

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号  
特許第6175682号  
(P6175682)

(45) 発行日 平成29年8月9日 (2017.8.9)

(24) 登録日 平成29年7月21日 (2017.7.21)

(51) Int.Cl.

F I

G O 6 F 3/0488 (2013.01)

G O 6 F 3/0488

請求項の数 10 (全 19 頁)

(21) 出願番号	特願2014-551382 (P2014-551382)	(73) 特許権者	314015767
(86) (22) 出願日	平成25年1月6日 (2013.1.6)		マイクロソフト テクノロジー ライセン
(65) 公表番号	特表2015-531086 (P2015-531086A)		シング, エルエルシー
(43) 公表日	平成27年10月29日 (2015.10.29)		アメリカ合衆国 ワシントン州 9805
(86) 国際出願番号	PCT/US2013/020430		2 レッドモンド ワン マイクロソフト
(87) 国際公開番号	W02013/103926		ウェイ
(87) 国際公開日	平成25年7月11日 (2013.7.11)	(74) 代理人	100107766
審査請求日	平成27年12月21日 (2015.12.21)		弁理士 伊東 忠重
(31) 優先権主張番号	2763316	(74) 代理人	100070150
(32) 優先日	平成24年1月6日 (2012.1.6)		弁理士 伊東 忠彦
(33) 優先権主張国	カナダ (CA)	(74) 代理人	100091214
(31) 優先権主張番号	13/363,046		弁理士 大貫 進介
(32) 優先日	平成24年1月31日 (2012.1.31)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 効率的なカスケードオペレーションの実現

(57) 【特許請求の範囲】

【請求項 1】

第 1 のカスケードオペレーションが定義されている要素に関連する入力を検出するステップと、

前記第 1 のカスケードオペレーションを前記要素に適用するステップと、

第 2 のカスケードオペレーションが定義されている、関連ツリー内の前記要素の少なくとも 1 つの上位要素に、前記第 2 のカスケードオペレーションを適用することに関連する処理コストを決定するステップと、

前記処理コストが処理コスト閾値を超えると判断したことに応答して、前記少なくとも 1 つの上位要素への前記第 2 のカスケードオペレーションの適用を終了するステップと、  
を含む、方法。

【請求項 2】

前記検出するステップは、タッチ入力を検出することを含む、請求項 1 に記載の方法。

【請求項 3】

前記検出するステップは、非接触入力を検出することを含む、請求項 1 に記載の方法。

【請求項 4】

前記第 1 のカスケードオペレーションは、少なくとも 1 つのカスケードスタイルシート (CSS) 疑似クラスにより定義されるカスケードオペレーションを備える、請求項 1 に記載の方法。

【請求項 5】

前記決定するステップ及び前記終了するステップに先行して、少なくとも1つの他のカスケードオペレーションを少なくとも1つの他の上位要素に適用するステップを更に含み

、  
前記少なくとも1つの他のカスケードオペレーションを適用するステップは、時間ベースのパラメータを使用して、前記少なくとも1つの他のカスケードオペレーションを適用すべき時を確認することによって実行される、請求項1に記載の方法。

【請求項6】

前記決定するステップ及び前記終了するステップに先行して、少なくとも1つの他のカスケードオペレーションを少なくとも1つの他の上位要素に適用するステップを更に含み

、  
前記少なくとも1つの他のカスケードオペレーションを適用するステップは、少なくとも時間ベースのパラメータ以外のパラメータを使用して、前記少なくとも1つの他のカスケードオペレーションを適用すべき上位要素を確認することによって実行される、請求項1に記載の方法。

【請求項7】

前記終了するステップに先行して、1つ又は複数の他のカスケードオペレーションを適用するステップを更に含む、請求項1に記載の方法。

【請求項8】

プロセッサによって実行されると、該プロセッサに、

第1のカスケードオペレーションが定義されている要素に関連する入力を検出するステップと、

前記検出に応答してタイマを開始するステップと、

前記第1のカスケードオペレーションを前記要素に適用するステップと、

第2のカスケードオペレーションが定義されている少なくとも1つの上位要素に、前記第2のカスケードオペレーションを適用することに関連する処理コストを決定するステップと、

前記タイマの期限が切れており、かつ前記入力はまだ検出されていることに応答し、前記処理コストが処理コスト閾値を超えないと判断したことに応答して、前記第2のカスケードオペレーションを、前記少なくとも1つの上位要素に適用するステップと、

を含む方法を実行させる、コンピュータプログラム。

【請求項9】

前記検出するステップは、タッチ入力を検出することを含む、請求項8に記載のコンピュータプログラム。

【請求項10】

システムにおいて、

1つ以上のプロセッサと；

前記1つ以上のプロセッサによって実行されると、当該システムに、

1つ以上のカスケードオペレーションが定義されているドキュメントオブジェクトモデル(DOM)ツリーから要素のサブセットを選択するステップと、

それぞれの前記カスケードオペレーションを前記DOMツリーからの前記要素のサブセットに適用するステップと、

それぞれの前記カスケードオペレーションを、前記DOMツリー内の前記要素のサブセットの1つ以上の上位要素に適用する平均時間を決定するステップと、

少なくとも1つのタイマの期限が切れた後、前記平均時間が予め定められた時間閾値未満であるという決定に応答して、追加のカスケードオペレーションを前記DOMツリー内の前記要素のサブセットの前記1つ以上の上位要素に適用するステップと、

を実行させるカスケードオペレーションモジュールを記憶する1つ以上のコンピュータ読取可能記憶デバイスと；

を備える、システム。

【発明の詳細な説明】

10

20

30

40

50

## 【技術分野】

## 【0001】

本発明は、効率的なカスケードオペレーションを可能にすることに関する。

## 【背景技術】

## 【0002】

ウェブ上で使用することができるコンテンツを設計する人が直面する1つの課題は、対応するような速さで動作するには必ずしも設計されていない可能性のあるデバイスを含め、様々なタイプのデバイスにおいて、感知できるほど早く、そのようなコンテンツを実行することができるようにすることである。例えばウェブコンテンツは、典型的に、下位要素(descendent element)と上位要素(ancestor element)と有する要素を含む、階層型のドキュメントオブジェクトモデル(DOM)ツリーで表される。

10

## 【発明の概要】

## 【発明が解決しようとする課題】

## 【0003】

多くの場合、特定の要素に関して入力を受け取ると、その入力をベースとして使用して、関連する挙動又はファイイベントを「トンネル」又はカスケードし、他の上位要素へと階層型のツリーを上る。大きなツリーの場合、どの要素に特定の挙動を適用すべきかを決定することは、コストのかかるオペレーションであり得る。したがって、確実にユーザの経験を効率的で、より良く認識されるようにすることは引き続き課題である。

## 【課題を解決するための手段】

20

## 【0004】

この「発明の概要」の記載は、以下の「発明を実施するための形態」において更に説明される概念の選択を、簡単な形で紹介するために提供される。この「発明の概要」の記載は、特許請求に係る主題の重要な特徴又は本質的特徴を特定するようには意図されていない。

## 【0005】

様々な実施形態は、階層型ツリー内の指定された要素のうちの1つの要素に関連付けられた入力を受信することに対応して、カスケードオペレーションのサブセットを、それらの指定された要素に選択的に適用することによって、効率的なカスケードオペレーションが実行されることを可能にする。カスケードオペレーションのサブセットを実行した後、様々なパラメータに従ってカスケードオペレーションの完全なセットを実行することができる。

30

## 【0006】

1つ又は複数の実施形態において、受信した入力についてのカスケードオペレーションのサブセットは、限定ではなく例として、ほんの数例を挙げると、ユーザ対話のタイミング、所与の要素についてのカスケードオペレーションの複雑性、及び/又はカスケードオペレーションを適用することができる要素の数を含むパラメータに基づいて、決定され得る。

## 【0007】

以下の発明の詳細な説明は、添付の図面を参照しながら説明される。図面において、参照番号の左端の桁は、その参照番号が最初に現れる図面を特定する。説明及び図面の異なる例における同じ参照番号の使用は、同様又は同一のアイテムを示す可能性がある。

40

## 【図面の簡単な説明】

## 【0008】

【図1】1つ又は複数の実施形態に係る例示的実装の環境の例を示す図である。

【図2】図1に示される例示的実装におけるシステムの例をより詳細に示す図である。

【図3】カスケードオペレーションを適用することができる例示的な実施形態を示す図である。

【図4】カスケードオペレーションを適用することができる例示的な実施形態を示す図である。

50

【図 5】カスケードオペレーションを適用することができる例示的な実施形態を示す図である。

【図 6】カスケードオペレーションを適用することができる例示的な実施形態を示す図である。

【図 7】1つ又は複数の実施形態に係る方法のステップを示すフロー図である。

【図 8】1つ又は複数の実施形態に係る方法のステップを示すフロー図である。

【図 9】1つ又は複数の実施形態に係る実装例を示す図である。

【図 10】本明細書で説明される様々な実施形態を実装するのに使用することができる例示的なコンピューティングデバイスを示す図である。

【発明を実施するための形態】

10

【0009】

< 概要 >

様々な実施形態は、階層型ツリー内の指定された要素のうちの1つの要素に関連付けられた入力を受信することに応答して、カスケードオペレーションのサブセットを、それらの指定された要素に選択的に適用することによって、効率的なカスケードオペレーションが実行されることを可能にする。カスケードオペレーションのサブセットを実行した後、様々なパラメータに従ってカスケードオペレーションの完全なセットを実行することができる。

【0010】

1つ又は複数の実施形態において、受信した入力についてのカスケードオペレーションのサブセットは、限定ではなく例として、ほんの数例を挙げると、ユーザ対話のタイミング、所与の要素についてのカスケードオペレーションの複雑性、及び/又はカスケードオペレーションを適用することができる要素の数を含むパラメータに基づいて、決定され得る。

20

【0011】

以下の議論では、まず、本明細書で説明される技術を用いるのに動作可能な例示的環境について説明する。次に、様々な実施形態の例示的な実施例について説明する。これらの実施形態は、例示的な環境だけでなく他の環境においても用いられる可能性がある。したがって、例示的環境は、説明される実施形態を実行するものに限定されず、説明される実施形態は、例示的環境における実装に限定されない。

30

【0012】

< 例示的な動作環境 >

図 1 は、本明細書で説明される技術を用いるのに動作可能な例示的な実装の環境 100 の図である。図示される環境 100 は、様々な方法で構成され得るコンピューティングデバイス 102 の例を含む。例えばコンピューティングデバイス 102 は、図 2 に関連して更に説明されるような、従来のコンピュータ（例えばデスクトップ型のパーソナルコンピュータ、ラップトップコンピュータ等）、モバイルステーション、娯楽用機器、テレビに通信的に接続されるセットトップボックス、ワイヤレスフォン、ネットブック、ゲームコンソール、ハンドヘルドデバイス等として構成され得る。したがって、コンピューティングデバイス 102 は、相当なメモリ及びプロセッサリソースを有する完全リソースデバイス（例えばパーソナルコンピュータ、ゲームコンソール）から、限られたメモリ及び/又は処理リソースを有する低リソースデバイス（例えば従来のセットトップボックス、ハンドヘルドゲームコンソール）にまで及ぶことがある。コンピューティングデバイス 102 は、該コンピューティングデバイス 102 に、下述するような1つ又は複数のオペレーションを実行させるソフトウェアも含む。

40

【0013】

コンピューティングデバイス 102 は、カスケードオペレーションモジュール 104 を含む。カスケードオペレーションモジュール 104 は、以下で更に詳述されるように、ユーザの経験を向上するように設計される効果的な手法で、効率的なカスケードオペレーションを実行することができるように構成される。少なくとも一部の実施形態において、カ

50

スケードオペレーションモジュール104は、1つ又は複数のタイマの使用により、1つ又は複数の受信したジェスチャ入力に関連付けられた時間を測定し、これにตอบสนองして、カスケードオペレーションを実行することができる。例えば少なくとも一部の実施形態において、カスケードオペレーションモジュールは、階層型ツリー内の指定された要素のうちのいずれか1つに関連付けられたジェスチャ入力のような入力を受信したことにตอบสนองして、カスケードオペレーションのサブセットをそれらの指定された要素に選択的に適用することによって、効率的なカスケードオペレーションを実行できるようにすることが可能である。次いで、カスケードオペレーションのサブセットを実行した後に、様々なパラメータに従って、カスケードオペレーションの完全なセットを実行することができる。カスケードオペレーションのサブセットは、受信した入力について、限定ではなく例としてほんの数例を挙げると、ユーザ対話のタイミング、所与の要素に対するカスケードオペレーションの複雑性、及び/又はカスケードオペレーションを適用することができる要素の数を含むパラメータに基づいて決定され得る。

10

#### 【0014】

コンピューティングデバイス102は、ジェスチャモジュール105も含む。ジェスチャモジュール105は、1又は複数の指によって実行され得る入力ポイントジェスチャを認識し、該ジェスチャに対応するオペレーション又はアクションを実行させる。モジュール105によって様々な異なる方法でジェスチャが認識され得る。例えばジェスチャモジュール105は、タッチスクリーン機能を使用して、又は例えばニアフィールド技術を用いてディスプレイデバイス108に必ずしも物理的に接触しないユーザの指の近接を感知する機能を使用して、コンピューティングデバイス102のディスプレイデバイス108へ近接するユーザの手106aの指のようなタッチ入力を認識するように構成され得る。モジュール105を使用して、単一の指のジェスチャとベゼルジェスチャ、複数の指/同じ手のジェスチャとベゼルジェスチャ、及び/又は複数の指/異なる手のジェスチャとベゼルジェスチャを認識することができる。カスケードオペレーションモジュール104とジェスチャモジュール105は、別個のモジュールとして示されているが、双方によって提供される機能性を、単一の一体型ジェスチャモジュールとして実装することができる。モジュール104及び/又は105によって実装される機能性を、限定ではなく例としてウェブブラウザのような任意の適切に構成されたアプリケーションによって実装することができる。特許請求に係る主題の精神及び範囲から逸脱することなく、他のアプリケーションが使用される可能性もある。

20

30

#### 【0015】

コンピューティングデバイス102は、(例えばユーザの手106aの1又は複数の指によって提供される)タッチ入力と、(例えばスタイラス116によって提供される)スタイラス入力とを検出して区別するようにも構成され得る。この区別は、ユーザの手106aの指によって接触されるディスプレイデバイス108の量に対する、スタイラス116によって接触されるディスプレイデバイス108の量を検出すること等のような、様々な方法で実行することが可能である。

#### 【0016】

したがって、ジェスチャモジュール105は、スタイラスとタッチ入力と、並びに異なるタイプのタッチ入力と非接触入力との間の区別の認識及び利用を通して、様々な異なるジェスチャ技術をサポートし得る。

40

#### 【0017】

図2は、複数のデバイスが中央コンピューティングデバイスを通じて相互接続される環境において実装されるような、カスケードオペレーションモジュール104及びジェスチャモジュール105を示す例示システム200を図示している。中央コンピューティングデバイスは、それらの複数のデバイスに対してローカルであってもよく、それらの複数のデバイスからリモートに配置されてもよい。一実施形態において、中央コンピューティングデバイスは、ネットワーク又はインターネット又は他の手段を通じて複数のデバイスに接続される1つ又は複数のサーバコンピュータを備える、「クラウド」サーバファームで

50

ある。

【 0 0 1 8 】

一実施形態において、この相互接続アーキテクチャは、機能性が複数のデバイスにわたって伝達され、該複数のデバイスのユーザに共通のシームレスな経験が提供されることを可能にする。複数のデバイスはそれぞれ、異なる物理的要件及び能力を有することがあり、中央コンピューティングデバイスは、そのデバイスに対して調整されるが、依然として全てのデバイスに共通の経験をデバイスに伝達することを可能にするプラットフォームを使用する。一実施形態において、ターゲットデバイスの「クラス」を作成して、経験をデバイスの汎用クラスに対して調整する。あるデバイスのクラスを、複数のデバイスの物理的な特徴又は用法又は他の共通の特性によって定義してもよい。例えば上述のように、コンピューティングデバイス 1 0 2 は、モバイル 2 0 2 での使用、コンピュータ 2 0 4 での使用及びテレビジョン 2 0 6 での使用のような、様々な異なる方法で構成され得る。これらの構成の各々は一般に、対応する画面サイズを有し、したがって、コンピューティングデバイス 1 0 2 は、この例示システム 2 0 0 においてこれらのデバイスクラスのうちのいずれか 1 つとして構成され得る。例えばコンピューティングデバイス 1 0 2 は、携帯電話、音楽プレイヤー、ゲームデバイス等を含むデバイスのモバイル 2 0 2 のクラスを想定してもよい。コンピューティングデバイス 1 0 2 はまた、パーソナルコンピュータ、ラップトップコンピュータ、ネットブック等を含むデバイスのコンピュータ 2 0 4 のクラスを想定してもよい。テレビジョン 2 0 6 の構成は、略式の環境において表示を伴うデバイスの構成、例えばテレビジョン、セットトップボックス、ゲームコンソール等を含む。したがって、本明細書で説明される技術は、コンピューティングデバイス 1 0 2 のこれらの様々な構成によってサポートされ、以下のセクションにおいて説明される具体的な例には限定されない。

【 0 0 1 9 】

クラウド 2 0 8 は、ウェブサービス 2 1 2 のためのプラットフォーム 2 1 0 を含むように図示されている。プラットフォーム 2 1 0 は、クラウド 2 0 8 のハードウェア（例えばサーバ）及びソフトウェアリソースの基礎となる機能を抽象化し、したがって「クラウドオペレーティングシステム」として機能し得る。例えばプラットフォーム 2 1 0 はリソースを抽象化して、コンピューティングデバイス 1 0 2 を他のコンピューティングデバイスに接続してもよい。プラットフォーム 2 1 0 は、リソースのスケールを抽象化して、プラットフォーム 2 1 0 を介して実装されるウェブサービス 2 1 2 について直面する需要に対して、対応するレベルのスケールを提供するようにも機能し得る。サーバファームにおけるサーバの負荷分散や、悪意あるパーティ（例えばスパム、ウィルス及び他のマルウェア）に対する保護等の様々な他の例も考慮される。

【 0 0 2 0 】

したがって、クラウド 2 0 8 は、インターネット又は他のネットワークを介してコンピューティングデバイス 1 0 2 に対して利用可能となるソフトウェア及びハードウェアリソースに関するストラテジの一部として含まれる。

【 0 0 2 1 】

カスケードオペレーションモジュール 1 0 4 及びジェスチャモジュール 1 0 5 によってサポートされるジェスチャ技術は、モバイル 2 0 2 構成におけるタッチスクリーン機能、コンピュータ 2 0 4 構成におけるトラックパッド機能を使用することによって検出されることがあり、また、カメラによって、特定の入力デバイスとの接触を要しないナチュラルユーザインタフェース（NUI）のサポートの一部として検出されることが等がある。さらに、特定のジェスチャを識別する入力を検出及び認識するオペレーションの性能は、コンピューティングデバイス 1 0 2 によって、及び／又はクラウド 2 0 8 のプラットフォーム 2 1 0 によりサポートされるウェブサービス 2 1 2 によって等、システム 2 0 0 を通じて分散されることがある。

【 0 0 2 2 】

一般に、本明細書で説明される機能のいずれかを、ソフトウェア、ファームウェア、ハ

10

20

30

40

50

ードウェア（例えば固定の論理回路）、手動の処理又はこれらの実装の組み合わせにより実装することができる。本明細書で使用されるとき、「モジュール」、「機能」及び「ロジック」という用語は、概してソフトウェア、ファームウェア、ハードウェア又はこれらの組み合わせを表す。ソフトウェアの実装の場合、モジュール、機能又はロジックは、プロセッサ（例えば1つ又は複数のCPU）において又はプロセッサによって実行されると、指定されたタスクを実行するプログラムコードを表す。プログラムコードは、1つ又は複数のコンピュータ読取可能なメモリデバイスに格納され得る。下述されるジェスチャ技術の特徴は、プラットフォーム独立であり、これは、本技術が、様々なプロセッサを有する様々な市販のコンピューティングプラットフォームにおいて実装され得ることを意味する。

10

#### 【0023】

以下の議論では、様々なセクションにおいて様々な例示的实施形態を説明する。「例示的なカスケードオペレーション」というタイトルのセクションでは、1つ又は複数の実施形態に従ってカスケードオペレーションを実行することができる実施形態を説明する。この次の「単一のタイマの使用例」というタイトルのセクションでは、1つ又は複数の実施形態に従って単一のタイマを使用することができる例を説明する。次に、「複数のタイマの使用例」というタイトルのセクションでは、1つ又は複数の実施形態に従って複数のタイマを使用することができる例を説明する。これに続く「他のパラメータ」というタイトルのセクションでは、1つ又は複数の実施形態に従ってカスケードオペレーションを実行するために、タイミングパラメータに加えて又はこれとは別個に、他のパラメータを使用

20

#### 【0024】

効率的なカスケードオペレーション機能を利用することができる例示的な動作環境について説明してきたが、次に例示的な実施形態の議論を検討する。

#### 【0025】

##### < 例示的なカスケードオペレーション >

上述のように、様々な実施形態では、階層型ツリー内の指定された要素のうちの1つの要素に関連付けられた入力を受信することに応答して、カスケードオペレーションのサブセットを、それらの指定された要素に選択的に適用することによって、効率的なカスケードオペレーションが実行されることを可能にする。カスケードオペレーションのサブセットを実行した後、様々なパラメータに従ってカスケードオペレーションの完全なセットを実行することができる。1つ又は複数の実施形態において、カスケードオペレーションのサブセットは、受信した入力について、限定ではなく例として、ほんの数例を挙げると、ユーザ対話のタイミング、所与の要素についてのカスケードオペレーションの複雑性、及び/又はカスケードオペレーションを適用することができる要素の数を含むパラメータに基づいて決定され得る。

30

#### 【0026】

下述される実施形態を、カスケード効果を有する任意のタイプのオペレーションに適用することができる。あるタイプのカスケードオペレーションは、ユーザにより認識可能な可視表現（visualization）をもたらすオペレーションに関する。これらの可視表現を任意の適切な方法で定義することができる。そのような可視表現を定義することができる1つの方法は、疑似クラスを用いるカスケードスタイルシート（CSS）の使用によるものである。2つの例示的な疑似クラスは、:hover疑似クラスと:active疑似クラスである。しかしながら、そのようなCSS疑似クラスは、説明される実施形態の対象となり得る可視表現の一例を構成すること、より具体的には、ドキュメントオブジェクトモデル（DOM）ツリーのような階層型ツリー内に現れる要素に適用することができるカスケードオペレーションを構成することを認識され、理解されたい。したがって、可視表

40

50

現を提示するこれらを含め、特許請求に係る主題の精神及び範囲を逸脱することなく、他のカスケードオペレーションを使用することができる。

【0027】

上述のCSS疑似クラスに関して、以下で検討する。

【0028】

セレクトタにおけるCSSの: hover疑似クラスと: active疑似クラスは、セレクトタによって選択された、ホバリングされている(ポイントがある)か、アクティブ化されている(例えばクリックされたか、他の方法で指定されている)要素のうちのいずれかにフォーマットが適用されるのを許可するか、又はホバリングされているかアクティブ化されている下位要素を有するのを許可する。この挙動は典型的に、ポインティング手段 (pointing implement) が、(ディスプレイ画面に接触するか又は近くに近接して置かれている) ユーザの指、スタイラス、マウス又はナチュラルユーザインタフェースを通じて受信される入力のものであっても、ホバリング/アクティブ状態を、ポインティング手段によって指定されている要素から上位ツリーを上へと「トンネリング」することによって実装される。したがって、指定された要素の上位ツリー内のいずれの要素も、同じホバリング/アクティブ状態において指定された要素であると見なされる。

10

【0029】

大きなドキュメントツリー及び複雑なCSSセレクトタについて、どの要素がホバリング状態にあるかを決定すること及び適切なフォーマットを適用することは、上記のようにコストのかかるオペレーションである可能性がある。例えばタッチを伴うユーザ対話は、極めて簡潔である。例えば要素に対するタップは、ユーザが画面を触る時からユーザが離す時まで数十ミリ秒程度の短さであり得る。多くの場合において: hoverセレクトタ及び: activeセレクトタのためにフォーマットを適用するのにかかる時間は、実際にはユーザ対話の期間を超える可能性がある。これにより、ユーザは、その対話後の遅滞の後、ホバリング/アクティブスタイルの「点滅 (blink)」を見ることになる。

20

【0030】

1つ又は複数の実施形態において、一般にユーザの対話に応答して直ぐに認識される視覚的フィードバックを提供するため、ホバリング状態において要素のサブセットがレンダリングされるように、トンネリングオペレーションの一部を部分的に実行することができる。しばらくして、又は他のパラメータとの関連で、ユーザが依然として、ポインティングデバイスにより要素を指定していることがシステムによって感知されると、ホバリング/アクティブ状態を完全にトンネリングして、上位要素の全てに適用することができる。

30

【0031】

速い対話について、ユーザは、指定された要素に対する迅速な視覚的フィードバックを見ることとなり、これによりそのタッチ対話が成功であったことを確認する。所定の期間の後続く長期の対話について、予測されるようにフォーマットの完全な適用を見ることとなる。

【0032】

カスケードオペレーションのいくつかの例を検討したので、効率的なカスケードオペレーションを可能にするための単一のタイマの使用を用いる例を検討する。

40

【0033】

<単一のタイマの使用例>

図3は、1つ又は複数の実施形態に従って、単一のタイマを使用して、カスケードオペレーションをどのようにDOMツリー内の要素のツリーに適用することができるかを示す図表示である。最初に、全体的に300で示される同心円の集合を検討する。この例において、各円は、DOMツリー内の特定の要素を表す。それぞれの大きな円は、1つ又は複数の円を包含しており、包含されている円の上位であると見なすことができる。このため、この例において例示のウェブページは全体的に302で示されている。ウェブページ302は、304、306、308及び310として複数のアクティブ化可能な要素を含む。アクティブ化可能な要素は、ウェブページのトップに現れる可能性のあるアイテムを表

50



す。それに応じて、最も内側の円は要素 3 0 4 を表す。

【 0 0 3 4 】

また、同心円の集合によって表される要素について検討すると、カスケードオペレーションは、要素 3 0 4 について定義されている。このカスケードオペレーションは、DOM ツリー内に現れる要素 3 0 4 の他の上位要素へと上にトンネリングされ得る。この特定の例では、カスケードオペレーションが CSS の : h o v e r 疑似クラスの使用により定義されると想定する。

【 0 0 3 5 】

この時点では、ウェブページ 3 0 2 内に現れる要素のいずれに対しても入力はまだ受信されていない。

【 0 0 3 6 】

次に図 4 では、ユーザが、この例では示されているように要素 3 0 4 にタッチダウンすることによって要素 3 0 4 上をホバリングしていると想定する。しかしながら、任意の適切なジェスチャ入力を使用してホバリング機能を起動することができる。要素 3 0 4 を指定する入力が受信されると、単一のタイマを開始して、要素 3 0 4 について定義されているホバリングスタイルを、迅速に適用することができる。この特定の例では、ホバリングスタイルは、示されているように要素 3 0 4 の色の変更をもたらす。ある期間の後、例えば予め定義された時間又は動的に選択可能な時間が経過した後、要素 3 0 4 が依然としてジェスチャ入力によって指定されており、要素 3 0 4 の他の上位要素は、同様に該上位要素に適用される定義済みのスタイルを有する可能性がある。例として図 5 を検討する。

【 0 0 3 7 】

図 5 において、要素 3 0 4 は、その色が変更されたままであり、加えて DOM ツリー内の要素 3 0 4 の上位要素の 1 つ又は複数が、これらの要素に適用される、: h o v e r 疑似クラスによって定義されるスタイルを有する。この特定の例において、要素 3 0 4 の 1 つの上位要素は、ここで現れる選択可能なアイテムのメニューである。定義された 1 つ又は複数のスタイルの適用を、DOM ツリー内の指定された要素の親ツリーに現れる上位要素の全てに適用することができる。

【 0 0 3 8 】

図示及び説明される実施形態において、任意の適切な時間、例えば予め定義された時間を用いることができる。少なくとも一部の実施形態では、効率的なカスケードオペレーションを可能にするために、3 0 0 ミリ秒 ( m s ) という予め定義された時間を適用することができる。少なくとも一部の実施形態では、ほぼ全てのタップが 3 0 0 m s という期間よりも短いという調査結果が示されていることを理由として、3 0 0 m s が使用される可能性がある。

【 0 0 3 9 】

他方において、その時間又は予め定義された時間内に、例えばユーザが要素 3 0 4 から指を取り除くこと等によって、要素 3 0 4 を指定している入力が取り除かれる場合、スタイルは、その上位要素の集合に適用されない。この手法では、ユーザを、スタイルが定義された可視表現に徐々に没頭させることによって、必ずしもスタイルを、指定された要素の DOM ツリー内に現れる要素の全てに対して迅速に適用する必要なく効率性が得られる。このようにして、適切に構成されたウェブブラウザ ( 又は他のアプリケーション ) は、要素の DOM ツリー内に他のフォーマットをレンダリングすべきかどうかを決定するのに所定の期間待つ間、ローカル化されたフォーマットをレンダリングすることによって、基本的なレベルの瞬間的な視覚的フィードバックを提供することができる。加えて、タッチ保持ジェスチャのような、より長いユーザ対話により、アプリケーションのホバリング / アクティブフォーマットの完全な効果が得られるので、HTML / CSS プログラミングモデルとの互換性が保たれる。

【 0 0 4 0 】

この特定の例では、要素 3 0 4 を指定する入力を受信したことに応答して、スタイルフォーマットが最初に要素 3 0 4 に適用され、一方、タイマの係属中に、その要素の上位要

10

20

30

40

50

素について定義されたスタイルフォーマットが適用されなかった。少なくとも一部の実施形態において、要素の指定があると、その要素のDOMツリーを上方へトラバースして、定義済みの要素の数に一致するか又はこれを超える要素のサブセットを識別することができる。スタイルフォーマットをDOMツリーの他の上位要素に適用すべきかどうかを確認するために、タイマの期限の終了を待ちつつ、最初に又は後に、要素のこのサブセットについて定義されたスタイルフォーマットを一般的に迅速に適用することができる。

#### 【0041】

##### < 複数のタイマの使用例 >

1つ又は複数の実施形態において、指定された要素に関連付けられたDOMツリー内に現れる上位要素について定義されているスタイルフォーマットを適用すべき時を決定するために、複数のタイマを使用することができる。例として、それぞれが600で指定されている、一連の同心円の集合を示している図6を検討する。同心円の集合内の各円は、DOMツリー内の要素を表す。同心円のそれぞれ個々の集合は、指定された要素に関連付けられたDOMツリー内に異なる時間に現れる、同じ複数の要素を表す。第1の要素を指定する入力が発出されると、該第1の要素について定義されたスタイルフォーマットを適用することができる。これは、最も中心の円に影が付けられている、1番上の同心円の集合によって示されている。第1の要素が指定されると、第1のタイマが開始される。第1のタイマが終了した後、ホバリング選択の対象とすること等により、第1の要素が指定されたままである場合、1つ又は複数の追加の上位要素が、これらの要素に同様に適用されるスタイルフォーマットを有する可能性がある。これは、2つの追加の円に影が付けられている、2番目の同心円の集合によって示されている。第2のタイマが終了した後、第1の要素が指定されたままの場合、1つ又は複数の追加の上位要素は、同様に適用されるスタイルフォーマットを有する可能性がある。これは、追加の円に影が付けられている、同心円の3番目の集合によって示されている。追加の1つ又は複数のタイマが終了すると、更なる上位要素が、該要素に適用されるスタイルフォーマットを有する可能性がある。これは、全ての円に影が付けられている、同心円の一番下の集合によって示されている。

#### 【0042】

##### < 他のパラメータ >

上記のように、時間パラメータ以外に、又はこれに加えて他のパラメータを使用して、どのようにしてカスケードオペレーションを徐々に適用すべきかを確認することができる。例えばタイマーベースのアプローチでは、タイマは、ユーザ入力を受信される方法、及び/又はデバイスの動作特性などの事項に適合するように調整可能である。例えばタイマは、ユーザ入力が起こるスピードに相当するように調整されることがある。具体的には、一部のユーザは、タップ入力のような入力を他のユーザよりも早く提供することがある。これらの例において、より早い（又は遅い）入力に相当するように、1つ又は複数のタイマを下方に（又は上方に）調整することが可能である。あるいはまた、より遅い動作特性を有することが認識されるデバイスは、そのより遅い動作特性に相当するように拡張される、1つ又は複数の関連するタイマを有する可能性がある。

#### 【0043】

あるいはまた、様々なヒューリスティックを使用して、初期又は後続のスタイルフォーマットを適用すべき要素の数を選択することができる。例えば一部のヒューリスティックは、スタイルフォーマットの複雑なアプリケーションが、要素ごとに、又は複数の要素ごとにある方法を考慮することができる。具体的には、これらの例において、初期のスタイルフォーマットの適用について、設定された数の要素を、典型的にスタイルフォーマットの適用にかかる時間に関連付けられたデータに基づいて選択してもよい。例えば平均で1つの要素が1つのスタイルフォーマットを適用するのに100msかかる場合、初期のタイマとして300msという予め定義された時間を使用して、最初に、DOMツリー内に現れる第1の2つの要素を、スタイルフォーマットを適用するために選択してよい。このようにして、ユーザが視覚的フィードバックを認識するように、対話中の残り時間でフォーマットをレンダリングすることができる。1つのタイマのみが使用される場合、次いで

、このタイマが終了した後に、上位のツリー内の残りの要素を、スタイルフォーマットの適用のために選択してよい。複数のタイマの実施形態では、初期のタイマが終了した後、上位ツリー内に現れる要素の次のサブセットを選択等することができる。

【 0 0 4 4 】

あるいはまた、スタイルフォーマットを提供すべき設定された数の要素を選択する代わりに、上位のツリーがトラバースされると、スタイルフォーマットを特定の要素に適用することに関連するコストを計算することができる。コストが特定の閾値を超えると、スタイルフォーマットの適用は終了し得る。コストを計算するために、限定ではなく例として、作業 (work) の一般的単位、CPU サイクル等を含む任意の適切なパラメータを使用することができる。

10

【 0 0 4 5 】

カスケードオペレーションを実行することができる例示的な実施形態を検討したので、次に、1つ又は複数の実施形態に係るいくつかの例示的な方法を検討する。

【 0 0 4 6 】

< 例示的な方法 >

図7は、1つ又は複数の実施形態に係る方法のステップを示すフロー図である。本方法を、任意の適切なハードウェア、ソフトウェア、ファームウェア又はこれらに任意の組み合わせとの関連で実行することができる。少なくとも一部の実施形態において、本方法を、任意のタイプのコンピュータ読取可能記憶媒体において具現化される、コンピュータ読取可能命令の形式のソフトウェアによって実行することができ、この命令を、1つ又は複数のプロセッサの影響の下で実行することができる。説明される機能を実行することができるソフトウェアの例は、上述のカスケードオペレーションモジュール104及びジェスチャモジュール105である。

20

【 0 0 4 7 】

ステップ700において、カスケードオペレーションが定義されている要素に関連する入力を検出する。任意の適切なタイプの入力を使用することができる。そのような入力には、限定ではなく例として、タッチジェスチャ又はスタイラスジェスチャによって受信され得るタッチ入力が含まれる。あるいはまた、そのような入力を、ナチュラルユーザインタフェースを通じて提供されるジェスチャ、ニアフィールド技術を通じて確認され得るような非接触ジェスチャ及びマウスクリック等により受信することができる。加えて、限定

30

【 0 0 4 8 】

ステップ702は、ステップ700において検出された入力について、要素に関連するカスケードオペレーションを適用する。ステップ704は、1つ又は複数のカスケードオペレーションを、上記カスケードオペレーションが定義されている、関連するツリー内の全ての上位要素よりも少ない要素に適用する。このステップを、任意の適切なタイプのパラメータを使用して任意の適切な方法で実行して、カスケードオペレーションを適用すべき要素を確認することができる。そのようなパラメータは、限定ではなく例として、時間ベースのパラメータ、及び/又は時間ベースのパラメータ以外のパラメータを含むことができる。加えて、ステップ704を最初に実行した後、このステップを後で、追加の要素、及び関連するツリー内の全ての上位要素よりも少ない要素に、あるいは残りの上位要素の全てに対して実行することができる。

40

【 0 0 4 9 】

図8は、1つ又は複数の実施形態に係る別の方法のステップを示すフロー図である。本方法を、任意の適切なハードウェア、ソフトウェア、ファームウェア又はこれらに任意の組み合わせとの関連で実行することができる。少なくとも一部の実施形態において、本方法を、任意のタイプのコンピュータ読取可能記憶媒体において具現化される、コンピュータ読取可能命令の形式のソフトウェアによって実行することができ、この命令を、1つ又は複数のプロセッサの影響の下で実行することができる。説明される機能を実行すること

50

ができるソフトウェアの例は、上述のカスケードオペレーションモジュール 104 及びジェスチャモジュール 105 である。

#### 【0050】

ステップ 800 において、カスケードオペレーションが定義されている要素に関連する入力を検出する。任意の適切なタイプの入力を使用することができる。そのような入力には、限定ではなく例として、タッチジェスチャ又はスタイラスジェスチャによって受信され得るタッチ入力が含まれる。あるいはまた、そのような入力を、ナチュラルユーザインタフェースを通じて提供されるジェスチャ、ニアフィールド技術を通じて確認され得るような非接触ジェスチャ及びマウスクリック等により受信することができる。加えて、限定ではなく例として、CSS 疑似クラスにより定義されるカスケードオペレーションを含め、任意の適切なタイプのカスケードオペレーションを使用することができる。

10

#### 【0051】

ステップ 802 において、タイマを開始する。任意の適切なタイプのタイマを利用することができる。タイマの例は上記に提示されている。ステップ 804 において、ステップ 800 にて検出された入力に対する要素に関するカスケードオペレーションを適用する。ステップ 806 において、タイマが終了しているかどうかを確認する。タイマが終了している場合、ステップ 808 において、ステップ 800 からの入力がまだ検出されるかどうかを確認する。更なる入力が検出されない場合、ステップ 810 において、カスケードオペレーションの適用をやめる。一方、ステップ 808 において入力がまだ検出される場合、ステップ 812 において、適用されたカスケードオペレーションを継続し、ステップ 806 に戻る。

20

#### 【0052】

ステップ 806 においてタイマが終了していた場合、ステップ 814 において、ステップ 800 からの入力がその要素上でまだ検出されるかどうかを確認する。更なる入力が検出されない場合は、ステップ 816 において、適用されたカスケードオペレーションをやめる。一方、ステップ 800 からの入力がまだ検出される場合、ステップ 818 において、少なくとも 1 つのカスケードオペレーションを、ステップ 804 にてカスケードオペレーションが適用された要素の 1 つ又は複数のそれぞれの上位要素に適用する。ステップ 818 によって適用されたカスケードオペレーションを、上位要素のサブセットに適用することができる。あるいはまた、カスケードオペレーションを上位要素の全てに適用することもできる。これが行われる方法の例は上記に提示されている。

30

#### 【0053】

1 つ又は複数の実施形態に係る例示的な方法を説明してきたので、次に実装例を説明する。

#### 【0054】

##### < 実装例 >

上記のように、ウェブページにおいて、ユーザは、ポインティングデバイス又は何らかの他の手段を使用して要素を指定することができる。要素が指定されると、該要素は、CSS の「ホバリング」状態にあると見なされる。さらに、これも上記したように、CSS は、所与の要素が「ホバリング」状態にある場合、その上位要素も全てホバリング状態にあると見なされると定めるように定義を拡張する。「ホバリング」状態を、指定された要素の親又は上位要素まで伝播するプロセスは、「ホバリングのトンネリング」として知られる。コンテンツの作成者は：hover CSS 疑似セレクタを使用して、スタイルフォーマットのルールを、ホバリング状態の要素に適用する。例えば以下の CSS は、DIV 要素がホバリング状態にあるとき、全ての DIV 要素に境界を適用することになる。

40

```
div:hover {border: 1px solid black;}
```

所与の要素は、その親によって占有される空間のサブセットを必ずしも占有しないことに留意されたい。例えば以下のコードサンプルと関連して図 9 を検討する。

#### 【0055】

```
<!doctype html>
```

50

```

<html>
  <head>
    <title>CSS Hover</title>
    <style>
      div {
        width: 200px;
        height: 200px;
        border: 1px solid black;
      }
      #B {
        position: absolute;
        right: 0px;
        top: 0px;
      }
      a:hover { color: red; }
      #A:hover { background-color: blue; }
    </style>
  </head>
  <body>
    <div id="A">A
      <div id="B">B
        <a href="#">Click Me</a>
      </div>
    </div>
  </body>
</html>

```

ここで、900として指定される一番上の要素A及び要素Bにおいて、要素Aは要素Bの親要素である。しかしながら、これらの要素は、レンダリングにおいて完全に異なる空間を占有する。このため、及びCSSのホバリング状態がトンネリングするために、ポインティングデバイスによって特別に指定されているかどうかに関わらず、要素について、ホバリングスタイルフォーマットを有することが可能である。ユーザが要素Aに対する入力を提供するとき、そのフォーマットは、902として指定される中間の要素A及び要素Bのように適用される。次に、ユーザが、904として指定される一番下の要素A及び要素Bにおいて、要素Bで「Click Me」というリンクを指定する場合、これにより、フォーマットが「Click Me」リンクに適用され、ホバリングルールによる背景色のフォーマットが要素Aについて適用されることになる。

#### 【0056】

説明されるアプローチを使用すると、「Click Me」リンクのホバリングスタイルが最初に適用され、次の上位のホバリングスタイル（B及びAに対する）がタイマの後に適用される。一実施形態において、単一の要素が初期フォーマットの適用で使用する。

#### 【0057】

実装例について検討してきたが、次に、上述のような1つ又は複数の実施形態を実装するのに用いることができる例示的デバイスを検討する。

#### 【0058】

##### <例示的なデバイス>

図10は、本明細書で説明されるアニメーションライブラリの実施形態を実装するために、図1及び2に関連して説明したような任意のタイプのポータブル及び/又はコンピュータデバイスとして実装することができる例示的デバイス1000の様々なコンポーネントを示す。デバイス1000は、（例えば受信済みのデータ、受信中のデータ、放送用に

スケジュールされたデータ、データのデータパケット等) デバイスデータ 1004 の有線及び/又は無線通信を可能にする通信デバイス 1002 を含む。デバイスデータ 1004 又は他のデバイスコンテンツは、デバイスの構成設定、デバイス上に格納されたメディアコンテンツ及び/又はデバイスのユーザに関連付けられた情報を含むことができる。デバイス 1000 上に格納されるメディアコンテンツは、任意のタイプのオーディオ、ビデオ及び/又はイメージデータを含むことができる。デバイス 1000 は、1つ又は複数のデータ入力部 1006 を含み、該1つ又は複数のデータ入力部 1006 を介して、ユーザ選択可能な入力、メッセージ、音楽、テレビメディアコンテンツ、記録されたビデオコンテンツ並びに任意のコンテンツ及び/又はデータソースから受信される任意の他のタイプのオーディオ、ビデオ及び/又はイメージデータのような、任意のタイプのデータ、メディアコンテンツ及び/又は入力を受信することができる。

10

#### 【0059】

デバイス 1000 は、通信インタフェース 1008 も含む。通信インタフェース 1008 を、シリアル及び/又はパラレルインタフェース、無線インタフェース、任意のタイプのネットワークインタフェース、モデムのうちのいずれか1つ又は複数として、並びに任意の他のタイプの通信インタフェースとして実装することができる。通信インタフェース 1008 は、デバイス 1000 と、他の電子、コンピューティング及び通信デバイスがデバイス 1000 と通信するための通信ネットワークとの間の接続リンク及び/又は通信リンクを提供する。

#### 【0060】

20

デバイス 1000 は、1つ又は複数のプロセッサ 1010 (例えばマイクロプロセッサ、コントローラ等のいずれか) を含む。プロセッサ 1010 は、様々なコンピュータ実行可能命令又は読取可能命令を実行して、デバイス 1000 の動作を制御し、上述の実施形態を実装する。あるいはまた、デバイス 1000 を、ハードウェア、ファームウェア又は一般に 1012 で示される処理及び制御回路に関連して実装される固定の論理回路のうちのいずれか1つ又は組み合わせで実装することができる。図示されていないが、デバイス 1000 は、デバイス内の様々なコンポーネントを結合するシステムバス又はデータ伝送システムを含むことができる。システムバスは、メモリバス若しくはメモリコントローラ、周辺バス、ユニバーサルシリアルバス、及び/又は様々なバスアーキテクチャのいずれかを使用するプロセッサ若しくはローカルバスのような異なるバス構造のいずれか1つ又は組み合わせを含むことができる。

30

#### 【0061】

デバイス 1000 は、1つ又は複数のメモリコンポーネントのようなコンピュータ読取可能媒体 1014 も含み、その例には、ランダムアクセスメモリ (RAM)、非揮発性メモリ (例えば読取専用メモリ (ROM)、フラッシュメモリ、EPROM、EEPROM 等のいずれか1つ又は複数) 並びにディスク記憶デバイスが含まれる。ディスク記憶デバイスは、ハードディスクドライブ、記録可能及び/又は再書込可能なコンパクトディスク (CD)、任意のタイプのデジタル多用途ディスク (DVD) 等のような、任意のタイプの磁気又は光ストレージデバイスとして実装され得る。デバイス 1000 は、大容量記憶メディアデバイス 1016 も含むことができる。

40

#### 【0062】

コンピュータ読取可能媒体 1015 は、デバイスデータ 1004 だけでなく、様々なデバイスアプリケーション 1018 並びにデバイス 1000 の動作状態様に関係する他のタイプの情報及び/又はデータを記憶する、データ記憶機構を提供する。例えばオペレーティングシステム 1020 を、コンピュータ読取媒体 1014 によるコンピュータアプリケーションとして保持し、プロセッサ 1010 として実行することができる。デバイスアプリケーション 1018 は、デバイスマネージャ (例えば制御アプリケーション、ソフトウェアアプリケーション、信号処理及び制御モジュール、特定のデバイスにネイティブのコード、特定のデバイス用のハードウェア抽象化レイヤ等) だけでなく、ウェブブラウザ、イメージ処理アプリケーション、インスタントメッセージング・アプリケーションのよう

50

な通信アプリケーション、ワード処理アプリケーション及び様々な他の異なるアプリケーションを含むことができる。デバイスアプリケーション 1018 は、本明細書で説明される技術の実施形態を実装する任意のシステムコンポーネント又はモジュールも含む。この例において、デバイスアプリケーション 1018 は、ソフトウェアモジュール及び/又はコンピュータアプリケーションとして示される、インタフェースアプリケーション 1022 及びジェスチャ・キャプチャ・ドライバ 1024 を含む。ジェスチャ・キャプチャ・ドライバ 1024 は、タッチスクリーン、トラックパッド、カメラ等のような、ジェスチャをキャプチャするように構成されるデバイスとのインタフェースを提供するのに使用されるソフトウェアを表す。あるいはまた、インタフェースアプリケーション 1022 及びジェスチャ・キャプチャ・ドライバ 1024 を、ハードウェア、ソフトウェア、ファームウェア又はこれらの任意の組み合わせとして実装することができる。加えて、コンピュータ読取可能媒体 1014 は、上述のように機能するカスケードオペレーションモジュール 1025a 及びジェスチャモジュール 1025b を含むことができる。

10

#### 【0063】

デバイス 1000 は、オーディオ及び/又はビデオ入出力システム 1026 も含む。オーディオ及び/又はビデオ入出力システム 1026 は、オーディオデータをオーディオシステム 1028 に提供し、及び/又はビデオデータをディスプレイシステム 1030 に提供する。オーディオシステム 1028 及び/又はディスプレイシステム 1030 は、オーディオ、ビデオ及びイメージデータを処理し、表示し、又は他の方法でレンダリングする任意のデバイスを含むことができる。ビデオ信号及びオーディオ信号を、デバイス 1000 からオーディオデバイス及び/又はディスプレイデバイスに、RF（無線周波数）リンク、S-video リンク、コンポジットビデオリンク、コンポーネントビデオリンク、DVI（デジタルビデオインタフェース、アナログオーディオ接続又は他の同様の通信リンクを介して通信することができる。一実施形態において、オーディオシステム 1028 及び/又はディスプレイシステム 1030 は、デバイス 1000 の外部コンポーネントとして実装される。あるいは、オーディオシステム 1028 及び/又はディスプレイシステム 1030 は、例示デバイス 1000 の組み込みコンポーネントとして実装される。

20

#### 【0064】

##### < 結論 >

様々な実施形態は、階層型ツリー内の指定された要素のうちの 1 つの要素に関連付けられた入力を受信することに応答して、カスケードオペレーションのサブセットを、それらの指定された要素に選択的に適用することによって、効率的なカスケードオペレーションが実行されることを可能にする。カスケードオペレーションのサブセットを実行した後、様々なパラメータに従ってカスケードオペレーションの完全なセットを実行することができる。

30

#### 【0065】

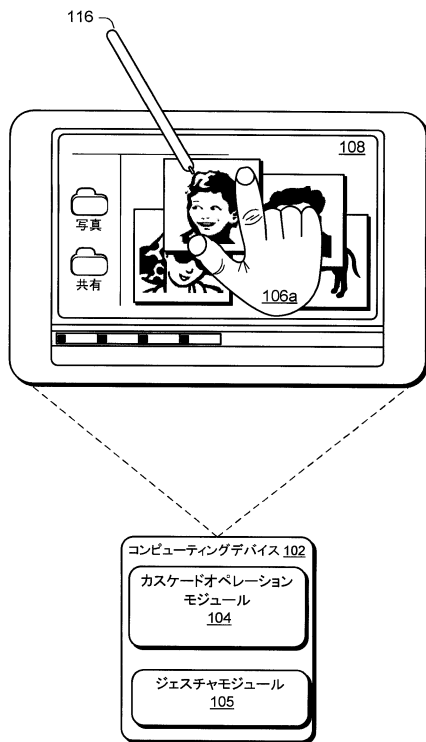
1 つ又は複数の実施形態において、受信した入力について、カスケードオペレーションのサブセットは、限定ではなく例としてほんの数例を挙げると、ユーザ対話のタイミング、所与の要素についてのカスケードオペレーションの複雑性、及び/又はカスケードオペレーションを適用することができる要素の数を含むパラメータに基づいて決定され得る。

40

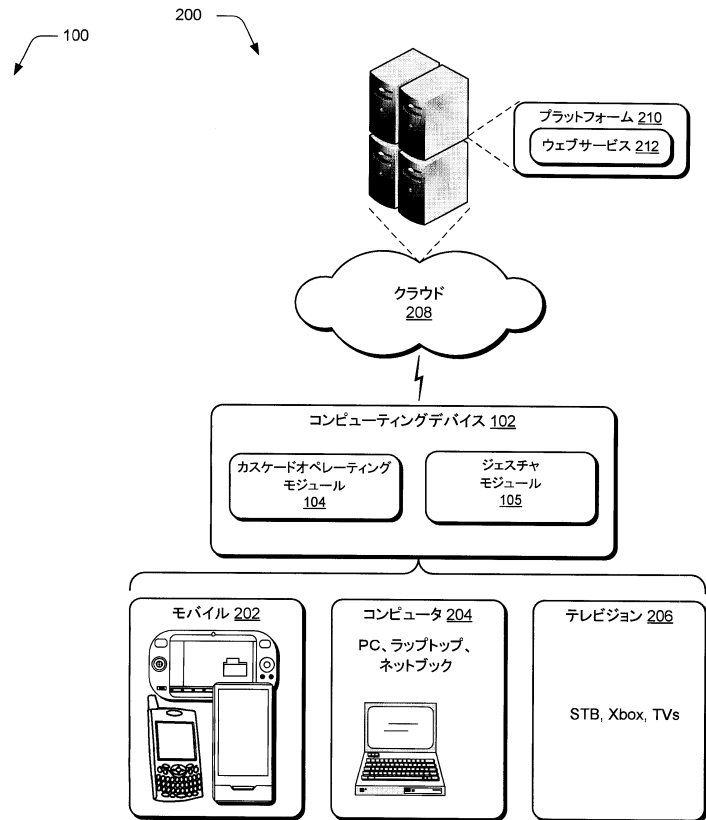
#### 【0066】

実施形態を、構造的特徴及び/又は方法的動作に特有の言葉で説明してきたが、特許請求の範囲において定義される実施形態は、必ずしも説明された具体的な特徴及び動作に限定されないことを理解されたい。むしろ、これらの具体的な特徴及び動作は、特許請求に係る実施形態を実装する例示的な形式として開示される。

【図 1】



【図 2】



【図 3】

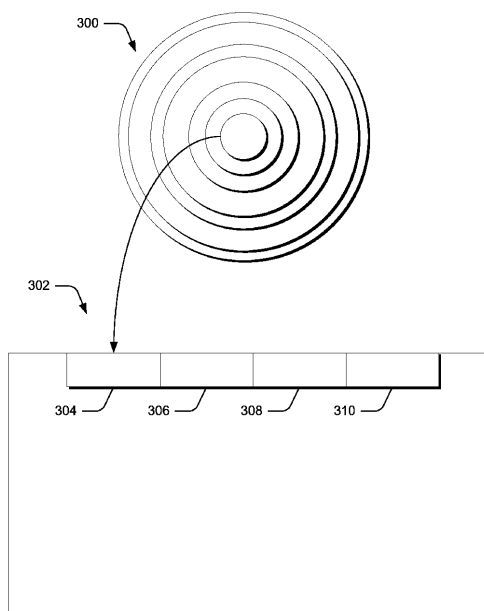


FIG. 3

【図 4】

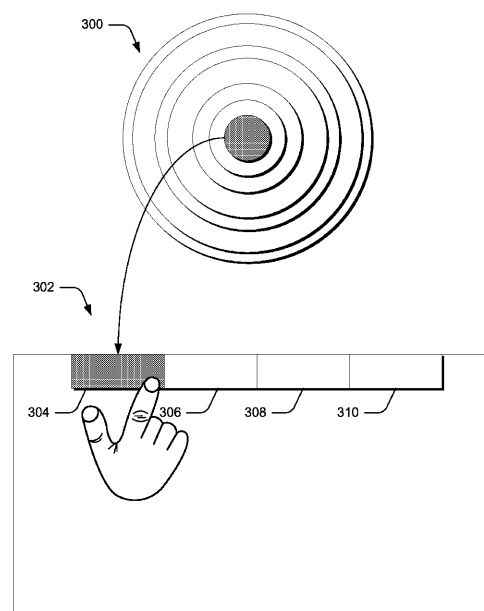


FIG. 4



【図 5】

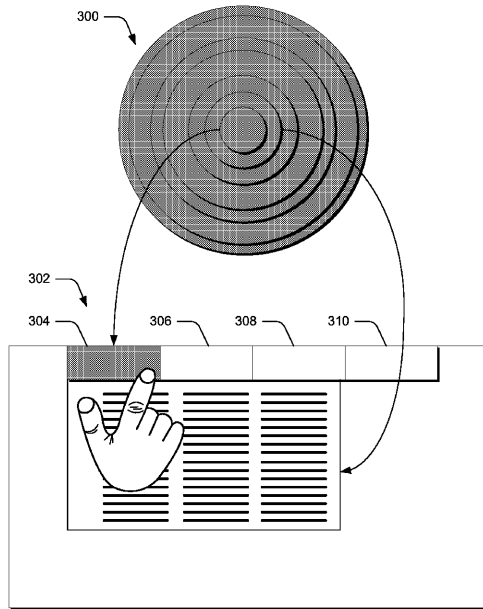
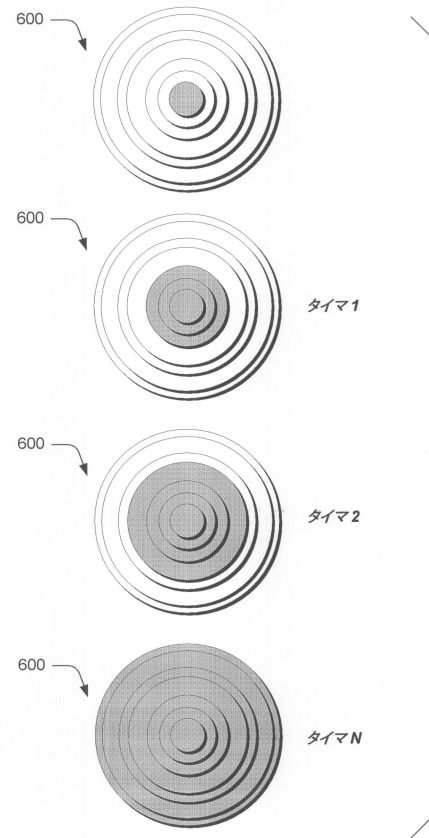
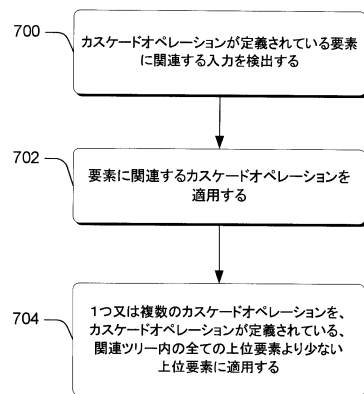


FIG. 5

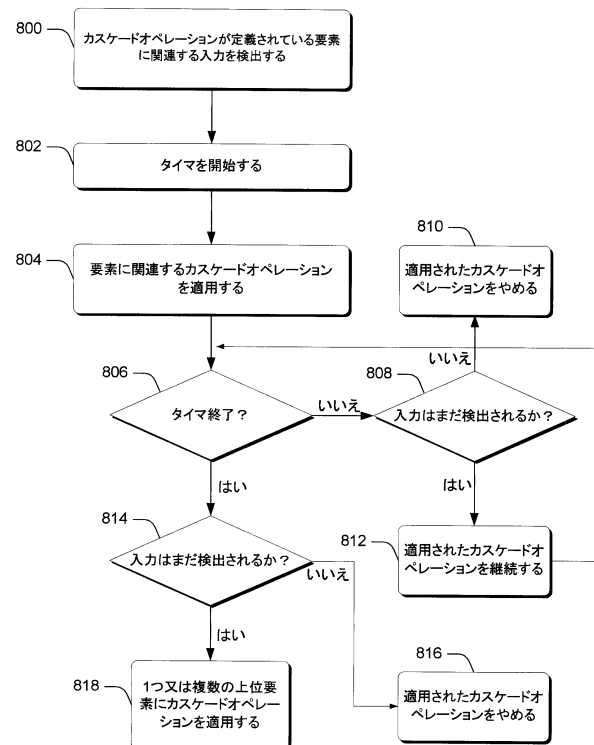
【図 6】



【図 7】



【図 8】



【図 9】

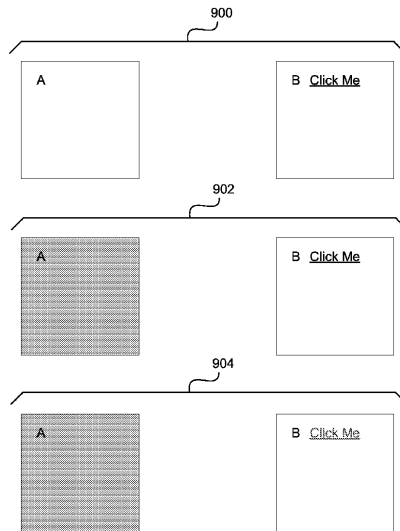
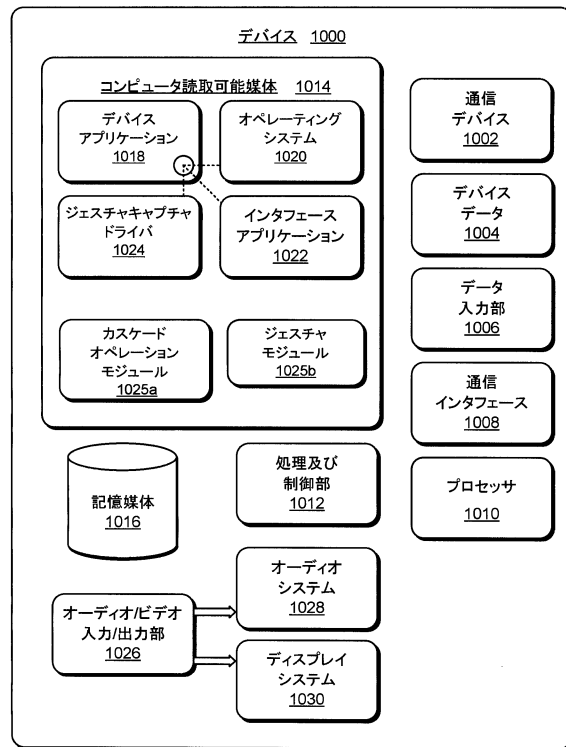


FIG. 9

【図 10】



## フロントページの続き

- (72)発明者 ロッシ, ジェイコブ エス.  
アメリカ合衆国 98052-6399 ワシントン州 レッドモンド ワン マイクロソフト  
ウェイ マイクロソフト コーポレーション エルシーエー - インターナショナル パテンツ 内
- (72)発明者 ロジャース, ジャスティン イー.  
アメリカ合衆国 98052-6399 ワシントン州 レッドモンド ワン マイクロソフト  
ウェイ マイクロソフト コーポレーション エルシーエー - インターナショナル パテンツ 内
- (72)発明者 フルトベングラー, ネイサン ジェイ.イー.  
アメリカ合衆国 98052-6399 ワシントン州 レッドモンド ワン マイクロソフト  
ウェイ マイクロソフト コーポレーション エルシーエー - インターナショナル パテンツ 内

審査官 池田 聡史

- (56)参考文献 特開2011-134212(JP, A)  
米国特許出願公開第2005/0114778(US, A1)  
外間かおり, “6-3 メニューアニメーション”, CSS3アニメーション・ブック for  
iPhone & Android, 日本, 株式会社ラトルズ, 2011年 4月30日, 初版  
, pp.175~185  
樋口泰行, “手のひらサイズでクリエイティブなサイトをつくる! Chapter 03 iPhone  
向けWebサイトをつくる8つの必須ポイント”, web creators, 日本, 株式会社  
エムディエヌコーポレーション, 2008年10月 1日, 第82巻, pp. 156~157  
浦野大輔ほか, “089 タッチパネルはマウスオーバーが使えない”, プロならば知っておく  
べきWebコーディング&デザインの定石100, 日本, 株式会社エムディエヌコーポレーショ  
ン 藤岡 功, 2010年11月 1日, 初版, pp.192~193

- (58)調査した分野(Int.Cl., DB名)  
G06F 3/0488  
JSTPlus(JDreamIII)