자 ID이다. 송신자 ID들은 상기 서버의 모든 세션들 중에서 고유하다. 상기 서버로부터 상기 클라이언트로 송신한 상기 클라이언트 ID 및 CR 메시지들은 -1과 같은 기본값으로 설정된 이들의 송신자 ID를 갖는다. 각 메시지 어레이의 제2 요소는 2글자로 된 코드이다. 이 코드는 어레이의 나머지 인수들(arguments)뿐만 아니라 의도된 액션(action)을 정의한다. -1의 송신자 ID와 함께 송신된 메시지들은 상기 서버로부터 발신된 메시지들이다.

[0077] 유효한 메시지 타입들

[0078] 다음은 본 명세서의 API 예시에 의해 지원되는 메시지들이다. 이러한 메시지들 중 다수는 다음 파라미터를 취한다:

[0079] sender-id: - - 상기 메시지를 송신하는 클라이언트의 ID, 또는 만약 상기 메시지가 서버로부터 발신된다면 -1

[0080] 클라이언트 ID 요청:

[0081] // server <-- client ["id", sessionId, zoomLevel, x1, y1, x2, y2]

[0082] 이 요청은 소켓 API와의 인터렉션을 인에이블하도록 사용된다.

[0083] 이 요청은 비동기화된 클라이언트 ID 요청/응답 핸드셰이크(asynchronous client-id request/response handshake)를 시작한다. 다음 섹션은 (제공된 클라이언트 ID를 포함하는)새로운 클라이언트의 비동기화된 확인(asynchronous acknowledgement)을 설명한다.

[0084] SessionId - - (string) 참여할 작업공간의 ID

[0085] zoomLevel - - (int) 이 클라이언트가 원하는 줌 레벨

[0086] x1, y1 - - (int, 선택적) 사용자 뷰포트에 대한 원점의 원하는 포인트

[0087] x2, y2 - - (int, 선택적) 사용자 뷰포트에 대한 범위의 원하는 포인트

[0088] 이 메시지와 함께 보내지는 송신자 ID는 존재하지 않는다.

[0089] 이 메시지에서 송신되는 줌 레벨은 이 클라이언트에 의해 선호되는 줌 레벨이다. 만약 상기 클라이언트가 비어있는 디스플레이 어레이에 ("ID" 메시지를 통해) 참여한다면, 클라이언트의 선호되는 줌 레벨은 디스플레이 어레이에 대한 초기 줌 레벨이 된다. 만약 클라이언트가 기존의 디스플레이 어레이에 참여한다면, 상기 디스플레이 어레이의 "ID" 메시지에 송신된 선호되는 줌 레벨은 무시되고, 기존의 디스플레이 어레이와 관련된 줌 레벨이 ("AV" 메시지에서) 송신된다.

[0090] 클라이언트 ID 응답:

[0091] // server --> client [-1, "id", client-id]

[0092] 클라이언트들은 후속의 소켓 요청들에서의 사용을 위해 할당된 클라이언트 ID를 저장하도록 요구된다. 새 클라이언트에게 이들의 ID를 알려준다. 이 경우에, 송신자 ID는 -1로 설정된다.

[0093] client-id - - (string) 새롭게 참여한 클라이언트의 ID

[0094] 공간에 참여:

[0095] // server <-- client [sender-id, "jr", room-id, [data]] [sender-id, "jr", "lobby"][sender]

[0096] 상기 서버에게 클라이언트에 의한 공간에 참여하려는 시도를 알려준다.

[0097] room-id - - 로비 또는 세션 중 하나를 포함한다

[0098] data - - 인수들의 와일드카드 세트이고, 이는 상기 공간을 초기화하기 위해서 사용되어야 한다.

[0099] 공간 데이터 인수들:

[0100] 세션은 상기 세션의 ID를 포함하는 "세션 ID"를 요구한다. 어레이는:

[0101] arrayId - - (string) 디스플레이 어레이의 ID

[0102] x - - (integer) 이 디스플레이의 x 오프셋

[0103] y - - (integer) 이 디스플레이의 y 오프셋

[0104] width - - (integer) 이 디스플레이의 폭

[0105] height - - (integer) 이 디스플레이의 높이

[0106] 서버는 이에 응답하여 "공간" 메시지로 응답할 것이다.

[0107] 공간 참여 응답:

[0108] // server --> client [-1, "room", [room-id], [databag]] [-1, "room", "lobby", {pin: pin}]

[0109] room-id - - 로비 또는 세션 중 하나를 포함

[0110] databag - - 변수들의 공간별 백(room-specific bag):

[0111] 로비는:

[0112] pin - - 벽 인증을 위한 핀을 포함한다

[0113] 을 제공한다.

[0114] 세션은:

[0115] sessionName - - 세션의 이름을 포함한다

[0116] 을 제공한다.

[0117] 공간 리스트 응답

[0118] // server --> client [-1, "rl", roomMembershipList]

[0119] 클라이언트에게 공간 멤버십들에 대해 알려준다. 공간 멤버십들은 당신과 동일한 공간을 방문하는 클라이언트들에 관한 정보를 포함한다

[0120] roomMembershipList - - (공간 멤버십 오브젝트들의 어레이)

[0121] 세션 요청:

[0122] // server <-- client [sender-id, "sr"]

[0123] 서버에게 클라이언트가 참여가능한 활성 세션들의 리스트를 원한다는 것을 알려준다.

[0124] 세션 리스트:

[0125] // server --> client [-1, "sl", sessionList]

[0126] 클라이언트에게 참여가능한 활성 세션들을 알려준다.

[0127] SessionList - - (array of strings)

[0128] 오브젝트 ID 예약:

[0129] 오브젝트를 생성하는 새로운 이력 이벤트들을 생성하기 위해 수용가능한 새로운 고유 오브젝트 ID를 생성하기 위해 이를 사용한다.

[0130] // server <-> client [sender-id, "oid"]

[0131] 서버는:

[0132] [-1, 'oid', <new-object-id>]

[0133] 로 응답한다.

[0134] 이력 이벤트

[0135] 모든 지속 이벤트들은 HistoryEvent에 송신된다. 이는: ** 이동 윈도우들 ** 텍스트 설정 ** 윈도우 제거 ** 윈도우 생성을 포함한다.

[0136] HistoryEvent들은 세션의 이력에 기록되고 이력이 검색될 때 반환된다.

[0137] HistoryEvent들은 eventID 없이 서버에 송신된다. 서버는 (발신 클라이언트를 포함하여) 모든 클라이언트들에 대해 eventID를 할당하고 상기 이벤트를 방송한다.

[0138] 새로운 오브젝트 ID들은 oid 메시지를 이용해서 예약된다.

[0139] 기본 메시지 포맷

[0140] // server <-- client [client-id, "he", target-id, event-type, event-properties]

[0141] client-id - - (string) 발신 클라이언트의 ID

[0142] target-id - - (string) 이 이벤트에 관련된 타겟 오브젝트/위젯/앱의 ID

[0143] event-type - - (string) 임의의 이벤트 타입

[0144] properties - - (object) 상기 이벤트에 대한 지속 키들/값들을 설명하는 JSON 오브젝트

[0145] // server --> client[client-id, "he", target-id, event-id, event-type, event-properties

[0146] client-id - - (string) 발신 클라이언트의 ID

[0147] target-id - - (string) 이 이벤트가 관련된 타겟 윈도우의 ID

[0148] event-id - - (string) 데이터베이스의 이벤트의 ID

[0149] event-type - - (string) 임의의 이벤트 타입

[0150] properties - - (object) 상기 이벤트에 대한 지속 키들/값들을 설명하는 JSON 오브젝트

[0151] 예시적인 인터렉션: 이동하는 오브젝트

[0152] 이력 이벤트/일시 이벤트 분류 중 일부를 설명하는 좋은 예시는 오브젝트를 움직이는 것이다. 오브젝트가 이동되거나 예를 들어 드래깅에 의해 사이즈가 조절되는 동안, 일련의 일시 이벤트들("휘발성 이벤트들 VEs"로 지칭됨)이 서버에 송신되고, 세션의 모든 클라이언트들에게 재방송된다. 일단 사용자가 오브젝트를 이동하는 것을 끝마치면, 클라이언트는 오브젝트의 rect 및 order를 특정하기 위해 이력 이벤트를 송신해야 한다.

[0153] ["511d6d429b4aee0000000003","he","511d6f9c9b4aee0000000039","position",{ "rect" }

[0154] 서버는 새롭게 지속되는 HE 기록으로 응답할 것이다. 기록의 eventID의 포함을 주목하자.

[0155] // server-> client format of 'he' is: [<clientId>, <messageType>, <targetId>, <eventId>,

[0156] 주목: eventID는 HTTP API를 통해 페치되는(fetched) 이력에 또한 포함된다.

[0157] 오브젝트/애플리케이션 타입 세션에 의한 이력 이벤트들

[0158] Create - - 작업 세션에 노트 또는 이미지를 추가

[0159] stroke - - 배경에 펜 또는 지우개 스트로크를 추가

[0160] 노트

[0161] text - - 텍스트 및/또는 노트의 텍스트 포맷화를 설정 또는 업데이트

[0162] delete - - 작업 세션으로부터 노트를 제거

[0163] position - - 작업 세션의 노트의 사이즈 또는 위치를 업데이트

[0164] pin - - 상기 노트를 핀(pin) 또는 핀 해제(unpin)함

[0165] stroke - - 이미지의 상단에 펜 또는 지우개 스트로크를 추가

[0166] 이미지

[0167] delete - - 작업 세션으로부터 노트를 제거

[0168] position - - 작업 세션의 노트의 사이즈 또는 위치를 업데이트

[0169] pin - - 상기 노트를 핀 또는 핀 해제함

[0170] stroke - - 이미지의 상단에 펜 또는 지우개 스트로크를 추가

[0171] 이력 이벤트 상세사항들

[0172] 텍스트(text)

[0173] 노트의 텍스트를 설정하고 스타일링한다. 텍스트 속성 및 스타일 속성 둘 모두는 선택적이다.

[0174] // server <-- client[client-id, "he", target-id, "text", { "text" : "abcdef",

[0175] 생성

[0176] 서버가 카드 생성(cc; card create) 메시지 또는 이미지 업로드를 수신할 때 클라이언트들에게 송신된다.

[0177] 메시지들의 생성에 대해서는 타겟 ID가 세션 ID이다.

[0178] // server --> client[client-id, "he", session-id, event-id, "create", { "id":"5123e7ebcd18d3ef5e000001"

[0179] 속성들

[0180] id - - (int) 윈도우에 대한 고유 식별자

[0181] baseName - - (string) 배경 이미지 파일 이름

[0182] ext - - (string) 배경 이미지 파일 확장자(file extension)

[0183] rect - - (object) 윈도우의 위치

[0184] actualWidth - - (int) 배경 이미지 폭

[0185] actualHeight - - (int) 배경 이미지 높이

[0186] order - - (int) z 차수

[0187] type - - (string) 텍스트를 갖는 오브젝트들에 대해서는 "note", 기타 오브젝트들에 대해서는 "image"

[0188] regionId - - (string) 만약 오브젝트가 캔버스 영역에 생성된다면 상기 캔버스 영역

[0189] hidden - - (boolean) 윈도우가 현재 감춰져 있는지 여부

[0190] 삭제(delete)

[0191] 윈도우가 세션으로부터 사라지도록 만들기 위해 사용된다. 삭제는 되돌리기 가능한 액션이다.

[0192] // server <-- client[client-id, "he", target-id, "delete", {"hidden":true}]// server -->

[0193] 위치(position)

[0194] 이동, 플링(fling) 또는 사이즈 조절 후의 윈도우의 위치를 저장하기 위해 사용된다.

[0195] // server <-- client[client-id, "he", target-id, "position", {"rect":[-1298,-390,-1018

[0196] 속성들

[0197] rect - - (object) 타겟 윈도우의 denlcl

[0198] order - - (int) 타겟 윈도우의 z-차수

[0199] 스트로크(stroke)

[0200] 스트로크를 저장하기 위해 사용된다.

[0201] / server <-- client[client-id, "he", target-id, "stroke", { "size": 10, "brush":

[0202] 속성들

[0203] locs - - (array) [10, 1, 10, 2, 12, 3]의 포맷의 스트로크 위치들, 여기서 좌표들은 어레이에서 [x, y, x, y, x, y] 로 쌍을 이룬다.

[0204] 핀(pin)

[0205] 해당 위치의 노트 또는 이미지에 핀을 하거나 현존하는 핀을 제거하기 위해 클라이언트들에 송신된다. 핀이 된 윈도우들은 핀이 해제될 때까지는 이동되거나 사이즈가 조절될 수 없다.

[0206] // server --> client[client-id, "he", session-id, event-id, "pin", {"pin": true}]

[0207] 속성들

[0208] pin - - (boolean) 참은 핀, 거짓은 핀 해제

[0209] **휘발성 이벤트(Volatile Event)**

[0210] 휘발성 이벤트들은 되돌리기/플레이백 이벤트 스트림에 기록되지 않은 일시 이벤트들이어서, 휘발성 이벤트들은 스크린 주위에서 카드를 드래깅하는 것과 같은 프로세싱 중인 스트리밍 이벤트들을 위해 좋고, 일단 사용자가 이들의 손가락을 들어올리면, HistoryEvent는 손가락의 최종 위치를 기록하기 위해 사용된다.

[0211] // server <--> client[client-id, "ve", target-id, event-type, event-properties]

[0212] client-id - - (string) 발신 클라이언트의 ID

[0213] target-id - - (string) 이 이벤트에 관련된 타겟 윈도우의 ID

[0214] event-type - - (string) 임의 이벤트 타입

[0215] properties - - (object) 이벤트에 대한 관련 키/값들을 설명하는 JSON 오브젝트

[0216] 오브젝트/애플리케이션 타입에 의한 휘발성 이벤트들

[0217] 세션

[0218] sb - - 스트로크를 시작한다. 다른 클라이언트에서 드로잉되는 동안 한 클라이언트에 스트로크들을 렌더링하기 위해 사용된다.

[0219] sc - - 포함할 다른 포인트를 부여함으로써 이전에 시작된 스트로크를 계속한다. 다른 클라이언트에서 드로잉되는 동안 스트로크들을 렌더링하기 위해 사용된다.

[0220] se - - 이전에 시작된 스트로크를 종료한다.

[0221] 노트

[0222] fling - - 작업 세션의 한 위치로부터 다른 위치로 슬라이딩하는 노트를 애니메이팅(animating)한다. 이는 사용자에 의한 플릭 또는 플링(flick or fling) 액션에 대한 시각적 응답이다.

[0223] position - - 다른 사용자에 의해 이동되는 동안 노트의 위치를 실시간으로 업데이트한다.

[0224] sb - - 스트로크를 시작한다. 다른 클라이언트에서 드로잉되는 동앉 한 클라이언트에 스트로크를 렌더링하기 위해 사용된다.

[0225] sc - - 포함할 다른 포인트를 부여함으로써 이전에 시작된 스트로크를 계속한다. 다른 클라이언트에서 드로잉되는 동안 스트로크들을 렌더링하기 위해 사용된다.

[0226] se - - 이전에 시작된 스트로크를 종료한다.

[0227] 이미지

[0228] fling - - 작업 세션의 한 위치로부터 다른 위치로 슬라이딩하는 이미지를 애니메이팅한다. 이는 사용자에 의한 플릭 또는 플링 액션에 대한 시각적 응답이다.

[0229] position - - 이미지가 다른 사용자에 의해 이동되는 동안 이미지의 위치를 실시간으로 업데이트한다.

[0230] sb - - 스트로크를 시작한다. 스트로크들이 다른 클라이언트에 의해 드로잉되는 동안 한 클라이언트에서 스트로크들을 렌더링하기 위해 사용된다.

[0231] sc - - 포함할 다른 포인트를 부여함으로써 이전에 시작된 스트로크를 계속한다. 다른 클라이언트에서 드로잉되는 동안 스트로크들을 렌더링하기 위해 사용된다.

[0232] se - - 이전에 시작된 스트로크를 종료한다.

[0233] 휘발성 이벤트들의 타입들

[0234] 플링

[0235] 모든 연결된 클라이언트들에 대한 플링 액션을 방송하기 위해 사용된다

[0236] // server <--> client[client-id, "ve", target-id, "fling", {"velocityX": 10, "velocityY"

[0237] 속성들

[0238] velocityX (int) 플링 벡터의 x 컴포넌트

[0239] velocityY (int) 플링 벡터의 y 컴포넌트

[0240] 위치 - ve

[0241] 윈도우 이동의 중간 단계들을 방송하기 위해 사용된다.

[0242] // server <--> client[client-id, "ve", target-id, "position", {"rect":[-1298,-390,-1018

[0243] 속성들

[0244] rect (object) 타겟 윈도우의 위치

[0245] order (int) 타겟 윈도우의 z 차수

[0246] sb:

[0247] 스트로크의 시작을 방송하기 위해서 사용됨

[0248] // server <--> client[client-id, "ve", target-id, "sb",{ "brush":1, "size":2, "color"

[0249] 속성들

[0250] x,y - - (int) 이 스트로크의 시작 포인트

[0251] strokeId - - (string) 새로운 스트로크의 ID

[0252] sc:

[0253] // server <--> client[client-id, "ve", target-id, "sc", {"x":100, "y":300, "strokeId"

[0254] 스트로크의 연속을 방송하기 위해 사용됨

[0255] 속성들

[0256] x,y - - (int) 스트로크의 새로운 종료점(end-point)

[0257] strokeId - - (string) 새로운 스트로크의 ID

[0258] se:

[0259] // server <--> client[client-id, "ve", target-id, "se", "strokeId" : "395523d316e942b496a2c8a6fe5f2cac"

[0260] 스트로크 ID에 의해 특정된 스트로크를 종료

[0261] stroke-id - - (string) 계속되는 스트로크의 ID

[0262] 스트로크 삭제

[0263] // server --> client[sender-id, "sd", stroke-id, target-id]

[0264] 스트로크를 삭제

[0265] stroke-id - - (string) 스트로크의 ID

[0266] target-id - - (string) 스트로크 타겟의 ID

[0267] 되돌리기(undo)

[0268] 마지막 되돌리기 가능한 액션을 되돌림(이동, 텍스트 설정, 스트로크 등)

[0269] // server <-- client

[0270] [sender-id, "un"]// server --> client

[0271] [client-id, 'undo', target-

[0272] 되돌리기 예시: 윈도우를 이동한 후 그 이동을 되돌림

[0273] 다음 예시는 이동을 보여주고, 그 이동이 어떻게 되돌려지는지를 보여준다.

[0274] // 클라이언트는 이동 메시지를 송신한 후 되돌리기 메시지를 송신한다.

[0275] 서버는 세션 이력으로부터 이동의 이력 이벤트를 제거하고, 이 기록을 이력 타임라인으로부터 꺼내면서 클라이언트에게 이 기록이 더 이상 세션의 공간 이벤트 맵의 부분이 되지 않을 것이라는 점을 통지한다. HTTP API를 통한 이력의 미래 요청들은 (되돌리기 후까지) 되돌리기 이벤트를 포함하지 않을 것이다.

[0276] 디스플레이 어레이 차원들:

[0277] // server --> client [-1, "dd", arrayId, width, height]

[0278] 클라이언트들에게 전체 디스플레이 어레이 폭 및 높이에 대한 변경들을 통지한다. 이는 공간 이벤트 맵의 일부들의 로컬 디스플레이를 관리할 자원들을 갖는 클라이언트측 네트워크 노드와는 활용되지 않을 것이다.

[0279] arrayID - - (string) 디스플레이 어레이의 ID

[0280] width, height - - (integers) 픽셀 단위로 디스플레이 어레이의 폭 및 높이

[0281] 팬 어레이(pan array):

[0282] // client --> server [sender-id, "pa", newArrayOffsetX, newArrayOffsetY]

[0283] 팬의 서버에게 새로운 위치를 통지한다.

[0284] newArrayOffsetX, newArrayOffsetY - - (int) 패닝(panning)된 후의 디스플레이 어레이의 새로운 위치

[0285] 세션 변경:

[0286] // server --> client [sender-id, "cs", sessionId]

[0287] 디스플레이 어레이의 시블링들(siblings)에 세션이 변경되었음을 통지한다.

[0288] SessionId - -(string) SessionId는 스위칭할 세션의 ID이다

[0289] 줌 변경(Zoom change):

[0290] // client --> server [sender-id, "zc", zoomLevel, zoomCenterX, zoomCenterY]

[0291] 서버에게 줌이 요청되었음을 통지한다.

[0292] zoomLevel (integer) 트랜지션할 줌 레벨, 1 부터 11까지

[0293] zoomCenterX (integer) 줌의 원점의 x 좌표

[0294] zoomCenterY (integer) 줌의 원점의 y 좌표

[0295] 맵-모드 변경(Map-mode change):

[0296] // client --> server

[0297] [sender-id, "mm", zoomLevel, zoomCenterX, zoomCenterY]

[0298] 서버에 맵-모드가 요청되었음을 통지한다. 피상적으로, 수십 또는 수백 개의 줌-변경 메시지들이 그래픽 처리에 있어서 이들 간에 트위닝(tweening)하면서 속사포같이(rapid-fire) 송신되도록 의도되는 반면, 맵-모드 메시지는 서로 다른 트랜지션 효과들을 갖는 단일 줌 스냅으로 의도된다는 점을 제외하고는, 이는 줌 변경 메시지와 거의 동일하게 동작한다.

[0299] zoomLevel - - (integer) 트랜지션할 줌 레벨, 1부터 11까지

[0300] zoomCenterX - - (integer) 줌의 원점의 x 좌표

[0301] zoomCenterY - - (integer) 줌의 원점의 y 좌표

[0302] 카드 생성

[0303] // server <-- client

[0304] [sender-id, "cc", templateId, regionId, x, y, x2, y2]

[0305] templateId - - (string) 사용될 카드 템플릿의 ID

[0306] regionId - - (string) (만약 존재한다면) 송신 이벤트의 캔버스 영역 ID

[0307] x, y, x2, y2 - - (int) 새로운 카드에 대한 바람직한 rect

[0308] 사용자 허가들

[0309] // server --> client [sender-id, "up", permissions]

[0310] 인증된 사용자가 그 허가를 갖고 있는지를 표시하기 위한 허가 타입들 및 참/거짓의 해쉬를 허가한다. 현재 유일한 허가는 세션을 다른 사람들과 공유할 수 있는 사용자들을 표시하는 "can_share"이다.

[0311] 위치 저장

[0312] // client --> server [-1, "sp", zoomLevel, x, y]

[0313] 현재 스크린 위치를 저장한다. 재연결시에, 클라이언트는 이들을 이 위치로 돌려놓을 'zc'(줌-변경) 및 'pa'(팬-어레이) 메시지를 수신할 것이다.

[0314] zoomLevel (integer) 디바이스가 현재 있는 줌 레벨

[0315] x (integer) 스크린의 원점의 x 좌표

[0316] y (integer) 스크린의 원점의 y 좌표

[0317] 스트로크 ID들

[0318] 스트로크 ID들은 클라이언트에 의해 선택된다. 현재 클라이언트 ID들은 도트(dot)에 의해 구별되는 증가하는 정수(increasing integer)로 구성되는 송신자 ID이다. 이는 클라이언트 ID를 모든 클라이언트들 간의 서버 컨텍스트 내에서 고유하게 만들기 위함이다.

[0319] 타겟 ID들

[0320] 스트로크는 주 세션(main session) 위에 플로팅(floating)하는 서브-캔버스("카드"로_불림)와 같이 세션의 특정 타겟에 부착될 수 있다. 스트로크가 카드에 속하는 경우에 있어서, 타겟 ID 필드는 카드 ID를 포함할 것이다. 세션의 주 캔버스에 대해 예정된(destined) 스트로크들은 세션 이름과 동일한 이들의 타겟 ID를 구비함으로써 지정된다.

[0321] 연결 설정

[0322] 클라이언트가 서버와 새로운 웹소켓 연결을 설정할 때, 서버는 먼저 고유 클라이언트 ID를 선택하고 상기 고유 클라이언트 ID를 클라이언트에 대한 "ID" 메시지로 송신한다. 클라이언트는 그 후에 송신자 ID가 -1로 설정된 "pa" 메시지를 송신한다.

[0323] 그 후의 대표적인 흐름은 클라이언트가 상기 세션의 카드들에 관한 정보를 수신하기 위해서 HTTP GET "/:sessionId/objects" (아래에 기록됨)을 수행하는 것이다. 그 후에 클라이언트는 history URL들의 어레이를 수신하는 "/:sessionId/history"를 요청한다 (마찬가지로 아래에 기록됨). 이들은 캐시가능성(cachability)을 향상하기 위해서 배치들(batches)로 나뉘어진다. 클라이언트는 그 후에 로컬 캐시에 저장되지 않은 URL 각각을 GET하고, 상기 서버는 이러한 이력 배치들에 대한 스트로크 데이터에 응답할 것이다.

[0324] 서비스 요청들

[0325] 이력 북마크들의 리스트를 얻는다. 북마크 각각은 캐시된 스트로크 이력의 범위(span)이다.

[0326] curl http://localhost:4545/<sessionId>/history

[0327] sessionId

[0328] 당신이 이력을 얻는 세션의 이름

[0329] 응답 헤더들

[0330] HTTP/1.1 200 OKX-Powered-By: ExpressAccess-Control-Allow-Origin: *Access-Control-Allow-Headers:

[0331] 응답

[0332] ["/<sessionId>/history/<startTime>/<endTime>?b=1"]

[0333] ["/<sessionId>/history/<startTime>/<endTime>?

[0334] sessionId - - (string) 스위칭할 세션의 ID

[0335] startTime - - (integer) 시작 타임스탬프

[0336] endTime - - (integer) 종료 타임스탬프

[0337] b - - 캐시 버스터(cache buster)

[0338] 이력의 블럭을 검색

[0339] 시작 시간과 종료 시간 사이의 이력을 얻는다. 이력의 반환된 범위 각각에 대해 요청이 이루어져야 한다.

[0340] curl

[0341] http://localhost:4545/<sessionId>/history/<startTime>/<endTime>?b=<cache-buster>

[0342] sessionId - - 당신이 이력을 얻는 세션의 ID

[0343] startTime - - 초기 이력 요청에 의해 부여된 시작시간

[0344] endTime - - 초기 이력 요청에 의해 부여된 종료시간

[0345] cacheBuster - - 클라이언트에 저장된 캐시가 더 이상 유효하지 않을 때마다 변경될 간단한 키

[0346] 응답 헤더

[0347] HTTP/1.1 200 OKX-Powered-By: ExpressAccess-Control-Allow-Origin: *Access-Control-Allow-Headers: X-Requested-With Content-Type: application/json

[0348] Content-Length: 2134

[0349] ETag: 1346968307576

[0350] Date: Fri, 14 Sep 2012 17:35:14 GMT

[0351] Connection: 유지중

[0352] Response

[0353]
```
[
```

[0354]
```
   [
```

[0355]
```
       4,
```

[0356]
```
       "sx",
```

[0357]
```
       "4.4",
```

[0358]
```
       [537, 650, 536, 649, 536, 648, ...],
```

[0359]
```
       {
```

[0360]
```
           "size": 10,
```

[0361]
```
           "color": [0, 0, 0, 1],
```

[0362]
```
           "brush": 1
```

[0363]
```
       },
```

[0364]
```
       1347644106241,
```

[0365]
```
       "cardFling"
```

[0366]
```
   ]
```

[0367]
```
]
```

[0368] (sx "스트로크-완성" 웹소켓 메시지에 대한 문서 참조)

[0369] 오브젝트 검색:

[0370] 요청된 세션에 대한 오브젝트들(카드들/이미지들)을 얻는다.

[0371] curl http://localhost:4545/<sessionId>/objects

[0372] sessionId 당신이 이력을 얻는 세션의 ID

[0373] 응답 헤더

[0374] HTTP/1.1 200 OK

[0375] X-Powered-By: Express

[0376] Access-Control-Allow-Origin: *

[0377] Access-Control-Allow-Headers: X-Requested-With

[0378] Content-Type: application/json; charset=utf-8

[0379] Content-Length: 285

[0380] Date: Fri, 14 Sep 2012 17:35:14 GMT

[0381] Connection: keep-alive

[0382] Response

[0383] [

[0384] {

[0385] "eventType": "oc",

[0386] "id": "50536840ce64b39439000005",

[0387] "baseName": "sessions/all/green",

[0388] "ext": "JPEG",

[0389] "rect": [-239, 49, 361, 649],

[0390] "arrayId": 3,

[0391] "clientId": 3,

[0392] "regionId": null,

[0393] "sessionId": "cardFling",

[0394] "actualWidth": 600,

[0395] "actualHeight": 600,

[0396] "order": null,

[0397] "_id": "50536840ce64b39439000005",

[0398] "type": "image"

[0399] },

[0400] {

[0401] "eventType": "oc",

[0402] "id": "50536b66ce64b39439000006",

[0403] "baseName": "sessions/all/orange",

[0404] "ext": "JPEG",

[0405] "rect": [-97, 190, 503, 790],

[0406] "arrayId": 5,

[0407] "clientId": 5,

[0408] "regionId": null,

[0409] "sessionId": "cardFling",

[0410] "actualWidth": 600,

[0411] "actualHeight": 600,

[0412] "order": null,

[0413] "_id": "50536b66ce64b39439000006",

[0414] "type": "image"

[0415] }

[0416]]

[0417] 카드 템플릿들:

[0418] 캐시된, 재사용 가능한 카드들을 생성하기 위한 글로벌 카드 템플릿들의 리스트를 얻는다. 이는 이 템플릿을 이용해 생성된 모든 카드들에 대해 동일한 배경 이미지가 사용되기 때문에 파일을 업로드하는 것과 다르다.

[0419] curl http://localhost:4545/card_templates.json

[0420] Response

[0421] [

[0422] {

[0423] "id":"50901cb0b9a18c190902a938",

[0424] "width":600,

[0425] "thumbnail":"card_templates/thumbnails/pink.jpeg"

[0426] },

[0427] {

[0428] "id":"50901cb0b9a18c190902a939",

[0429] "width":600,

[0430] "thumbnail":"card_templates/thumbnails/green.jpeg"

[0431] }

[0432]]

[0433] 이러한 값들은 카드 메시지를 송신하기 위해 사용될 수 있다:

[0434] // 위의 핑크 템플릿을 이용해서 새로운 카드를 생성

[0435] ["cc", "50901cb0b9a18c190902a938", <regionIdOrNull>, <x>, <y>]

[0436] 업로드:

[0437] 상기 세션에 놓일 서버에 이미지를 송신한다.

[0438] curl -F "file=@photo.JPG" -F "x=236" -F "y=832" -F "clientId=10" -F "sessionId=cardFling"

[0439] -F "arrayId=10" -F "order=23" -F "x2=899" -F "y2=1495" -F "filename=photo.jpg"

[0440] http://localhost:4545/<sessionId>/object/upload

[0441] 파라미터들

[0442] x: 드랍(drop)의 x 위치

[0443] y: 드랍(drop)의 y 위치

[0444] clientId: 클라이언트 Id

[0445] sessionId: 세션 Id

[0446] arrayId: 어레이 식별자

[0447] order: z 차수

[0448] x2: 드랍의 하단 우측 코너의 x 위치

[0449] y2: 드랍의 하단 우측 코너의 y 위치

[0450] filename: 업로드된 파일의 이름

[0451] 위에 설명된 API는 하나의 예시적인 메시지 구조를 제공한다. 특정한 구현에 맞는 다른 구조들이 또한 활용될 수 있다.

[0452] 협업 시스템의 일 실시예에서, 애플리케이션 프로그램 인터페이스 API는 도 6을 참조로 하여 제안되는 바와 같이 디스플레이 클라이언트 각각에 대해 2개의 통신 채널들에 기초해서 협업 서버(105) 및 디스플레이 클라이언트들에 의해 실행된다.

[0453] 일부 실시예들에서 연합 디스플레이가 배치될 수 있다. 연합 디스플레이에서, 디스플레이 클라이언트들(711-714) 각각은 서버와 2개의 채널들을 독립적으로 유지한다. 제1 채널은 실시간 이벤트들에 관한 통신들을 하도록 구성된 시스템에 기초한 메시지이다. 일 예시에서, 제1 채널은 협업 서비스(601)와 웹소켓 채널을 통해 소켓 요청들의 세트를 이용해서 구현되며, 일단 연결되면 관련 데이터(새로운 스트로크들, 카드들, 클라이언트들 등)를 이용해서 클라이언트들을 업데이트 하기 위해 서버에 의해 사용된다. 상기 메시지에 기초한 채널은 또한 초기 연결 핸드셰이크를 핸들링할 수 있다. 제2 채널은 n HTTP/REST 인터페이스와 같이 캐시 가능한 응답들뿐만 아니라 데이터(예를 들어, 이미지들 및 카드들)를 포스팅하기 위해 사용되는 포털(602)과의 더욱 비상태인(more stateless) 타입 링크이다. 또한 디스플레이 클라이언트에 대한 작업 공간 데이터의 초기 로딩은 캐시를 지원하기 위해서 메시지에 기초한 채널(웹소켓)보다는 HTTP 요청들을 이용해서 수행될 수 있다.

[0454] 도 8은 분산된 협업 시스템에서 클라이언트측 기능들(예를 들어, 컴퓨터 시스템(110)) 또는 서버측 기능들(예를 들어, 서버(105))을 구현하기 위해서 사용되는 컴퓨터 시스템 또는 네트워크 노드의 단순화된 블럭도이다. 컴퓨터 시스템은 통상적으로 버스 서브시스템(1012)을 통해 다수의 주변 디바이스들과 통신하는 프로세서 서브시스템(1014)을 포함한다. 이러한 주변기기 디바이스들은 메모리 서브시스템(1026) 및 파일 저장 서브시스템(1028), 사용자 인터페이스 입력 디바이스들(1022), 사용자 인터페이스 출력 디바이스들(1020) 및 네트워크 인터페이스 서브시스템(1016)을 포함하는 저장 서브시스템(1024)을 포함한다. 입력 및 출력 디바이스들은 컴퓨터 시스템과의 사용자 인터랙션을 가능하게 한다. 통신 모듈(1016)은 통신 네트워크(104)에 대한 인터페이스를 포함하여 외부 네트워크들에 인터페이스들에 대한 물리적 지원 및 통신 프로토콜 지원을 제공하고, 다른 컴퓨터 시스템들의 대응하는 통신 모듈들에 통신 네트워크(104)를 통해 결합된다. 통신 네트워크(104)는 많은 상호연결된 컴퓨터 시스템들 및 통신 링크들을 포함한다. 이러한 통신 링크들은 유선 링크들, 광학 링크들, 무선 링크들, 또는 정보의 통신을 위한 어떤 다른 메커니즘들일 수 있지만, 적어도 통신 네트워크의 맨 끝에서 통상적으로 이러한 통신 링크들은 IP에 기초한 통신 네트워크이다. 비록 일 실시예에서, 통신 네트워크(104)는 인터넷이지만, 다른 실시예들에서 통신 네트워크(104)는 어떤 적합한 컴퓨터 네트워크이다.

[0455] 네트워크 인터페이스들의 물리적 하드웨어 컴포넌트는 비록 이들이 카드의 형태일 필요는 없지만, 종종 네트워크 인터페이스 카드들(NIC들)로서 언급된다. 예를 들어 이들은 집적 회로들(IC들) 및 마더보드에 직접 맞춰진 연결부들의 형태일 수 있거나, 컴퓨터 시스템의 다른 컴포넌트들과 함께 단일 집적 회로 칩에 제작된 매크로셀의 형태일 수 있다.

[0456] 사용자 인터페이스 입력 디바이스들(1022)은 키보드, 마우스, 트랙볼, 터치패드와 같은 포인팅 디바이스들, (넓은 포맷의 디지털 디스플레이(102c)의 터치 감응 부분들을 포함하는) 디스플레이에 통합된 터치 스크린, 음성 인식 시스템들, 마이크로폰들 및 다른 타입들의 유형의 입력 디바이스들과 같은 오디오 입력 디바이스들을 포함한다. 일반적으로, "입력 디바이스"라는 용어의 사용은 컴퓨터 시스템에 또는 컴퓨터 네트워크(104)에 정보를 입력하기 위한 많은 가능한 타입들의 디바이스들 및 방식들을 포함하도록 의도된다.

[0457] 사용자 인터페이스 출력 디바이스들(1020)은 디스플레이 서브시스템, 프린터, 팩스 머신, 또는 오디오 출력 디바이스들과 같은 비-시각 디스플레이들을 포함한다. 디스플레이 서브시스템은 CRT(cathode ray tube), LCD(liquid crystal display)와 같은 플랫 패널 디바이스, 프로젝션 디바이스, 또는 가시 이미지를 생성하기 위한 어떤 다른 메커니즘을 포함한다. 도 1b의 실시예에서, 디스플레이 서브시스템은 넓은 포맷의 디지털 디스플레이(102c)의 디스플레이 기능들을 포함한다. 디스플레이 서브시스템은 또한 오디오 출력 디바이스들과 같은 비-시각적 디스플레이를 제공한다. 일반적으로, "출력 디바이스"라는 용어의 사용은 상기 컴퓨터 시스템으로부터 사용자에게 또는 다른 머신 혹은 컴퓨터 시스템에 정보를 출력하기 위한 모든 가능한 타입들의 디바이스들 및 방식들을 포함하도록 의도된다.

[0458] 저장 서브시스템(1024)은 본 발명의 특정 실시예들의 기능성을 제공하는 기초 프로그래밍 및 데이터 구성물들을 저장한다.

[0459] 서버측 네트워크 노드들의 구현을 위해 사용될 때의 저장 서브시스템(1024)은 작업 공간에서 이벤트들을 위치시

키는 공간 이벤트 맵을 포함하는 기계 판독가능 데이터 구조를 저장하는 비일시적 컴퓨터 판독가능 매체를 포함하는 물(product)을 포함하고, 상기 공간 이벤트 맵은 이벤트들의 로그를 포함하며, 상기 로그의 입력들은 작업 공간의 이벤트의 그래픽 타겟의 위치 및 시간을 갖는다. 또한, 저장 서브시스템(1024)은 서버측 네트워크 노드와 관련된 본 명세서에서 설명된 절차들을 수행하기 위한 실행가능한 명령어들을 포함하는 물을 포함한다.

[0460] 클라이언트측 네트워크 노드들의 구현을 위해 사용될 때의 저장 서브시스템(1024)은 아래에 설명된 바와 같이 캐시된 복사본의 형태의 공간 이벤트 맵을 포함하는 기계 판독가능 데이터 구조를 저장하는 비일시적인 컴퓨터 판독가능 매체를 포함하는 물을 포함하고, 상기 공간 이벤트 맵은 작업공간에 이벤트들을 위치시키고, 공간 이벤트 맵은 이벤트들의 로그를 포함하며, 로그의 입력들은 작업 공간의 이벤트의 그래픽 타겟의 위치 및 시간을 갖는다. 또한, 저장 서브시스템(1024)은 클라이언트측 네트워크 노드와 관련된 본 명세서에서 설명된 절차들을 수행하기 위한 실행가능한 명령어들을 포함하는 물을 포함한다.

[0461] 예를 들어, 본 발명의 특정 실시예들의 기능성을 구현하는 다양한 모듈들은 저장 서브시스템(1024)에 저장된다. 이러한 소프트웨어 모듈들은 일반적으로 프로세서 서브시스템(1014)에 의해 실행된다.

[0462] 메모리 서브시스템(1026)은 통상적으로 프로그램 실행 동안 명령어들 및 데이터의 저장을 위한 주 랜덤 액세스 메모리(RAM)(1030) 및 고정 명령어들이 저장된 판독 전용 메모리(ROM)(1032)를 포함하는 다수의 메모리들을 포함한다. 파일 저장 서브시스템(1028)은 프로그램 및 데이터 파일들에 대한 지속 저장소를 제공하고, 하드 디스크 드라이브, 관련된 이동식 매체와 함께 플로피 디스크 드라이브, CD ROM 드라이브, 광학 드라이브 또는 이동식 매체 카트리지들을 포함한다. 본 발명의 특정 실시예들의 기능성을 구현하는 데이터베이스들 및 모듈들은 하나 이상의 CD-ROM들과 같은 컴퓨터 판독가능 매체에 제공되었고, 파일 저장 서브시스템(1028)에 의해 저장된다. 호스트 메모리(1026)는 다른 것들 중에서도 특히, 프로세서 서브시스템(1014)에 의해 실행될 때 컴퓨터 시스템으로 하여금 본 명세서에서 설명된 기능들을 동작 또는 수행하도록 하는 컴퓨터 명령어들을 포함한다. 본 명세서에서 사용되는 바와 같이, "호스트" 또는 "컴퓨터"에서 또는 그 상에서 실행된다고 언급되는 프로세스들 및 소프트웨어는, 그러한 명령어들 및 데이터에 대한 어떤 기타 로컬 또는 원격 저장소를 포함하는 호스트 메모리 서브시스템(1026)의 컴퓨터 명령어들 및 데이터에 응답하여 프로세서 서브시스템(1014)에서 실행된다.

[0463] 버스 서브시스템(1012)은 컴퓨터 시스템의 다양한 컴포넌트들 및 서브시스템들이 의도된 대로 서로 통신하도록 하는 메커니즘을 제공한다. 비록 버스 서브시스템(1012)은 개략적으로 단일 버스로서 도시되지만, 버스 서브시스템의 대안적인 실시예들은 다수의 버스들을 이용한다.

[0464] 컴퓨터 시스템 자체는 퍼스널 컴퓨터, 휴대용 컴퓨터, 워크스테이션, 컴퓨터 단말, 네트워크 컴퓨터, 텔레비전, 메인프레임, 서버 팜(server farm) 또는 어떤 기타의 데이터 프로세싱 시스템 또는 사용자 디바이스를 포함하는 다양한 타입들일 수 있다. 일 실시예에서, 컴퓨터 시스템은 몇몇의 컴퓨터 시스템들을 포함하고, 컴퓨터 시스템 각각은 넓은 포맷의 디스플레이(102c)를 구성하는 타일들 중 하나를 제어한다. 컴퓨터들 및 네트워크들의 계속 변화하는 속성(ever-changing nature) 때문에, 도 8에 묘사된 컴퓨터 시스템(110)의 설명은 본 발명의 바람직한 실시예를 설명하기 위한 목적들을 위해서 특정 예시로서만 의도된다. 도 8에 묘사된 컴퓨터 시스템보다 많거나 적은 컴포넌트들을 갖는 컴퓨터 시스템의 많은 다른 구성들이 가능하다. 동일한 컴포넌트 및 변형들은 또한 도 1의 협업 환경에서 기타 디바이스들(102) 각각을 구성할 뿐만 아니라, 협업 서버(105) 및 디스플레이 데이터베이스(106)를 구성한다.

[0465] 디지털 디스플레이(102c) 상의 활성인 드로잉 영역들에 관한 특정 정보는 디스플레이 클라이언트의 컴퓨터 시스템(110)에 액세스 가능한 데이터베이스에 저장된다. 상기 데이터베이스는 MongoDB 데이터베이스, XML 데이터베이스, 관계형 데이터베이스(relational database), 또는 객체 지향 데이터베이스(object oriented database)를 포함하나 이에 한정되지 않는 많은 형태들의 서로 다른 실시예들을 취할 수 있다. 도 9는 데이터베이스가 포함하는 특정 정보 및 상기 데이터 간의 특정 관계를 설명하는 개략도이다.

[0466] 본 명세서에서 설명되는 실시예들에서, 드로잉 영역 각각은 툴바의 차일드(child)인 것으로 고려된다. 벽 배경의 포인트의 터치는 툴바를 스포닝(spawning)하고, 이는 또한, (비록 드로잉 영역이 열릴 때까지는 툴바가 필연적으로 가시적일 필요는 없지만) 드로잉 영역을 스포닝한다. 유사하게, 드로잉 영역을 닫기 위해서, 사용자는 드로잉 영역의 툴바의 "닫기" 아이콘을 터치한다. 그러므로 도 9에서, 데이터베이스는 하나 이상의 툴바 ID들(1110)에 의해 헤딩된다(headed). 각 툴바 ID(1110)는 툴바의 수평 위치, 상기 툴바의 드로잉 영역의 좌측 에지의 수평 위치 및 상기 드로잉 영역에 대한 드로잉 속성들의 세트를 표시하면서 데이터의 각각의 블럭(1112)을 포함하거나 포인팅한다. 좌측보다는 드로잉 영역의 우측 에지 위치를 특정하고, 드로잉 영역 폭보다는 반대 에지 위치를 특정하는 것과 같은 많은 변형들이 가능함이 이해될 것이다. 일 실시예에서, 상기 툴바는 동일한 수

직 위치에 항상 남아 있기 때문에, 툴바 위치는 수평값만을 갖는다. 다른 실시예에서 수평 및 수직 위치들 둘 모두가 특정될 수 있다.

[0467] 드로잉 속성들은 드로잉 속성들의 어레이(1114)를 포함하거나 이를 포인팅하며, 드로잉 속성 각각은 하나 이상의 값들과 관련된다. 도 9의 드로잉 속성들은 브러시 타입을 포함하고, 브러시 타입의 값은 예를 들어, "페인트," "잉크," "크레용," "마커" 또는 "지우개"를 표시하며, 이들 각각은 디스플레이(102c)에 드로잉될 때 서로 다른 겉보기 특성을 갖는다. 도 9의 드로잉 속성들은 브러시 폭을 또한 포함하고, 이는 이용 가능한 값들의 범위의 어떤 값을 취할 수 있다. 도 9의 드로잉 속성은 브러시 색상을 포함하고, 브러시 색상은 3개의 관련된 값들을 갖는다: 빨간색, 녹색 그리고 파란색 컨텐츠. 본 명세서에서 사용되는 바와 같이, 3개의 속성들인 브러시 타입, 브러시 폭 및 브러시 색상은 "라인 겉보기 속성들"을 구성하는 것으로 고려된다. 다양한 실시예들에서 드로잉 속성들(1114)은 또한 라인의 위치 또는 라인의 부분의 위치에 영향을 주는 기타 속성들을 포함한다. 이러한 속성들은 코너-라운딩 반경(corner-rounding radius) 또는 Bezier 곡선 파라미터들과 같은 속성들을 포함한다. 도 11에서 볼 수 있는 바와 같이, (라인 겉보기 속성들을 포함하여) 서로 다른 드로잉 영역들에 대한 드로잉 속성들이 반드시 동일해야 한다는 요건은 존재하지 않는다. 이들은 서로 독립적으로 설정되어서, 이들이 동일할 필요는 없다. 통상의 경우에 이들은 동일하지 않을 것이다.

[0468] 디스플레이(102c)에 라인을 드로잉하기 위해서, 사용자는 라인의 드로잉을 표시하는 "드로잉 사용자 입력"을 제공한다. 다른 실시예들이 사용자로 하여금 손가락으로 드로잉하도록 하는 반면에, 도 1의 실시예에서, 라인의 드로잉을 표시하기 위해 스타일러스 만이 사용될 수 있다. 직관적으로, 사용자는 그래서 디스플레이(102c) 표면에 스타일러스를 터치하고 스타일러스를 라인에 대해 원하는 위치들을 따라 드래깅함으로써 드로잉 영역 내에 표시한다. 라인 드로잉 동작의 종료는 디스플레이(102c) 표면으로부터 스타일러스를 들어올림으로써 표시된다. 로컬 컴퓨터 시스템(110)은 라인의 포인트들이 어디에 위치될지를 사용자 입력으로부터 결정하고, 이들을 디스플레이(102c)에 디스플레이한다. 컴퓨터 시스템(110)은 또한 협업 서버(105)에 스트로크 정보를 송신하고(도 1b), 협업 서버(105)는 상기 정보를 협업 서버의 디스플레이 데이터베이스(106)에 기록하고 상기 세션을 공유하는 다양한 디바이스들(102)에 상기 정보를 다시 송신한다. 상기 디바이스들(102) 각각은 그 후에 상기 라인을 디스플레이하여(상기 라인이 디바이스의 뷰포트를 구분하는 한), 모든 그러한 디바이스들(102)은 대략적으로 동일한 시간에 상기 라인을 보여줄 것이다.

[0469] 도 10 내지 14는 컴퓨터 시스템들에 액세스 가능하고 서버, 디스플레이 클라이언트 또는 둘 모두에 의해 실행되는 로직을 설명하는 흐름도이다. 상기 로직은 프로세서들, 필드 프로그래밍 가능한 집적 회로들을 포함하는 전용 로직 하드웨어, 전용 로직 하드웨어와 컴퓨터 프로그램들의 조합들에 의해 실행 가능한 메모리에 저장된 컴퓨터 프로그램들을 이용하여 프로그래밍 되는 프로세서들을 이용하여 구현된다. 본 명세서의 흐름도들 모두와 함께, 상기 단계들 중 많은 단계들은 조합되고, 병렬로 수행되거나 달성되는 기능들에 영향을 줌이 없이 다른 시퀀스로 수행될 수 있음이 이해될 것이다. 일부 경우들에 있어서, 독자가 이해할 바와 같이, 단계들의 재정렬은 특정한 다른 변화들이 또한 이루어지는 경우에만 동일한 결과들을 달성할 것이다. 다른 경우들에 있어서, 독자가 이해할 바와 같이, 단계들의 재정렬은 특정한 조건들이 만족되는 경우에만 동일한 결과들을 달성할 것이다. 더욱이, 본 명세서의 흐름도들은 본 발명의 이해에 관한 단계들만을 도시함이 이해될 것이며, 기타 기능들을 달성하기 위한 다수의 추가적인 단계들이 도시된 단계들 전에, 후에, 그리고 그 사이에 수행될 수 있음이 이해될 것이다.

[0470] 도 10은 사용자가 지속적 작업 공간의 일부로서 세션에 참여할 때 서버측에서 실행되는 기초 로직을 설명한다. 흐름도는 사용자가 퍼스널 컴퓨터, 터치패드, 스마트폰 등과 같은 사용자에 의해 소유된 디바이스를 통해 웹 포털 액세스에 사용자 식별자를 입력하는, 사용자에 의한 로그인으로 시작한다(1210). 다음으로, 사용자 인증 프로토콜이 실행되고(1212), 여기서 프로토콜은, 예를 들어, 사용자로 하여금 개인 패스워드를 입력하도록 요구하고, 사용자가 실제로 사용자 식별자를 입력한 사람임을 검증하도록 요구하는 것을 포함한다. 다음으로, 예를 들어, 포털 머신을 이용하는 협업 서버는 인증된 사용자가 참여하도록 인증된 작업 공간들에 링크들을 제시한다(1214). 다음으로, 협업 서버는 선택된 디스플레이 클라이언트 및 사용자에 대한 선택된 작업 공간을 결정한다(1216). 이 결정은 사용자 소유 머신들 사이에 메시지들의 교환에 의해 이뤄지며, 상기 포털은 상기 인증 프로토콜이 실행되는 통신 채널을 이용한다. 디스플레이 클라이언트 및 작업 공간이 식별될 때, 협업 서버는 디스플레이 클라이언트가 선택된 작업 공간에 대한 데이터를 다운로드하도록 인에이블한다(1218). 또한, 협업 서버는 선택된 작업 공간에 대한 실시간 이벤트 채널에 클라이언트를 추가한다(1220).

[0471] 도 11은 사용자가 작업 공간에 참여할 때 클라이언트측에서 실행되는 기초 2채널 로직을 설명한다. 흐름도는 사용자가 사용자 식별자를 입력하고 사용자 식별자를 웹 포털에 송신하는, 제1 채널에서 실행되는 사용자에 의한

로그인(1230)으로 시작한다. 다음으로, 사용자 인증 프로토콜이 실행된다(1232). 사용자는 그 후에 인증 프로토콜이 실행되는 통신 채널을 이용해서 포털의 페이지를 열고, 상기 페이지는 인증된 작업 공간들에 대한 링크들을 디스플레이한다(1234). 다음으로, 사용자는 제1 채널을 이용해서 현재 세션에서 사용될 선택된 작업 공간 및 디스플레이 클라이언트를 식별하는 데이터를 입력한다(1236). 상기 서버가 선택된 디스플레이 클라이언트를 인에이블한 후에, 사용자 활동은 상기 디스플레이 클라이언트와 서버 간의 채널에 전달될 수 있고, 상기 채널은 상기 선택된 세션에 대한 작업 공간 데이터를 다운로드한다(1238). 디스플레이 클라이언트는 그 후에 작업공간 데이터를 트래버싱(traversing)하고 디스플레이 클라이언트에 의해 관리되는 디스플레이 영역에 대한 이미지를 구성한다(1240). 또한, 디스플레이 클라이언트는 그 후에 실시간 이벤트 채널에 참여한다(1242). 클라이언트측 네트워크 노드 및 서버측 네트워크 노드는 세션의 설정 동안 공간 이벤트 맵 데이터의 암호화 및 암호화 해제에 대한 프로토콜을 설정한다.

[0472] 일 예시에서, 작업 공간 데이터를 다운로드하는 프로세스는 각 디스플레이 클라이언트에 상기 세션에 대한 이벤트 오브젝트들을 전달하는 것을 포함한다. 상기 작업 공간 데이터에 포함된 현재 사용자 위치가 제공된다. 대안적으로, 작업 공간 데이터가 전달되고, 이는 작업 공간 데이터의 중심과 같은 기본 위치로부터 사용자와 관련된 현재 위치로의 오프셋을 어떻게 계산할지를 디스플레이 클라이언트에 식별하는 메시지들의 시퀀스에 의해 후속된다. 각 디스플레이 클라이언트는 그 후에 디스플레이 클라이언트에 의해 관리되는 디스플레이 영역에 매핑되는 세션 위치들을 갖는 그러한 오브젝트들을 식별하기 위해서 이벤트 오브젝트들을 트래버싱한다. 이벤트 오브젝트들을 트래버싱하기 위한 로직은 예를 들어, 디스플레이 영역에 매핑되는 작업 공간의 오브젝트들을 검색하도록 구성되는 R-TREE 검색을 포함한다. 식별된 오브젝트들은 그 후에 렌더링되고, 가능하게는 디스플레이에 의해 관리되는 디스플레이 영역에 오브젝트에 관련된데이터를 획득하기 위해 상기 포털과 통신한다.

[0473] 도 12는 작업 공간 데이터를 다운로드하는 것과 관련된 클라이언트측의 기초 로직을 설명한다. 로직은 협업 서버로부터의 작업공간 데이터의 다운로드로 시작한다(1250). 디스플레이 클라이언트는 사용자 핵심(user focus) 주위의 디스플레이 클라이언트의 범위 내에 있는 오브젝트들을 렌더링하고(1252), 상기 사용자 핵심은 서버에 의해 제공되거나 클라이언트측 네트워크 노드에서 유지되는 작업 공간 내의 위치로부터 결정된다. 디스플레이 클라이언트는 사용자 핵심에 가까운 작업 공간 데이터의 현재 위치들을 갖는 오브젝트들을 포함하는 작업 공간 데이터 또는 작업 공간 데이터의 적어도 부분들을 보유한다. 상기 세션 동안, 사용자 입력 또는 기타 데이터에 응답하여, 디스플레이 클라이언트는 작업 공간의 현재 위치를 결정하기 위해서 작업 공간의 위치들을 트래버싱한다(1254). 디스플레이 클라이언트는 그 후에 트래버싱된 위치들 주위의 범위 내의 오브젝트들을 렌더링한다(1256). 세션의 종료 시에, 디스플레이 클라이언트에 의해 매핑되는 작업 공간 내의 마지막 위치는 협업 서버에 사용자 위치로서 저장된다(1258).

[0474] 도 13은 액세스가 많은 사용자들 사이에서 공유되는 디스플레이들에 대한 액세스를 관리하기 위한 서버측에 의해 실행되는 로직을 설명한다. 이 예시에서, 상기 서버는 서버에 활성 링크들을 갖는 프리 디스플레이 벽들(free display walls)의 리스트를 유지한다. 이러한 링크들은 디스플레이 벽들이 켜질 때 셋업되고, 디스플레이 벽이 아이들(idle) 상태인 대기 기간동안 유지된다. 상기 서버는 프리 디스플레이 벽이 서버에 링크를 갖는지 여부를 결정하기 위한 로직을 포함한다(1260). 만약 서버에 링크를 갖는 벽이 검출된다면, 상기 벽은 1회용 식별 코드 또는 PIN에 할당된다. 만약 식별 코드가 할당되면 상기 벽은 "로비"에 추가되고, 상기 로비는 협업 시스템의 이용 가능한 벽들(예를 들어, 프리 벽들)의 리스트를 포함한다(1264). 서버는 또한 인증된 사용자 로그인들을 대기하는 루프를 실행한다(1266). 로그인이 검출될 때, 서버는 상기 사용자가 작업 공간을 선택하고 벽 - 상기 벽에 대해 사용자가 인증됨-을 선택하도록 프롬프트(prompt)한다(1268). 서버는 그 후에 사용자에게 상기 선택된 벽과 관련된 일회용 식별 코드를 입력하도록 요구한다(1270). 만약 서버가 상기 선택된 벽에 대한 매칭되는 식별 코드를 수신하지 않는다면(1272), 에러 신호가 발행된다(1273). 서버가 상기 선택된 벽에 대한 매칭되는 식별 코드를 수신할 때(1272), 상기 선택된 벽과 관련된 디스플레이 클라이언트 또는 클라이언트들은 상기 세션에 대한 실시간 이벤트 채널에 링크되고, 벽이 점유된 동안 상기 일회용 식별 코드는 디스에이블되거나 변경된다(1274). 또한, 상기 서버는 작업공간 데이터를 선택된 디스플레이 클라이언트 또는 클라이언트들에 송신한다(1276). 상기 사용자는 그 후에 작업공간 데이터를 수신한 후에 상기 세션과 협업할 수 있다(1278). 상기 사용자가 상기 세션을 로그오프할 때, 디스플레이 벽이 프리된다(1280). 만약 디스플레이 벽이 이용가능한 채로 있다면, 단계들(1260, 1262, 1264)에 후속하여 디스플레이 벽은 서버에 대해 프리 디스플레이 벽인 것으로 표시되고, 새로운 식별 코드로 상기 로비에 추가된다. 일부 실시예들에서, 식별 코드는 타임아웃 인터벌의 만료 시에 변경되어서, 사용 중이 아닌 벽으로부터 식별 코드를 훔칠 수 있는 침입자들에 의한 로그인들에 대한 보안을 제공한다.

[0475] 도 14는 연합 디스플레이 어레이를 관리하기 위한 서버측에서 실행되는 기초 로직을 설명한다. 이 흐름도의 제1 단계는 어레이의 디스플레이 클라이언트들 각각에 작업공간 데이터를 다운로드하는 것을 수반한다(1302). 각 디스플레이 클라이언트는 사용자 위치로부터 오프셋된 클라이언트 주위의 디스플레이 클라이언트들의 범위내에 있는 오브젝트들을 렌더링한다(1304). 서버는 클라이언트 이벤트 메시지들을 모니터링한다(1306). 서버가 어레이의 디스플레이 클라이언트들 중 하나로부터 클라이언트 이벤트 메시지를 수신할 때, 상기 서버는 상기 메시지가 작업공간 데이터에 관한 것인지 또는 단지 어레이에 관한 것인지를 결정한다(1308). 어레이 메시지들은 연합 디스플레이 어레이에 참여하는 이러한 디스플레이 클라이언트들만이 메시지들을 수신하도록 어레이 채널에서 방송된다(1310). 작업 공간 데이터 메시지들은 협업 채널에서 방송되어서, 작업 공간 데이터와 세션에 참여하는 디스플레이 클라이언트들 모두는 적절한 대로 업데이트된다(1312). 연합 디스플레이 어레이에만 관련된 이러한 메시지들은 위에 설명된 바와 같이 툴바들 및 드로잉 영역들의 위치를 업데이트하는 메시지들과 같은 메시지들을 포함한다. 또한, 작업 공간의 오브젝트들의 위치를 변경하지 않고 작업 공간 데이터의 부분인 오브젝트들을 생성 또는 수정하지 않는 메시지들은 로컬 어레이 전용 메시지들인 것으로 결정된다.

[0476] 도 15는 도 1b의 스타일로 널리 분산된 디스플레이들이 존재하는 분산된 디스플레이 협업을 지원하는 시스템을 설명한다. 상기 시스템은 작업 공간 데이터를 저장하는 관련된 디스플레이 데이터베이스(106)를 갖는 협업 서버(105)를 포함한다. 협업 서버는 예를 들어, Chicago, Los Angeles 및 Sao Paulo에 위치된 복수 개의 벽들(1502a, 1502b 및 1502c)에 통신 링크들(104)에 의해 연결된다. 협업 서버(105)는 또한 알려진 사용자의 소유에 있는 것으로 예측되는 터치패드 또는 기타 퍼스널 컴퓨팅 플랫폼과 같은 사용자 디바이스(1504)에 결합된다. 도 13과 함께 위에 언급된 바와 같이, 협업 서버(105)는 "로비"(109)로서 언급되는 데이터 구조의 프리 디스플레이 벽들의 리스트를 유지한다. 디스플레이 벽들 각각과 관련된 것은 Chicago의 디스플레이 벽과 관련된 OT-PIN#1, Los Angeles의 디스플레이 벽과 관련된 OT-PIN#2, 및 Sao Paulo의 디스플레이 벽과 관련된 OT-PIN#3을 포함하는 일회용 식별 코드이다. 개인 디바이스(1504)를 소유하는 사용자는, 사용자 인증의 목적을 위해서 사용자 ID 및 사용자 패스워드를 입력하여 협업 서버(105)에 의해 관리되는 포털에 로그인한다. 그 후에, 개인 디바이스(1504)를 소유하는 사용자는 사용자가 작업 공간 데이터를 디스플레이하기를 원하는 디스플레이 벽에 대한 작업 공간 식별자 및 식별 코드를 제공한다. 협업 서버가 사용자를 성공적으로 인증하고, 상기 사용자가 사용자가 인증된 디스플레이 벽, 사용자가 인증된 작업 공간을 식별했음을 결정할 때, 식별된 디스플레이와 관련된 디스플레이 클라이언트는 협업 이벤트 채널에 링크되고 작업 공간 데이터를 다운로드하도록 인에이블된다. 디스플레이 디바이스가 주어진 세션에 대해 인에이블될 때, 이는 로비(109)로부터 제거되고, 일회용 식별 코드는 삭제 또는 변경된다. 디스플레이가 로비(109)에 추가될 때마다, 새로운 일회용 식별 코드가 작업 공간 데이터에 기여(contribute)하는 사용자 입력을 승인하도록 계산된다. 상기 시스템은 작업 공간 데이터에 대해 인증된 사용자가 선택된 디스플레이에 물리적 액세스를 가짐을 보장하는 프로토콜에 기초해서 선택된 디스플레이들에 작업 공간 데이터를 제공하는 관리 로직을 포함한다.

[0477] 도 16은 클라이언트 프로세서(1600), 디스플레이 구동기(1601), 로컬 디스플레이 및 터치스크린(1602)과 같은 사용자 인터페이스, 스택에 의해 제어되는 통신 인터페이스를 포함하는 프로토콜 스택(1604), 실시간 공간 이벤트 맵의 캐시 복사본(cache copy), 이미지들의 캐시 및 디스플레이 가능한 영역을 렌더링하는 데 사용되는 그래픽 구성물들을 저장하는 로컬 메모리(1605), 터치스크린 또는 마우스와 같은 유형의 사용자 입력 디바이스로부터의 입력을 커맨드 해석기(1606)에 의해 사용가능한 형태로 번역(translate)하는 입력 프로토콜을 실행하는 입력 프로토콜 디바이스(1607)를 포함하는 클라이언트측 네트워크 노드의 단순화된 도면이다. 적합한 입력 프로토콜 디바이스(1607)는 예를 들어, 디스플레이 벽과의 유형의 그리고 다중 터치의 인터렉션의 해석을 위해서 TUIO 산업 표준과 호환되는 소프트웨어를 포함한다. 프로토콜 스택(1604)은 클라이언트 프로세서로부터 API 컴플라이언트 메시지들(API compliant messages) 및 인터넷 메시지들을 수신하고 위에 논의된 대로 API 컴플라이언트 메시지들이 교환되는 협업 서버에 채널(1611)을 설정하기 위한 자원들을 포함하고, 로컬 디스플레이(1602)를 서빙(serving)하는 기타 통신들을 지원하는 인터넷에 대한 링크(1610)를 포함한다. 디스플레이 구동기(1601)는 로컬 디스플레이(1602)에서 디스플레이 가능한 영역(1603)을 제어한다. 디스플레이 가능한 영역(1603)은 클라이언트 프로세서에 의해 로직으로 구성되거나 클라이언트측 네트워크 노드의 기타 프로그래밍 자원들에 의해 로직으로 구성된다. 또한, 디스플레이 가능한 영역(1603)의 물리적 사이즈는 로컬 디스플레이의 주어진 구현에 대해 고정될 수 있다. 클라이언트 프로세서(1600)는 브라우저와 같은 프로세싱 자원, 디스플레이 가능한 영역(1603)과 작업 공간의 위치들 간에 번역을 위해 사용되는 매핑 로직(mapping logic), 및 API 절차들을 구현하기 위한 로직을 포함한다.

[0478] 본 명세서에서 설명되는 협업용 시스템은 도 16의 클라이언트측 네트워크 노드와 같은 클라이언트측 네트워크

노드를 포함한다. 클라이언트측 네트워크 노드는 물리적 디스플레이 공간을 갖는 디스플레이, 사용자 입력 디바이스, 프로세서 및 통신 포트를 포함한다. 클라이언트측 네트워크 노드는, 서버측 네트워크 노드에 링크를 설정하고; 서버측 네트워크 노드로부터 작업 공간의 위치들을 갖는 그래픽 타겟들에 관한 이벤트들의 공간 이벤트 로그, 이벤트의 그래픽 타겟의 작업 공간의 위치를 포함하는 로그의 입력들, 이벤트의 시간, 그리고 그래픽 타겟의 타겟 식별자의 적어도 일부를 검색하고; 물리적 디스플레이 공간의 디스플레이 가능한 영역을 작업 공간 내의 매핑된 영역에 매핑하고, 매핑된 영역 내의 공간 이벤트 로그의 입력들을 식별하고, 디스플레이 가능한 영역에 식별된 입력들에 의해 식별된 그래픽 타겟들을 렌더링하고; 디스플레이 가능한 영역 내에 디스플레이되는 그래픽 타겟들의 수정 및 생성에 관한 이벤트들을 생성하는 사용자 입력 디바이스로부터 입력 데이터를 승인하고, 상기 이벤트들에 기초해서 서버측 네트워크 노드에 메시지들을 송신하도록 로직으로 구성된다.

[0479] 도 16에 도시된 클라이언트측 네트워크 노드는 서버측 네트워크 노드와 통신하는 프로세스를 포함하는 애플리케이션 인터페이스를 포함하는 예시를 설명한다.

[0480] 도 16에 도시된 클라이언트측 네트워크 노드는 API에 따라 구성되는 예시를 설명하고, 여기서 이벤트들은 다른 클라이언트측 네트워크 노드들에 배포되고 서버측 네트워크 노드의 공간 이벤트 로그에 추가될 이력 이벤트들로서 지정되는 이벤트의 제1 클래스 및 다른 클라이언트측 네트워크 노드들에 배포되지만 서버측 네트워크 노드의 공간 이벤트 로그에는 추가되지 않는 일시적인 것으로 지정된 이벤트의 제2 클래스를 포함한다.

[0481] 도 17은 클라이언트측 네트워크 노드에 의해 실행되는 절차의 단순화된 흐름도이다. 단순화된 흐름도에 설명되는 순서는 설명의 목적을 위해서 제공되고, 특정한 구현에 맞게 수정될 수 있다. 예를 들어 많은 단계들은 병렬적으로 실행될 수 있다. 이 절차에서, 클라이언트 로그인이 실행되고(1700), 로그인에 의해 클라이언트가 특정한 협업 세션 및 협업 세션의 공간 이벤트 맵에 액세스를 부여받는다. 협업 서버는 공간 이벤트 맵의 식별자 또는 공간 이벤트 맵의 일부의 식별자들을 제공하고, 상기 식별자는 상기 클라이언트가 협업 서버로부터 공간 이벤트 맵을 검색하기 위해 사용된다(1710). 클라이언트는 제공된 식별자 또는 식별자들을 이용하여 협업 서버로부터 공간 이벤트 맵, 또는 공간 이벤트 맵의 적어도 일부들을 검색한다(1702).

[0482] 예를 들어, 클라이언트는 주어진 작업 공간에 대한 모든 이력을 요청하고, 상기 주어진 작업 공간에 대해서 클라이언트는 다음과 같이 액세스를 부여받는다:

[0483] curl http://localhost:4545/<sessionId>/history

[0484] 서버는 모든 청크들(chunks)을 이용해서 응답할 것이다(청크 각각은 서버의 시간의 고유 섹션이다.)

[0485] ["/<sessionId>/history/<startTime>/<endTime>?b=1 "]

[0486] ["/<sessionId>/history/<startTime>/<endTime>?b=1 "]

[0487] 각 청크에 대해서, 클라이언트는 이벤트들을 요청할 것이다:

[0488] Curl http: //localhost:4545/<sessionId>/history/<startTime>/

[0489] <endTime>?b=<cache-buster>

[0490] 각 응답 청크는 이벤트들의 어레이이며 클라이언트에 의해 캐시가능하다:

[0491] [

[0492] [

[0493] 4,

[0494] "sx",

[0495] "4.4",

[0496] [537, 650, 536, 649, 536, 648, ...],

[0497] {

[0498] "size": 10,

[0499] "color": [0, 0, 0, 1],

[0500] "brush": 1

[0501] },

[0502] 1347644106241,

[0503] "cardFling"

[0504]]

[0505]]

[0506] 개별 메시지들은 스크린상의 위치, 색상, 스트로크의 폭, 생성된 시간 등의 정보를 포함한다.

[0507] 클라이언트는 예를 들어 서버에 의해 제공되는 핵심 포인트(focus point), 및 로컬 디스플레이에 대한 디스플레이 경계들을 이용해서, 그 후에 작업 공간의 위치를 결정한다(1703). 공간 이벤트 맵의 로컬 복사본은 공간 이벤트 맵 입력들에 대한 디스플레이 데이터를 수집하기 위해 트래버싱(traversing)되고, 상기 디스플레이 데이터는 로컬 디스플레이에 대한 디스플레이 가능한 영역에 매핑된다. 일부 실시예들에서, 작업 공간 내의 주밍(zooming) 및 패닝(panning)과 같은 예측되는 사용자 인터렉션들을 지원하기 위해서, 클라이언트는 로컬 디스플레이에 대한 디스플레이 가능한 영역보다 넓은 영역을 정의하는 컬링 경계(culling boundary) 내의 공간 이벤트 맵 입력들에 대한 디스플레이를 렌더링하는 것을 지원하는 추가적인 데이터를 수집할 수 있다(1704). 클라이언트 프로세서는 공간 이벤트 맵 이벤트들, 일시 이벤트들 및 디스플레이 경계 내에 해당되는 공간 이벤트 맵의 일부들을 렌더링하기 위한 디스플레이 데이터를 이용한 프로세스를 실행한다(1705). 이 프로세스는, 예를 들어, TUIO 구동기로부터와 같이, 로컬 사용자 인터페이스 메시지들을 검색한다(1706). 또한, 이 프로세스는 협업 서버로부터 소켓 API 메시지들을 수신한다(1710). 로컬 사용자 인터페이스 메시지들에 응답하여, 상기 프로세스는 입력들을 이력 이벤트들 및 일시 이벤트들로서 분류하고, API에 의해 특정되는 이력 이벤트와 일시 이벤트 둘 모두에 대해 협업 서버에 소켓의 API 메시지들을 송신하고, 공간 이벤트 맵의 캐시된 부분들을 이력 이벤트들로 업데이트하고, 이력 이벤트들과 일시 이벤트들 둘 모두에 대한 디스플레이 데이터를 생산한다(1707). 소켓 API 메시지들에 응답하여, 프로세스는 서버측 네트워크 노드에 의해 식별되는 이력 이벤트들로 공간 이벤트 맵의 캐시된 부분을 업데이트하고, API에 의해 특정되는 소켓의 API 메시지들에 응답하고, 소켓 메시지들에 의해 통지되는 이력 이벤트들 및 일시 이벤트들 둘 모두에 대한 디스플레이 데이터를 생산한다(1711).

[0508] 도 18은 클라이언트측 네트워크 노드에 의해 실행되는 사용자 입력을 해석하기 위한 프로세스의 단순화된 흐름도이다. 단순화된 흐름도에 설명된 순서는 설명의 목적을 위해 제공되고, 특정 구현에 맞게 수정될 수 있다. 많은 단계들은 예를 들어, 병렬적으로 수행될 수 있다. 상기 프로세스는 디스플레이의 물리적 좌표와 함께, 유형의 사용자 입력 디바이스로부터 네이티브 입/출력 이벤트 타입 메시지(native I/O event type message)를 수신하는 것으로 시작한다(1800). 프로세스는 디스플레이의 물리적 좌표를 작업 공간의 좌표에 매핑하고 공간 이벤트 맵의 로컬 캐시된 복사본을 이용해서, 공간 이벤트 맵의 오브젝트들을 식별하고, 만약 존재한다면, 상기 오브젝트들은 작업 공간 좌표들에 매핑된다(1801). 그 후에, 작업 공간 좌표, 식별된 오브젝트, 컨텍스트 및 I/O 이벤트 타입에 기초해서, 이벤트가 이력 이벤트 또는 일시 이벤트인지 여부를 포함하여, 프로세서는 API 이벤트 타입을 결정한다(1802). 클라이언트측 프로세서는 그 후에 API 컴플라이언트 메시지를 생산하고, 공간 이벤트 맵의 로컬 캐시된 복사본을 업데이트하고, 디스플레이 데이터를 업데이트한다(1803). 본 명세서에서 설명되는 바와 같이, 서버측 네트워크 노드 및 클라이언트측 네트워크 노드를 이용해서, 일부 기초적 절차들은 로그인 및 클라이언트측 네트워크 노드에 세션에 대한 공간 이벤트 맵의 다운로드, 그리고 실시간 작업공간 공간 이벤트들의 작업 공간 채널에 연결하는 것을 포함한다.

[0509] 로그인 및 공간 이벤트 맵의 다운로드

[0510] 1. 클라이언트는 협업 세션에 참여하기 위한 인증을 요청하고, 작업 공간을 오픈한다.

[0511] 2. 서버는 클라이언트가 세션에 참여하도록 인증하고, 작업 공간에 대한 공간 이벤트 맵을 로딩하기 시작한다.

[0512] 3. 클라이언트는 세션과 관련된 공간 이벤트 맵의 "컨텐츠들의 테이블"과 같은 식별을 요청한다.

[0513] 4. 컨텐츠들의 테이블의 식별된 공간 이벤트 맵의 각 부분은 클라이언트에 의해 요청된다. 공간 이벤트 맵의 이러한 부분들은 함께 작업 공간을 작업 공간 시간의 시작으로부터 현재까지의 이벤트들의 선형 시퀀스로서 나타낸다. "작업 공간 시간의 시작"은 협업 세션의 시작 시간으로부터 경과된 시간으로 간주될 수 있거나, 세션과 관련되어 기록된 절대 시간으로 간주될 수 있다.

[0514] 5. 클라이언트는 로컬 메모리에 공간 이벤트 맵의 캐시된 복사본을 어셈블링(assembling)한다.

[0515] 6. 클라이언트는 로컬 디스플레이의 현재 디스플레이 가능한 영역 또는 뷰포트가 주어질 때 무엇이 관련있는지를 결정하기 위해서 작업 공간의 공간 이벤트 맵을 이용해서 작업 공간의 적절한 영역을 디스플레이한다.

[0516] 실시간 공간 이벤트 맵 이벤트들의 세션 채널에 연결:

[0517] 1. 인증 후에, 클라이언트는 작업 공간 채널에 참여할 것을 요청한다.

[0518] 2. 서버는 작업공간 채널들을 통해서 업데이트들을 수신하기 위해서 작업공간 참가자들의 리스트에 클라이언트를 추가한다.

[0519] 3. 클라이언트는 이력 이벤트들과 일시 이벤트들 둘 모두를 전달하는 작업공간으로부터의 실시간 메시지들을 수신하고, 채팅방과 같은 통신 패러다임을 수신한다. 예를 들어, 일시 이벤트들의 시퀀스 및 이력 이벤트는 공간 이벤트 맵의 새로운 위치에 공간 이벤트 맵의 오브젝트를 이동시키는 것과 관련될 수 있다.

[0520] 4. 클라이언트는 공간 이벤트 맵의 메시지들의 로컬 복사본을 변경함으로써 그리고 메시지들의 로컬 디스플레이를 다시 렌더링함으로써 서버측 네트워크 노드로부터 실시간 메시지들에 반응한다.

[0521] 5. 실시간 메시지들은 기한 없이 지속되는 공간 이벤트 맵의 double형의 기록된 이벤트들인 "이력" 이벤트들 및 세션의 이력의 부분이 되지 않는 정보의 피스들인 "일시" 이벤트로 구성된다.

[0522] 6. 클라이언트가 클라이언트의 로컬 디스플레이와 인터렉션에 의해 오브젝트를 생성, 수정, 이동, 또는 삭제할 때, 새로운 이벤트가 클라이언트측 네트워크에 의해 생성되고 작업공간 채널을 통해서 서버측 네트워크 노드에 송신된다. 서버측 네트워크 노드는 세션에 대한 공간 이벤트 맵에 이력 이벤트들을 저장하고, 이력 이벤트들과 일시 이벤트들 모두를 세션의 모든 활성 클라이언트들에 배포한다.

[0523] 7. 세션을 종료할 때, 클라이언트는 작업공간 채널로부터 연결 해제된다.

[0524] 협업 세션은 공유된 협업 서버에 의해 관리되는 작업공간 데이터에 기초한 이미지들을 디스플레이하기 위해서, 그리고 작업공간 데이터에 기여하는 사용자 입력을 승인하기 위해서, 많은 분산된 디지털 디스플레이들을 가질 수 있고, 한편 각 디스플레이로 하여금 세션 이력, 실시간 로컬 입력 및 기타 디스플레이들로부터 실시간 입력에 기초해서 디스플레이할 이미지를 빠르게 구성하도록 할 수 있다.

[0525] 본 명세서에서 사용되는 바와 같이, 정보의 아이템의 "식별"은 그 정보의 아이템의 직접적인 명세(specification)를 필연적으로 요구하는 것은 아니다. 정보는 간접적인 행동의 하나 이상의 레이어들을 통해서 실제 정보를 단순히 참조함으로써, 또는 실제 정보의 아이템을 결정하기 위해 충분하게 함께 서로 다른 정보의 하나 이상의 아이템들을 식별함으로써 "식별될" 수 있다. 이에 더하여, 본 명세서에서 사용되는 "표시한다(indicate)"라는 용어는 "식별한다"와 동일한 의미로 사용된다.

[0526] 또한, 본 명세서에서 사용되는 바와 같이, 만약 선행 신호, 이벤트 또는 값이 주어진 신호, 이벤트 또는 값에 영향받는다면, 주어진 신호, 이벤트 또는 값은 선행(predecessor) 신호, 이벤트 또는 값에 "응답"할 수 있다. 만약 개입하는 프로세싱 요소, 단계 또는 기간이 존재한다면, 주어진 신호, 이벤트 또는 값은 여전히 선행 신호, 이벤트 또는 값에 "응답"할 수 있다. 만약 개입하는 프로세싱 요소 또는 단계가 2개 이상의 신호, 이벤트 또는 값을 조합한다면, 프로세싱 요소 또는 단계의 신호 출력은 신호, 이벤트 또는 값 입력들 각각에 "응답"하는 것으로 간주된다. 만약 주어진 신호, 이벤트 또는 값이 선행 신호, 이벤트 또는 값과 동일하다면, 이는 주어진 신호, 이벤트 또는 값이 여전히 선행 신호에 "응답"하는 것으로 간주되는 축퇴된 경우(degenerate case)이고, 이 경우 주어진 신호, 이벤트 또는 값은 여전히 선행 신호, 이벤트 또는 값에 "응답"하는 것으로 간주된다. 주어진 신호, 이벤트 또는 값의 다른 신호, 이벤트 또는 값에 대한 "의존성"은 유사하게 정의된다.

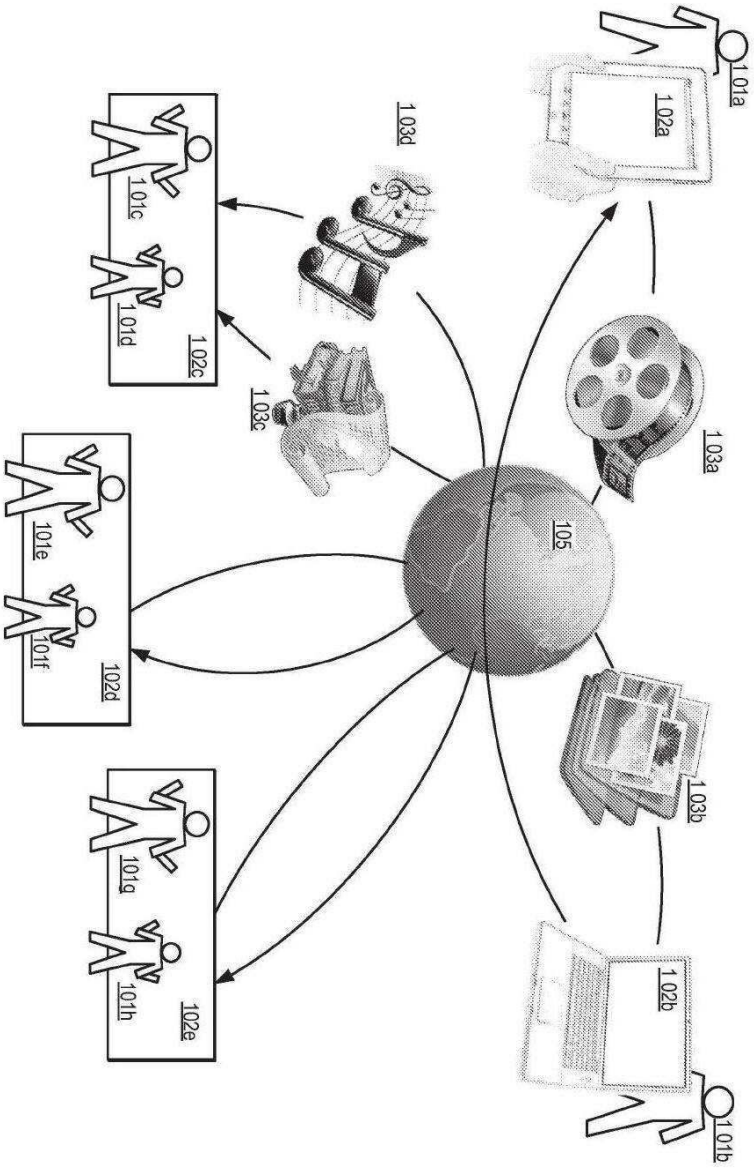
[0527] 출원인은 그러한 피처들 또는 피처들의 조합들이 본 명세서에 개시된 어떤 문제들을 해결하는지와는 무관하게, 그리고 특허청구범위의 범위에 대한 한정 없이, 그러한 피처들 또는 조합들이 통상의 기술자의 흔한 일반적 지식에 비추어 전체로서 본 명세서에 기초해서 수행될 수 있는 범위에서, 본 명세서에 의해 본 명세서에 설명된 각각의 개별 피처들 및 2개 이상의 그러한 피처들의 어떤 조합을 별개로 개시한다. 출원인은 본 발명의 양상들이 그러한 피처들 또는 피처들의 조합을 구성함을 표시한다. 전술한 설명에 비추어 보면, 다양한 수정들이 본 발명의 범위 내에서 이루어질 수 있음이 통상의 기술자에게 명백할 것이다.

[0528] 본 발명의 바람직한 실시예들의 전술한 설명은 설명의 목적들을 위해 제공되었다. 이는 포괄적인(exhaustive) 것으로 의도되지 않았고 개시된 특정 형태들로 본 발명을 한정하려고 의도되지 않았다. 명백하게, 많은 수정들

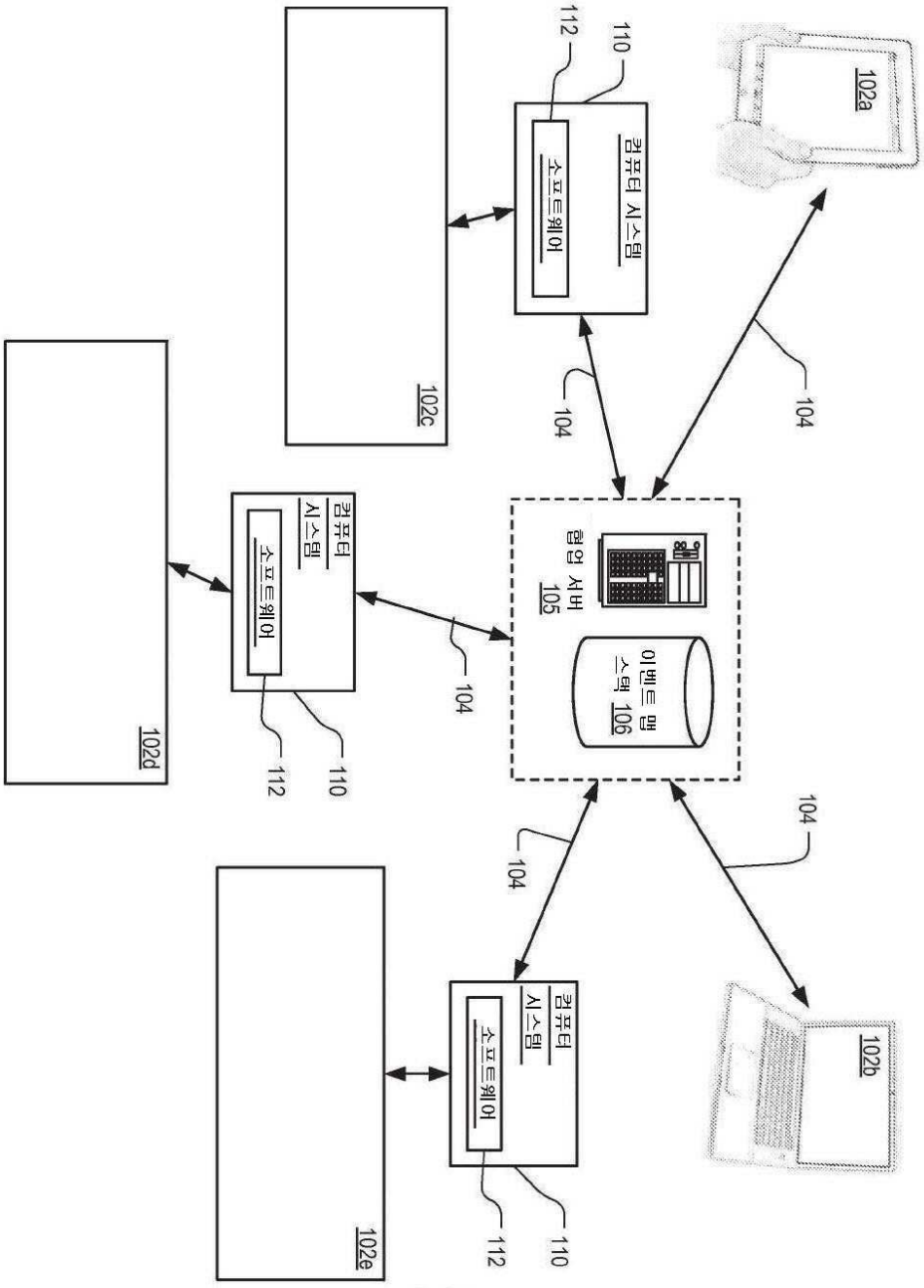
및 변형들이 통상의 기술자들에게 명백할 것이다. 예를 들어, 비록 본 명세서에서 설명된 디스플레이들이 넓은 포맷이지만, 비록 다수의 드로잉 영역들이 적어도 12 피트의 폭을 갖는 디스플레이들에 대해 더 유용하지만, 좁은 포맷의 디스플레이들 또한 다수의 드로잉 영역들을 사용하기 위해 배열될 수 있다. 특별히, 그리고 한정 없이, 본 특허출원의 배경기술 섹션 또는 참조에 의해 통합된 문헌들(materials)에 의해 설명되고 제안된 어떤 그리고 모든 변형들은 본 발명의 실시예들의 본 명세서의 설명에 참조에 의해 구체적으로 통합된다. 이에 더하여, 어떤 하나의 실시예에 관해 설명되고, 제안되고 본 명세서에 참조에 의해 통합된 어떤 그리고 모든 변형들은 또한 모든 다른 실시예들에 관해 교시된 것으로 간주된다. 본 명세서에서 설명된 실시예들은 본 발명의 원리들 및 본 발명의 실제적인 응용을 가장 잘 설명하기 위해서 선택되고 설명되었고, 이에 의해 다른 통상의 기술자들로 하여금 다양한 실시예들 및 고려되는 특정한 용도에 맞는 다양한 수정들에 대해 본 발명을 이해하는 것을 인에이블한다. 본 발명의 범위는 다음의 특허청구범위 및 이의 균등물들에 의해 정의되는 것으로 의도된다.

도면

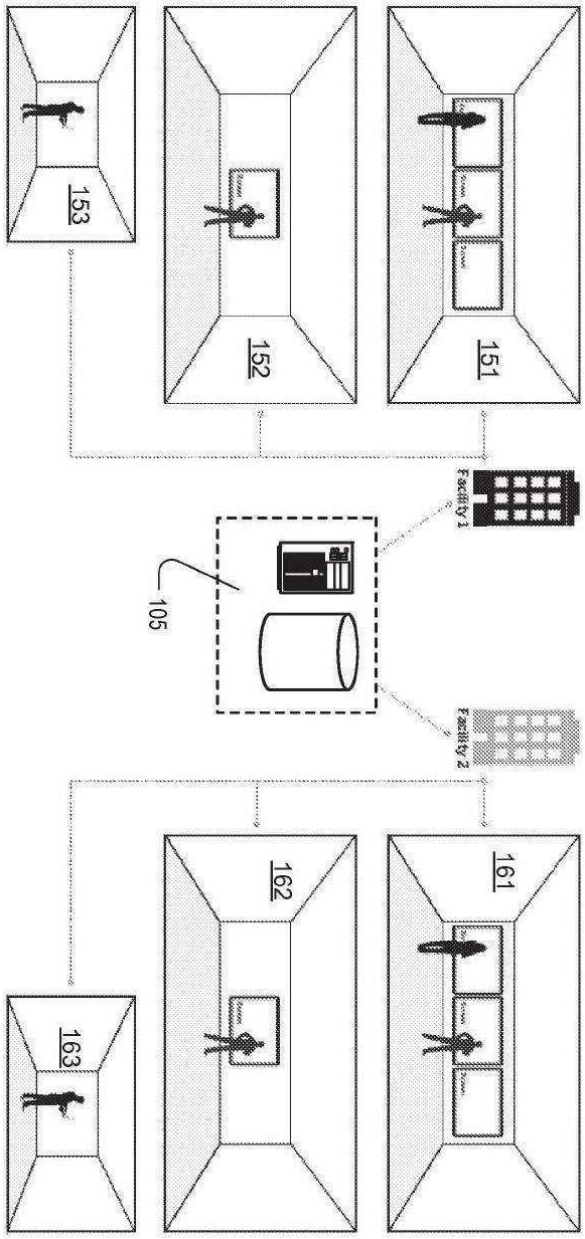
도면1a



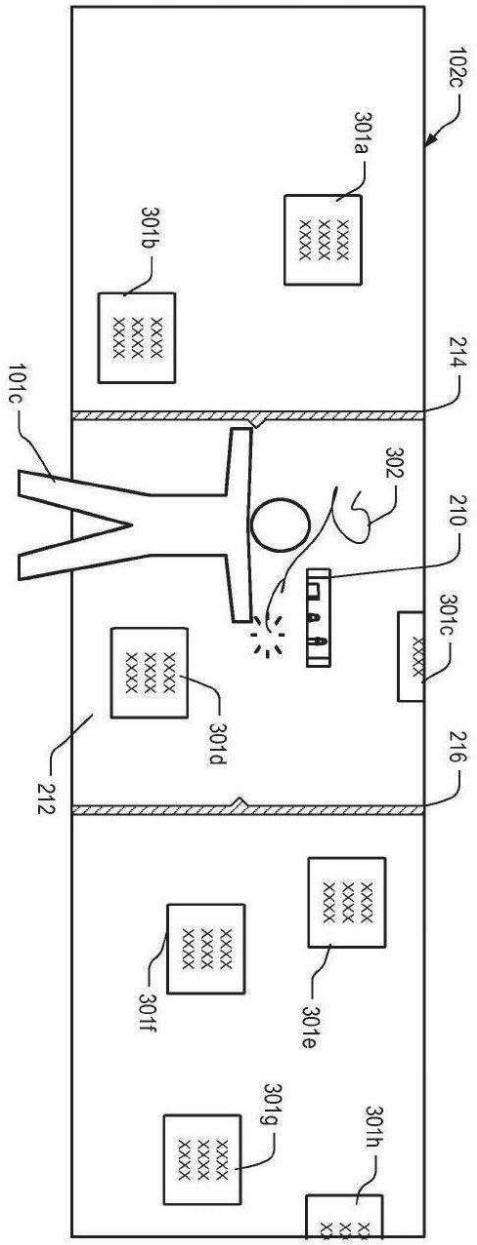
도면1b


102a
102b
102c
102d
102e
컴퓨터 시스템
소프트웨어
110
112
104
협업 서버
105
이벤트 맵
스택 106

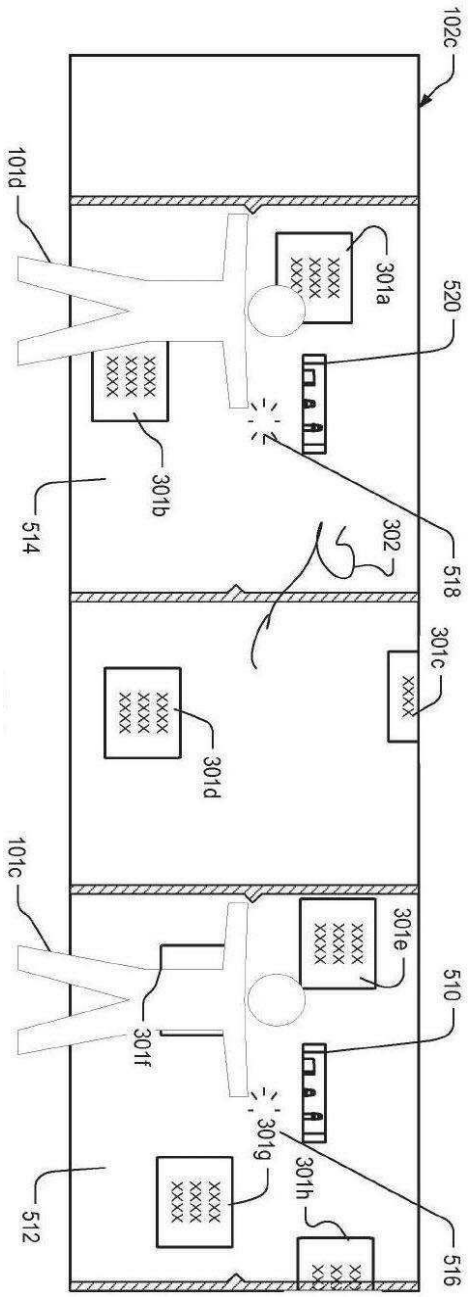
도면2



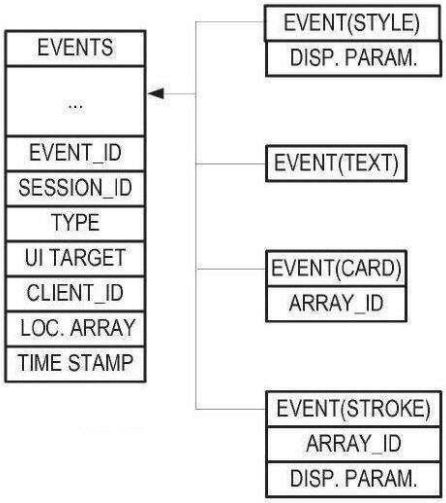
도면3


102c
301a
301b
214
101c
302
210
301c
301d
216
212
301e
301f
301g
301h

도면4

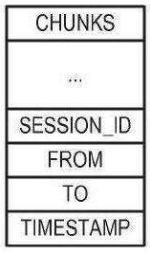

102c
101d
301a
520
301b
514
302
518
301c
301d
101c
301e
510
301f
301g
301h
512
516

도면5a



도면5b

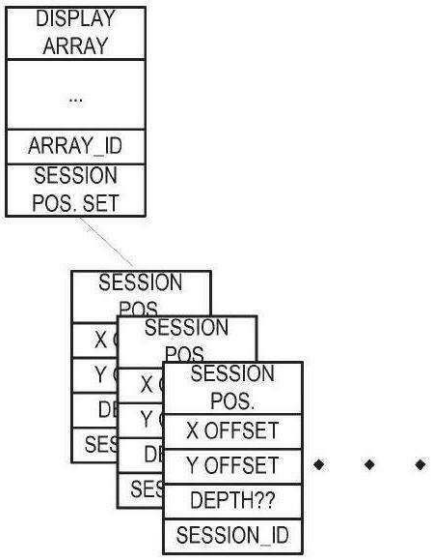
도면5c



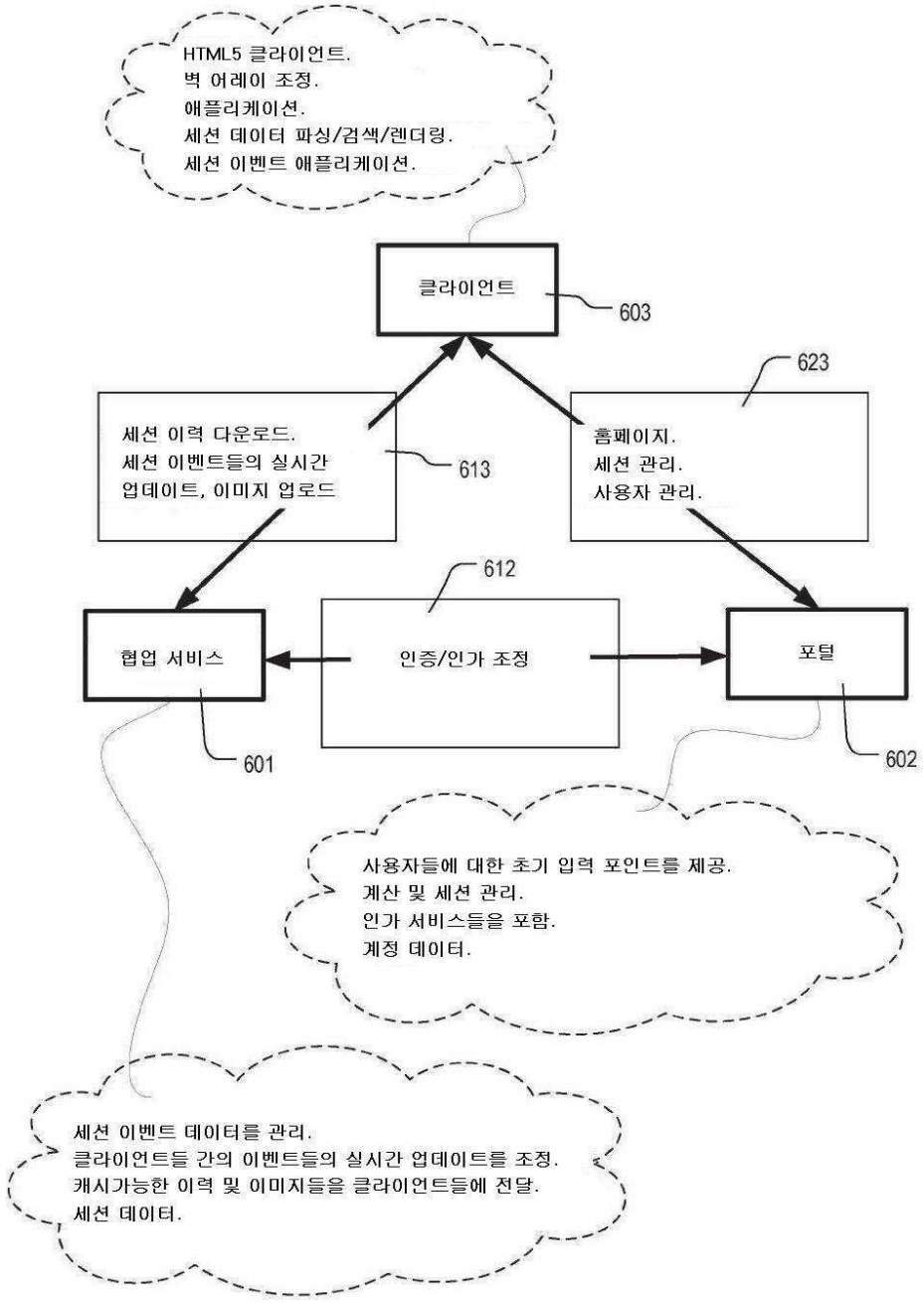
도면5d


USER SESSION
...
ACCESS TOKEN
SESS.CLIENT_ID
SESS.USER_ID
LAST ACCESS
EXPIRES
COOKIE

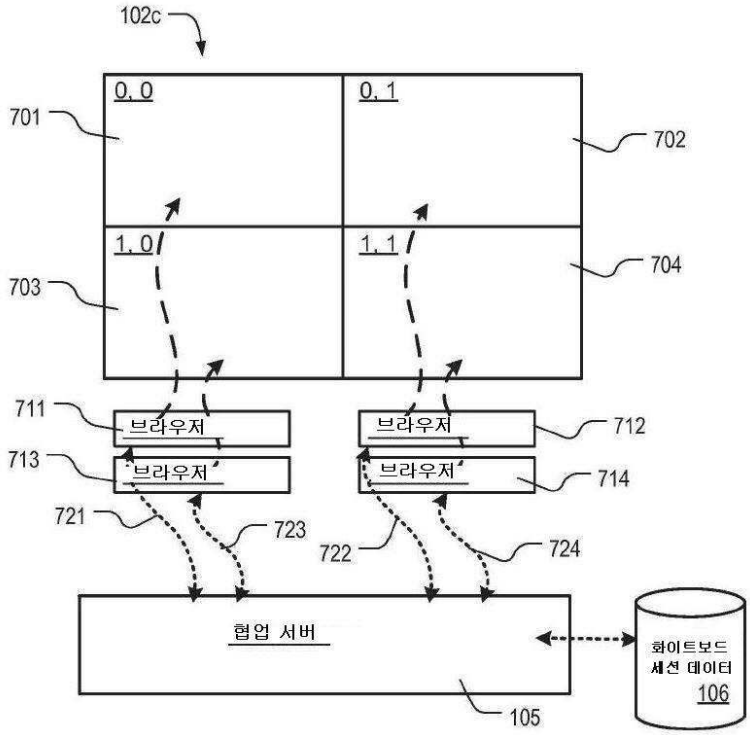
도면5e


DISPLAY ARRAY
...
ARRAY_ID
SESSION POS. SET
SESSION POS.
X OFFSET
Y OFFSET
DEPTH??
SESSION_ID

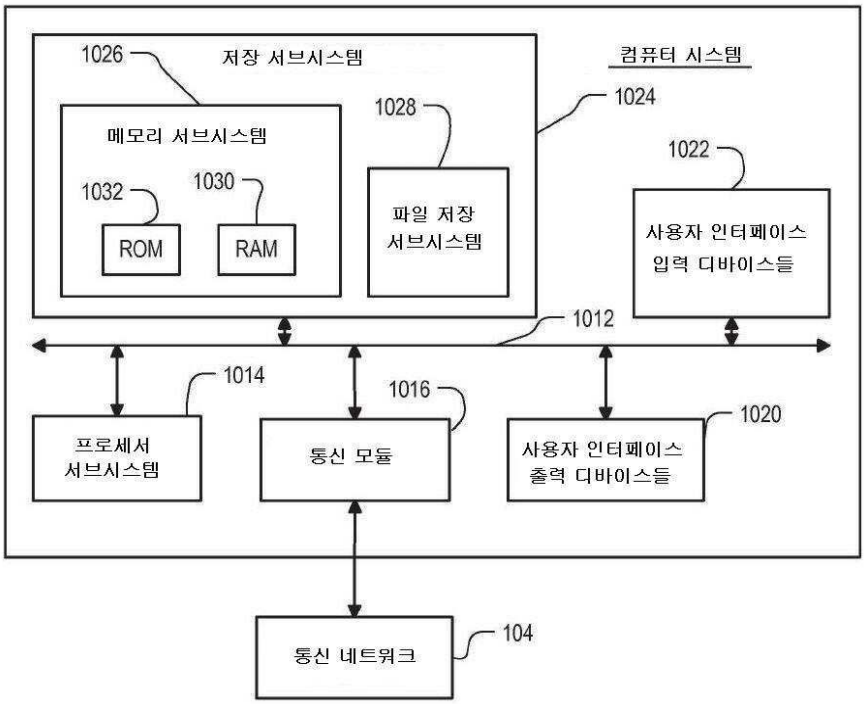
도면6


HTML5 클라이언트.
벽 어레이 조정.
애플리케이션.
세션 데이터 파싱/검색/렌더링.
세션 이벤트 애플리케이션.
클라이언트
603
623
세션 이력 다운로드.
세션 이벤트들의 실시간
업데이트, 이미지 업로드
613
홈페이지.
세션 관리.
사용자 관리.
612
협업 서비스
인증/인가 조정
포털
601
602
사용자들에 대한 초기 입력 포인트를 제공.
계산 및 세션 관리.
인가 서비스들을 포함.
계정 데이터.
세션 이벤트 데이터를 관리.
클라이언트들 간의 이벤트들의 실시간 업데이트를 조정.
캐시가능한 이력 및 이미지들을 클라이언트들에 전달.
세션 데이터.

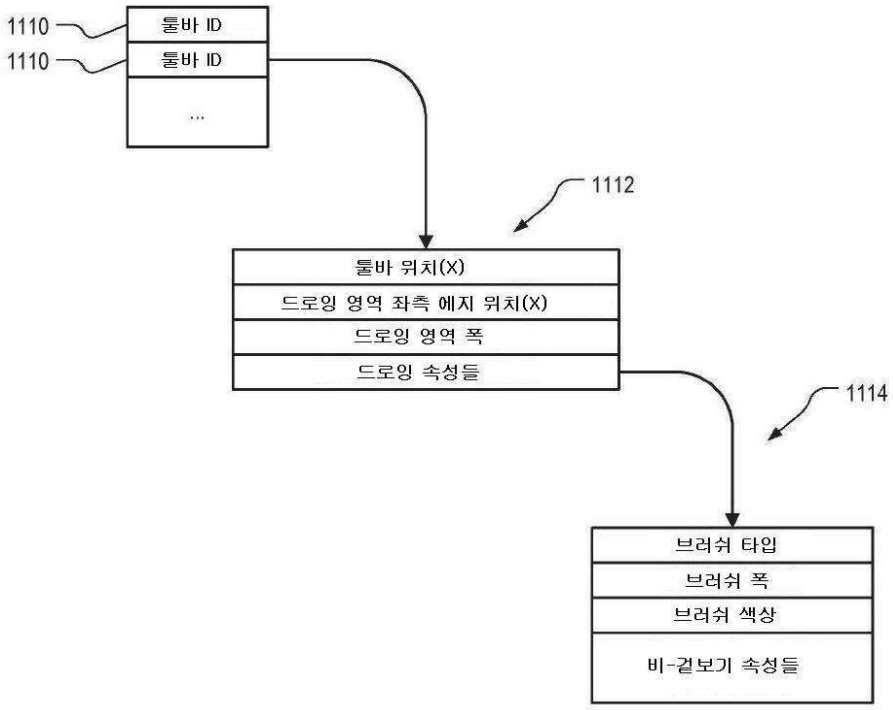
도면7


102c
0, 0
0, 1
1, 0
1, 1
701
702
703
704
711
712
713
714
721
722
723
724
브라우저
브라우저
브라우저
브라우저
협업 서버
화이트보드 세션 데이터
106
105

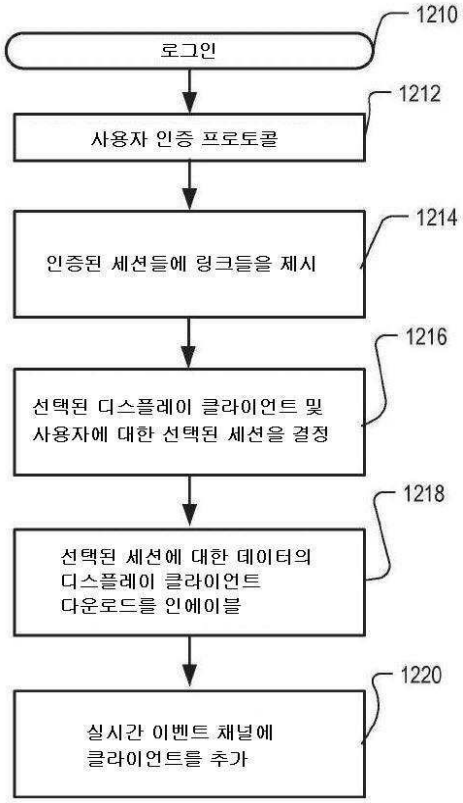
도면8


1026
저장 서브시스템
메모리 서브시스템
1032
ROM
1030
RAM
1028
파일 저장 서브시스템
컴퓨터 시스템
1024
1022
사용자 인터페이스 입력 디바이스들
1012
1014
1016
1020
프로세서 서브시스템
통신 모듈
사용자 인터페이스 출력 디바이스들
통신 네트워크
104

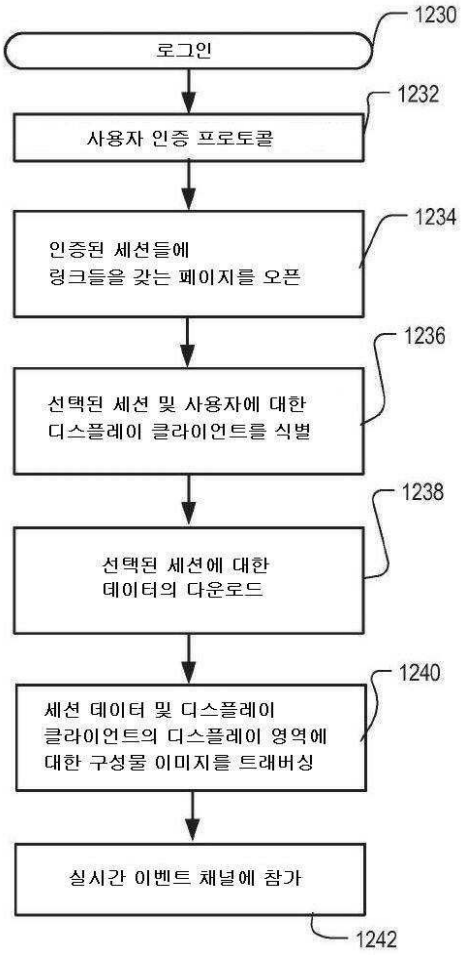
도면9


1110
툴바 ID
1110
툴바 ID
...
1112
툴바 위치(X)
드로잉 영역 좌측 에지 위치(X)
드로잉 영역 폭
드로잉 속성들
1114
브러쉬 타입
브러쉬 폭
브러쉬 색상
비-겉보기 속성들

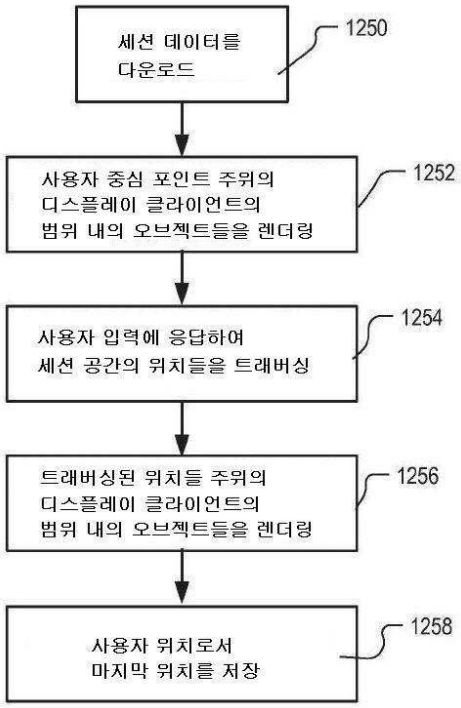
도면10


1210
로그인
1212
사용자 인증 프로토콜
1214
인증된 세션들에 링크들을 제시
1216
선택된 디스플레이 클라이언트 및
사용자에 대한 선택된 세션을 결정
1218
선택된 세션에 대한 데이터의
디스플레이 클라이언트
다운로드를 인에이블
1220
실시간 이벤트 채널에
클라이언트를 추가

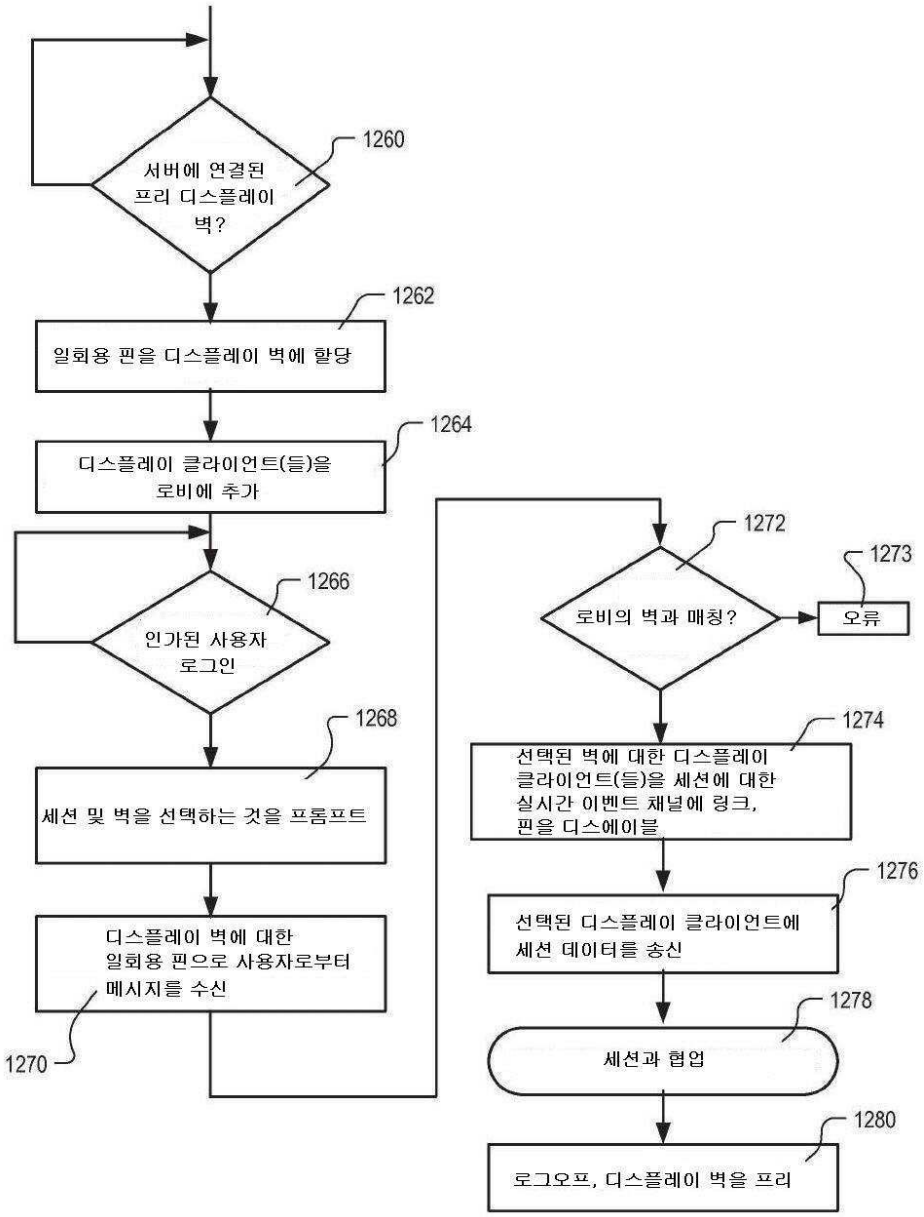
도면11


1230
로그인
1232
사용자 인증 프로토콜
1234
인증된 세션들에
링크들을 갖는 페이지를 오픈
1236
선택된 세션 및 사용자에 대한
디스플레이 클라이언트를 식별
1238
선택된 세션에 대한
데이터의 다운로드
1240
세션 데이터 및 디스플레이
클라이언트의 디스플레이 영역에
대한 구성물 이미지를 트래버싱
실시간 이벤트 채널에 참가
1242

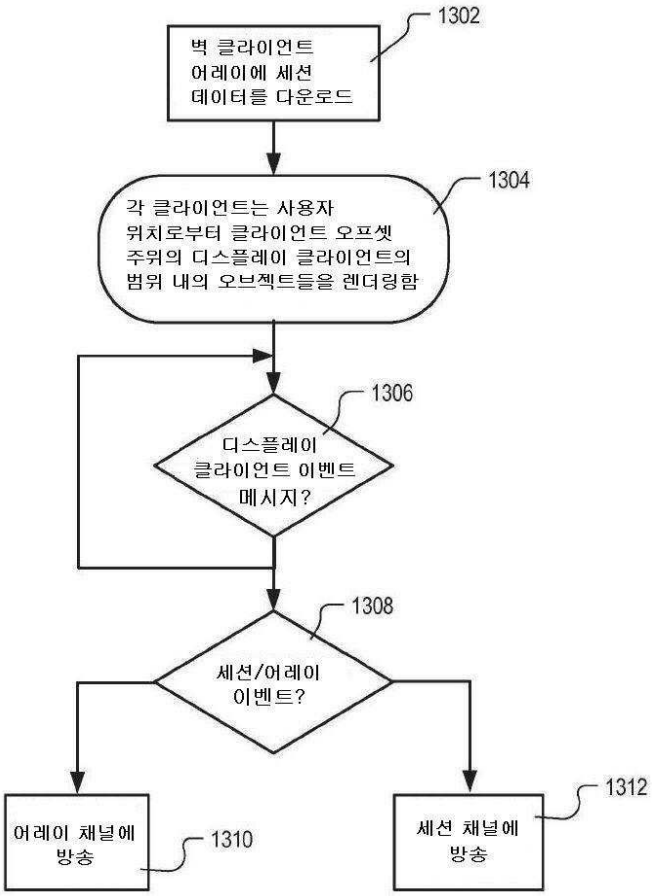
도면12


세션 데이터를 다운로드
1250
사용자 중심 포인트 주위의 디스플레이 클라이언트의 범위 내의 오브젝트들을 렌더링
1252
사용자 입력에 응답하여 세션 공간의 위치들을 트래버싱
1254
트래버싱된 위치들 주위의 디스플레이 클라이언트의 범위 내의 오브젝트들을 렌더링
1256
사용자 위치로서 마지막 위치를 저장
1258

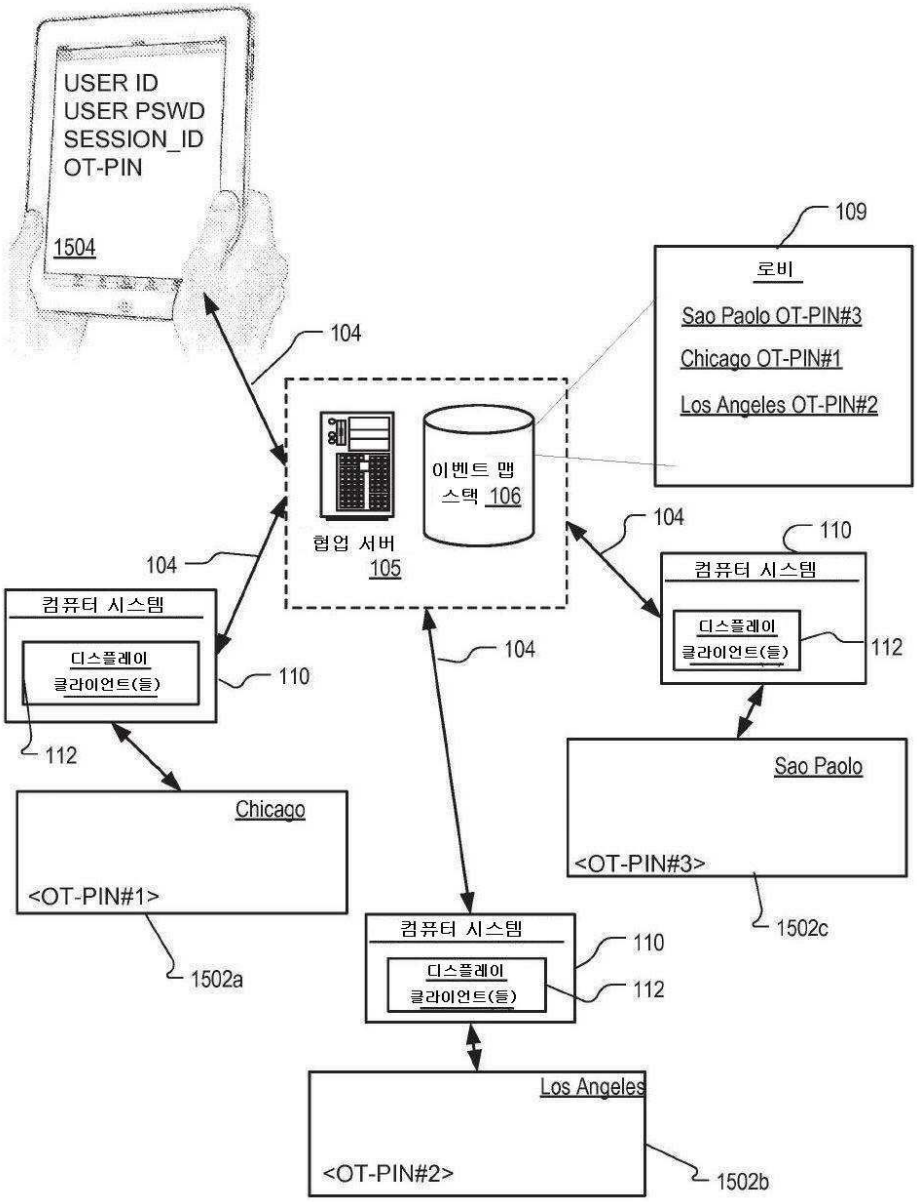
도면13


서버에 연결된 프리 디스플레이 벽?
1260
일회용 핀을 디스플레이 벽에 할당
1262
디스플레이 클라이언트(들)을 로비에 추가
1264
인가된 사용자 로그인
1266
세션 및 벽을 선택하는 것을 프롬프트
1268
디스플레이 벽에 대한 일회용 핀으로 사용자로부터 메시지를 수신
1270
로비의 벽과 매칭?
1272
오류
1273
선택된 벽에 대한 디스플레이 클라이언트(들)을 세션에 대한 실시간 이벤트 채널에 링크, 핀을 디스에이블
1274
선택된 디스플레이 클라이언트에 세션 데이터를 송신
1276
세션과 협업
1278
로그오프, 디스플레이 벽을 프리
1280

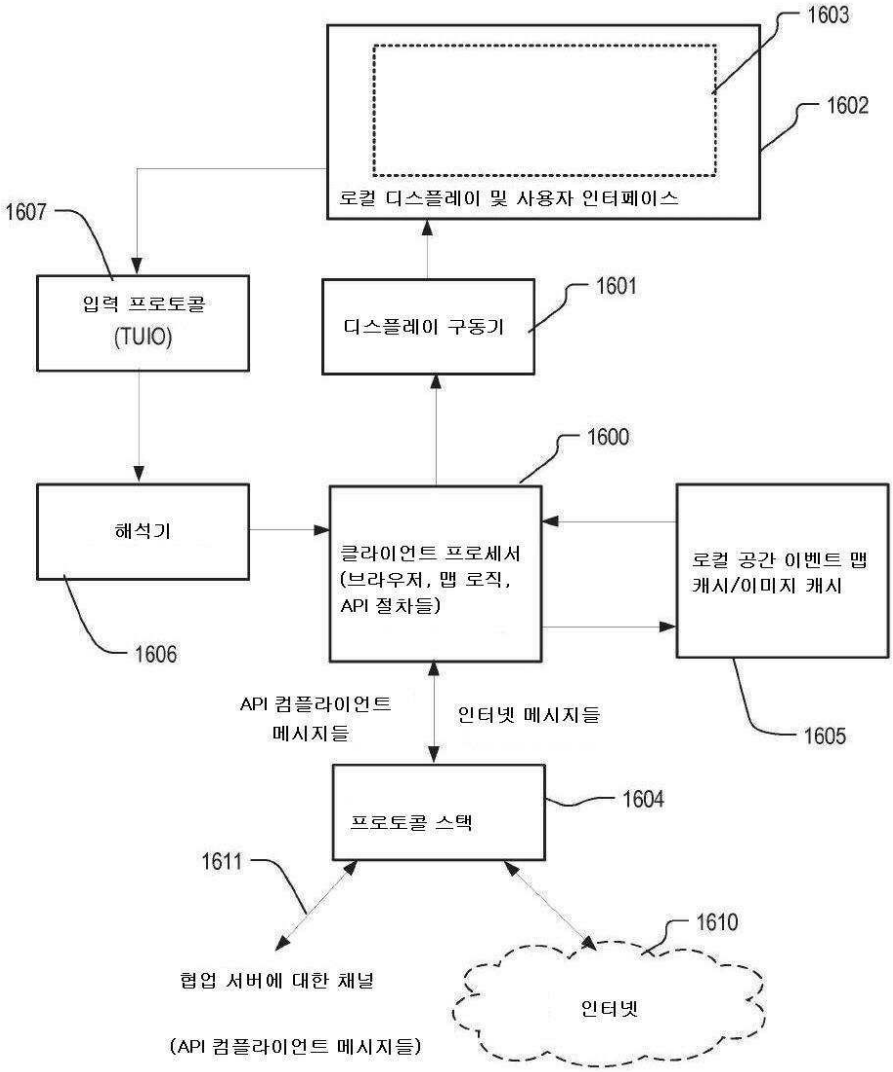
도면14


1302
벽 클라이언트 어레이에 세션 데이터를 다운로드
1304
각 클라이언트는 사용자 위치로부터 클라이언트 오프셋 주위의 디스플레이 클라이언트의 범위 내의 오브젝트들을 렌더링함
1306
디스플레이 클라이언트 이벤트 메시지?
1308
세션/어레이 이벤트?
어레이 채널에 방송
1310
세션 채널에 방송
1312

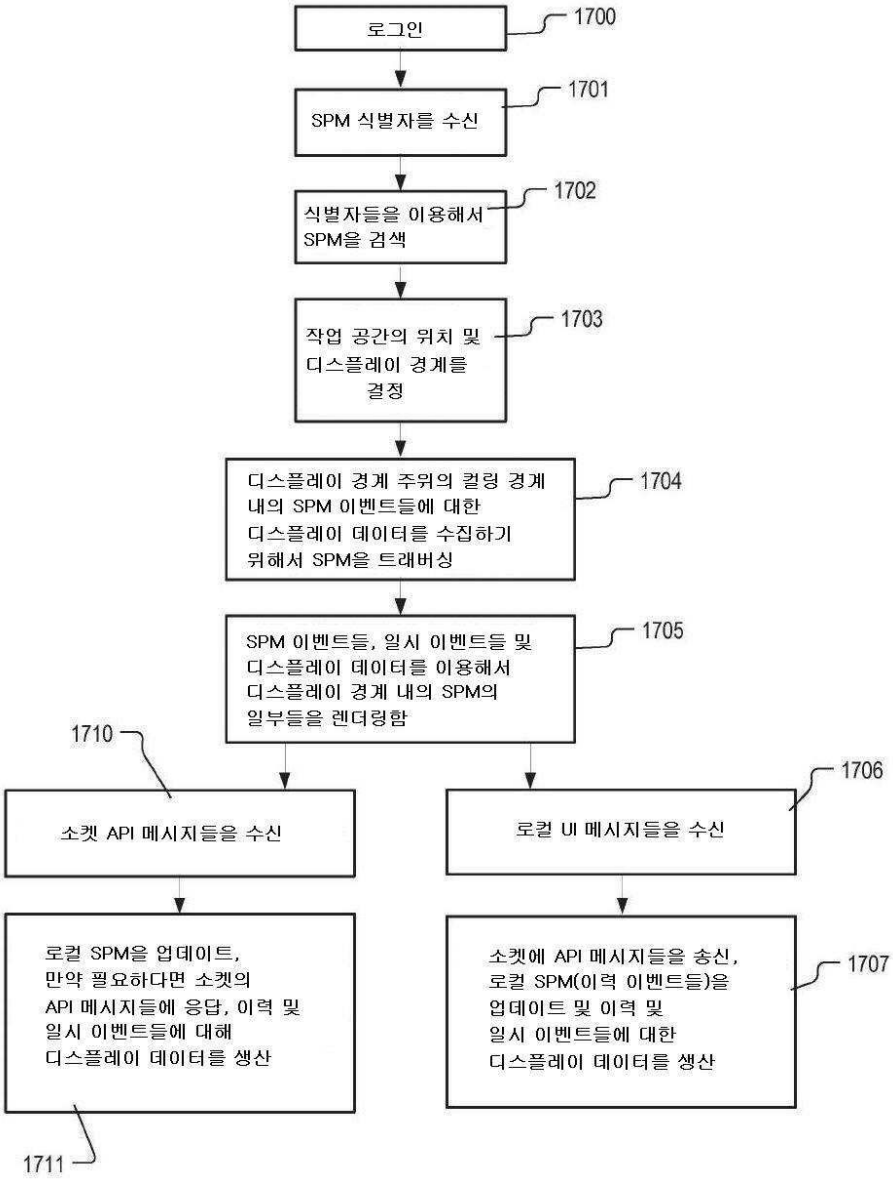
도면15


USER ID
USER PSWD
SESSION_ID
OT-PIN
1504
104
109
로비
Sao Paolo OT-PIN#3
Chicago OT-PIN#1
Los Angeles OT-PIN#2
이벤트 맵
스택 106
협업 서버
105
104
104
110
컴퓨터 시스템
디스플레이
클라이언트(들)
112
110
컴퓨터 시스템
디스플레이
클라이언트(들)
112
104
Sao Paolo
<OT-PIN#3>
1502c
Chicago
<OT-PIN#1>
1502a
컴퓨터 시스템
디스플레이
클라이언트(들)
110
112
Los Angeles
<OT-PIN#2>
1502b

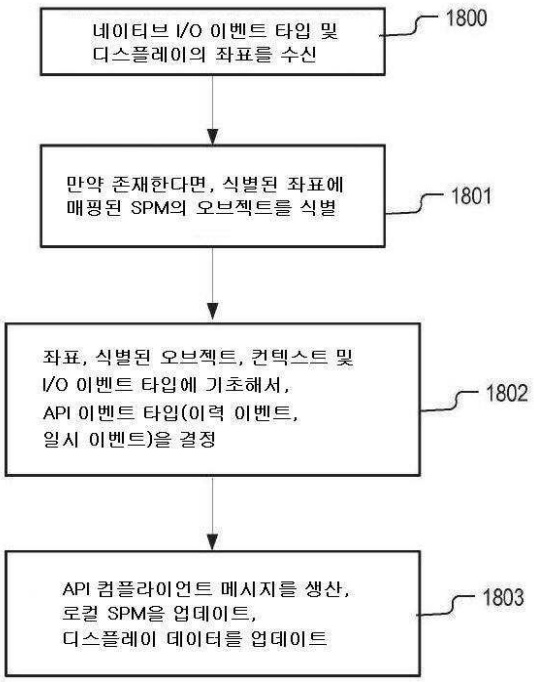
도면16


1603
1602
로컬 디스플레이 및 사용자 인터페이스
1607
입력 프로토콜
(TUIO)
디스플레이 구동기
1601
1600
해석기
클라이언트 프로세서
(브라우저, 맵 로직,
API 절차들)
로컬 공간 이벤트 맵
캐시/이미지 캐시
1606
1605
API 컴플라이언트
메시지들
인터넷 메시지들
프로토콜 스택
1604
1611
1610
협업 서버에 대한 채널
(API 컴플라이언트 메시지들)
인터넷

도면17


로그인
1700
SPM 식별자를 수신
1701
식별자들을 이용해서 SPM을 검색
1702
작업 공간의 위치 및 디스플레이 경계를 결정
1703
디스플레이 경계 주위의 컬링 경계 내의 SPM 이벤트들에 대한 디스플레이 데이터를 수집하기 위해서 SPM을 트래버싱
1704
SPM 이벤트들, 일시 이벤트들 및 디스플레이 데이터를 이용해서 디스플레이 경계 내의 SPM의 일부들을 렌더링함
1705
1710
소켓 API 메시지들을 수신
로컬 UI 메시지들을 수신
1706
로컬 SPM을 업데이트, 만약 필요하다면 소켓의 API 메시지들에 응답, 이력 및 일시 이벤트들에 대해 디스플레이 데이터를 생산
소켓에 API 메시지들을 송신, 로컬 SPM(이력 이벤트들)을 업데이트 및 이력 및 일시 이벤트들에 대한 디스플레이 데이터를 생산
1707
1711

도면18


네이티브 I/O 이벤트 타입 및
디스플레이의 좌표를 수신
1800
만약 존재한다면, 식별된 좌표에
매핑된 SPM의 오브젝트를 식별
1801
좌표, 식별된 오브젝트, 컨텍스트 및
I/O 이벤트 타입에 기초해서,
API 이벤트 타입(이력 이벤트,
일시 이벤트)을 결정
1802
API 컴플라이언트 메시지를 생산,
로컬 SPM을 업데이트,
디스플레이 데이터를 업데이트
1803