

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6417408号  
(P6417408)

(45) 発行日 平成30年11月7日(2018.11.7)

(24) 登録日 平成30年10月12日(2018.10.12)

(51) Int. Cl. F I  
G 0 6 F 1 3 / 0 0 ( 2 0 0 6 . 0 1 ) G 0 6 F 1 3 / 0 0 6 5 0 A

請求項の数 30 (全 46 頁)

(21) 出願番号	特願2016-518315 (P2016-518315)	(73) 特許権者	513199154 ハワース, インコーポレイテッド HAWORTH, INC. アメリカ合衆国 49423 ミシガン州 ホランド, ワン ハワース センター
(86) (22) 出願日	平成26年2月25日 (2014. 2. 25)	(74) 代理人	100114476 弁理士 政木 良文
(65) 公表番号	特表2016-525736 (P2016-525736A)	(72) 発明者	フォーリー, デイブ, エム. アメリカ合衆国 98122 ワシントン 州 シアトル, スイート 202, 9 00 イー. パイン
(43) 公表日	平成28年8月25日 (2016. 8. 25)	(72) 発明者	ピアソン, アダム アメリカ合衆国 98122 ワシントン 州 シアトル, スイート 202, 9 00 イー. パイン
(86) 国際出願番号	PCT/US2014/018375		
(87) 国際公開番号	W02014/197015		
(87) 国際公開日	平成26年12月11日 (2014. 12. 11)		
審査請求日	平成29年2月23日 (2017. 2. 23)		
(31) 優先権主張番号	14/090, 830		
(32) 優先日	平成25年11月26日 (2013. 11. 26)		
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	61/832, 106		
(32) 優先日	平成25年6月6日 (2013. 6. 6)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 空間的イベントマップを備えた共同システム

(57) 【特許請求の範囲】

【請求項 1】

作業空間内にイベントを配置する空間的イベントマップを維持し、共同のための複数のディスプレイに通信リソースを提供するデータプロセッサ、

共同のためのディスプレイを使用するネットワークノードから、イベントを識別するメッセージを受信するためのロジックであって、識別されるイベントが、履歴イベントとして分類されるイベントと一時的イベントとして分類されるイベントを含み、前記作業空間内の位置と前記識別されるイベントの時刻を有する図形ターゲットを生成又は修正するイベントであるロジック、及び、

前記一時的イベントとして分類されるイベントを、前記一時的イベントに対応するエントリーを前記空間的イベントマップに追加せずに、前記作業空間の共同のためのディスプレイを使用するネットワークノードへ送信し、前記履歴イベントとして分類されるイベントを、前記作業空間の共同のためのディスプレイを使用するネットワークノードへ送信し、前記履歴イベントに対応するエントリーを前記空間的イベントマップに追加するロジックを備えたことを特徴とする空間的イベントマップシステム。

【請求項 2】

空間的イベントマップは、イベントのログを含み、前記ログ内のエントリーは、前記イベントに係る図形ターゲットの前記作業空間内の位置及び時刻を有することを特徴とする請求項 1 に記載のシステム。

【請求項 3】

10

20

作業空間に位置を有する図形ターゲットに関するイベントのログ、前記イベントに係る前記図形ターゲットの前記作業空間内の位置を含む前記ログ内のエントリ、前記イベントの時刻、及び、前記イベントに係る前記図形ターゲットのターゲット識別子を保存するメモリを備えたサーバ側ネットワークノードを有し、

前記ネットワークノードが、

複数の稼働中のクライアント側ネットワークノードとの接続を確立し、前記作業空間に位置を有する図形ターゲットの修正及び生成に関するイベントを識別するメッセージを受信し、前記メッセージに応答して前記ログにイベントを追加し、且つ、特定のクライアント側ネットワークノードから受信したメッセージで識別されたイベントに関するメッセージを他の稼働中のクライアント側ネットワークノードに配信するロジックを含み、

前記イベントが、履歴イベントとして分類され、前記ログに保存され且つ他のクライアント側ネットワークノードに配信される第1種のイベントと、一時的イベントとして分類され、他のクライアント側ネットワークノードに配信されるが前記ログに保存されない第2種のイベントを含むことを特徴とするシステム。

#### 【請求項4】

前記ロジックが、前記ログの一部を運搬するメッセージをクライアント側ネットワークノードへ送信し、且つ、前記作業空間内に位置を有する図形ターゲットに関するイベントを識別するデータを運搬するメッセージをクライアント側ネットワークノードから受信するためのアプリケーションプログラムインタフェースを含むことを特徴とする請求項3に記載のシステム。

#### 【請求項5】

クライアント側ネットワークノードから受信したイベントを他のクライアント側ネットワークノードへ配信するロジックを含むことを特徴とする請求項3に記載のシステム。

#### 【請求項6】

物理的表示空間を有するディスプレイ、ユーザ入力デバイス、プロセッサ及び通信ポートを備えたクライアント側ネットワークノードを有し、

前記クライアント側ネットワークノードが、

サーバ側ネットワークノードとの接続を確立し、

前記サーバ側ネットワークノードから、作業空間内に位置を有する図形ターゲットに関するイベントに係る空間的イベントログの少なくとも一部、イベントに係る前記図形ターゲットの前記作業空間内の位置を含む前記ログのエントリ、前記イベントの時刻、及び、前記図形ターゲットのターゲット識別子を検索し、

前記物理的表示空間内の表示可能領域を前記作業空間内の被マップ領域にマップし、前記被マップ領域内にある前記空間的イベントログのエントリを識別し、前記識別されたエントリにより特定される図形ターゲットを前記表示可能領域上に描画し、

前記表示可能領域内に表示される図形ターゲットの修正及び生成に関するイベントを生成する前記ユーザ入力デバイスからの入力データを受け入れ、前記イベントに基づいて前記サーバ側ネットワークノードへメッセージを送信するロジックにより構成され、

前記イベントが、履歴イベントに指定され、他のクライアント側ネットワークノードに配信され且つ前記サーバ側ネットワークノード内の前記空間的イベントログに保存される第1種のイベントと、一時的イベントに指定され、他のクライアント側ネットワークノードに配信されるが前記サーバ側ネットワークノード内の前記空間的イベントログに保存されない第2種のイベントを含むことを特徴とするシステム。

ていることを特徴とするシステム。

#### 【請求項7】

前記ロジックが前記サーバ側ネットワークノードと通信する処理を含むアプリケーションプログラムインタフェースを含むことを特徴とする請求項6に記載のシステム。

#### 【請求項8】

前記ロジックが、前記作業空間に対して前記表示可能領域をパン及びズームする手順を含むことを特徴とする請求項6に記載のシステム。

## 【請求項 9】

前記図形ターゲットがテキスト及びストロークを含むことを特徴とする請求項 3 または 6 に記載のシステム。

## 【請求項 10】

前記ログ上の前記エントリが、前記図形ターゲットをディスプレイ上に描画するのに用いられる図形構成物を識別するパラメータ（例えば、url 又は実際のファイル）を含むことを特徴とする請求項 3 または 6 に記載のシステム。

## 【請求項 11】

前記ログ上の前記エントリが、前記イベントに関連付けられたユーザを識別するパラメータを含むことを特徴とする請求項 3 または 6 に記載のシステム。

10

## 【請求項 12】

前記作業空間が、實際上無限の仮想空間を覆っていることを特徴とする請求項 1、3 及び 6 の何れか 1 項に記載のシステム。

## 【請求項 13】

サーバ側ネットワークノード内で、作業空間に位置を有する図形ターゲットに関するイベントのログ、前記イベントに係る前記図形ターゲットの前記作業空間内の位置を含む前記ログ内のエントリ、前記イベントの時刻、及び前記イベントに係る前記図形ターゲットのターゲット識別子を保存する工程、

複数の稼働中のクライアント側ネットワークノードとの接続を確立する工程、及び、

前記確立された接続上で、前記作業空間内の位置を有する前記図形ターゲットの修正及び生成に関するイベントを識別するメッセージを受信し、前記メッセージにตอบสนองして前記ログ内にイベントを追加し、且つ、前記確立された接続上で、特定のクライアント側ネットワークノードから受信したメッセージで識別されたイベントに関するメッセージを他の稼働中のクライアント側ネットワークノードに配信する工程を有し、

20

前記イベントが、履歴イベントとして分類され、前記ログに保存され且つ他のクライアント側ネットワークノードに配信される第 1 種のイベントと、一時的イベントとして分類され、他のクライアント側ネットワークノードに配信されるが前記ログに保存されない第 2 種のイベントを含むことを特徴とする方法。

## 【請求項 14】

前記ログの一部を運搬するメッセージをクライアント側ネットワークノードへ送信し、且つ、前記作業空間内に位置を有する図形ターゲットに関するイベントを識別するデータを運搬するメッセージをクライアント側ネットワークノードから受信するためのアプリケーションプログラムインタフェースを実行する工程を備えることを特徴とする請求項 13 に記載の方法。

30

## 【請求項 15】

クライアント側ネットワークノードからサーバ側ネットワークノードへの接続を確立する工程と、

前記クライアント側ネットワークノードにおいて、前記サーバ側ネットワークノードから作業空間内に位置を有する図形ターゲットに関するイベントに係る空間的イベントログの少なくとも一部、イベントに係る前記図形ターゲットの前記作業空間内の位置を含む前記ログのエントリ、前記イベントの時刻、及び、前記図形ターゲットのターゲット識別子を検索する工程、

40

前記クライアント側ネットワークノードにおいて、物理的表示空間内の表示可能領域を作業空間内の被マップ領域にマップし、前記被マップ領域内にある前記空間的イベントログのエントリを識別し、前記識別されたエントリにより特定される図形ターゲットを前記表示可能領域上に描画する工程、

前記クライアント側ネットワークノードにおいて、前記表示可能領域内に表示される図形ターゲットの修正及び生成に関するイベントを生成するユーザ入力デバイスからの入力データを受け入れる工程、及び、

前記イベントに基づき、前記確立された接続上で前記サーバ側ネットワークノードへメ

50

ッセージを送信する工程を有し、

前記イベントが、履歴イベントに指定され、他のクライアント側ネットワークノードに配信され且つ前記サーバ側ネットワークノード内の前記空間的イベントログに保存される第1種のイベントと、一時的イベントに指定され、他のクライアント側ネットワークノードに配信されるが前記サーバ側ネットワークノード内の前記空間的イベントログに保存されない第2種のイベントを含むことを特徴とする方法。

【請求項16】

前記サーバ側ネットワークノードで実行されるアプリケーションプログラムインタフェースに準拠したメッセージを送信する工程を含むことを特徴とする請求項15に記載の方法。

10

【請求項17】

前記クライアント側ネットワークノードにおいて、前記作業空間に対して前記表示可能領域をパン及びズームする手順を実行する工程を含むことを特徴とする請求項15に記載の方法。

【請求項18】

前記図形ターゲットがテキスト及びストロークを含むことを特徴とする請求項13または15に記載の方法。

【請求項19】

前記ログ上の前記エントリが、前記図形ターゲットをディスプレイ上に描画するのに用いられる図形構成物を識別するパラメータを含むことを特徴とする請求項13または15

20

【請求項20】

前記ログ上の前記エントリが、前記イベントに関連付けられたユーザを識別するパラメータを含むことを特徴とする請求項13または15に記載の方法。

【請求項21】

前記作業空間が、實際上無限の仮想空間を覆っていることを特徴とする請求項13または15に記載の方法。

【請求項22】

コンピュータ読み出し可能な持続性メモリ、

作業空間内に位置を有する図形ターゲットに関するイベントのログ、前記イベントに係る前記図形ターゲットの前記作業空間内の位置を含む前記ログ内のエントリ、前記イベントの時刻、及び前記イベントに係る前記図形ターゲットのターゲット識別子を含み、前記メモリに保存される機械読み出し可能なデータ構造、

30

複数の稼働中のクライアント側ネットワークノードとの接続を確立するための実行可能な命令、及び、

前記確立された接続上で、前記作業空間に位置を有する図形ターゲットの修正及び生成に関するイベントを識別するメッセージを受信し、前記メッセージに応答して前記ログ内にイベントを追加し、且つ、前記確立された接続上で、特定のクライアント側ネットワークノードから受信されたメッセージで識別されたイベントに関するメッセージを他の稼働中のクライアント側ネットワークノードに配信する実行可能な命令を有し、

40

前記イベントが、履歴イベントとして分類され、前記ログに保存され且つ他のクライアント側ネットワークノードに配信される第1種のイベントと、一時的イベントとして分類され、他のクライアント側ネットワークノードに配信されるが前記ログに保存されない第2種のイベントを含むことを特徴とする製品。

【請求項23】

前記ログの一部を運搬するメッセージをクライアント側ネットワークノードへ送信し、且つ、前記作業空間内に位置を有する図形ターゲットに関するイベントを識別するデータを運搬するメッセージをクライアント側ネットワークノードから受信するためのアプリケーションプログラムインタフェースの実行を含むことを特徴とする請求項22に記載の製品。

50

## 【請求項 2 4】

コンピュータ読み出し可能な持続性メモリ、  
クライアント側ネットワークノードからサーバ側ネットワークノードへの接続を確立するための実行可能な命令、

前記クライアント側ネットワークノードにおいて、前記サーバ側ネットワークノードから、作業空間内に位置を有する図形ターゲットに関するイベントに係る空間的イベントログの少なくとも一部、イベントに係る前記図形ターゲットの前記作業空間内の位置を含む前記ログ内のエントリ、前記イベントの時刻、及び、前記図形ターゲットのターゲット識別子を検索するための実行可能な命令、

前記クライアント側ネットワークノードにおいて、物理的表示空間内の表示可能領域を前記作業空間内の被マップ領域にマップして、前記被マップ領域内にある前記空間的イベントログのエントリを識別し、前記識別されたエントリにより特定される図形ターゲットを前記表示可能領域上に描画する実行可能な命令、

前記クライアント側ネットワークノードにおいて、前記表示可能領域内に表示される図形ターゲットの修正及び生成に関するイベントを生成するユーザ入力デバイスからの入力データを受け入れる実行可能な命令、及び、

前記イベントに基づき、前記確立された接続上で前記サーバ側ネットワークノードへメッセージを送信するための実行可能な命令を有し、

前記イベントが、履歴イベントに指定され、他のクライアント側ネットワークノードに配信され且つ前記サーバ側ネットワークノード内の前記空間的イベントログに追加される第 1 種のイベントと、一時的イベントに指定され、他のクライアント側ネットワークノードに配信されるが前記サーバ側ネットワークノード内の前記空間的イベントログに追加されない第 2 種のイベントを含むことを特徴とする製品。

## 【請求項 2 5】

前記サーバ側ネットワークノードにおいて実行されるアプリケーションプログラムインタフェースに準拠したメッセージを送信するための実行可能な命令を含むことを特徴とする請求項 2 4 に記載の製品。

## 【請求項 2 6】

前記クライアント側ネットワークノードにおいて、前記作業空間に対して前記表示可能領域をパン及びズームする手順を実行するための実行可能な命令を含むことを特徴とする請求項 2 4 に記載の製品。

## 【請求項 2 7】

前記図形ターゲットがテキスト及びストロークを含むことを特徴とする請求項 2 2 または 2 4 に記載の製品。

## 【請求項 2 8】

前記ログ上の前記エントリが、前記図形ターゲットをディスプレイ上に描画するのに用いられる図形構成物を識別するパラメータを含むことを特徴とする請求項 2 2 または 2 4 に記載の製品。

## 【請求項 2 9】

前記ログ上の前記エントリが、前記イベントに関連付けられたユーザを識別するパラメータを含むことを特徴とする請求項 2 2 または 2 4 に記載の製品。

## 【請求項 3 0】

前記作業空間が、實際上無限の仮想空間を覆っていることを特徴とする請求項 2 2 または 2 4 に記載の製品。

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

本発明はデジタル共同のための装置、方法、及びシステムに関し、特に、複数ユーザが同時にグローバル作業空間データへアクセスするのを容易にするデジタルディスプレイシステムに関する。

10

20

30

40

50

**【背景技術】****【0002】**

デジタルディスプレイは、ホワイトボードと類似の方法で、対話的プレゼンテーションや他の目的のためによく使用される。一方のディスプレイ上で表示画像に加えられた修正が別のディスプレイ上で複製されるように、ディスプレイ同士がネットワーク接続され、共同制作のための利用を可能としたものもある。大規模ディスプレイは1を超えるユーザが、同時に、同じ表面上で説明し、注釈を付ける機会を提供する。しかしながら、複数ユーザの調整に問題が生じることがあり、ある状況において、単一ディスプレイを彼らが使用することが彼らの表現の柔軟性を制限することがある。

**【0003】**

また、デジタルディスプレイは、大面積の対話表面を提供するように構成された、一部屋内の大画面スクリーン又はスクリーンの配列を含むことができる。これより、大面積デジタルディスプレイは、異なる時刻の多数のユーザに、異なる共同制作のために共有され得ることが予期される。共同のための作業空間データはアクセスが承認されたユーザに制限されるように秘密にされるが、ユーザが対話するデジタルディスプレイは多数の場所に分散しており、必ずしもユーザによる排他的制御の下にあるわけではなく、共同制作物へのアクセスのセキュリティに問題が生じる。

**【0004】**

加えて、システムの分散的性質は、異なる場所にいる複数のユーザが、同時に及び他のユーザが当該作業データを見ていない時に、同じ作業空間データと対話し、及び作業空間データを変更できる可能性を導く。これは複数の場所における同時並行性に問題を生じさせ、作業空間データの現在の状態についての情報の共有に問題を生じさせる。

**【発明の概要】****【発明が解決しようとする課題】****【0005】**

このため、ディスプレイの分散ネットワークにおいて、共同制作物の機密性を保護する適切なセキュリティを提供しつつ、夫々のユーザが自身のアイデアを表現し、リアルタイムでアイデアを交換する最大の自由を有するようにして、複数ユーザで作業空間データを共有するのを可能とする方法を見出すことが望ましい。故に、当該問題についての堅固な解法を生み出す機会が生じている。より良いアイデア、共同及び結果が実現され得る。

**【課題を解決するための手段】****【0006】**

空間的イベントマップを実装した共同システムが説明される。共同システムは、空間的イベントマップに基づいて画像を表示するのに用いられる分散配置した多数のデジタルディスプレイを有することができる。空間的イベントマップは、また、単一の場所に単一のディスプレイを有したシステムにも採用され得る。

**【0007】**

複数の機器及び場所を介してアクセスが可能で、セッション毎に本質的に無限の二次元及び三次元作業空間を支持するシステムが開示される。

**【0008】**

作業空間内にイベントを配置するエントリを含んだ空間的イベントマップに基づいた共同システムが説明される。空間的イベントマップはイベントのログを含むことができ、ログ内のエントリはイベントに係る図形ターゲットの作業空間内の位置と時刻を有している。また、ログ内のエントリは、当該図形ターゲットをディスプレイ上に描画するのに用いる図形構成物を識別するパラメータ(例えば、url又は実際のファイル)を含むことができる。共同システムを形成するように対話するサーバ側ネットワークノードとクライアント側ネットワークノードが記載され、これによって空間的イベントマップは承認されたクライアントにアクセス可能となり、クライアントはローカルの表示領域を描画し、イベントを生成するために空間的イベントマップを利用することができる。生成されるイベントは、空間的イベントマップに追加され、且つ他のクライアントと共有され得る。

10

20

30

40

50

## 【0009】

特定の共同セッションに関連付けられる作業空間は、時間及び仮想共同空間においてイベントを配置するための、具体的な境界のない参照の一フレームを提供する無限の仮想領域として表現される。作業空間は、クライアント側ネットワークノードがその境界を越えて航行する可能性を無視できるだけの十分大きなサイズを有するという点において、実際上無限の仮想領域を覆う。例えば、1,000,000×1,000,000画素を含む物理的表示空間にマップする仮想領域を覆うサイズは、いくつかの設定において、実際上無限と考えられる。いくつかの例では、仮想空間内の位置を識別するのに用いるアドレス指定方式の範囲によってのみ、サイズが制限されるという点で、作業空間は本質的に「無限」である。また、システムは多数の作業空間を備え、夫々の作業空間がユーザ又はユーザグループにより個別にアクセス可能に構成され得る。

10

## 【0010】

共同システムは、サーバ側ネットワークノードとクライアント側ネットワークノードが共同イベントについて通信を行うためのアプリケーションプログラムインタフェース（API）に従って構成されることができる。メッセージは、作業空間内の位置及び時刻を有する図形ターゲットを生成または修正するイベントを識別するものとして定義され得る。イベントは、履歴イベントと一時的イベントに分類されることができ、履歴イベントは空間的イベントマップに保存される。一時的イベントは空間的イベントマップに永久に保存されるわけではないが、共同セッションの他のクライアントに配信される。

20

## 【0011】

一組の処理及び処理のためのパラメータを含み、空間的イベントマップを利用する作業空間への参加を提供するアプリケーションプログラムインタフェース（API）が提供される。処理の組はソフトウェア的に実装された機能を用いてデータプロセッサ内に実装されることができるが、ハードウェアにより補助され、パラメータを利用してAPIの処理を実行するように構成されることができる。

## 【0012】

上記の要約は本明細書に記載された共同システムのいくつかの態様についての基本的理解を提供するために提供される。この要約は、発明の鍵となる或いは重要な要素を特定し、又は発明の範囲を詳説することを意図するものではない。

30

## 【図面の簡単な説明】

## 【0013】

本発明はその具体的な実施形態について説明される。図面には参照が付けられているが、縮尺を示すために描かれたものではない。

【図1A】（図1A、図1Bを併せて図1とする）デジタルディスプレイ共同環境の態様の例を示す図

【図1B】デジタルディスプレイ共同環境の態様の例を示す図

【図2】地理的に分散した複数の壁型ディスプレイを備え、作業空間データが承認されたユーザの利用のために配送される共同システムを説明する図

【図3】図1のディスプレイの描画領域の作用の態様を示す図

【図4】図1のディスプレイの描画領域の作用の態様を示す図

40

【図5A】（図5A～5Eを併せて図5とする）作業空間の作業空間データの部分のデータ構造の簡略化された図

【図5B】作業空間の作業空間データの部分のデータ構造の簡略化された図

【図5C】作業空間の作業空間データの部分のデータ構造の簡略化された図

【図5D】作業空間の作業空間データの部分のデータ構造の簡略化された図

【図5E】作業空間の作業空間データの部分のデータ構造の簡略化された図

【図6】本明細書に記載されたシステムの一例における作業空間内の役者（actor）の機能特徴を説明する図

【図7】連合ディスプレイを用いて実装されたデジタルディスプレイの略図

【図8】コンピュータシステム110、例えばクライアントデバイスコンピュータシステ

50

ム(図1B)の簡略化された図

【図9】クライアントデバイスコンピュータシステム110(図1B)にアクセス可能に保存されるデータベースの模式図

【図10】共同システムのために用いられ得るユーザログイン手順のサーバ側論理(logic)の態様を説明するフローチャート

【図11】共同システムのために用いられ得るユーザログイン手順のクライアント側論理の態様を説明するフローチャート

【図12】共同セッションのために用いられる壁内の表示クライアントのクライアント側論理の態様を説明するフローチャート

【図13】共同システムにおける分散した壁型ディスプレイの利用を管理するサーバ側論理の態様を説明するフローチャート

10

【図14】共同システムのディスプレイとして利用される連合ディスプレイシステムのためのクライアント側論理の態様を説明するフローチャート

【図15】図1Bの様式において、ディスプレイが広範囲に分散した分散型ディスプレイ共同を支援するシステムを説明する図

【図16】クライアント側ネットワークノードとディスプレイの簡略化された機能ブロック図

【図17】図16に類似のクライアント側ネットワークノードの処理を説明するフローチャート

【図18】図16に類似のクライアント側ネットワークノードのロジックにより実行される手順を説明するフローチャート

20

【発明を実施するための形態】

【0014】

以下の記載は当業者が本発明を構成し、利用するのを可能とするために提示され、特定の応用とそれに必要な条件の文脈で提供される。開示された実施形態の様々な修正は、当業者にとって明白であり、本明細書に定義される一般原理は、本発明の精神及び範囲を逸脱することなく他の実施形態及び応用に適用され得る。したがって、本発明が示される実施形態に限定されることは意図されず、本明細書に開示の原理及び特徴と矛盾しない最も広い範囲が認められる。

【0015】

30

「無限の作業空間」問題はどのようにして人と機器が作業空間と時間とともに対話するかを追跡する必要がある。この核心的問題の解決のため、我々は、我々が空間的イベントマップと呼ぶものを創造した。空間的イベントマップは、作業空間内のオブジェクト及びイベントを定義するために必要な情報を含む。空間、イベント、当該空間内のイベントのマップ、及び、複数同時ユーザを含む複数のユーザによる当該空間へのアクセスの点から、技術を考慮することは有益である。

【0016】

空間： 所定の共同セッションに対し無限量の空間的情報を支援するため、例えば二次元の直交座標平面によって特徴付けられ、1つまたは両方の次元において本質的に無限の空間となっている作業空間と称する仮想空間を、例えば、新しい内容が追加されると、その内容が空間内に配置及び再配置され、ユーザが空間の一部から別の一部へと航行することができ、空間内の必要なものを必要な時に容易に見つけることができるように、組織する方法を提供する。

40

【0017】

イベント： 作業空間との対話はイベントとして処理される。人、及びシステムは、触知できるユーザインタフェース機器を介して、作業空間と対話する。イベントは物理ディスプレイ上に表示されるターゲット図形オブジェクトを定義又は指し示すデータ、ターゲット図形オブジェクトの作業空間内における生成、変更、移動及び削除としてのアクション、及び、これらに関連したメタデータを有する。メタデータは起案者、日付、時間、作業空間内の位置、イベントタイプ、セキュリティ情報、及び他のメタデータ等の情報を含

50

み得る。

【 0 0 1 8 】

作業空間内のイベントの追跡は、空間的イベントの作業空間内の現在の状態を提示するだけでなく、それを複数ディスプレイ上の複数ユーザと共有し、その内容に係る関連する外部情報を共有することを可能にし、且つ、空間的データが時間とともにどのように発展するかの理解を可能とする。また、空間的イベントマップは境界のない作業空間を定義するものでありながら、必要なデータ量としては合理的なサイズを有することができる。

【 0 0 1 9 】

システムにおいて、いくつかの異なる種類のイベントがあり得る。イベントは永続的イベントに分類され得る。永続的イベントは、履歴イベントとも呼ばれ、永久に、或いはシステムが作業空間を維持するために必要な時間、その有用寿命の間保存される。イベントは、短期間においてのみ有用又は関係があり、セッションに参加している他のクライアント同士でライブ共有される一時的イベントに分類され得る。永続的イベントは取り消し/再生イベントストリーム内に保存される履歴イベントを含み得る。当該イベントストリームはセッションの空間的イベントマップと同じか、空間的イベントマップから派生したものであり得る。一時的イベントはシステムの取り消し/再生イベントストリームに保存されないイベントを含み得る。(複数の)空間的イベントマップは、永続的イベントと一時的イベントの両方がシステムの作業空間上に設けられた実施形態において、作業空間内の時刻と場所を追跡するために共同システムによって利用され得る。

【 0 0 2 0 】

マップ： 作業空間内のイベントのマップは個別の空間的イベント全ての合計を含むことができる。作業空間に対し永続的な空間的イベントが利用できる時、その作業空間は特定のサイズの表示可能領域を有し、表示可能領域に表示される作業空間内の場所又は領域を識別するディスプレイ又はスクリーンに「マップ」される。

【 0 0 2 1 】

複数ユーザアクセス： 1つの鍵となる特性は、同時に作業空間上で作業している全てのユーザ、又は複数ユーザが、リアルタイムに近い方法で他のユーザの対話内容を把握する必要のある点である。空間的イベントマップは異なる物理的位置にディスプレイを有するユーザが、彼らの個々の表示可能領域内で、任意の所定の作業空間上にいる全てのユーザに対し、永続的イベントと一時的イベントの両方について、リアルタイムに近いイベントを経験することを可能にする。

【 0 0 2 2 】

図 1 A はデジタルディスプレイ共同環境の態様の一例を説明するものである。この例では、複数のユーザ 1 0 1 a ~ h (総称して 1 0 1) が、複雑な画像、音楽、動画、文書、及び/又は他のメディア(全て、図 1 A において 1 0 3 a ~ d (総称して 1 0 3)として示されている)の創作においてお互いに共同を望むことがあり得る。図示された例におけるユーザは、互いに共同するために、例えばタブレット 1 0 2 a、パーソナルコンピュータ(PC) 1 0 2 b、及び、多数の大画面ディスプレイ 1 0 2 c、1 0 2 d、1 0 2 e (総称してデバイス 1 0 2)のような、電子的ネットワークノードとして構成された種々のデバイスを使用する。図示された例において大画面ディスプレイ 1 0 2 c は、本明細書中で「壁」と呼ばれることもあるが、1より多いユーザ(例えば、ユーザ 1 0 1 c と 1 0 1 d、ユーザ 1 0 1 e と 1 0 1 f、及び、1 0 1 g と 1 0 1 h)に対応する。クライアント側ネットワークノードと呼ばれるユーザデバイスは、作業空間内のイベントを表示するための表示可能領域が割り当てられたディスプレイを有している。所定のユーザの表示可能領域は、事実上無限の空間の作業空間と比較して限られた領域又は空間を有する、ディスプレイのスクリーン全体、スクリーンの一部、スクリーンに表示されるウィンドウ等を含み得る。

【 0 0 2 3 】

図 1 B は図 1 A と同じ環境を図示する。図 1 B に示されるように、「壁」と呼ばれることもある大画面ディスプレイ 1 0 2 c、1 0 2 d、1 0 2 e は、サーバ側物理ネットワー

10

20

30

40

50

クノードとして構成された中央の共同サーバ105とネットワーク通信を行う夫々のクライアント側物理ネットワークノード10によって制御され、中央共同サーバ105は1以上の作業空間のための空間的イベントスタックを記憶するデータベース106とアクセス可能である。本明細書中において、物理ネットワークノードはネットワークに取り付けられた稼働中の電子デバイスであり、通信チャネルを介して情報を送信し、受信し、又は転送することが可能である。ネットワークノードとして採用され得る電子デバイスの例は、任意の種類のコピュータ、ワークステーション、ラップトップコンピュータ、ハンドヘルドコンピュータ、及びスマートフォンを含む。本明細書中において、用語「データベース」は必ずしも任意の単一構造を意味しない。例えば、2つ以上の別々のデータベースは、一緒にまとめることを考慮すると、本明細書の用語における「データベース」を構成する。

10

#### 【0024】

共同サーバ105で稼働するアプリケーションは、Apache又はnginxのようなウェブサーバソフトウェアを用いて実行されることができ、例えば、Linuxのようなオペレーティングシステムが稼働する仮想マシン上で実行されることがあり得る。サーバ105は図1Bにおいて単一のコンピュータとして一般的に図示されている。しかしながら、サーバ構成には、大規模クラウドベースサービスに一般的な、サーバアプリケーションが稼働する多数のコンピュータからなるシステムを含み得る。サーバ構成は、共同セッション中の夫々のクライアント毎に1より多いチャネルを含んで、様々なタイプの通信チャネル用に構成されることのできる通信モジュールを含む。例えば、ネットワークを介したリアルタイムに近い更新のために、クライアントソフトウェアは、例えばWeb Socketプロトコルに基づいたメッセージベースのチャネルを使ってサーバ通信モジュールと通信することができる。最初の大容量作業空間データの受信と同様、ファイルのアップロードに際して、クライアントソフトウェアはサーバ通信モジュールとHTTPを介して通信する。サーバは、例えばRuby on Railsを実装したJavaScriptで記述されたフロントエンドプログラムを実行し、例えばOAuthに基づいた認証/承認に対応し、多数の分散したクライアント同士の調整を支援することができる。サーバ通信モジュールは、作業空間データ内のユーザのアクションを記録し、ユーザアクションを適用可能として他のクライアントに中継する機能を実行する、Web Socketアプリケーションのようなメッセージベース通信プロトコルスタックを含み得る。このシステムは例えばNode.jsプラットフォーム上、或いは高負荷ソケットアプリケーションを扱うために設計された他のサーバ技術上で稼働する。

20

30

#### 【0025】

データベース106は、夫々のセッションの空間的イベントマップについて、ディスプレイキャンバス上に表示可能なオブジェクトに関するイベントを含み、又は識別することのできる作業空間データセットのデジタル表現を保存している。作業空間データセットは空間的イベントスタックの形式で実装されることができ、少なくとも永続的空間イベントが、取消動作の間、先入れ後出しのパターンで、スタックに追加され(プッシュ)、スタックから取り出される(ポップ)ように管理される。多数の異なる作業空間毎に作業空間データセットがあり得る。所定の作業空間のためのデータセットはデータベース内に構成されるか、又は作業空間と結びついた機械により可読な文書として構成され得る。作業空間は無限の、又は事実上無限の大きさを有し得る。作業空間データは、表示クライアントによってディスプレイ壁上の表示領域内に表示可能なオブジェクトを識別するイベントデータ構造を含み、作業空間内の時間と場所を当該イベントデータ構造によって識別されたオブジェクトと関連付ける。デバイス102の夫々は全体の作業空間のうち一部だけを表示する。ディスプレイ壁はオブジェクトを表示するための表示領域を有し、表示領域は作業空間内の対応する領域にマップされる。当該対応する領域は、作業空間内のユーザ位置を中心とする、或いはそうでない場合作業空間内のユーザ位置から定まる領域に対応する。作業空間内の対応する領域への表示領域のマッピングは、表示クライアントにより描画されるべき表示領域内にある作業空間データ内のオブジェクトを識別し、ディスプレイ上

40

50

の表示領域内の位置におけるユーザのタッチ入力とリンクするオブジェクトを識別するのに利用できる。

【 0 0 2 6 】

サーバ 1 0 5 及びデータベース 1 0 6 は、作業空間内に位置を有する図形ターゲットに関するイベントのログ、イベントに係る図形ターゲットの作業空間内の位置を含む当該ログ内のエントリ、イベントの時刻、及び、イベントに係る図形ターゲットのターゲット識別子を保存するメモリを備え、サーバ側ネットワークノードを構成することができる。サーバは、複数の稼働中のクライアント側ネットワークノードとの接続を確立し、作業空間内の位置を有する図形ターゲットの修正及び生成に関するイベントを識別するメッセージを受信し、当該メッセージに回答してログにイベントを追加し、及び、特定のクライアント側ネットワークノードから受信したメッセージのうち識別されたイベントに関するメッセージを他の稼働中のクライアント側ネットワークノードへ配信するための論理（ロジック）を含み得る。

10

【 0 0 2 7 】

サーバ 1 0 5 の論理は、アプリケーションプログラムインタフェースを含むことができ、規定の手順及びパラメータの組を備え、これによってログの一部を運搬するメッセージをクライアント側ネットワークノードに送信し、作業空間内の位置を有する図形ターゲットに関するイベントを識別するデータを運搬するメッセージをクライアント側ネットワークノードから受信することができる。

【 0 0 2 8 】

20

また、サーバ 1 0 5 の論理は、あるクライアント側ネットワークノードから受信したイベントを他のクライアント側ネットワークノードへ配信する処理を備えたアプリケーションインタフェースを含み得る。

【 0 0 2 9 】

A P I に準拠したイベントは、ログに保存され、且つ他のクライアント側ネットワークノードへ配信される第 1 種のイベント（履歴イベント）と、他のクライアント側ネットワークノードへ配信されるがログに保存されない第 2 種のイベント（一時的イベント）を含み得る。

【 0 0 3 0 】

サーバ 1 0 5 は複数の作業空間に対し作業空間のデータセットを保持でき、セッションに参加する表示クライアントに作業空間データを提供することができる。作業空間データは、その後、表示クライアントソフトウェアを含む適当なソフトウェア 1 1 2 を備えたコンピュータシステム 1 1 0 によって、ディスプレイ上に表示する画像を決定し、対話対象のオブジェクトにディスプレイ面上の位置を割り当てるのに利用される。サーバ 1 0 5 は、異なる共同セッションに対して、多数の作業空間を保持及び維持する。夫々の作業空間はユーザのグループにより関連付けられ、グループ内の承認されたユーザのみがアクセス可能に構成することができる。

30

【 0 0 3 1 】

いくつかの代替する構成では、サーバ 1 0 5 は、そのデバイス上で視認可能なキャンパス内の部分を示す「ビューポート」をデバイス 1 0 2 毎に追跡し、当該ビューポートを表示するのに必要なデータをデバイス 1 0 2 毎に提供することができる。

40

【 0 0 3 2 】

描画オブジェクトの表示、ユーザ入力の処理、及びサーバとの通信を担うクライアントデバイスで稼働するアプリケーションソフトウェアは、HTML 5 或いは他のマークアップベースの手法に基づくものであることができ、ブラウザ環境で実行されるものであり得る。これは多数の異なるクライアントのオペレーティングシステム環境への容易な対応を可能とする。

【 0 0 3 3 】

データベース 1 0 6 に保存されるユーザインタフェースデータは、イメージビットマップ、ビデオオブジェクト、多ページ文書、スケーラブルベクタグラフィックス等の図形構

50

成物を含む様々な種類のオブジェクトを含む。デバイス102は、夫々、共同サーバ105とネットワーク104を介して通信している。ネットワーク104はLAN、WAN、ルータ、スイッチ、WiFiコンポーネント、セルラーコンポーネント、有線及び光学コンポーネント、及びインターネット等の全ての形式のネットワークコンポーネントを含み得る。ある状況では2以上のユーザ101は同じ部屋内にあり、彼らのデバイス102がWiFiを介して共同サーバ105と通信する。別の状況では2以上のユーザ101は何千マイルも互いに離れていて、彼らのデバイス102がインターネットを介して共同サーバ105と通信する。壁102c、102d、102eは、画像を表示するだけでなく、スタイラス又は1以上の指のような体の一部の何れかで表示面に触れることによって提示されるユーザジェスチャを検知することができるマルチタッチデバイスであり得る。いくつかの実施形態において、壁(例えば102c)は一以上の指(或いは、例えば腕全体)によるタッチとスタイラスによるタッチを区別することができる。ある実施形態では、壁は赤外光を出射し受光した光を検出することによって、ユーザの指からの反射光が壁が受光した周辺光と区別する特性を有することから、タッチを検知する。スタイラスは、壁が周辺光とユーザの指からの反射光の両方を区別できる方法で、自身の赤外光を出射する。壁102は、例えば、フィンランド国ヘルシンキに所在地を有するMultiTouch社製造のモデルNo. MT533UTBL MultiTactionセルが、水平方向及び垂直方向にタイルされた配列であり得る。表現手段の多様性を提供するため、壁102cは「状態」を維持するように動作される。即ち、それは所定の入力に対し、とりわけ入力に依存して異なった反応をし得る。例えば、ツールバーを使用して、ユーザは多数の利用可能なブラシスタイル及び色の何れかを選択できる。一度選択されると、壁は、続くスタイラスによるストロークが選択されたスタイル及び色を使用して線を描く状態にある。

10

20

#### 【0034】

図示された実施形態において、ディスプレイ配列は、合計で高さ5フィート幅30フィートの、複数ユーザが壁の異なる部分に立って同時に操作するには十分な大きさの表示可能領域を有することができる。しかしながら、この実施形態では、壁は異なるユーザの指同士を区別しない、或いは異なるユーザによって操作されるスタイラス同士を区別しないので、壁における表現の柔軟性は複数ユーザがいる状況で制限されることがある。故に、あるユーザが壁をある所望の状態に置くと、第2のユーザの入力が異なって扱われるべきと認識する方法を壁は有してないので、第2のユーザは同じ状態を使用するように制限される。

30

#### 【0035】

この制限を避けるため、クライアント側ネットワークノードは壁102c上の「描画領域」を定義できる。本明細書中において、描画領域は、壁の状態の少なくとも一態様が壁上の他の領域とは独立に変化することのできる領域である。本実施形態において、描画領域間で異ならせることのできる状態の態様はスタイラスを用いて壁上に描く線の特性を含む。指のタッチ行動に対するシステムの応答のような他の状態の態様には、描画領域の影響を受けないものがあり得る。

#### 【0036】

図2は、地理的に分散した、表示クライアントが配置された多数の施設(例えば、施設1及び施設2)を接続する共有サーバ105を備えた分散型共同システムを図示するものである。例えば、施設1はニューヨーク市にあり、施設2はロサンゼルスにあることがあり得る。共同システムにおいて使用可能な表示クライアントが設置される多くの他の物理的場所があり得る。この例では、施設1は第1室151、第2室152、及び第3室153を備える。施設2は第1室161、第2室162、及び、第3室163を備える。施設1の第1室151は複数のディスプレイを用いて実装された大規模ディスプレイを備える。施設1の第2室152は単一スクリーンの中規模ディスプレイを備える。施設1の第3室153は個室の事務室或いは他の部屋であり、パーソナルコンピュータ又はラップトップが、選択された作業空間内で対話するセッションのための表示クライアントとして利用

40

50

され得る。この図において、施設 2 は施設 1 と同様であり、第 1 室 1 6 1、第 2 室 1 6 2、及び第 3 室 1 6 3 を備える。施設 2 の第 1 室 1 6 1 は複数のディスプレイを用いて実装された大規模ディスプレイを備える。施設 2 の第 2 室 1 6 2 は単一スクリーンの中規模ディスプレイを備える。施設 2 の第 3 室 1 6 3 は個室の事務室或いは他の部屋であり、パーソナルコンピュータ、ラップトップ、携帯パッド、又は携帯電話が、セッションのための表示クライアントとして利用し得る。

【 0 0 3 7 】

図 2 は、遠く離れて配置された大規模又は中規模ディスプレイ（或いは壁）を頼りにした分散型共同システムに関連して生じる問題を示している。大規模又は中規模ディスプレイは、一般に、個々のユーザによる排他的制御下にあるわけではない。それ故、共同サーバ 1 0 5 は、所定の時刻にディスプレイにアクセスした人について、情報を有しないことがあり得る。

10

【 0 0 3 8 】

図 3 は壁 1 0 2 c を示している。この例における壁は高さが 6 フィートで幅が 3 0 フィートである。それは初めは既定の背景色又は背景画像であり、壁全体にわたって既定の描画状態を有している。また、作業空間データは、壁の表示領域内の物理的位置にマップされる作業空間内の位置を有する複数のオブジェクト 3 0 1 a ~ 3 0 1 h（オブジェクト 3 0 1 と総称する）を定義できる。オブジェクト 3 0 1 は例えばテキスト、画像又は図面を含んだカードを含むことができ、表示領域に表示される。また、オブジェクト 3 0 1 は、ユーザがカードの内容と対話するのを可能にする特徴、又は、タッチスクリーンのような、ディスプレイのユーザインタフェースツールの使用によりカードと結びついた機能を含むことができる。また、図 3 に示すように、描画オーバレイオブジェクト 3 0 2 が壁の表示領域内に表示されている。

20

【 0 0 3 9 】

描画状態は領域に表示されるオブジェクト 3 0 1、3 0 2 に独立な領域の特性であることができ、図 3 の実施形態におけるブラシタイプ、ブラシサイズや色等の線の外観の特性を含む線描画特性によって定義される。例を挙げると、システムは、スタイラス又は 1 以上の指（本明細書において、筆記用具と総称することがある）の何れかを用いて、ユーザ 1 0 1 c が壁に触れると、その近くにツールバーが現れ、描画領域 2 1 2 が定義されるように構成され得る。タッチポイントに触れることは、本明細書中で「ユーザ入力を開く」と呼ばれることがあるものの一実施形態であり、他の実施形態は読者にとって明らかであろう。新しく定義された描画領域の初期の描画状態は（ブラシ種 = インク、線幅 = 5 mm、色 = 白のような）予め設定された既定値であるが、種々の実施形態において、壁の他の領域の既定の状態と一致していることも一致していないこともあり得る。図 2 の実施形態では、ある描画領域に対して確立された描画特性はその描画領域について適用される。線描画はプログラムが壁 1 0 2 c の任意の該当領域の所有者であるか否かに拘らず、コンピュータシステム 1 1 0 上で稼働し得る任意のアプリケーションプログラムよりも論理的に上のレイヤーで、壁上で実行される。

30

【 0 0 4 0 】

図 3 の実施形態では、描画領域は垂直方向において壁の全域を覆っているが、他の実施形態では、領域はより短く、及び / 又は矩形でない形状を有することがあり得る。また、図 3 の実施形態では、左手及び右手境界 2 1 4 及び 2 1 6 により描画領域の境界を知覚できるようにしているが、別の実施形態では領域を区画するのに背景の陰影付け（シェーディング）等の他の手段を利用し得る。更に別の実施形態では領域の境界はユーザに知覚できない。左側と右側に十分な空間がある場合、クライアント側コンピュータシステム 1 1 0 は、ユーザのタッチ位置を中心とする位置において描画領域を生成することができる。描画領域は最小の幅  $W_{min}$  と理想的な幅  $W_{ideal}$  を有する。最小の幅は、好ましくは、合理的に自由な表現が可能な最小の幅となるように選択され、図 3 の実施形態において 4 フィートである。理想的な幅は、好ましくは、平均的なユーザが両腕を水平方向に伸ばした時の最も広い幅となるように選択され、図 3 の実施形態において 6 フィートである

40

50

## 【0041】

ユーザのタッチ点の両側に十分な空間がある場合、コンピュータシステム110は初期領域の幅をWidealに設定できる。これは図3に示された状況である。ユーザのタッチ位置が、それを中心とする新たな描画領域との関係で壁の終端に近すぎる場合、コンピュータシステム110は新しい描画領域を壁端で終端させる。それでも、十分な空間があれば、新たな描画領域はユーザのタッチ位置を中心として置かれなければならないことにより、新たな描画領域は幅Widealを有する。一方、タッチ位置を中心とする描画領域を生成できる程度にユーザのタッチ位置が壁端から十分離れてはいるが、新たな描画領域が壁端からWmin未満の距離になる場合には、壁端と新たな描画領域との間のギャップ空間は利用不可能とみなされる。この場合コンピュータシステム110は当該利用不可能な空間を覆うように新たな描画領域を拡大する。

10

## 【0042】

図4は、例として、二人のユーザ101cと10dが同時に壁を使用できる状況を説明するものである。初めに、ユーザ101cがタッチ位置516において壁102cに触れると、これに応答してコンピュータシステム110が描画領域512をツールバー510と一緒に生成する。随意に、ユーザ101cは領域512内の線の外観特性を変更するためにツールバー510に触れることができる。次に、第2のユーザ101dが壁102cの背景内にある(即ち、全ての既存の描画領域外にある)タッチ位置518において壁102cに触れると、第2描画領域514が、ツールバー520と一緒に、コンピュータシステム110によって生成される。ユーザ101dがこの時に領域514内で線を描く場合、コンピュータシステム110は、描画領域512に対してユーザ101cによって設定済みの線種よりはむしろ既定の線特性で、それを塗る。その後、ユーザ101dは、領域514内の線の外観特性を変更するため、ツールバー520のコントロールに随意的に触れる。その後に領域514に描かれる線には新しい線外観特性が適用される。領域512の線外観特性は変化しない。

20

## 【0043】

描画領域はまた、スタイラスの移動を自動追跡するために作成されることができる。多数の可能な追跡アルゴリズムは読者にとって明白であるが、下記の最低限の規則に従うものが好ましい:(1)スタイラスが相対的に当該領域の中心近傍にある限り領域は移動せず、(2)スタイラスが領域の境界に接近すると、境界がスタイラスの前方を維持するように領域が移動する。

30

## 【0044】

描画領域は局所的なディスプレイ壁に効果を及ぼすことのできるユーザ対話の一例を提供するが、大域的な作業空間データには効果を及ぼさない。この例に示されているように、オブジェクト301、302の位置は、表示領域、ツールバー、及び当該領域内の描画オーバレイの割り当ての影響を受けない。もちろん、他のタイプのユーザインタフェース対話において、オブジェクト301、302の位置は移動され得るが、その移動は大域的作業空間データ内のオブジェクトに関連したイベントであり得る。

40

## 【0045】

局所的な壁との対話に基づいたユーザ入力の解釈に関する様々な動作が、「デジタルホワイトボードのための領域ダイナミクス」と題する2013年2月4日出願の同時係属中の米国特許出願番号13/758,984に記載され、本願明細書中において参照により取り込まれる。これらの動作は、いくつかの実施形態において共同サーバで維持される共有の作業空間データに殆ど或いは全く影響を与えずに、ローカルのコンピュータシステム110により実行され得る、壁でのユーザ入力及び画像データの局所的処理を図示的に説明するものである。

## 【0046】

図5A~5Eは、共同サーバ105のデータベースによって維持される作業空間データの一部となり得るデータ構造を表している。図5Aにおいて、あるイベントデータ構造が

50

図示されている。イベントは作業空間データとの対話であり、当該作業空間データの変化を招き得る。したがって、イベントはイベント識別子、時刻印 (timestamp)、セッション識別子、イベント型パラメータ、クライアント識別子、及び作業空間内の位置の配列を含み、対応するイベントの1以上を含むことができる。例えば時刻印は、競争するイベントが同一オブジェクトに影響を及ぼす競争的状況の可能性を最小にするために、ミリ秒或いはより細かい分解能を有することが望ましい。また、イベントデータ構造は、クライアントディスプレイでのタッチスクリーン上のストローク (stroke) と結びついている作業空間内のオブジェクトを識別するUIターゲットを含み得る。イベントは、例えばストロークの表示パラメータを示すスタイルイベントを含み得る。イベントは、テキストオブジェクトの作業空間内における登録、修正又は移動を示すテキスト型イベントを含み得る。イベントは、カード型オブジェクトの作業空間内における生成、修正又は移動を示すカード型イベントを含み得る。イベントは、ストロークの位置配列、及び、例えば色や線幅等のストロークの表示パラメータを識別するストローク型イベントを含み得る。

10

**【0047】**

イベントは永続的な履歴イベントと一時的イベントに分類される。作業空間データに追加するためのイベントの処理、及びユーザ間での共有は、イベントの分類に依存し得る。この分類はイベント型パラメータに固有のものであるか、イベントデータ構造内の付加的なフラグ又はフィールドが分類を示すために使用され得る。

**【0048】**

空間的イベントマップは履歴イベントのエントリを有するイベントのログを含み、夫々のエントリは図5Aに図解されるような構造を含んでいる。サーバ側ネットワークノードは一時的及び履歴イベントを運搬するメッセージをクライアント側ネットワークノードから受信し、当該一時的イベントに対応するエントリをログ内に追加することなく他のクライアント側ネットワークノードに送信し、当該履歴イベントに対応するエントリをログ内に追加したうえで他のクライアント側ネットワークノードに送信するための論理 (ロジック) を含む。

20

**【0049】**

図5Bはカードデータ構造を図示している。カードデータ構造は、セッション識別子、カード型識別子、配列識別子、クライアント識別子、カードの大きさ、カードに関連付けられたファイルの形式、及び、作業空間内のセッションの位置を含む、作業空間内のオブジェクトについての現在状態の情報を識別する属性のキャッシュを提供し得る。

30

**【0050】**

図5Cはチャンクと呼ばれる、多数のイベント及びオブジェクトを1つのキャッシュ可能な組み合わせに統合したデータ構造を図示している。データ構造はセッションID、チャンク内に含まれるイベントの識別子、及び、チャンクが生成された時刻印を含む。

**【0051】**

図5Dは選択された作業空間内のセッションに参加しているユーザへのリンクのためのデータ構造を図示している。このデータ構造はアクセストークン、セッションに係る表示クライアントのクライアント識別子、表示クライアントとリンクしたユーザ識別子、ユーザがセッションにアクセスした最後の時刻、有効期限、及び、セッションについての種々の情報を運搬するクッキーを含み得る。この情報は例えば、ユーザの作業空間内の現在位置を維持することができ、ユーザがログインした夫々の時刻において、ログインと関連付けられた表示クライアントに表示するための作業空間データを決定するのに利用し得る。

40

**【0052】**

図5Eはディスプレイ配列データ構造を図示しており、夫々が表示クライアントを有する連合ディスプレイにより実装される大規模ディスプレイに関連して利用され得る。当該連合ディスプレイの表示クライアントは単一のディスプレイとして機能するように協力する。作業空間データは配列IDによってディスプレイの配列を識別し、夫々のディスプレイのセッション位置を識別するディスプレイ配列データ構造を維持することができる。夫々のセッション位置は連合ディスプレイの領域内におけるxオフセットとyオフセット、

50

セッション識別子、及び深さを含み得る。

【 0 0 5 3 】

システムはクライアント側ネットワークノードとの通信を暗号化し、空間的イベントマップが保存されるデータベースを暗号化することができる。また、クライアント側ネットワークノード上において、いくつかの実施形態では、クライアント側コンピュータへのアクセスを得た侵入者によるデータの不正アクセスを防ぐため、空間的イベントマップのキャッシュコピーが暗号化される。

【 0 0 5 4 】

図 6 は、作業空間に係る作業空間データを生成、修正、配信、及び表示するのに使用される分散型共同システムの機能構成を表現した図である。基本構成はサーバ 1 0 5 等のサーバにより実行されるイベントデータを管理する共同サービス 6 0 1、サーバ 1 0 5 等のサーバにより実行される、又はピアネットワークノード等の当該サーバがアクセス可能な他のコンピュータシステムに設けられたポータルサービス 6 0 2、及び、クライアント側ネットワークノードに設けられ、ユーザ対話を有効とした表示クライアント 6 0 3 を含む。表示クライアント 6 0 3 は共同サービス 6 0 1 及びポータル 6 0 2 と通信を行う。表示クライアント 6 0 3 と共同サービス 6 0 1 との間の通信チャンネル 6 1 3 は、セッション履歴のダウンロード、及びセッションイベントのライブ更新を管理する。また、このチャンネル 6 1 3 を介して、表示クライアント 6 0 3 はイベントに関連付けられ得る画像を共同サービス 6 0 1 にアップロードし得る。表示クライアント 6 0 3 は通信チャンネル 6 2 3 を介してポータル 6 0 2 と通信を行う。ポータル 6 0 2 は作業空間データ、セッション管理、及び、ユーザ管理のためのホームページを管理する。このポータルはユーザログイン、認証、及び、通信チャンネル 6 1 3 に代替して或いは通信チャンネル 6 1 3 と並行して画像ファイル等を配送するのに利用され得る。共同サービス 6 0 1 及びポータル 6 0 2 はチャンネル 6 1 2 を介して通信する。共同サービス 6 0 1 及びポータル 6 0 2 は認証及び承認プロトコルを管理し、セッション管理を整理し、作業空間データ管理を行う。

【 0 0 5 5 】

表示クライアント 6 0 3 は、アクセス可能なメモリに保存され、共同セッションを支援するための、HTML 5 クライアント、ディスプレイ配列実現のための壁配列調整、作業データの解析、検索及び描画、サーバ及びディスプレイ壁において作業空間データとのライブ対話を管理するセッションイベントアプリケーションを含む論理（ロジック）を提供するコンピュータプログラムを有した物理又は仮想コンピュータシステムを含み、クライアント側ネットワークノードの一部であり得る。

【 0 0 5 6 】

ポータル 6 0 2 は、アクセス可能なメモリに保存され、ユーザの共同サーバへのアクセスを支援するための論理を提供するコンピュータプログラムを有した物理又は仮想コンピュータシステムを含み、サーバ側ネットワークの一部であり得る。当該思考は、ログインリソース付きウェブページのような初期エントリーポイントをユーザに提供するアプリケーション、ユーザアカウント及びセッションの参加の管理、OAuth に基づくサービス等の承認サービスの提供、及びアカウントデータを含み得る。

【 0 0 5 7 】

共同サーバ 6 0 1 は、サーバ側ネットワークノードの一部であり得るとともに、セッションイベントデータを管理し、クライアント間で更新されたイベントの調整を行い、キャッシュ可能な履歴及び画像をクライアントに配送し、作業空間データ内に保存されているデータベースへのアクセスを制御し得る。

【 0 0 5 8 】

図 7 は、連合して制御される複数のディスプレイ 7 0 1 ~ 7 0 4 により実装されたディスプレイに基づく、ディスプレイ壁の実現に関する随意的な技術を図示している。この例では、夫々のディスプレイ 7 0 1 ~ 7 0 4 は対応する表示クライアント 7 1 1 ~ 7 1 4 と関連付けられている。夫々の表示クライアントは、オブジェクトを作業空間から表示領域上に描画するのに使用されるブラウザを実行する。表示領域は、複数のディスプレイの

10

20

30

40

50

夫々に対応する表示領域のサブセットを複数有している。表示クライアントは、例えばオフセットパラメータ（例えばディスプレイ701に対して0, 0、ディスプレイ702に対して0, 1、ディスプレイ703に対して1, 0、ディスプレイ704に対して1, 1）を保持することによって、夫々、セッションに際して表示領域のあるサブセット内のディスプレイを管理するように構成し得る。

**【0059】**

表示クライアント711～714の夫々は、作業空間データベース106と結合している共同サーバ105との通信チャネル721～724を維持する。共同サーバ105、及び/又は、クライアントは、承認されたユーザ夫々について、作業空間内のユーザ位置を維持し得る。承認されたユーザがログインし、図7に示すようなディスプレイ配列を表示キャンパスとして選択すると、共同サーバは表示クライアント711～714の夫々をセッション及びユーザによって関連付けられたグループに結びつける。共同サーバは、その後、当該グループに属する表示クライアントの夫々に対し、作業空間内の現在のユーザ位置を表示可能領域又はキャンパスへダウンロードし得る。当該グループに属する表示クライアントは、自身のオフセットパラメータを独立に適用し、当該オフセットパラメータにより指定される作業空間のサブセット上にマップするためのセッション位置を識別し得る。代わりに、共同サーバは、表示クライアント711～714の夫々と通信を行い、配列特徴に応じたオフセットとしての現在のユーザ位置をクライアントの夫々に配送することによって、オフセット計算を行うことができる。

**【0060】**

複数のディスプレイの中の単一のディスプレイの調整を支援するため、さらに、表示クライアント711～714の夫々は、他の表示クライアントの夫々と、表示領域の管理に局所的であって大域的な作業空間データに効果を及ぼさないイベントを介して、通信することができる。代わりに、表示クライアント711～714は、共同サーバ105と単独で通信し、共同サーバ105が局所イベントをセッションに関連付けられた表示クライアントのグループに向け、大域的イベントを作業空間に関係する稼働中のセッション内の全ての表示クライアント及び作業空間データを記憶しているデータベースに向けることができる。

**【0061】**

連合ディスプレイで構成された単一のディスプレイの表示クライアントは、対応するディスプレイと結合した個々のコンピュータシステムにより実装されるか、対応するディスプレイと結合した仮想マシンを有する単一コンピュータシステムを使用して実装され得る。

**【0062】**

また、単一ディスプレイのドライバは、単一ディスプレイ壁を形成するように配列した物理的ディスプレイの集合の全表面を制御するように構成され得る。

**【0063】**

空間的イベントマップシステムは、任意の数の物理及び仮想マシンを含むクライアント側及びサーバ側のリソースの協調によって実行されるAPIを含み得る。APIの一例が下記に記載される。APIは、（複数の）サーバ側ネットワークノードにおける空間的イベントマップの維持、及び、一または複数の稼働中のクライアント側ネットワークノード同士での空間的イベントマップの共有を支援する要素を含む一方で、種々の方法により定義されることができる。この例では、APIは以下の2つのサーバによって管理されるプロセスに分解される：

**【0064】**

ソケット要求サーバ（ウェブソケット）： 接続後、関連するデータ（新しいストローク、カード、クライアント等）についてクライアントを更新するのに使用される。また、初期接続におけるハンドシェイクを処理する。

**【0065】**

サービス要求サーバ（HTTP/REST）： データ（即ち、画像及びカード）の投

10

20

30

40

50

稿のほか、キャッシュ可能な応答に使用される。

【 0 0 6 6 】

クライアント側ネットワークノードは当該 A P I に従って構成され、対応するソケット要求クライアント及びサービス要求クライアントを含む。

【 0 0 6 7 】

・ソケット要求：

ソケットサーバは、ウェブソケット ( W e b s o c k e t ) を介した接続を維持するネットワークプロトコルを実行し得る。A P I 内で使用されるメッセージはウェブソケットプロトコル内でカプセル化される。メッセージは夫々、U T F - 8 で符号化された J S O N 配列であり得る。

10

【 0 0 6 8 】

履歴の初期読み出しは、共同セッションにおける空間的イベントマップの全部であれ一部であれ、クライアント側ネットワークノードにおいて、キャッシングをサポートしたウェブソケットよりはむしろ、サービス要求サーバを介した H T T P 要求によりなされることができる。

【 0 0 6 9 】

・ソケット接続

`http://localhost:4545/<sessionId>/socket?device=<device>`

`sessionId` -- ( 文字列 ) 参加するセッションの I D

`device` -- ( 文字列 ) 壁或いはデスクトップ等のデバイスタイプ

20

【 0 0 7 0 】

・メッセージ構造

夫々のメッセージ配列の第 1 の要素は、メッセージの源となったクライアントを特定する送信者 I D ( sender-id ) である。送信者 I D はサーバ上の全てのセッション間で固有である。サーバからクライアントに送信される i d 及び c r メッセージは既定値 ( 例えば - 1 ) に設定された送信者 I D を有している。夫々のメッセージ配列の第 2 の要素は、2 文字の符号である。この符号は意図される動作のほか、配列内の残りの引数を定義する。- 1 の送信者 I D 付きで送信されるメッセージは、サーバから発されたメッセージである。

【 0 0 7 1 】

・有効なメッセージ型

以下は本明細書中の A P I の例によってサポートされるメッセージである。これらメッセージの多くは以下のパラメータを取る：

`sender-id` -- メッセージを送信するクライアントの I D であり、サーバにより発された場合 - 1 である。

【 0 0 7 2 】

・クライアント I D 要求：

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

この要求はソケット A P I との対話を可能とするために使用される。

この要求は非同期のクライアント I D 要求 / 応答ハンドシェイクを開始させる。次節では、( 与えられたクライアント I D を含む ) 新規クライアントの非同期確認を説明する。

40

`sessionId` -- ( 文字列 ) 参加する作業空間の I D

`zoomLevel` -- ( 整数 ) このクライアントにおいて望まれる拡大レベル

`x1, y1` -- ( 整数、任意 ) ユーザのビューポートの所望の原点の位置

`x2, y2` -- ( 整数、任意 ) ユーザのビューポートの所望の大きさの位置

このメッセージは、送信者 I D が一緒に送信されない。

このメッセージにおいて送信される拡大レベルは、クライアントにおいて好ましい拡大レベルである。クライアントが ( 「 i d 」 メッセージを介して ) 空のディスプレイ配列に参加する場合、当該クライアントの好ましい拡大レベルはそのディスプレイ配列の初期拡大レベルとなる。クライアントが既存のディスプレイ配列に参加する場合、「 i d 」メッ

50

ページ中の送信された拡大レベルは無視され、当該既存のディスプレイ配列に関連付けられた拡大レベルが(「av」メッセージ内に)送信される。

【0073】

・クライアントID応答:

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

クライアントは割り当てられたクライアントIDを続くソケット要求において使用するために保存しておくことが要求される。

新しいクライアントに自分のIDを通知する。この場合、送信者IDは-1に設定される。

client-id -- (文字列) 新たに参加するクライアントのID

10

【0074】

・室への参加:

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

クライアントの室への参加の試みをサーバに通知する。

room-id --ロビー又はセッションの1つを含み得る。

data --引数のワイルドカードによる組み合わせであり、室を初期化するのに使用される。

室データの引数:

セッションはセッションのIDを含むセッションID (session-id) を必要とする。配列は、下記のものが必要とする。

20

arrayId -- (文字列) ディスプレイ配列のID

x -- (整数) このディスプレイのxオフセット

y -- (整数) このディスプレイのyオフセット

width -- (整数) このディスプレイの幅

height -- (整数) このディスプレイの高さ

サーバは「room」メッセージによって応答する。

【0075】

・室への参加に対する応答:

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

30

room-id --ロビー又はセッションの1つを含み得る。

databag --室に特有の変数の集まりである。

lobbyは、下記を提供する:

pin --壁認証のための個人識別番号を含む。

sessionは、下記を提供する:

sessionName --セッションの名前を含む。

【0076】

・室リスト応答

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

40

クライアントに室の会員を通知する。室の会員は、当該クライアントと同じ室を訪れたクライアントに関する情報を含む。

roomMembershipList -- (室会員オブジェクトの配列)

【0077】

・セッション要求:

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

クライアントが参加可能な稼働中セッションのリストを欲していることをサーバに通知する。

【0078】

・セッションリスト:

50

```
// server --> client [-1, "sl", sessionList]
クライアントに参加可能な稼働中セッションを通知する。
SessionList -- (文字列の配列)
```

## 【 0 0 7 9 】

## ・オブジェクトIDの予約：

オブジェクトを生成する新しい履歴イベントを生成するに際して、受け入れ可能な新たな固有のオブジェクトIDを生成するために、これを使用する。

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

サーバは、下記により応答する：

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

10

## 【 0 0 8 0 】

## ・履歴イベント

全ての永続的イベントは履歴イベント (HistoryEvent) として送信される。これは、\* \*ウィンドウの移動\* \*テキストの設定\* \*ウィンドウの削除\* \*ウィンドウの生成を含む。

履歴イベントは、セッションの履歴に書き込まれ、当該履歴が検索されたときに復帰する。

履歴イベントはイベントID (eventId) を付すことなくサーバに送信される。サーバがイベントIDを割り当て、当該イベントを (送信元のクライアントを含む) 全てのクライアントへ放送する。

20

新たなオブジェクトIDはoidメッセージを使用して予約されることができる。

## 【 0 0 8 1 】

## ・基本メッセージ形式：

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

client-id -- (文字列) 送信元クライアントのID

target-id -- (文字列) このイベントに関連するターゲットのオブジェクト / ウィジェット / アプリのID

event-type -- (文字列) 任意のイベント型

properties -- (オブジェクト) イベントの関連するキー / 値を記述するJSONオブジェクト

30

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

client-id -- (文字列) 送信元クライアントのID

target-id -- (文字列) このイベントに関連するターゲットウィンドウのID

event-id -- (文字列) データベース内におけるイベントのID

event-type -- (文字列) 任意のイベント型

properties -- (オブジェクト) イベントの関連するキー / 値を記述するJSONオブジェクト

40

## 【 0 0 8 2 】

## ・対話の例：オブジェクトの移動

永続的イベント / 一時的イベントの分類を説明する良い例は、オブジェクトの移動である。オブジェクトがドラッグによって移動中である、又は例えばリサイズ中であるとき、一連の一時的イベント (「揮発性イベントVE」と呼ぶ) がサーバに送信され、セッション内の全てのクライアントに再放送される。ユーザがオブジェクトの移動を完了させると、クライアントが履歴イベントを送信し、オブジェクトの矩形及び順位を特定する：

```
["511d6d429b4aee0000000003", "he", "511d6f9c9b4aee00000000039", "position", { "rect" } ]...
```

サーバは新たな永続的HEレコードによって応答する。当該レコードのイベントIDの

50

包含に留意せよ。

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

イベントIDは、HTTP APIを介して取り出される履歴内にも含まれることになる。

#### 【0083】

- ・オブジェクト/アプリケーション型別の履歴イベント
- ・セッション

Create --作業セッション上にノート(note)又は画像(image)を追加する。

stroke --背景にペン又は消しゴムストロークを追加する。

10

- ・ノート

text --テキスト、及び/又は、ノートのテキストフォーマットを設定または更新する。

delete --作業セッションからノートを取り除く。

position --作業セッションにおけるノートのサイズ又は位置を更新する

pin --ノートをピン止め、又はピン止め解除する。

stroke --画像の上にペン又は消しゴムストロークを追加する。

- ・画像

delete --作業セッションからノートを取り除く。

position --作業セッションにおけるノートのサイズ又は位置を更新する。

20

pin --ノートをピン止め、又はピン止め解除する。

stroke --画像の上にペン又は消しゴムストロークを追加する。

- ・履歴イベントの詳細

#### 【0084】

- ・text

ノートのテキストについて、設定及びスタイルの設定を行う。テキスト属性及びスタイル属性は両方とも随意的である。

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

30

#### 【0085】

- ・create

サーバがカード生成(cc)メッセージ又は画像アップロードを受信した場合に、クライアントに送信する。

メッセージ生成の場合、ターゲットIDはセッションIDとなる。

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

プロパティ

id -- (整数) ウィンドウに固有の識別子

baseName -- (文字列) 背景画像のファイル名

40

ext -- (文字列) 背景画像ファイルの拡張子

rect -- (オブジェクト) ウィンドウの位置

actualWidth -- (整数) 背景画像の幅

actualHeight -- (整数) 背景画像の高さ

order -- (整数) z方向の順位

type -- (文字列) テキストを有するオブジェクトの場合「ノート」であり、他のオブジェクトの場合「画像」となる。

regionId -- (文字列) オブジェクトがキャンバス領域に生成された場合、当該キャンバス領域

hidden -- (ブール値) ウィンドウが現在隠れているか否か

50

## 【 0 0 8 6 】

## ・ delete

ウィンドウをセッションから消失させるために使用する。これは取り消しできない動作である。

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

## 【 0 0 8 7 】

## ・ Position

移動、フリング (fling) 動作、又はリサイズ後のウィンドウの位置を保存するために使用する。

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

## プロパティ

rect -- (オブジェクト) ターゲットウィンドウの位置

order -- (整数) ターゲットウィンドウの z 方向の順位

## 【 0 0 8 8 】

## ・ stroke

ストロークを保存するのに使用する。

```
/ server <-- client[client-id, "he", target-id, "stroke", { "size": 10, "bru
sh":
```

## プロパティ

locs -- (配列) [10, 1, 10, 2, 12, 3]の形式で、ストロークの位置。配列内において、座標が[x, y, x, y, x, y]の組にされている。

## 【 0 0 8 9 】

## ・ pin

ノート又は画像を定位置にピン止めし、或いは既存のピンを取り除くためにクライアントに送信される。ピン止めされているウィンドウは、ピン止めが解除されるまで移動、又はリサイズができない。

```
// server --> client[client-id, "he", session-id, event-id, "pin", {"pin": t
rue}]
```

## プロパティ

pin -- (ブール値) trueの場合ピン止めされ、falseの場合ピン止めされていない。

## 【 0 0 9 0 】

## ・ 揮発性イベント

揮発性イベントは取消 / 再生イベントストリームに記録されない一時的イベントであり、スクリーン周囲でのカードのドラッグのような進行中のストリーミングイベントにおいて効果的である。一旦ユーザが指を持ち上げると、その最後の場所を記録するために履歴イベントが使用される。

```
// server <--> client[client-id, "ve", target-id, event-type, event-properti
es]
```

client-id -- (文字列) 送信元のクライアントの ID

target-id -- (文字列) このイベントに関連するターゲットウィンドウの ID

event-type -- (文字列) 任意のイベント型

properties -- (オブジェクト) イベントの関連するキー / 値を記述する J S O

N オブジェクト

## 【 0 0 9 1 】

## ・ オブジェクト / アプリケーション型別の揮発性イベント

## ・ セッション

sb -- ストロークを開始する。一のクライアントでストロークを描画しつつ、そ

10

20

30

40

50

れが別のクライアント上でも描画されるために使用する。

sc --以前に開始したストロークを、ストロークに含めるべき別の点を与えることにより継続する。ストロークを描画しつつ、それが別のクライアント上で描画されるために使用する。

se --以前に開始したストロークを終了する。

・ノート

fling --ノートを作業セッション内のある場所から別の場所へスライドさせるアニメーションを行う。これはユーザによるフリック又はフリング動作に対する視覚的な応答である。

position --別のユーザによって移動中のノートの位置を自動更新する。

sb -- ストロークを開始する。一のクライアントでストロークを描画しつつ、それが別のクライアント上で描画されるために使用する。

sc -- 以前に開始したストロークを、ストロークに含めるべき別の点を与えることにより継続する。ストロークを描画しつつ、それが別のクライアント上で描画されるために使用する。

se --以前に開始したストロークを終了する。

・画像

fling -- 画像を作業セッション内のある場所から別の場所へスライドさせるアニメーションを行う。これはユーザによるフリック又はフリング動作に対する視覚的な応答である。

position --別のユーザによって移動中の画像の位置を自動更新する。

sb -- ストロークを開始する。一のクライアントでストロークを描画しつつ、それが別のクライアント上で描画されるために使用する。

sc -- 以前に開始したストロークを、ストロークに含めるべき別の点を与えることにより継続する。ストロークを描画しつつ、それが別のクライアント上で描画されるために使用する。

se --以前に開始したストロークを終了する。

・揮発性イベントの種類

【 0 0 9 2 】

・Fling

フリング動作を全ての接続クライアントに放送するために使用される。

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

プロパティ

velocityX (整数)フリングベクトルのx成分

velocityY (整数)フリングベクトルのy成分

【 0 0 9 3 】

・position - ve

ウィンドウ移動の途中ステップを放送するために使用される。

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

プロパティ

rect (オブジェクト)ターゲットウィンドウの位置

order (整数)ターゲットウィンドウのz方向の順位

【 0 0 9 4 】

・sb:

ストロークの開始を放送するために使用される

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

10

20

30

40

50

## プロパティ

x,y -- (整数) ストロークの開始点  
strokeId -- (文字列) 新規stroークのID

【0095】

・sc:

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

stroークの継続を放送するために使用される。

## プロパティ

x,y -- (整数) ストロークの新しい終点  
strokeId -- (文字列) 新たなstroークのID

10

【0096】

・se:

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

stroークIDにより特定されるstroークを終了する。

stroke-id -- (文字列) 継続したstroークのID

【0097】

・Delete stroke:

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

20

stroークを削除する。

stroke-id -- (文字列) stroークのID

target-id -- (文字列) stroークターゲットのID

【0098】

・Undo:

直近の取消可能な動作(移動、テキスト設定、stroーク等)を取り消す。

```
// server <-- client
```

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

```
[client-id, 'undo', target-
```

30

取消例: ウィンドウを移動し、その後その移動を取り消す。

下記の例は移動と、如何にその移動が取り消されるかを示す。

//クライアントは移動を送信し、その後取消メッセージを送信する。

サーバはセッション履歴から当該移動の履歴イベントを取り除き、この記録はもはやセッションの空間的イベントマップの一部ではなく、履歴の時系列から取り除かれたことをクライアントに通知する。HTTP APIを介したその後の履歴要求は、(再びやり直される後まで)取消イベントを含まない。

【0099】

・ディスプレイ配列の大きさ:

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

40

ディスプレイ配列の全体の幅及び高さの変更をクライアントに通知する。これは、クライアント側ネットワークノードが空間的イベントマップの部分の局所的ディスプレイを管理するためのリソースを有している場合には、適用されないことがある。

arrayID -- (文字列) ディスプレイ配列のID

width, height -- (整数) ディスプレイ配列の画素単位での幅及び高さ

【0100】

・パン配列:

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

新たな位置に対するパンをサーバに通知する。

50

newArrayOffsetX, newArrayOffsetY -- ( 整数 ) パン後のディスプレイ配列の新たな位置

【 0 1 0 1 】

・セッション変更 :

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

セッションが変更されたことをディスプレイ配列の兄弟に通知する。

sessionId -- ( 文字列 ) 切り替え後のセッションの ID

【 0 1 0 2 】

・ズーム変更 :

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

ズームが要求されたことをサーバに通知する。

zoomLevel ( 整数 ) 遷移後の拡大レベル、1 から 11 まで

zoomCenterX ( 整数 ) ズームの起点の x 座標

zoomCenterY ( 整数 ) ズームの規定の y 座標

【 0 1 0 3 】

・マップモード変更 :

```
// client --> server
```

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

マップモードが要求されたことをサーバに通知する。一見すると、これはズーム変更メッセージと殆ど同一の動作を行うが、グラフィック処理において数ダース又は数百のズーム変更メッセージがそれら同士でさえずりながら矢継ぎ早に送信されるのに対し、マップモードメッセージは異なる遷移効果を有した単一ズームスナップに用いられる。

zoomLevel -- ( 整数 ) 遷移後の拡大レベル、1 から 11 まで

zoomCenterX -- ( 整数 ) ズームの起点の x 座標

zoomCenterY -- ( 整数 ) ズームの起点の y 座標

【 0 1 0 4 】

・カード生成

```
// server <-- client
```

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

templateId -- ( 文字列 ) 使用されるカードテンプレートの ID

regionId -- ( 文字列 ) 元となるイベントのキャンパス領域 ID ( あれば )

x, y, x2, y2 -- ( 整数 ) 新カードの所望の rect 値

【 0 1 0 5 】

・ユーザ許可

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

認証されたユーザが許可を有しているかを示す許可タイプのハッシュ及び真 / 偽を許可する。現在のところ、唯一の許可はセッションを他人と共有することのできるユーザを示す「can\_share」である。

【 0 1 0 6 】

・位置の保存 :

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

現在のスクリーン位置を保存する。再接続の際に、クライアントはこの位置に戻って送信された「zc」(ズーム変更)及び「pa」(パン配列)メッセージを受信する。

zoomLevel ( 整数 ) 現在動作中のデバイスの拡大レベル

x ( 整数 ) スクリーンの起点の x 座標

y ( 整数 ) スクリーンの起点の y 座標

【 0 1 0 7 】

・ストローク ID

ストローク ID はクライアントによって選択される。現在のところ、それは送信者 ID に、ドットにより分離される増加する整数値を組み合わせたものである。これにより、ス

10

20

30

40

50

トロークIDはサーバ内において全てのクライアント間で固有とする。

#### 【0108】

##### ・ターゲットID

ストロークは、主キャンパス上を浮遊するサブキャンパス（「カード」と呼ばれる）のような、セッション内の特定のターゲットに貼り付けられることがある。カードに属するストロークの場合、ターゲットIDフィールドはカードIDを含む。セッション内の主キャンパスに向かうストロークは、そのターゲットIDがセッション名と同じとなるように指定される。

#### 【0109】

##### ・接続の確立

クライアントが新たなウェブソケット接続をサーバと確立すると、サーバは先ず固有のクライアントIDを選択し、「id」メッセージにおいてそれをクライアントに送信する。その後、サーバは送信者IDが-1に設定された「pa」メッセージを送信する。

この後の代表的なフローはクライアントがHTTP GET “/:sessionId/objects”（下記に記載される）を実行し、セッション内のカードについての情報を受信することである。その後、クライアントは“/:sessionId/history”（同様に、下記に記載される）を要求し、履歴urlの配列を受信する。これらはキャッシュ可能性を向上させるためバッチに分けられる。クライアントはローカルキャッシュに保存されていない夫々のurlを取得し、サーバはこれらの履歴バッチに対しストロークデータで応答する。

#### 【0110】

##### ・サービス要求

##### ・履歴：

履歴ブックマークのリストを取得する。夫々のブックマークはキャッシュされたストローク履歴の大きさである。

```
curl http://localhost:4545/<sessionId>/history
      sessionId 履歴を取得しようとするセッションの名前
```

##### ・応答ヘッダ

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

##### ・応答

```
["/<sessionId>/history/<startTime>/<endTime>?b=1"]
["/<sessionId>/history/<startTime>/<endTime>?
  sessionId -- (文字列) 切り替え後のセッションのID
  startTime -- (整数) 時刻印の開始
  endTime -- (整数) 時刻印の終了
  b -- キャッシュ破棄
```

##### ・履歴ブロックの検索

開始時刻と終了時刻の間の履歴を取得する。要求は履歴の戻りの範囲の夫々についてなされる必要がある。

```
curl http://localhost:4545/<sessionId>/history/<startTime>/<endTime>?b=<cache-buster>
  sessionId --履歴を取得しようとするセッションのID
  startTime --最初の履歴要求で与えられた開始時刻
  endTime --最初の履歴要求で与えられた終了時刻
  cacheBuster --クライアントが保存するキャッシュがもはや有効でないときに変更される単純なキー
```

10

20

30

40

50

- ・ 応答ヘッダ

```
HTTP/1.1 200 OKX-Powered-By: ExpressAccess-Control-Allow-Origin: *Access-Control-Allow-Headers: X-Requested-With Content-Type: application/json
Content-Length: 2134
ETag: 1346968307576
Date: Fri, 14 Sep 2012 17:35:14 GMT
Connection: keep-alive
```

- ・ 応答

```
[
  [
    4,
    "sx",
    "4.4",
    [537, 650, 536, 649, 536, 648, ...],
    {
      "size": 10,
      "color": [0, 0, 0, 1],
      "brush": 1
    },
    1347644106241,
    "cardFling"
  ]
]
```

( s x 「ストローク完了」ウェブソケットメッセージの記載を参照 )

【 0 1 1 1 】

- ・ オブジェクト検索

要求されたセッションのオブジェクト ( カード / 画像 ) を取得する。

```
curl http://localhost:4545/<sessionId>/objects
sessionId 履歴を取得しようとするセッションの I D
```

- ・ 応答ヘッダ

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Content-Type: application/json; charset=utf-8
Content-Length: 285
Date: Fri, 14 Sep 2012 17:35:14 GMT
Connection: keep-alive
```

- ・ 応答

```
{
  "eventType": "oc",
  "id": "50536840ce64b39439000005",
  "baseName": "sessions/all/green",
  "ext": "JPEG",
  "rect": [-239, 49, 361, 649],
  "arrayId": 3,
```

```

    "clientId": 3,
    "regionId": null,
    "sessionId": "cardFling",
    "actualWidth": 600,
    "actualHeight": 600,
    "order": null,
    "_id": "50536840ce64b39439000005",
    "type": "image"
  },
  {
    "eventType": "oc",
    "id": "50536b66ce64b39439000006",
    "baseName": "sessions/all/orange",
    "ext": "JPEG",
    "rect": [-97, 190, 503, 790],
    "arrayId": 5,
    "clientId": 5,
    "regionId": null,
    "sessionId": "cardFling",
    "actualWidth": 600,
    "actualHeight": 600,
    "order": null,
    "_id": "50536b66ce64b39439000006",
    "type": "image"
  }
]

```

10

20

### 【 0 1 1 2 】

#### ・カードテンプレート：

キャッシュされた再利用可能なカードを生成するための大域的カードテンプレートのリストを取得する。これは、このテンプレートで生成された全てのカードについては同じ背景画像が用いられる点で、ファイルのアップロードとは異なる。

30

```
curl http://localhost:4545/card_templates.json
```

#### ・応答

```

[
  {
    "id": "50901cb0b9a18c190902a938",
    "width": 600,
    "thumbnail": "card_templates/thumbnails/pink.jpeg"
  },
  {
    "id": "50901cb0b9a18c190902a939",
    "width": 600,
    "thumbnail": "card_templates/thumbnails/green.jpeg"
  }
]

```

40

これらの値はカード生成メッセージを送信するために用いられる：

```
// 上記のピンクテンプレートを用いて新規カードを生成する
["cc", "50901cb0b9a18c190902a938", <regionIdOrNull>, <x>, <y>]
```

50

## 【 0 1 1 3 】

・アップロード：

セッション内に配置されるべき画像をサーバに送信する。

```
curl -F "file=@photo.JPG" -F "x=236" -F "y=832" -F "clientId=10" -F "sessionId=cardFling"
-F "arrayId=10" -F "order=23" -F "x2=899" -F "y2=1495" -F "filename=photo.jpg"
http://localhost:4545/<sessionId>/object/upload
```

パラメータ

x: ドロップの x 位置  
y: ドロップの y 位置  
clientId: クライアント I D  
sessionId: セッション I D  
arrayId: 配列識別子  
order: z 方向の順位  
x2: ドロップの右下隅の x 位置  
y2: ドロップの右下隅の y 位置  
filename: アップロードされるファイル名

10

## 【 0 1 1 4 】

上記の A P I はメッセージ構造の一例を提供する。特定の実装に合うように、他の構造も同様に利用され得る。

20

## 【 0 1 1 5 】

共同システムの一実施形態において、図 6 から示唆されるように、アプリケーションプログラムインタフェース A P I は共同サーバ 1 0 5 及び表示クライアントによって、表示クライアントの夫々につき 2 つの通信チャンネルに基づいて実行される。

## 【 0 1 1 6 】

いくつかの実施形態では、連合ディスプレイが採用される。連合ディスプレイにおいて、表示クライアント 7 1 1 ~ 7 1 4 の夫々は独立にサーバとの 2 つのチャンネルを維持し得る。第 1 のチャンネルはライブイベントの通信用に構成されたメッセージベースのシステムである。一例では、この第 1 チャンネルは、共同サービス 6 0 1 とのウェブソケットチャンネル上の 1 組のソケット要求を用いて実装され、接続後に関連するデータ（新たなストローク、カード、クライアント等）でクライアントを更新するに際してサーバにより使用される。メッセージベースのチャンネルはまた、初期接続のハンドシェイクを処理することができる。第 2 のチャンネルは、H T T P / R E S T インタフェースのような、ポータル 6 0 2 との状態把握のより少ない接続であり、データ（即ち、イメージ及びカード）の投稿のほか、キャッシュ応答に利用され得る。また、表示クライアントへの作業空間データのロードは、キャッシングに対応するため、メッセージベースのチャンネル（ウェブソケット）よりはむしろ H T T P 要求を利用してなされ得る。

30

## 【 0 1 1 7 】

図 8 は、分散型共同システムにおいて、クライアント側の機能（例えば、コンピュータシステム 1 1 0 ）又はサーバ側の機能（例えば、サーバ 1 0 5 ）を実現するために使用され得るコンピュータシステム又はネットワークノードの簡略化されたブロック図である。コンピュータシステムは、典型的に、多数の周辺機器とバスサブシステム 1 0 1 2 を介して通信するプロセッササブシステム 1 0 1 4 を含む。周辺機器は、メモリサブシステム 1 0 2 6 及びファイルストレージサブシステム 1 0 2 8 を備えたストレージサブシステム 1 0 2 4、ユーザインタフェース入力デバイス 1 0 2 2、ユーザインタフェース出力デバイス 1 0 2 0、及び、ネットワークインタフェースサブシステム 1 0 1 6 を含み得る。入力及び出力デバイスはコンピュータシステムとのユーザ対話を可能にする。通信モジュール 1 0 1 6 は、通信ネットワーク 1 0 4 とのインタフェースを含んで、外部ネットワークと

40

50

のインタフェースとして物理的な通信プロトコルサポートを提供し、他のコンピュータシステムの対応する通信モジュールと通信ネットワーク104を介して接続している。通信ネットワーク104は、多数の相互接続されたコンピュータシステム及び通信接続を含み得る。これら通信接続は、有線接続、光接続、無線接続、或いは他の任意の情報通信のための機構であり得るが、典型的には、少なくともその肢部において、IPベースの通信ネットワークである。一実施形態においては、通信ネットワーク104はインターネットであり、他の実施形態では、通信ネットワーク104は任意の適したコンピュータネットワークであり得る。

**【0118】**

ネットワークインタフェースの物理ハードウェア要素は、時々ネットワークインタフェースカード(NIC)と呼ばれるが、それらは必ずしもカードの形をしている必要はなく、例えばマザーボード上に直接取り付けられた集積回路(IC)とコネクタの形式、或いはコンピュータシステムの他の要素と一緒に単一の集積回路上に実装されるマクロセルの形式であり得る。

10

**【0119】**

ユーザインタフェース入力デバイス1022はキーボード、マウス、トラックボール、タッチパッド、又はグラフィックタブレット、スキャナ、ディスプレイに組み込まれたタッチスクリーン(大規模デジタルディスプレイ102cのタッチセンサ部分を含む)のような指示デバイス、音声認識システム、マイクのような音声入力デバイス、及び他の形の触知できる入力デバイスを含み得る。一般に、用語「入力デバイス」の使用は、コンピュータシステム内に、又はコンピュータネットワーク104上に情報を入力するための全ての可能な型式のデバイス及び方法を含むことが意図される。

20

**【0120】**

ユーザインタフェース出力デバイス1020はディスプレイサブシステム、プリンタ、ファックス機、或いは音声出力デバイスのような視覚以外のディスプレイを含み得る。ディスプレイサブシステムは陰極線管(CRT)、液晶ディスプレイ(LCD)のようなフラットパネルデバイス、投影デバイス、或いは他の可視画像を生成する機構を含み得る。図1Bの実施形態において、それは大規模デジタルディスプレイ102cの表示機能を含む。ディスプレイサブシステムは、音声出力デバイスを介するもののような、視覚以外のディスプレイを提供し得る。一般に、用語「出力デバイス」の使用は、コンピュータシステムからユーザ又は他の機械又はコンピュータシステムに情報を出力するための全ての可能な型式のデバイス及び方法を含むことが意図される。

30

**【0121】**

ストレージサブシステム1024は、本発明の特定の実施形態の機能性を提供する基本プログラム及びデータ構造体を保存する。

**【0122】**

ストレージサブシステム1024は、サーバ側ネットワークノードの実装に使用される場合、イベントを作業空間内に配置する空間的イベントマップを含む機械読み出しが可能なデータ構造を記憶し、コンピュータ読み出し可能な持続性媒体を含んだ製品を備える。当該空間的イベントマップは、イベントのログ、イベントに係る図形ターゲットの作業空間内の位置及び時刻を有する当該ログのエントリを含む。また、ストレージサブシステム1024は、サーバ側ネットワークノードに関連して本明細書中に記載された手続を実行するための実行可能な命令を含んだ製品を備える。

40

**【0123】**

ストレージサブシステム1024は、クライアント側ネットワークノードの実装に使用される場合、イベントを作業空間内に配置する空間的イベントマップを含む機械読み出しが可能なデータ構造を、以下に説明されるキャッシュコピーの形で記憶し、コンピュータ読み出し可能な持続性媒体を含んだ製品を備える。当該空間的イベントマップは、イベントのログ、イベントに係る図形ターゲットの作業空間内の位置及び時刻を有する当該ログのエントリを含む。また、ストレージサブシステム1024は、クライアント側ネットワ

50

ークノードに関連して本明細書中に記載された手続を実行するための実行可能な命令を含んだ製品を備える。

【0124】

例えば、本発明の特定の実施形態の機能性を実現する種々のモジュールが、ストレージサブシステム1024内に保存され得る。これらのソフトウェアモジュールは一般にプロセッササブシステム1014によって実行される。

【0125】

メモリサブシステム1026は、典型的には、プログラム実行の間命令及びデータを保持するための主ランダムアクセスメモリ(RAM)1030、及び、固定の命令が保存された読み出し専用メモリ(ROM)を含む多数のメモリを備える。ファイルストレージサブシステム1028はプログラム及びデータファイルの永久的な保管を提供し、ハードディスクドライブ、フロッピーディスクドライブと関連する着脱可能媒体、CDROMドライブ、光学ドライブ、又は着脱可能媒体カートリッジを含み得る。本発明の特定の実施形態の機能性を実現するデータベース及びモジュールは、1以上のCDROMのようなコンピュータ読み出し可能な媒体で提供されており、ファイルストレージサブシステム1028によって保存され得る。ホストメモリ1026は、とりわけ、プロセッササブシステム1014により実行されると、コンピュータシステムに本明細書に記載の機能を動作又は実行させるためのコンピュータ命令を含む。本明細書中において、「ホスト」又は「コンピュータ」内又は上で稼働するとしてプロセス及びソフトウェアは、命令及びデータのための他の任意のローカルの又はリモートのストレージを含んだホストメモリサブシステム1026内のコンピュータ命令及びデータにตอบสนองして、プロセッササブシステム上で実行される。

【0126】

バスサブシステム1012は、コンピュータシステムの種々の要素を互いに意図したとおりに通信させるための機構を提供する。バスサブシステム1012は模式的に単一バスとして示されているが、バスサブシステムの代替実施形態は複数バスを使用してもよい。

【0127】

コンピュータシステムそれ自身は、パーソナルコンピュータ、ポータブルコンピュータ、ワークステーション、コンピュータ端末、ネットワークコンピュータ、テレビ、メインフレーム、サーバファーム、又は他のデータ処理システム或いはユーザデバイスを含み、様々な種類があり得る。一実施形態において、コンピュータシステムは、夫々が大規模ディスプレイ102を構成するタイルの1つを制御する複数個のコンピュータシステムを含む。コンピュータ及びネットワークの絶えず変化する性質によって、図8に示されたコンピュータシステムの記載は本発明の好ましい実施形態を図示するための具体的な一例を意図したに過ぎない。図8に示したコンピュータシステムよりも要素の多いまたは少ないコンピュータシステムの他の構成が多数可能である。同一の要素及び変形は、共同サーバ105と表示データベース106のほか、図1の共同環境における他のデバイス102の夫々を構成することもできる。

【0128】

デジタルディスプレイ102c上において有効な描画領域についての特定の情報は表示クライアントのコンピュータシステム110にアクセス可能なデータベースに保存される。データベースは異なる環境における多数の形式をとることができ、MongoDBデータベース、XMLデータベース、リレーショナルデータベース、又はオブジェクト指向データベースを含むが、これらに限られるものではない。図9は、データベースに含まれる特定の情報、及びデータ間の関連性を図説する模式図である。

【0129】

本明細書に記載する実施形態において、描画領域の夫々はツールバーの子であると考えられる。壁の背景の一点へのタッチがツールバーを生成し、(ツールバーは描画領域が開くまで必ずしも視覚できる必要はないが)その後ツールバーが描画領域を生成する。同様に、描画領域を閉じるため、ユーザは描画領域のツールバーの「閉じる」アイコンにタッ

10

20

30

40

50

チする。したがって、図9において、データベースは1以上のツールバーID1110によって先導される。ツールバーIDの夫々は、描画領域の幅、及び当該描画領域における描画特性の組とともに、ツールバーの水平方向の位置、ツールバーの描画領域の左端の水平方向の位置を示すデータの夫々のブロック1112を含み、又は指し示している。描画領域の左よりも右端の位置を指定することや、描画領域の幅の代わりに反対側の端の位置を指定することのような多数の変形が可能である。一実施形態において、ツールバーの位置は同じ垂直方向の位置を維持するので、水平方向の値しか有していないが、別の実施形態では水平方向と垂直方向の位置の両方が指定され得る。

#### 【0130】

描画特性は、夫々が1以上の値と関連付けられた、描画属性の配列1114を含み又は指し示している。図9の描画特性は、ブラシ種類、例えば「ペイント」、「インク」、「クレヨン」、「マーカ」或いは「消しゴム」を示し得る値を含み、夫々、ディスプレイ102c上に描画されるときに異なる外観特性を有する。図9の描画特性はまた、実現可能な値の範囲内で任意の値を取り得るブラシ幅を含む。図9の描画特性はまた、ブラシ色を含み、赤、緑、及び青に関連付けられた3つの値を有する。本明細書中において、ブラシ種類、ブラシ幅、及びブラシ色の3属性が「線外観特性」を構成すると考えられる。描画属性1114は、種々の実施形態において、線の配置や線の一部の配置に影響を与える他の属性を含み得る。これらの特性は、角を丸める半径や、ベジェ曲線のパラメータのような属性を含み得る。図11から分かる通り、異なる描画領域における描画属性（線外観特性を含む）を同じとする要請はない。それらは互いに独立に確立されるものであるので、それらを同一とする必要はない。典型的な場合、それらは同じではない。

#### 【0131】

ディスプレイ102c上に線を描くため、ユーザは線の描画を示す「描画ユーザ入力」を提供する。他の実施形態ではユーザが指で描くのを可能とし得るが、図1の実施形態では、スタイラスのみが線の描画を示すために使用され得る。直観的に、ユーザはスタイラスをディスプレイ102c表面に接触させ、描画領域内で、スタイラスを線のための所望の位置に沿ってドラッグすることによって、そのように指示する。線描画動作の終了はスタイラスをディスプレイ102c表面から離すことによって示される。ローカルのコンピュータシステム110は、どこに線上の点を位置決めすべきかを、ユーザ入力から決定し、それをディスプレイ102c上に表示する。コンピュータシステム110はまた、ストローク情報を共同サーバ105（図1B）に送信し、共同サーバ105はその情報を表示データベース106に書き込み、セッションを共有中の様々なデバイス102に送信し返す。それから、夫々のデバイス102cは線を（当該線が各デバイスの表示域と重なる限り）表示し、全てのデバイス102が線をおおよそ同時に示す。

#### 【0132】

図10～14はサーバ、表示クライアント、又はその両方によって実行される論理（ロジック）を図示するフローチャートである。当該論理はコンピュータシステムにアクセス可能なメモリに記憶され、プロセッサにより実行可能なコンピュータプログラムを用いてプログラムされるプロセッサを用いて、フィールドプログラマブル集積回路を含む専用のロジックハードウェア、及び、専用ロジックハードウェアとコンピュータプログラムの組み合わせにより実装される。明細書中の全てのフローチャートについて、多数のステップが組み合わせられ、並列に実行され、又は実現される機能に影響を与えることなく異なる順序で実行されることができると理解される。いくつかの場合には、特定の他の変更がされている場合であっても、ステップの再配置により同じ結果が得られることを読者は理解できる。他の場合には、特定の条件が満足される場合に限り、ステップの再配置により同じ結果が得られることを読者は理解できる。さらに、本明細書中のフローチャートは本発明の理解に関係するステップだけを示しており、他の機能を実現するための多数の追加ステップが、示されたステップの前後及びそれらの間に実行されることが理解される。

#### 【0133】

図10は永続的な作業空間の部分としてのセッションにユーザが参加するときサーバ

10

20

30

40

50

側で実行される基本ロジックを示している。フローチャートはユーザのログイン（1210）で始まり、ユーザはパーソナルコンピュータ、タッチパッド、スマートフォン等のユーザにより所有されているデバイスを通して、ウェブポータルアクセスにユーザ識別子を入力し得る。次に、ユーザ認証プロトコルが実行される（1212）。プロトコルは、たとえば、ユーザが実際にユーザ識別子を入力した人物であるかを検証するため、ユーザに個人パスワードの入力を要求することを含み得る。次に、例えばポータルマシンを使用して、共同サーバが、認証されたユーザが参加を承認された作業空間へのリンクを提示し得る（1214）。次に、共同サーバはユーザに対し選択される表示クライアント及び選択される作業空間を決定し得る（1216）。この決定はユーザが所有するマシンと認証プロトコルが実行される通信チャネルを利用するポータルとの間のメッセージ交換によってなされ得る。表示クライアントおよび作業空間が識別されると、共同サーバは、当該表示クライアントに選択された作業空間に対するデータのダウンロードを許可し得る（1218）。また、共同サーバは選択された作業空間のライブイベントチャンネルにクライアントを追加し得る（1220）。

10

#### 【0134】

図11はユーザが作業空間に参加したときにクライアント側で実行される基本の2チャンネルロジックを示している。フローチャートは第1のチャンネル上で実行されたユーザによるログイン（1230）で始まり、ユーザはユーザ識別子を入力し、それをウェブポータルに送信し得る。次に、ユーザ認証プロトコルが実行される（1232）。その後、ユーザは認証プロトコルが実行される通信チャネルを利用してポータルのページを開く（1234）。当該ページには、承認された作業空間へのリンクが表示されている。次に、ユーザは今回のセッションで利用する選択される作業空間及び表示クライアントを識別するデータを、第1のチャンネルを利用して入力する（1236）。サーバが選択された表示クライアントを有効にした後、ユーザの活動は表示クライアントとサーバ間のチャンネルに移り、選択されたセッションに対する作業空間データをダウンロードし得る（1238）。表示クライアントは作業空間データをトラバース（traverse）し、表示クライアントによって管理される表示領域のための画像を構成し得る（1240）。また、表示クライアントはライブイベントチャンネルに参加し得る（1242）。クライアント側ネットワークノードとサーバ側ネットワークノードは、セッションを確立している間、空間的イベントマップの暗号化及び復号化のプロトコルを確立し得る。

20

30

#### 【0135】

一例において、作業空間データをダウンロードする処理はセッションに係るイベントオブジェクトを夫々の表示クライアントに配送することを含む。作業空間データとともに、現在のユーザ位置が提供され得る。代わりに、作業空間データの中心位置のような、表示クライアントに初期位置からのオフセットを如何にして計算するかを識別する一連のメッセージの後、ユーザに関連する現在位置に作業空間データが配送され得る。それから、夫々の表示クライアントはイベントオブジェクトをトラバースし、当該表示クライアントによって管理される表示領域へマップするセッション位置を有するオブジェクトを識別する。イベントオブジェクトをトラバースするロジックは、例えば、作業空間内で表示領域にマップするオブジェクトを探し出すように構成されたR木による探索を含むことができる。識別されたオブジェクトは、その後、ディスプレイにより管理される表示領域上に描画されるが、オブジェクトに関連するデータを取得するためにポータルと通信することがある。

40

#### 【0136】

図12は作業空間データのダウンロードに関するクライアント側の基本ロジックを示している。ロジックは共同サーバから作業空間データのダウンロードで始まる（1250）。表示クライアントは、当該表示クライアントの範囲内で、ユーザフォーカスの周囲にあるオブジェクトを描画する（1252）。ユーザフォーカスは、サーバにより提供される作業空間内の位置から決定されるか、又はクライアント側ネットワークノードに維持されている。表示クライアントは、作業空間データ内に現在の位置を有し、ユーザフォーカス

50

に近い位置にあるオブジェクトを含む作業空間データ又は作業空間データの少なくとも一部分を保持する。セッションの間、ユーザ入力または他のデータに応答して、表示クライアントは作業空間内の複数位置をトラバースし、作業空間内の現在位置を決定する(1254)。表示クライアントはトラバースされた位置周囲の範囲内にあるオブジェクトを描画する(1256)。セッションの終了時に、表示クライアントによってマップされた作業空間内の最後の位置が共同サーバ上のユーザ位置として保存される(1258)。

#### 【0137】

図13は、多数のユーザ同士でアクセスが共有されるディスプレイへのアクセスを管理するためのサーバ側で実行されるロジックを示している。この例では、サーバは当該サーバと有効なリンクを有しているフリーのディスプレイ壁のリストを維持している。これらのリンクはディスプレイ壁が電源オンされると設定され、ディスプレイ壁がアイドル状態である待ち時間の間維持される。サーバはフリーのディスプレイ壁がサーバとのリンクを有しているか否かを判断するロジックを含むことができる(1260)。壁がサーバとのリンクを有していると検出されると、一回限りの識別コード又はPINがそれに割り当てられる。識別コードが割り当てられると、当該壁は共同システムで利用可能な壁(フリー壁)のリストを含む「ロビー」に追加される(1264)。サーバはまた、承認されたユーザログインを待つループを実行する(1266)。ログインが検出されると、サーバはユーザが承認されるべき作業空間及び壁の選択を促す(1268)。その後、サーバはユーザに選択された壁に関連付けられた一回限りの識別コードの入力を要求する(1270)。当該選択された壁と一致する識別コードをサーバが受信しない場合(1272)、エラー信号が発行される(1273)。当該選択された壁と一致する識別コードをサーバが受信した場合(1272)、選択された壁に関連付けられた(1又は複数の)表示クライアントが、壁が占有されている間、セッションのライブイベントチャネルと接続され、一回限りの識別コードは無効に又は変更される(1274)。また、サーバは選択された(1又は複数の)表示クライアントに作業空間データを送信する(1276)。作業空間データの受信後、ユーザはセッションを介して共同することが可能となる(1278)。ユーザがセッションからログオフすると、ディスプレイ壁は解放される(1280)。ディスプレイ壁が利用可能状態を維持していれば、それはフリーディスプレイ壁であることがサーバに示され、ステップ1260、1262、1264に従い、新たな識別コードを付してロビーに追加される。いくつかの実施形態では、使用されていない壁から識別コードを盗んだ侵入者によるログインに対するセキュリティを提供するため、タイムアウト期間の満了により識別コードは変更される。

#### 【0138】

図14は連合ディスプレイ配列を管理するためにサーバ側で実行される基本ロジックを示している。このフローチャートの第1ステップは作業空間データを配列内の夫々の表示クライアントにダウンロードすることを含む(1302)。夫々の表示クライアントは、当該表示クライアントの範囲内で、ユーザ位置からクライアントオフセットの周囲にあるオブジェクトを描画する(1304)。サーバはクライアントイベントメッセージを監視する(1306)。配列内の表示クライアントの1つからクライアントイベントメッセージをサーバが受信すると、サーバはそのメッセージが作業空間データに関連するものか、又は配列のみに関連するものかを判断する(1308)。配列メッセージは、連合ディスプレイ配列に参加している表示クライアントだけがメッセージを受け取るように、アレイチャネル上で放送される(1310)。作業空間データメッセージは、作業空間データに係るセッションに参加している全ての表示クライアントが必要に応じて更新されるように、共同チャネル上で放送される(1312)。連合ディスプレイ配列のみに関係するこれらのメッセージは、上述したツールバーの位置や描画領域を更新するメッセージを含み得る。また、作業空間内のオブジェクトの位置を変更せず、作業空間の一部のオブジェクトを生成又は修正しないメッセージは、局所的な配列のみに関連するメッセージと判断され得る。

#### 【0139】

図15は、図1Bの様式において、ディスプレイが広範囲に分散した分散型ディスプレイ共同を支援するシステムを図説している。システムは、共同サーバ105と、作業空間データを記憶する関連表示データベース106を備える。共同サーバは通信リンク104によって、例えばシカゴ、ロサンゼルス、サンパウロに設けられ得る複数の壁1502a、1502b、1502cと接続されている。共同サーバ105はまた、既知ユーザの所有物であることが期待され得るタッチ패드又は他のパーソナルコンピューティングプラットフォームのようなユーザデバイス1504と接続している。図13との関係で上述したように、共同サーバ105はフリーなディスプレイ壁のリストを、「ロビー」109と呼ぶデータ構造内で維持し得る。シカゴに設けられたディスプレイ壁に関連付けられたOT-PIN#1、ロサンゼルスに設けられたディスプレイ壁に関連付けられたOT-PIN#2、サンパウロに設けられたディスプレイ壁に関連付けられたOT-PIN#3を含む、1回限りの識別コードが、ディスプレイ壁の夫々と関連付けられている。パーソナルデバイス1504を所有するユーザは、ユーザ認証のためにユーザID及びユーザパスワードを入力することで、共同サーバ105によって管理されるポータルにログインし得る。その後、パーソナルデバイス1504を所有するユーザは、作業空間識別子、及び、ユーザが作業空間データを表示させたいディスプレイ壁の識別コードを提供し得る。共同サーバがユーザの認証に成功し、ユーザ承認されるべきディスプレイ壁及びユーザ承認されるべき作業空間をユーザが識別したと判断すると、識別されたディスプレイに関連付けられた表示クライアントが共同イベントチャンネルと接続され、作業空間データのダウンロードが許可される。所定のセッションに対して表示デバイスが許可されると、それはロビー109から取り除かれ、1回限りの識別コードは削除または変更される。ディスプレイがロビーに追加される度に、新たな識別コードが、作業空間データに貢献することのできるユーザ入力を受け入れる為に計算され得る。システムは、作業空間データについて承認されたユーザが選択されたディスプレイへの物理アクセスを有しているのを保証するプロトコルに基づき、選択されたディスプレイに作業空間データを提供する管理ロジックを含み得る。

#### 【0140】

図16は、クライアントプロセッサ1600、ディスプレイドライバ1601、タッチスクリーン1602のようなローカルディスプレイ及びユーザインタフェース、スタックにより制御される通信インタフェースを含むプロトコルスタック1604、ライブ空間的イベントマップのキャッシュコピー及び表示可能領域の描画に用いられる画像や他の図形構成物のキャッシュを保存するローカルメモリ1605、及び、タッチスクリーンやマウス等の実際のユーザ入力デバイスからの入力をコマンドインタプリタ1606によって利用可能な形式に翻訳する入力プロトコルを実行する入力プロトコルデバイス1607を備えたクライアント側ネットワークノードの簡略化された図である。適当な入力プロトコルデバイス1607は、例えばディスプレイ壁との触覚によるマルチタッチ対話の解釈のために、TUIO工業標準と互換性のあるソフトウェアを含み得る。プロトコルスタック1604はクライアントプロセッサ1600からAPIに準拠したメッセージ及びインターネットメッセージを受信し、上述の通り、APIに準拠したメッセージが交換され得る共同サーバへのチャンネル1611を確立するためのリソース、及び、ローカルディスプレイ1602に提供する他の通信を支援するため、インターネットへの接続1610を確立するためのリソースを含む。ディスプレイドライバ1601はローカルディスプレイ1602上の表示可能領域1603を制御する。表示可能領域1603はクライアントプロセッサ又はクライアント側ネットワークノード内の他のプログラミングリソースにより論理構成される。また、表示可能領域1603の物理サイズは、ローカルディスプレイの所定の実装のために固定にし得る。クライアントプロセッサ1600は、ブラウザ、表示可能領域1603上と作業空間上の間で位置を変換するためのマッピングロジック、及び、API手続を実行するためのロジック等のプロセッシングリソースを含み得る。

#### 【0141】

本明細書に記載の共同のためのシステムは図16に類似のクライアント側ネットワーク

10

20

30

40

50

ノードを含み得る。クライアント側ネットワークノードは物理的表示空間を有したディスプレイ、ユーザ入力デバイス、プロセッサ及び通信ポートを備える。クライアント側ネットワークノードは下記のロジックにより構築される：

サーバ側ネットワークノードとの接続を確立し、

作業空間内に位置を有する図形ターゲットに関するイベントに係る空間的イベントログの少なくとも一部、イベントに係る当該図形ターゲットの作業空間内における位置を含むログのエントリ、イベントの時刻、及び、当該図形ターゲットのターゲット識別子をサーバ側ネットワークノードから検索し、

物理的表示空間内の表示可能領域を作業空間内の被マップ領域にマップし、当該被マップ領域内にある空間的イベントログのエントリを識別し、識別されたエントリにより特定される図形ターゲットを表示可能領域上に描画し、

表示可能領域内に表示される図形ターゲットの修正及び生成に関するイベントを生成するユーザ入力デバイスからのユーザ入力を受け入れ、当該イベントに基づいてサーバ側ネットワークノードへメッセージを送信する。

【 0 1 4 2 】

図 1 6 に示されたクライアント側ネットワークノードは、サーバ側ネットワークノードと通信する処理を含むアプリケーションインタフェースを備えた一例を図示するものである。

【 0 1 4 3 】

図 1 6 に示されたクライアント側ネットワークノードは、API に従って構築される一例であり、イベントが、他のクライアント側ネットワークノード間に配信され、サーバ側ネットワークノード内の空間的イベントログに追加される履歴イベントに指定された第 1 種のイベントと、他のクライアント側ネットワークノード間で配信されるが、サーバ側ネットワークノード内の空間的イベントログに追加されない一時的イベントに指定された第 2 種のイベントを含む。

【 0 1 4 4 】

図 1 7 はクライアント側ネットワークノードにより実行される手順の簡略化されたフロー図である。この簡略化されたフロー図に示された順序は、説明のために提示されたものであり、特定の実装に適合するように修正されることがある。多くのステップは、例えば、並列に実行され得る。この手順では、クライアントのログインが実行され ( 1 7 0 0 )、これによりクライアントは特定の共同セッション及びその空間的イベントマップへのアクセスを得る。共同サーバは空間的イベントマップの識別子、或いは空間的イベントマップの一部の識別子を提供し、当該識別子が、クライアントによって、共同サーバから空間的イベントマップを検索するために使用され得る ( 1 7 0 1 )。クライアントは、提供された ( 複数の ) 識別子を用いて、共同サーバから空間的イベントマップを検索、或いはその少なくとも一部を検索する ( 1 7 0 2 )。

【 0 1 4 5 】

例えば、下記のように、クライアントはアクセス権が与えられた所定の作業空間の全ての履歴を要求することができる：

```
curl http://localhost:4545/<sessionId>/history
```

【 0 1 4 6 】

サーバは ( 夫々が時間で区分された ) 全てのチャンクで応答する：

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

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

【 0 1 4 7 】

チャンク毎に、クライアントはイベントを要求する：

```
Curl http://localhost:4545/<sessionId>/history/<startTime>/
```

```
<endTime>?b=<cache-buster>
```

【 0 1 4 8 】

応答されるチャンクの夫々はイベントの配列であり、クライアントでキャッシュ可能で

10

20

30

40

50

ある：

```
[
  [
    4,
    "sx",
    "4.4",
    [537, 650, 536, 649, 536, 648, ...],
    {
      "size": 10,
      "color": [0, 0, 0, 1],
      "brush": 1
    },
    1347644106241,
    "cardFling"
  ]
]
```

10

#### 【 0 1 4 9 】

個々のメッセージは、スクリーン上の位置、色、ストロークの幅、生成時刻等と同様の情報を含み得る。

#### 【 0 1 5 0 】

20

その後、クライアントは、例えばサーバから提供されたフォーカス点を用いて、作業空間内の位置、及びローカルディスプレイに対する表示境界を設定する（1703）。空間的イベントマップのローカルコピーは、ローカルディスプレイの表示可能領域へマップする空間的イベントマップエントリについての表示データを収集するためにトラバースされる。いくつかの実施形態では、クライアントは、作業空間内のズームやパンといった予測されるユーザ対話の支援に備えるために、ローカルディスプレイの表示可能領域より大きな領域を定義する選抜境界（culling boundary）内において空間的イベントマップエントリについてのディスプレイの描画を支援する追加のデータを収集し得る（1704）。クライアントプロセッサは空間的イベントマップイベント、一時的イベント及び表示データを用いて、表示境界の範囲内に収まる空間的イベントマップの部分を描画する処理を実行する（1705）。この処理はTUIODライバからのような、ローカルのユーザインタフェースメッセージを受信する（1706）。また、この処理は共同サーバからソケットAPIメッセージを受信する（1710）。ローカルユーザインタフェースメッセージに回答して、履歴イベントと一時的イベントに入力を分離する処理を行い、APIにより特定される通りに、履歴イベントと一時的イベントの両方に対し、ソケット上に共同サーバに向けてAPIメッセージを送信し、空間的イベントのキャッシュ部分を履歴イベントで更新し、履歴イベントと一時的イベントの両方の表示データを生成する（1707）。ソケットAPIメッセージに回答して、空間的イベントのキャッシュ部分をサーバ側ネットワークノードにより識別された履歴イベントで更新し、APIにより特定される通りに、ソケット上のAPIメッセージに回答し、ソケットメッセージにより通知された履歴イベントと一時的イベントの両方の表示データを生成する（1711）。

30

40

#### 【 0 1 5 1 】

図18はクライアント側ネットワークノードにより実行されるユーザ入力を解釈する処理の簡略化されたフロー図である。この簡略化されたフロー図に示された順序は、説明のために提示されたものであり、特定の実装に適合するように修正されることがある。多くのステップは、例えば、並列に実行され得る。処理は、ディスプレイ上の物理座標と一緒に、実際のユーザ入力デバイスからネイティブIOイベント型メッセージを受信することから開始する（1800）。処理はディスプレイ上の物理座標を作業空間内の座標にマップし、空間的イベントマップのローカルキャッシュコピーを用いて、作業空間座標にマップする空間的イベントマップ内のオブジェクトがあればそれを識別する（1801）。その

50

後、作業空間座標、識別されたオブジェクト、コンテキスト及びI/Oイベント型に基づき、プロセッサはイベントが履歴イベントかそれとも一時的イベントかを含むAPIイベント型を判定する(1802)。その後、クライアント側プロセッサはAPI準拠メッセージを生成し、空間的イベントマップのローカルキャッシュコピーを更新し、表示データを更新する(1803)。上述したサーバ側ネットワークノード及びクライアント側ネットワークノードを用いて、いくつかの基本手続は、ログインし、セッションに係る空間的イベントマップをクライアント側ネットワークノードにダウンロードすること、及び、ライブ作業空間の空間的イベントの作業空間チャンネルに接続することを含む。

#### 【0152】

- ・ログイン及び空間的イベントマップのダウンロード 10
  1. クライアントは共同セッションへの参加の承認を要求し、作業空間を開く。
  2. サーバはクライアントのセッションへの参加を承認し、作業空間に係る空間的イベントマップの読み込みを開始する。
  3. クライアントは、「目次(table of contents)」のような、セッションに関連付けられた空間的イベントマップの識別を要求する。
  4. 空間的イベントマップの目次により識別される夫々の部分がクライアントにより要求される。これらの空間的イベントマップの部分は、全体で、作業空間時間の開始から現在までのイベントを順番に配列したものであるものとしての作業空間を表す。「作業空間時間の開始」は、共同セッションの初期化時刻からの経過時間、又はセッションに関し記録された絶対時刻を考慮することができる。 20
  5. クライアントは空間的イベントマップのキャッシュコピーをローカルメモリ内にまとめる。
  6. クライアントは、ローカルディスプレイ上の現在の表示可能領域又は表示域(viewport)を考慮して、空間的イベントマップを用いて関連する項目を決定し、これにより作業空間内の適切な領域を表示させる。

#### 【0153】

- ・ライブ空間的イベントマップイベントのセッションチャンネルへの接続：
  1. 承認の後、クライアントは作業空間チャンネルへの参加を要求する。
  2. サーバは、クライアントを、作業空間チャンネルを介して更新を受け取る作業空間参加者のリストに加える。 30
  3. クライアントは履歴イベントと一時的イベントの両方を運ぶライブメッセージを作業空間から受信し、チャットルームと同類の通信枠組みを受信する。例えば、一連の一時的イベントと、履歴イベントは、空間的イベントマップ内のオブジェクトの当該空間的イベントマップ内の新たな位置への移動と関連付けられ得る。
  4. クライアントは、空間的イベントマップのローカルコピーを入れ替え、ローカルディスプレイを再描画することによってサーバ側ネットワークからのライブメッセージに反応する。
  5. ライブメッセージは、取消可能で、空間的イベントマップに記録されるイベントとして永続すべき「履歴」イベントと、セッションの履歴の一部にならない情報である「一時的」イベントからなる。 40
  6. クライアントがローカルディスプレイとの対話によりオブジェクトを生成、修正、移動又は削除すると、クライアント側ネットワークノードにより新たなイベントが生成され、作業空間チャンネルを介しサーバ側ネットワークノードへ送信される。サーバ側ネットワークノードはセッションに係る空間的イベントマップ内の履歴イベントを保存し、履歴イベントと一時的イベントの両方をセッション内の全ての稼働中のクライアントへ配信する。
  7. セッションから退出するに際し、クライアントは作業空間チャンネルとの接続を切断する。

#### 【0154】

共同システムは、共有される共同サーバによって管理される作業空間データに基づき画 50

像を表示すると、作業空間データに貢献し得るユーザ入力を受け付けるのとの両方に用いられる多数の分散型デジタルディスプレイを有することができ、夫々のディスプレイがセッション履歴、リアルタイムのローカル入力及び他のディスプレイからのリアルタイムの入力に基づいて表示するための画像を迅速に構成する。

【0155】

本明細書中において、一項目の情報の「識別」は、情報の当該項目について直接の特定を必ずしも必要とするものではない。情報は、1以上の間接レイヤを介した実際の情報を単に参照することによって、或いは、組み合わせることで情報の実際の項目を判断するのに十分となる異なる情報の1以上の項目を識別することによって、フィールド内で「識別」され得る。加えて、用語「示す(indicate)」は、本明細書中において「識別する(identify)」と同じ意味で用いられる。

10

【0156】

また、本明細書中において、所定の信号、イベント又は値は、先行する信号、イベント又は値が当該所定の信号、イベント又は値に影響を与えるものであれば、当該先行する信号、イベント又は値に「応答」する。さらに、介在する処理要素、ステップ又は期間があるときも、当該所定の信号、イベント又は値は、先行する信号、イベント又は値に「応答」する。介在する処理要素又はステップが1を超える信号、イベント又は値を組み合わせる場合、当該処理要素又はステップの信号出力は「夫々の」信号、イベント又は値入力に対して「応答」すると考えられる。所定の信号、イベント又は値が、先行する信号、イベント又は値と一致する場合は、単に劣化した場合に過ぎず、当該所定の信号、イベント又は値が、当該先行する信号、イベント又は値に「応答」すると考えられる。他の信号、イベント又は値に対する所定の信号、イベント又は値の「依存性」については同様に定義される。

20

【0157】

出願人は、本明細書に記載された個々の特徴の夫々、及び、当該特徴の2以上の組み合わせを、当業者の共通の一般的知識に照らして全体として現明細書に基づいて実現が可能な当該特徴又は組み合わせの範囲内において、当該特徴又は特徴の組み合わせが本明細書に記載された課題を解決するかに拘らず、また、特許請求の範囲が限定されることなく、ここに、分離して開示する。出願人は本発明の態様が任意の当該特徴又は特徴の組み合わせから構成され得ることを示す。前述の記載を鑑み、本発明の範囲内で種々の修正がなされ得ることは、当業者にとって明白である。

30

【0158】

本発明の好ましい実施形態についての前述の記載は、図示及び説明のために提供されたものである。網羅的であること、又は発明を開示された正にその形式に限定することを意図するものではない。言うまでもなく、多数の修正や変形は熟練の実施者にとって明白であろう。例えば、本明細書中に記載したディスプレイは大型のものであるが、小型のディスプレイを配列させて複数の描画領域を利用することも可能である。もっとも、複数の描画領域は少なくとも幅12フィートの大きさを有するディスプレイにおいてより有益である。特に、限定なく、本願の背景技術の欄、又は参考文献に含まれる内容により記載され、示唆される任意の及び全ての変形が、本明細書の発明の実施形態の記載中に参照により具体的に取り込まれる。加えて、任意のある実施形態に関して本明細書に記載され、示唆され、又は参照により取り込まれた任意の及び全ての変形は、他の全ての実施形態に対しても教示されると考えられる。本明細書中に記載の実施形態は、本発明の原理及びその実際の応用を最も良く説明し、これにより、様々な実施形態について、及び熟考される特定の使用に適した様々な修正について、本発明の理解を他の当業者に可能とするために選択され記載されたものである。本発明の範囲は以下の請求の範囲とその均等物によって規定されることが意図される。

40

【図1A】

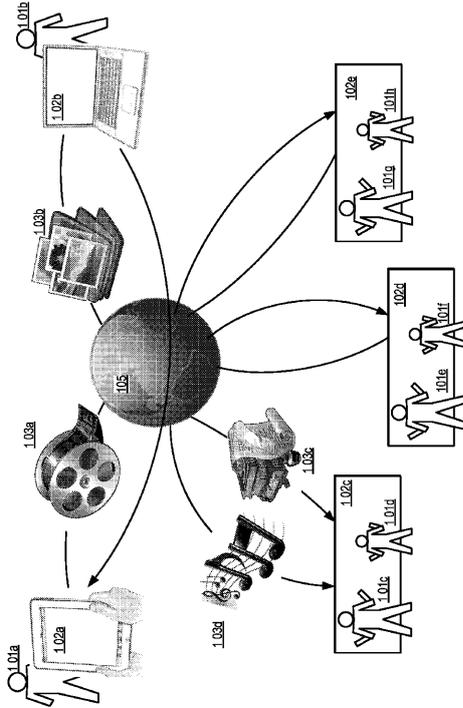
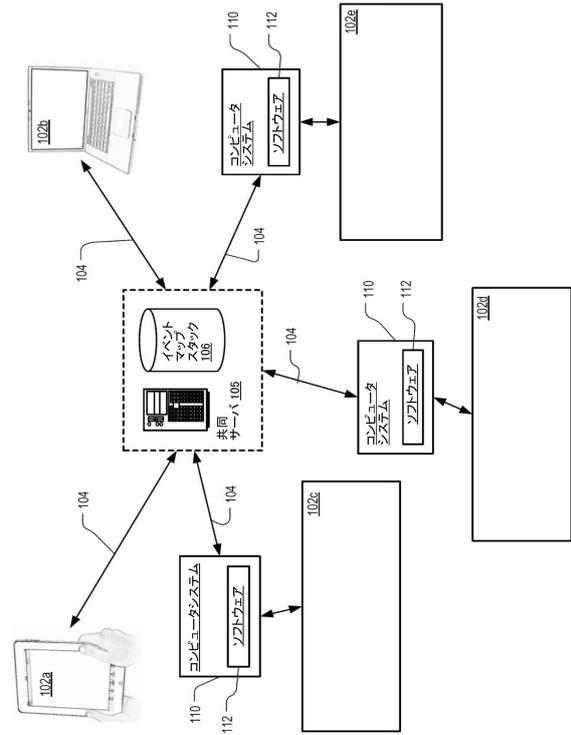
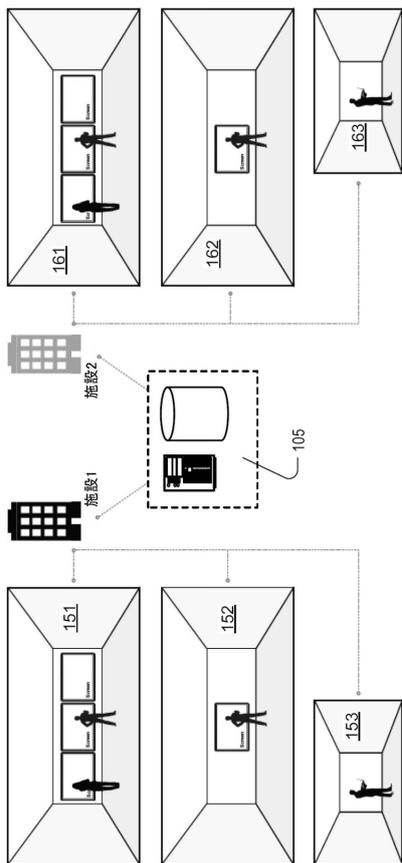


FIG. 1A

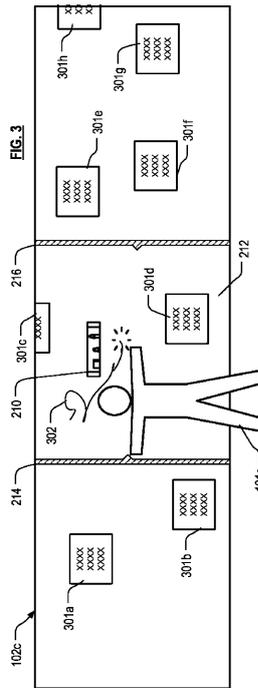
【図1B】



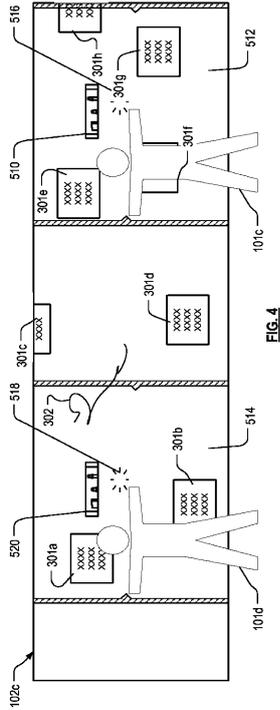
【図2】



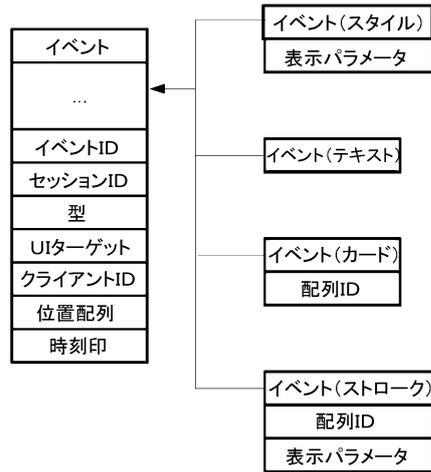
【図3】



【 図 4 】



【 図 5 A 】



【 図 5 B 】

カード
...
セッションID
カード型
配列ID
クライアントID
大きさ
ファイル型
セッション位置

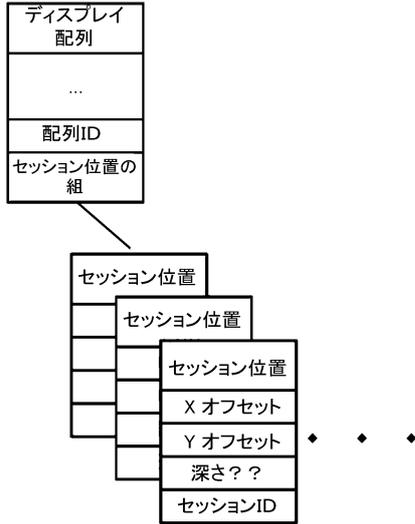
【 図 5 D 】

ユーザセッション
...
アクセストークン
セッションクライアントID
セッションユーザID
最新のアクセス
有効期限
クッキー

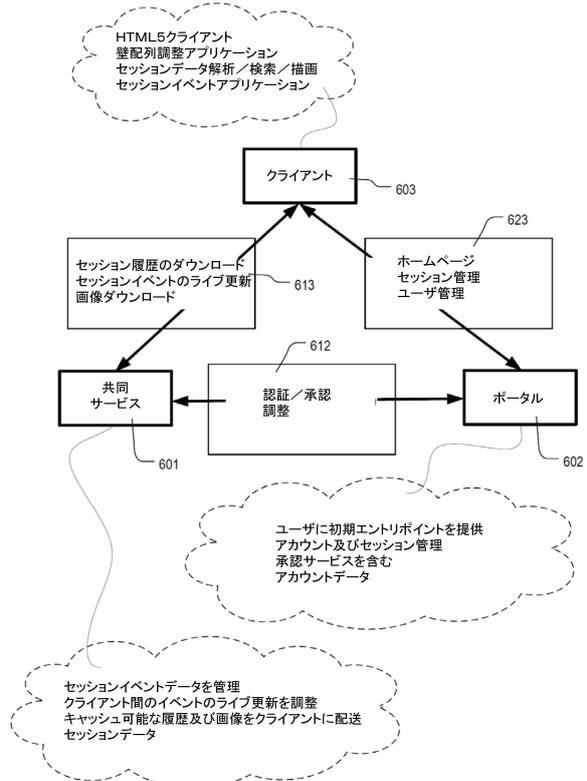
【 図 5 C 】

チャンク
...
セッションID
受信元
送信先
時刻印

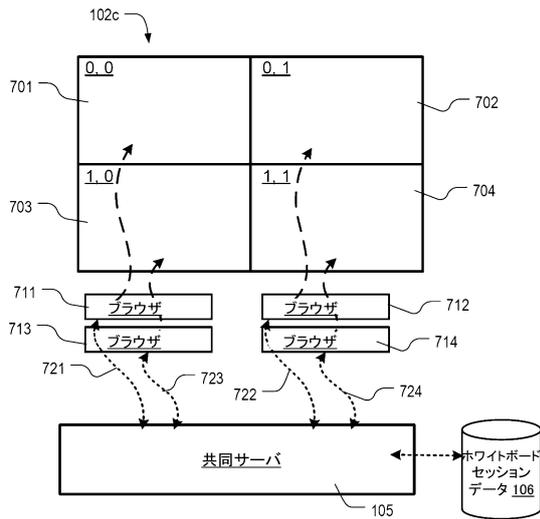
【図5E】



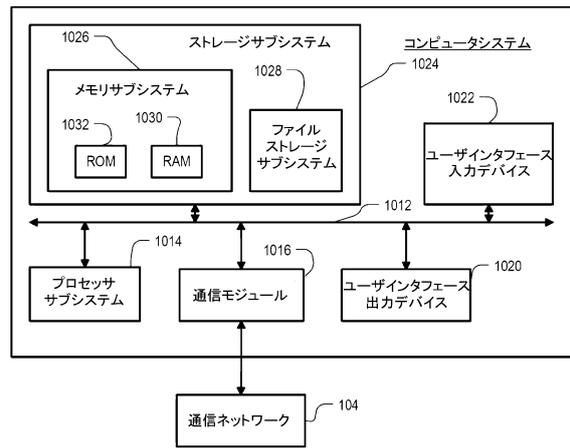
【図6】



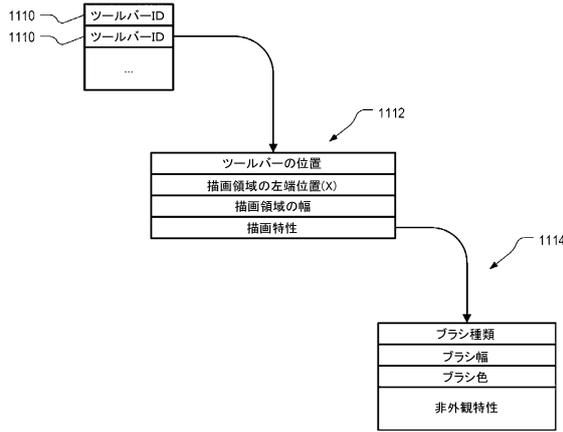
【図7】



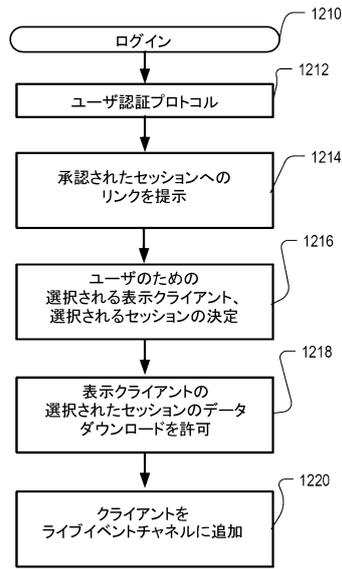
【図8】



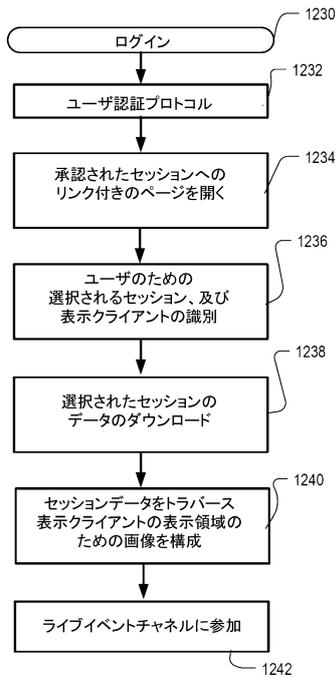
【図 9】



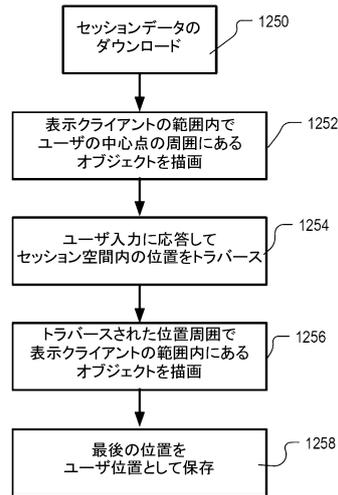
【図 10】



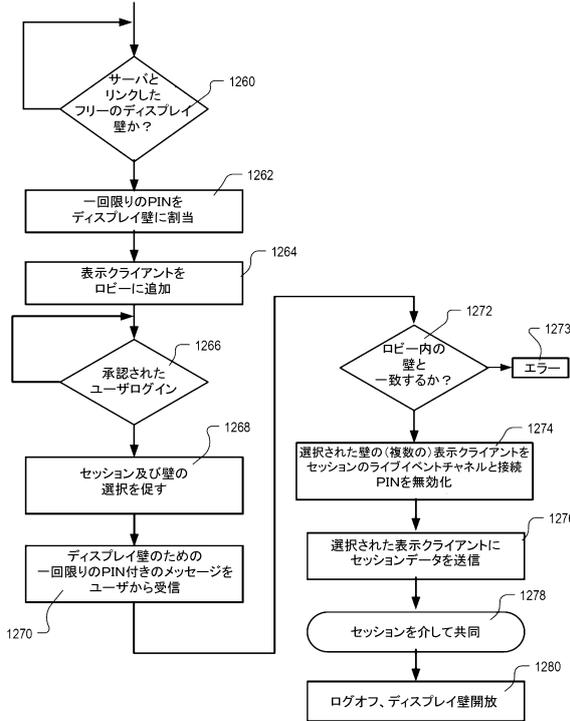
【図 11】



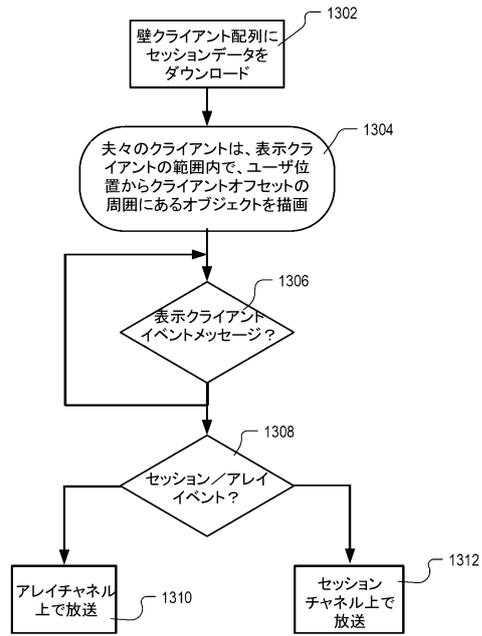
【図 12】



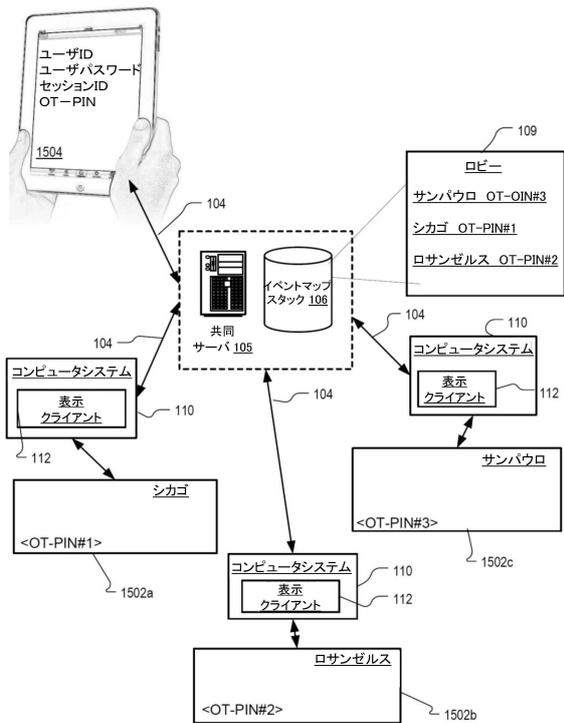
【図13】



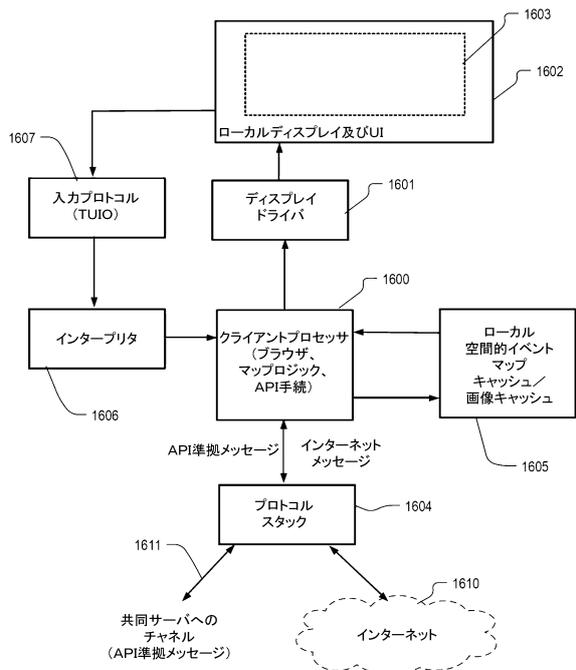
【図14】



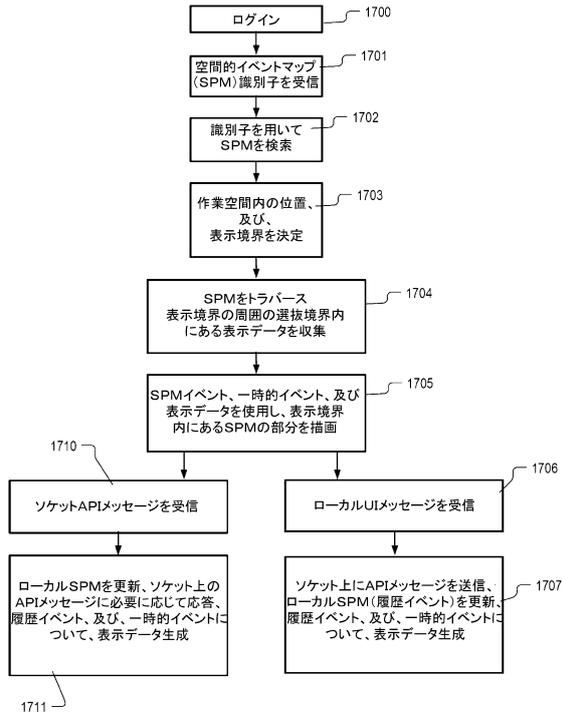
【図15】



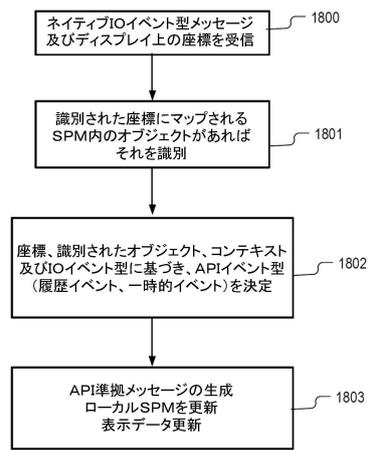
【図16】



【図17】



【図18】



---

フロントページの続き

(72)発明者 エントレキン, デミアン  
アメリカ合衆国 49423 ミシガン州 ホランド, ワン ハワース センター

審査官 佐々木 洋

(56)参考文献 特開2010-165178(JP,A)  
特開2002-116996(JP,A)  
特開2010-218313(JP,A)  
米国特許出願公開第2012/0320073(US,A1)  
特開2005-258877(JP,A)  
特開2010-129093(JP,A)

(58)調査した分野(Int.Cl., DB名)  
G06F 13/00