

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5753908号
(P5753908)

(45) 発行日 平成27年7月22日 (2015. 7. 22)

(24) 登録日 平成27年5月29日 (2015. 5. 29)

(51) Int. Cl. F I
G O 6 F 9 / 4 4 5 (2 0 0 6 . 0 1) G O 6 F 9 / 0 6 6 5 0 C

請求項の数 18 (全 45 頁)

(21) 出願番号	特願2013-540092 (P2013-540092)	(73) 特許権者	507103802
(86) (22) 出願日	平成23年11月18日 (2011. 11. 18)		グーグル・インコーポレーテッド
(65) 公表番号	特表2013-543202 (P2013-543202A)		アメリカ合衆国・カリフォルニア・940
(43) 公表日	平成25年11月28日 (2013. 11. 28)		43・マウンテン・ビュー・アンフィシア
(86) 国際出願番号	PCT/US2011/061560		ター・パークウェイ・1600
(87) 国際公開番号	W02012/068570	(74) 代理人	100108453
(87) 国際公開日	平成24年5月24日 (2012. 5. 24)		弁理士 村山 靖彦
審査請求日	平成26年4月16日 (2014. 4. 16)	(74) 代理人	100110364
(31) 優先権主張番号	61/415, 243		弁理士 実広 信哉
(32) 優先日	平成22年11月18日 (2010. 11. 18)	(74) 代理人	100133400
(33) 優先権主張国	米国 (US)		弁理士 阿部 達彦
早期審査対象出願		(72) 発明者	ジョン・ニコラス・ジトコフ
前置審査			アメリカ合衆国・カリフォルニア・943
			01・パロ・アルト・エンバーカデロ・ロ
			ード・505

最終頁に続く

(54) 【発明の名称】 コンピューティングオブジェクトのコンテキストの履歴

(57) 【特許請求の範囲】

【請求項1】

コンピュータに実装される方法であって、
 コンピューティングデバイスでオペレーティングシステムのオブジェクトを初めてインスタンス化するステップと、
 前記オペレーティングシステムのオブジェクトがインスタンス化されるときに、前記インスタンス化されたオペレーティングシステムのオブジェクト以外の前記コンピューティングデバイス上で開いているオブジェクトの状態を定義するコンテキストメタデータを自動的に特定するステップと、
 前記特定したコンテキストメタデータを前記オペレーティングシステムのオブジェクトと関連させて記憶するステップと、
 前記オペレーティングシステムのオブジェクトが閉じられているときに、前記オペレーティングシステムのオブジェクトをインスタンス化するその後の要求を受信するステップと、
 前記コンテキストメタデータで特定される副次的なオブジェクトを呼び出すステップとを含み、
 前記コンテキストメタデータは、前記オペレーティングシステムのオブジェクトがインスタンス化されるときに前記オペレーティングシステムでアクティブである1つまたは複数のオブジェクトを特定し、
 前記コンピューティングデバイス上で開いている前記オブジェクトの状態を定義する前

10

20

記コンテキストメタデータが、前記オペレーティングシステムのオブジェクトがインスタンス化されている間に前記コンピューティングデバイスを使用する1人または複数の人を特定する

方法。

【請求項2】

副次的なオブジェクトを呼び出すステップが、前記オペレーティングシステムのオブジェクトが最初にインスタンス化されたときに開いていた1つまたは複数のアプリケーションを起動するステップを含む請求項1に記載の方法。

【請求項3】

前記その後の要求が受信されるときにアクティブであるオブジェクトに対応する情報を、記憶されたコンテキストメタデータに追加するステップをさらに含む請求項1に記載の方法。

10

【請求項4】

前記オペレーティングシステムのオブジェクトがインスタンス化されている間に前記オペレーティングシステムのオブジェクト以外のオブジェクトとのユーザのインタラクションを特定するステップと、前記ユーザのインタラクションについての情報を記憶されたコンテキストメタデータに追加するステップとをさらに含む請求項1に記載の方法。

【請求項5】

前記オペレーティングシステムのオブジェクトが閉じられているときに、前記オペレーティングシステムのオブジェクトをインスタンス化するその後の要求を受信するステップと、前記コンテキストメタデータで特定される副次的なオブジェクトを呼び出すステップとをさらに含む請求項1に記載の方法。

20

【請求項6】

前記オペレーティングシステムのオブジェクトが閉じられているときに、前記コンテキストメタデータで特定される1人または複数の人が存在するという指示を受信するステップと、前記指示を受信するステップにตอบสนองして、前記オペレーティングシステムのオブジェクトをインスタンス化するステップとをさらに含む請求項1に記載の方法。

【請求項7】

前記オペレーティングシステムのオブジェクトが閉じられているときに、前記コンテキストメタデータで特定される2人以上の人が存在するという指示を受信するステップと、前記指示を受信するステップにตอบสนองして、前記オペレーティングシステムのオブジェクトをインスタンス化するステップとをさらに含む請求項1に記載の方法。

30

【請求項8】

コンピュータに実装されるシステムであって、
コンピュータのオペレーティングシステムと、
前記コンピュータのオペレーティングシステムによって定義され、前記オペレーティングシステムを実行するデバイスのユーザがアクセス可能な複数のオブジェクトと、

前記オブジェクトのそれぞれに関するコンテキストメタデータの複数のインスタンスを含むコンテキスト情報のデータストアとを備え、

コンテキストメタデータの各インスタンスが、前記オブジェクトのうちの1つまたは複数に相互に関連付けられ、

40

コンテキストメタデータの各インスタンスが、前記コンテキストメタデータに相互に関連付けられるオブジェクトが前記オペレーティングシステムで最初にインスタンス化されたときに前記オペレーティングシステムでアクティブであった1つまたは複数のオブジェクトを特定し、

前記コンピュータが、前記オペレーティングシステムのオブジェクトが閉じられているときに、前記オペレーティングシステムのオブジェクトをインスタンス化するその後の要求を受信された場合、前記コンテキストメタデータで特定される副次的なオブジェクトを呼び出し、

コンテキストメタデータの各インスタンスが、前記オペレーティングシステムのオブジ

50

エクトがインスタンス化されている間に前記デバイスを使用する1人または複数の人を特定する システム。

【請求項 9】

前記コンテキストメタデータの1つまたは複数のインスタンスが、前記コンテキストメタデータの前記1つまたは複数のインスタンスに相互に関連付けられる前記オブジェクトが、前記オブジェクトの最初のインスタンス化の後、前記オペレーティングシステムでインスタンス化されたときにアクティブであった1つまたは複数のオブジェクトを特定する請求項8に記載のシステム。

【請求項 10】

前記コンテキストメタデータの1つまたは複数のインスタンスが、コンテキストメタデータの前記1つまたは複数のインスタンスに相互に関連付けられるオブジェクトが前記オペレーティングシステムでアクティブであったときのある期間中の、コンテキストメタデータの前記1つまたは複数に相互に関連付けられる前記オブジェクト以外のオブジェクトに関して行われた動作を特定する請求項8に記載のシステム。

【請求項 11】

コンピュータに実装されるシステムであって、
 コンピュータのオペレーティングシステムと、
 前記コンピュータのオペレーティングシステムによって定義され、前記オペレーティングシステムを実行するデバイスのユーザがアクセス可能な複数のオブジェクトと、
 コンテキストメタデータのインスタンスを前記複数のオブジェクトのそれぞれと相互に関連付けるための手段と、

前記オペレーティングシステムのオブジェクトが閉じられているときに、前記オペレーティングシステムのオブジェクトをインスタンス化するその後の要求が受信された場合、前記コンテキストメタデータで特定される副次的なオブジェクトを呼び出す手段とを備え、

コンテキストメタデータの各インスタンスが、前記コンテキストメタデータに相互に関連付けられるオブジェクトが前記オペレーティングシステムで最初にインスタンス化されたときに前記オペレーティングシステムでアクティブであった1つまたは複数のオブジェクトを特定し、

コンテキストメタデータの各インスタンスが、前記オペレーティングシステムのオブジェクトがインスタンス化されている間に前記デバイスを使用する1人または複数の人を特定する

システム。

【請求項 12】

コンピュータに実装される方法であって、
 コンピューティングデバイス上で開いている第1のオペレーティングシステムのオブジェクトに対する第1の事前に定義された動作の第1の発生を記録するステップと、
 前記第1のオペレーティングシステムのオブジェクトと同時に前記コンピューティングデバイス上で開いている第2のオペレーティングシステムのオブジェクトに対する第2の事前に定義された動作の第1の発生を記録するステップと、

前記第1の事前に定義された動作と前記第2の事前に定義された動作とを、同時に発生したものとして相互に関連付けるステップと、

前記第1のオペレーティングシステムのオブジェクトおよび前記第2のオペレーティングシステムのオブジェクトに関連する相互に関連付けられた動作を含む相互に関連付けられた状態を記憶するステップと、

相互に関連付けられた状態を記憶するためのデータベースを保守するステップと、

前記第1のオペレーティングシステムのオブジェクトに対する前記第1の事前に定義された動作の第2の発生を特定するステップを含む、前記コンピューティングデバイスで動作を監視するステップと、

前記データベースから前記記憶された相互に関連付けられた状態を取り出すステップと

10

20

30

40

50

前記第1の事前に定義された動作の前記第2の発生を特定することに応答して前記第2のオペレーティングシステムのオブジェクトの状態を前記第2の事前に定義された動作の状態に変更するステップと

を含み、

相互に関連付けられた状態が、オペレーティングシステムのオブジェクトがインスタンス化されている間に前記コンピューティングデバイスを使用する1人または複数の人を特定する方法。

【請求項13】

前記第2のオペレーティングシステムのオブジェクトの状態を前記第2の事前に定義された動作の状態に変更するステップは、前記第2のオペレーティングシステムのオブジェクトをインスタンス化するステップを含む請求項12に記載の方法。

【請求項14】

前記第1のオペレーティングシステムのオブジェクトが閉じられているときに前記第1のオペレーティングシステムのオブジェクトをインスタンス化するその後の要求を受信するステップと、

前記第1のオペレーティングシステムのオブジェクトをインスタンス化する前記要求を、前記データベース内の相互に関連付けられた状態と比較するステップと

をさらに含む請求項12に記載の方法。

【請求項15】

前記比較するステップに基づいて、前記第2のオペレーティングシステムのオブジェクトを、前記第1のオペレーティングシステムのオブジェクトに関連するものとして特定するステップと、

前記第2のオペレーティングシステムのオブジェクトをインスタンス化するステップとをさらに含む請求項14に記載の方法。

【請求項16】

第2のオペレーティングシステムのオブジェクトに対する前記第2の事前に定義された動作が、前記第2のオペレーティングシステムのオブジェクトを閉じる要求である請求項12に記載の方法。

【請求項17】

前記第2のオペレーティングシステムのオブジェクトの状態を前記第2の事前に定義された動作の状態に変更するステップは、前記第2のオペレーティングシステムのオブジェクトが開いている場合に、前記第2のオペレーティングシステムのオブジェクトを閉じるステップ

を含む請求項16に記載の方法。

【請求項18】

前記第2のオペレーティングシステムのオブジェクトをインスタンス化するステップの前に、前記第2のオペレーティングシステムのオブジェクトをインスタンス化するためのユーザの許可を要求するステップをさらに含む請求項13に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照

本出願は、2010年11月18日に出願した「User Interaction in a Computer Operating System」と題した米国仮出願第61/415,243号の優先権を主張するものであり、この仮出願の内容は、本明細書にすべて記載されているかのように参照により組み込まれる。

【0002】

本明細書は、コンピュータのオペレーティングシステムのユーザとのインタラクションのためのシステムおよび技法に関する。

10

20

30

40

50

【背景技術】

【0003】

コンピュータ(例えば、デスクトップPC、ラップトップ、ネットブック、タブレット、またはスマートフォン)が動作する中核的な構造は、基本入出力システム、すなわちBIOS、オペレーティングシステム、およびコンピュータを動作させるための基本的機能を提供するソフトウェアの「スタック(stack)」を占有すると見なされ得るその他のコンポーネントを含み得る。BIOSは、コンピュータが最初に電源を入れられ、ブートされるときにアクセスされ得、コンピュータをブートするためにさらに必要とされるコードを記憶するストレージデバイスを含むシステムのデバイスの特定、テスト、および初期化などのコンピュータに関する基本的な機能を提供することができる。オペレーティングシステムに関するコードは、そのような追加的なデバイスに記憶され得、コンピュータが完全にブートされ、動作する準備が整うまでコンピュータのさらなるブートを行うことができる。

10

【0004】

一般に、オペレーティングシステムは、コンピュータのハードウェアとコンピュータのユーザとの間、およびハードウェアとユーザがコンピュータにロードするアプリケーションとの間のインターフェースとして働く。オペレーティングシステムは、さまざまな機能を提供することができる。例えば、オペレーティングシステムは、コンピュータのユーザがコンピュータからの出力を受け取ることができ、コンピュータに入力を与えることができるグラフィカルユーザインターフェース(GUI)を提供することができる。また、オペレーティングシステムは、さまざまなサードパーティアプリケーションが実行されるプラットフォームを提供することができ、オペレーティングシステムは、アプリケーションによって必要とされるサービスを提供し、さらに、アプリケーションがその他のアプリケーション、周辺機器(例えば、プリンタ、カメラなど)などのその他のリソースと、およびオペレーティングシステム自体が提供するサービスと通信することができる機構を提供する。

20

【発明の概要】

【課題を解決するための手段】

【0005】

本明細書は、コンピューティングデバイスのためのオペレーティングシステムの一部として、またはいくつかの異なるコンピューティングデバイスを含むシステムで実装され得るシステムおよび技法を記載する。例えば、さまざまな機構が、コンピューティングデバイスをクラウドベースのシステムに記憶されたデータと同期させるために使用可能であり、それによって、ホストされるコンピュータサーバシステムが、システムにアクセスするために一般のメンバーが利用することができるようにされ、ひいてはシステムは、データの記憶およびバックアップ、文書の記憶、電子メールのルーティングおよび処理、ならびにその他の有用なサービスなどのさまざまなサービスを提供する。コンピューティングデバイスは、比較的小さなローカルストレージを有し、したがって、ユーザのデータをホストされるサーバシステムに記憶する形態のものであることができる。さらに、デバイスは、基本的に常にネットワーク(例えば、ワイヤレスネットワーク)に接続され、そのネットワークを介してインターネットに接続されているように構成され得る。その結果、デバイスのさまざまなコンポーネントが、ほとんど常時オンである手法によって動作するように構成され得る。

30

40

【0006】

例えば、以下で説明される特定の実装形態においては、コンピュータが、どのオブジェクト(例えば、アプリケーション、プログラム、ファイル)が同時に使用されているのかを記録するためにさまざまな時点で「スナップショット」を取ることができる。これらの「スナップショット」から、ユーザがそのユーザのコンピュータで特定のオブジェクトを同時に開いておくことを好むという推定がなされ得る。例えば、ユーザは、ワードプロセッサ文書に対して作業しているときに音楽を聴くことを好む可能性がある。したがって、ユーザがワードプロセッサ文書を開くとき、コンピュータは、音楽プログラムとワードプロセッサ文書の両方が同時に開いている多くの「スナップショット」が取得されたことを認

50

識する。別の例において、ユーザは、特定のワードプロセッサ文書に対して作業している可能性がある。前に記録された「スナップショット」は、ワードプロセッサ文書が作成されるときに、持続的検索(persistent search)オブジェクトもまた、このコンピュータにおいてアクティブであったことを示し得る。そのような合同作用は、文書が、持続的検索によって生成された情報を含むようにユーザによって生成されたことを示す可能性がある。例えば、ユーザが情報を持続的検索の結果からクリップボードにコピーし、文書を作成し、次いで、検索の結果またはその他のコピーされたデータを文書にペーストした場合、その点に関してさらに強い推定がなされ得る。この情報も、ワードプロセッサ文書に関連してメタデータとして記憶され得る。

【0007】

より具体的には、電子的なレコーダが、コンピューティングデバイスのオペレーティングシステムの一部として実装可能である。レコーダは、デバイス上のプロセスを監視することができ、デバイスにログオンするための関連する活動と見なされる活動のリストを調べるようにプログラムされ得る。例えば、デバイスにおける任意のアプリケーションのインスタンス化、メディアファイルのロード(例えば、歌もしくは動画の再生)、またはチャットセッションなどの通信セッションの開始、およびその他のそのようなイベントが、レコーダにより調べることができる。レコーダは、関連するイベントを認識するとき、その他のどのアプリケーションが開いているか、どのウェブページが現在ロードされているか、その他のどのユーザが現在の通信相手であるかなどを記録するなど、デバイスの現在の状態のスナップショットを取得することができる。レコーダは、現在の状態についてのそのような点をログに残すことが可能であり、その他のそのようなイベントがシステム上で認識されるときに繰り返しそうすることができる。また、実質的に同時に、そのようなイベントが発生するときに、システムは、同じまたは同様のイベントが起きた前のインスタンスのログを検索することができ、それらのイベントが起きたときのデバイスの状態の詳細を特定することができる。そのとき、システムは、それらの詳細の一部を自動的にインスタンス化することができる(例えば、前に開いていたアプリケーションを開き、前に開いていたウェブページまたはその他のコンテンツのソースを開き、前に会話していたユーザに連絡する)。あるいは、オペレーティングシステムは、前の状態のそのような観点のリストを表示することができ、ユーザは、それらの観点のうちの1つまたは複数を選択することができる、次いで、システムは、選択された観点をインスタンス化することができる。

【0008】

特定の実装形態において、そのようなシステムおよび技法は、1つまたは複数の利点を提供することができる。例えば、コンピュータシステムとのユーザのインタラクションは、ユーザの好みを送るようにユーザに明示的に依頼することなく、コンピュータシステムを使用しているときのユーザの個人的な好みに関するより多くの情報をもたらすことが可能である。その代わりに、そのような好みは、行動の傾向から推定され得る。加えて、コンピュータが、ユーザが特定のオブジェクトを同時に開いておくことを好むことを認識し、ユーザが第1のオブジェクトを開いた後にそのユーザのためにその他のオブジェクトを自動的に開くので、ユーザは、複数のオブジェクトの代わりに1つのオブジェクトをクリックして開くことしか必要としないので時間を節約することができる。

【0009】

一部の実施形態においては、コンピュータにより実装されるオブジェクト追跡方法が、コンピューティングデバイスでオペレーティングシステムのオブジェクトを初めてインスタンス化するステップと、オペレーティングシステムのオブジェクトがインスタンス化されるときに、インスタンス化されるオペレーティングシステムのオブジェクト以外のコンピューティングデバイス上で開いているオブジェクトの状態を定義するコンテキストメタデータ(contextual meta data)を自動的に特定するステップと、特定されたコンテキストメタデータをオペレーティングシステムのオブジェクトと関連させて記憶するステップとを含み、コンテキストメタデータは、オペレーティングシステムのオブジェクトがインス

10

20

30

40

50

タンス化されるときにオペレーティングシステムでアクティブである1つまたは複数のオブジェクトを特定する。一部の実装形態は、オペレーティングシステムのオブジェクトが閉じられているときに、オペレーティングシステムのオブジェクトをインスタンス化するその後の要求を受信するステップと、コンテキストメタデータで特定される副次的なオブジェクトを呼び出すステップとをさらに含む。

【0010】

場合によっては、副次的なオブジェクトを呼び出すステップは、オペレーティングシステムのオブジェクトが最初にインスタンス化されたときに開いていた1つまたは複数のアプリケーションを起動するステップを含む。方法は、その後の要求が受信されるときにアクティブであるオブジェクトに対応する情報を、記憶されたコンテキストメタデータに追加するステップをさらに含み得る。さらに、方法は、オペレーティングシステムのオブジェクトがインスタンス化されている間にオペレーティングシステムのオブジェクト以外のオブジェクトとのユーザのインタラクションを特定するステップと、ユーザのインタラクションについての情報を、記憶されたコンテキストメタデータに追加するステップとをさらに含み得る。

10

【0011】

1つまたは複数の実施形態の詳細が、添付の図面および以下の説明に記載されている。その他の特徴および利点は、それらの説明および図面から、ならびに特許請求の範囲から明らかになるであろう。

【図面の簡単な説明】

20

【0012】

【図1】コンピューティングデバイスの遅らせたロックを行うためのプロセスの概念図である。

【図2A】コンテキストオブジェクト(contextual object)を使用するオペレーティングシステムの概念図である。

【図2B】コンテキストオブジェクトを使用するオペレーティングシステムの代替的な実施形態の概念図である。

【図3】コンピューティングデバイスのメモリの制御を維持するためのシステムの概念図である。

【図4】コンピュータプロセス間のメッセージの受渡しを伴うスレッドアフィニティ(thread affinity)を提供するシステムの概念図である。

30

【図5】ステートレス環境で状態情報を提供するシステムの概念図である。

【図6】ネットワークを介してコンピューティングデバイスのイメージを取得するシステムの概念図である。

【図7】コンピューティングデバイスのリモートの監視および制御を行うシステムの概念図である。

【図8】ホストされるコンピュータシステムに集中的に記憶されたデータのコンピューティングデバイスへのキャッシュを行うためのシステムの概念図である。

【図9】コンピューティングデバイスの遅らせたロックを行うためのプロセスの流れ図である。

40

【図10A】オペレーティングシステムでコンテキストオブジェクトを管理するためのプロセスの流れ図である。

【図10B】オペレーティングシステムでコンテキストオブジェクトを管理するための代替的なプロセスの流れ図である。

【図11】コンピューティングデバイスでメモリの制御を維持するためのプロセスの流れ図である。

【図12】コンピュータプロセス間のメッセージの受渡しを伴うスレッドアフィニティを提供するためのプロセスの流れ図である。

【図13】ステートレス環境で状態情報を提供するためのプロセスの流れ図である。

【図14】ネットワークを介してコンピューティングデバイスのイメージを取得するプロ

50

セスの流れ図である。

【図15】コンピューティングデバイスのリモートの監視および制御を行うためのプロセスの流れ図である。

【図16】ホストされるコンピュータシステムに集中的に記憶されるデータのコンピューティングデバイスへのキャッシュを行うためのプロセスの流れ図である。

【図17】本明細書に記載の技法を実装するために使用され得るコンピュータデバイスおよびモバイルコンピュータデバイスの例を示す図である。

【発明を実施するための形態】

【0013】

さまざまな図面における同様の参照符号は、同様の要素を示す。

10

【0014】

本明細書は、例えば、デバイスで動作するオペレーティングシステムの要素を使用してモバイルスマートフォンなどのコンピューティングデバイスのユーザとのインタラクションを行うためのシステムおよび技法を説明する。システムおよび技法は、1つまたは複数のホストされるサーバシステムと通信し、そのようなサーバシステムからサービスを提供され得るさまざまなオペレーティングシステムのコンポーネントを提供することができる。特に、オペレーティングシステムは、実行されている時間のほとんどでインターネットに接続しながら実行されるように設計され得る。その結果、オペレーティングシステムによって実行される動作のうちの多くは、ネットワーク接続が利用可能であることを想定するように、およびネットワーク接続が回復することができるまでキャッシュ技法またはその

20

【0015】

図1は、コンピューティングデバイスの遅らせたロックを行うためのシステムのプロットである。概して、システムは、ポータブルコンピューティングデバイスのユーザが、デバイスを直ちにシャットダウンすることなくポータブルコンピューティングデバイスをシャットダウンする措置を取ることを可能にするための機構を提供する。その結果、ユーザがそのような措置を取った後ですぐに考えを変える場合、ユーザは、この特徴がなければ、ユーザがそのような措置を取ったら直ちにデバイスがシャットダウンされることになっていたならば別途必要とされたであろうその他のステップを行うことを必要とせずデバイス

30

【0016】

この図においては、ポータブルコンピューティングデバイス、この場合はクラムシェル設計のラップトップコンピュータの3つの異なる状態が、示されている。第1の段階102において、コンピュータは開き、動作しており、時間は昼の12時である。コンピュータのディスプレイ108は、ロックタイマーが10秒に設定されたことを示す。本明細書においては、ディスプレイ108が例として与えられるが、実際の使用においては、デバイスは、タイマーの残りの量を表示せず、タイマーは、状態102によって示された時点で動作を開始して

40

【0017】

状態104において、5秒後に、ディスプレイがコンピュータの本体に向かって閉じられ、通常、ハイバネーションまたはその他の形態の非アクティブモードへの移行をコンピュータに直ちに開始させる措置である。そのような変化は、コンピュータのマイクロプロセッサを停止することと、ディスプレイ108をオフにすることと、冷却ファンおよびその他の関連する機構をオフにすることと、グラフィックス処理ユニット(GPU)を停止することと、デバイスでその他の省電力技法を実行することを含む。状態104によって示されていないが、この例においては、デバイスがシャットダウンシーケンスをさらに始める前に10秒の固有の遅延を有する(およびさらに、ロック解除パスワードを入力することなどの実

50

質的なユーザの関与なしに再アクティブ化され得る時点をデバイスが越えるまで、そのシーケンスが始まった後、いくらかの未確定の時間が存在する可能性がある)ので、デバイスは、ほんの数秒閉じられたときには完全に停止されていないか、または停止し始めている。その代わりに、デバイスが停止されているという印象をユーザに与えるように、例えば、ディスプレイ108のスイッチを切ることによって、限られた数の特徴が停止され得る。しかし、デバイスを再アクティブ化するのに、この時点では、デバイスを再び開くこと以外、何も必要としないこともあり得る。

【 0 0 1 8 】

14秒後かつデバイスを閉じた後9秒後に、状態106において、例えば、デバイスのユーザがデバイスの使用を中断することを望まず、むしろ、デバイスを用いてさらに作業を行う必要があると判断したために、デバイスが開かれる。タイマーは10秒後に切れるように設定されたので、デバイスは、まだそのデバイスのハイバネーションまたはその他の停止された状態に遷移していない。その代わりに、再び起動するために長い時間がかかるデバイスのすべてのシステムは、起動されたままであり、ディスプレイ108などの限られたシステムのみが、停止されている。また、この例においては、タイマーが再び10秒にリセットされており、したがって、ユーザがデバイスを再び閉じる場合、デバイスは、閉じた後の10秒間はハイバネーションまたはその他の非アクティブモードへの移行を開始しない。さらに特定の実施形態において、デバイスは、ある入力にตอบสนองしてスリープに直ちに移行し、別の入力にはตอบสนองしないことができる。例えば、制御キーの組合せの入力が、デバイスをスリープに直ちに移行させることが可能であり、そのような組合せの入力は、ユーザによる意図がよりはっきりしていると考えられてもよい。対照的に、クラムシェルデバイスを閉じることは、そのような動作が(例えば、ある場所から次の場所にデバイスを移動させるときの)一時的な動作として意図されている見込みがより高い可能性があるので、所定の遅延を実施することができる。

【 0 0 1 9 】

デバイスを停止するプロセスを開始するための特定の遅延時間は、デバイスのユーザによって設定され得る。例えば、ユーザがバッテリーを最大限節約することを望んでおらず、頻繁にそのユーザのコンピュータを閉じ、次いで、その後直ちにまたはすぐにそのユーザのコンピュータで再び作業し始めたいと決断する場合、ユーザは、デバイスがスリープまたはハイバネーションモードに移行するためにそのデバイスのさまざまなシステムを停止し始める前にタイマーがカウントダウンする比較的長い期間を設定することができる。

【 0 0 2 0 】

本明細書において、デバイスは、クラムシェルが閉じているときにスイッチが押下され得、コンピュータが停止モードに移行することを決定できるようにコンピュータのヒンジの近くにスイッチを含むクラムシェル構成を有するものとして示されているが、その他の実装形態も使用され得る。例えば、フラットタッチスクリーンタブレットまたはスレートコンピュータは、デバイスの表面、裏面、または周囲の縁の電源ボタンを押すことによってユーザがそのフラットタッチスクリーンタブレットまたはスレートコンピュータを停止することを可能にすることができる。そのような動作は、デバイスが完全に停止しようとしているという印象をユーザに与えるようにデバイスのディスプレイを直ちにオフにすることができる。しかし、マイクロプロセッサ、メモリコントローラ、グラフィカル処理ユニット、およびその他のそのようなサブシステムなどのデバイスのその他のサブシステムは、上で検討されたように、タイムアウト遅延の継続時間の間はそのままの状態を保つことができる。

【 0 0 2 1 】

システムを停止すること以外に、タイマーは、図示されたコンピューティングデバイスのようなコンピューティングデバイスでセキュリティ装置の発動がリセットされるのを遅らせるために使用され得る。例えば、特定のコンピューティングデバイスは、スリープモードまたはその他の非アクティブモードにされるとき、デバイスを完全なアクティブモードに戻すためにユーザによってパスワードまたはその他のセキュリティ機構が必要とされ

10

20

30

40

50

るように構成され得る。本明細書において検討されるタイマーがないと、ユーザは、それらのユーザのデバイスを閉じるか、またはその他の方法でそのデバイスを非アクティブ化し、それからすぐに、やはりデバイスを使うことを必要とすることを思い出す場合に、それらのユーザのパスワードを再入力することを強られる可能性がある。本明細書において検討されるタイマーの実装形態などのタイマーがあれば、ユーザは、それらのユーザのデバイスを非アクティブ化するための何らかの行為を行った後すぐにボタンを押すかまたはそのデバイスを開くことができ、それらのユーザがデバイスのロックを解除するためにそれらのユーザのパスワードまたはその他の証明情報を再入力することを必要とせず、ホームページまたはデバイスのデスクトップをほとんどすぐにそれらのユーザに対して表示させることができる。要するに、デバイスにおけるロック解除状態からのロック状態への遷移は、決められた、ユーザが選択可能な期間だけ意図的に遅らされること可能である。

10

【0022】

このようにして、本明細書に記載のシステムおよび方法は、コンピューティングデバイスによるより使いやすいユーザエクスペリエンスをもたらすことができる。デバイスは、そのデバイスの停止を、ユーザが停止のコマンドを示した少し後に遅らせることができ、そのような遅延は、ユーザがすぐに考えを変える場合にデバイスが素早く起動することを可能にするために使用され得る。同時に、停止の遅延は、デバイスのために余計なバッテリー電力が使われないように比較的短い場合もある。

【0023】

20

図2Aは、コンテキストオブジェクトを使用するオペレーティングシステムの概念図である。概して、コンテキストオブジェクトは、オペレーティングシステムで第1のオブジェクトが生成され、かつ/またはアクティブであったときにオペレーティングシステムに存在し、アクティブであったその他のオブジェクトを記載することによるなどして、オブジェクトがオペレーティングシステムに前に存在したコンテキストについての情報をそのオブジェクトとともに、またはそのオブジェクトに関して記憶したオペレーティングシステムのオブジェクトである。例えば、以下で説明される特定の実装形態においては、コンピュータが、どのオブジェクト(例えば、アプリケーション、プログラム、ファイル)が同時に使用されているのかを記録するためにさまざまな時点で「スナップショット」を取ることができる。これらの「スナップショット」から、ユーザがそのユーザのコンピュータで特定のオブジェクトを同時に開いておくことを好むという推定がなされ得る。

30

【0024】

図において、コンピュータ202は、基本的なクラムシェルラップトップコンピュータの形態で示されているが、コンピュータ202は、スマートフォン、タッチスクリーンタブレット、またはスレートデバイスなどのその他の形態を取ってもよい。オブジェクトのいくつかの視覚的な表示がコンピュータ202のスクリーン上に示され、コンピュータ202にロードされたアプリケーションのアイコンおよび表示を含む。例えば、アイコン208は、コンピュータ202、またはコンピュータ202からアクセス可能なサーバシステムの永続的なストレージに記憶されることなどによってコンピュータ202からアクセス可能な特定のワードプロセッサ文書を表す。例えば、コンピュータ202は、コンピュータ202がそのコンピュータ202自体をサーバシステムに対して特定して、コンピュータ202のユーザに対して登録されたサーバシステムでのアカウントを示すことができるクッキーまたはその他の機構を記憶することが可能である。そのような機構は、アイコン208の文書を表すデータを取得するためなど、サーバシステムから情報を得るためにコンピュータ202によって使用され得る。次いで、ユーザは、アイコン208を選択することによって文書呼び出すことができる。

40

【0025】

ウィジェットまたはガジェット212が、コンピュータ202のディスプレイ上の時計によって表され、コンピュータ202上にやはり表示され得るオブジェクトの種類を示す。ウィジェットまたはガジェット212は、さまざまな同様の形態を取ってもよく、デバイス202に口

50

ードされたオペレーティングシステムで異なる形で利用可能な機能を補足するためのアプリケーションを立案するサードパーティからのコードとして提供され得る。ニュースフィード214は、最近の新しい出来事の情報コンピュータ202のユーザに示すニュースアグリゲータ(news aggregator)の形態でデバイス202で実行されているアクティブなアプリケーションを表す。

【0026】

アイコン204は、コンピュータ202によって実行されるべき持続的検索の形態のオブジェクトを表す。持続検索は、コンピュータ202などのデバイスによって自動的に繰り返される検索である。例えば、ヨーロッパでの休暇を計画しているユーザは、フライトが特定の価格で買えるようになった場合に直ちに通知され得るようにヨーロッパへの航空会社のフライトの持続的検索を定めることができる。

10

【0027】

ボックス206、210は、オブジェクト208および204のうちの1つに関連して保存され得るコンテキストデータを示す。例えば、ボックス210は、文書または文書208などのワードプロセッサ文書に関するコンテキスト情報を示す。オブジェクトとともに保存され得るコンテキスト情報の種類を示すさまざまなフィールドがボックス210に示される。例えば、オブジェクトは、そのオブジェクトのオブジェクト種類を示す名前と、コンピュータ202に記憶されたすべてのその他のオブジェクトに対してこのオブジェクトを一意に特定する識別番号とを含む。

【0028】

20

また、ボックス210は、オブジェクトが最初に生成された時間と、オブジェクト208が生成されたときにコンピュータ202でやはり開いていたオブジェクトのリストとを示す。加えて、ボックス210は、オブジェクト208が最後に使用されたときに開いていたその他のオブジェクトのリストを含む。この例において、ユーザは、ワードプロセッサオブジェクト208が生成されたときと、そのオブジェクト208が最後に使用されたときの両方で、ウェブサイトwww.test.comを見ていた。また、ワードプロセッサ文書が生成されたとき、持続的検索オブジェクト204が、コンピュータ202でアクティブであった。そのような合同作用は、文書が、持続的検索によって生成された情報を含むようにユーザによって生成されたことを示すことができる。例えば、ユーザがその情報を持続的検索の結果からクリップボードにコピーし、文書を生成し、次いで、検索の結果またはその他のコピーされたデータを文書にペーストした場合、その点に関してさらに強い推定がなされ得る。そのような情報も、(例えば、文書が最後に保存されるときに、そのとき開いているその他のアプリケーションによって文書内のどの情報が共有されているかを判定し、それによって、文書とオブジェクトとの間でコピーされたコンテンツを示すことによって)ボックス210に示されるようにオブジェクト208に関連して記憶され得る。

30

【0029】

ボックス210は、文書オブジェクト208がインタラクションしたオブジェクトも示す。この例において、文書208は、アプリケーションウェブプリンタ(application web printer)とインタラクションし、そのことは、文書208がそのプリンタで印刷したことを示し得る。オブジェクト間の特定の直接的なインタラクションが記憶され得るのは、それらのインタラクションがオブジェクトとその他のオブジェクトとの間の特に強い接続を示す可能性があるからである。そのような接続は、特定の状況においてオブジェクトの意図されたユーザであるユーザを特定するために使用され得る。

40

【0030】

ボックス206は、オブジェクト204に関連して記憶されたコンテキスト情報を示す。やはり、名前を付けられたオブジェクトが、一意の識別番号を、オブジェクトが生成された時間とともに含む。ボックス206によって示されるように、持続的検索オブジェクト204が生成されたとき、4つの異なるその他のオブジェクトがアクティブであった。それらのその他のオブジェクトは、ワードプロセッサ文書、表計算文書、および持続的検索オブジェクトが生成されたときにコンピュータ202上で見られていた2つの異なるウェブページを含む

50

。また、ボックス206は、持続的検索オブジェクトがコンピュータ202で地図アプリケーションおよび連絡先管理アプリケーションとインタラクションしたことを示す。さらに、アクセスログが、オブジェクトがユーザによってアクセスされた時間を示し、ユーザがそのオブジェクトで何を行ったかを示す情報も含み得る。

【0031】

本明細書において示される情報を用いて、さまざまなサービスが、コンピュータ202によってユーザに提供され得る。例えば、オブジェクト208がコンピュータ202で次に起動される時、コンピュータ202は、情報のボックス210(この図においてはボックス210がそのようなデータを表すが、実際にはコンピュータ202または別のデバイスに記憶されたデータ)を見て、デバイス202のユーザが、文書処理のために特定の文書も開いていたときにww
w.test.comのウェブページをブラウザで頻繁に開いていたと判定することができる。その
10
ような判定は、適切な状況で、その文書が開かれるときに、そのウェブページがブラウザでアクティブになるようにしてブラウザを自動的に起動するために使用され得る。そのようにして、ユーザは、複数の関連が推定されるアプリケーションのうちの1つに関する1つのアイコンのみを選択することによってそれらのアプリケーションがアクティブ化されるようにすることができる。一部の実装形態においては、コンピュータ202は、記憶されたメタデータに応じてオブジェクトをインスタンス化する前にユーザの許可を要求することができる。ユーザが許可を与える場合、オブジェクトがインスタンス化される。ユーザが許可を与えない場合、プログラムはインスタンス化されない。コンピュータ202は、その
20
ような許可の付与または拒否のユーザの履歴をメタデータにさらに記憶することができる。

【0032】

図2Bは、図2Aに似た、コンテキストオブジェクトを使用するオペレーティングシステムの概念図である。この図において、アイコン228は、コンピュータ202、またはコンピュータ202からアクセス可能なサーバシステムの永続的なストレージに記憶されることなどによってコンピュータ202からアクセス可能な特定の画像ファイルを表す。例えば、コンピュータ202は、コンピュータ202がそのコンピュータ202自体をサーバシステムに対して特定して、コンピュータ202のユーザに対して登録されたサーバシステムでのアカウントを示すことができるクッキーまたはその他の機構を記憶することができる。そのような機構は、アイコン228の文書を表すデータを取得するためなど、サーバシステムから情報を得
30
るためにコンピュータ202によって使用され得る。そのとき、ユーザは、アイコン228を選択することによって文書呼び出すことができる。

【0033】

アイコン224は、チャットウィンドウの形態のオブジェクトを表す。チャットウィンドウは、ユーザがコンピュータまたはスマートフォンなどのコンピューティングデバイスの別のユーザとコミュニケーションすることを可能にする。場合によっては、チャットプログラムは、2人以上の人が文章によるメッセージを互いに書き込むことによってコミュニケーションすることを可能にする。その他の場合、チャットプログラムは、チャットのための音声または映像の送受信を容易にする。

【0034】

ボックス226、230は、オブジェクト228および224のうちの1つに関連して保存され得るコンテキストデータを示す。例えば、ボックス230は、文書または画像ファイル228などのワードプロセッサ文書に関するコンテキスト情報を示す。オブジェクトとともに保存され得るコンテキスト情報の種類を示すさまざまなフィールドが、ボックス230に示される。例えば、オブジェクトは、そのオブジェクトのオブジェクト種類を示す名前と、コンピュータ202に記憶されたすべてのその他のオブジェクトに対してこのオブジェクトを一意に特定する識別番号とを含む。

【0035】

また、ボックス230は、オブジェクトが最初に生成された時間と、オブジェクト228が生成されたときにコンピュータ202でやはり開いていたオブジェクトのリストとを示す。加
50

えて、ボックス230は、オブジェクト228が最後に使用されたときに開いていたその他のオブジェクトのリストと、オブジェクトが生成されたときおよび最後に使用されたときに存在した人のリストとを含む。オブジェクトが生成されたときおよび使用されたときに存在した人を判定するために、さまざまな方法が使用され得る。例えば、オブジェクトを実行するプログラムは、存在する人に関するユーザ入力を要求することが可能である。別の例において、コンピュータ202は、記憶されたデータベースに基づいてユーザの声を感知することによって、または記憶されたデータベースに基づいて特定の個人に関連するデバイスを検出することによってどの人が存在するかを自動的に判定することも可能である。

【0036】

図2Bを参照すると、ボックス230によって示されるように、オブジェクトが生成されたとき、ユーザXはユーザYおよびユーザZとチャットしていた。ユーザXは、オブジェクトが最後に使用されたとき、ユーザZとチャットしていた。ボックス230は、オブジェクトが生成されたときにユーザXが存在したと、オブジェクトが最後に使用されたときにユーザXとユーザYが両方とも存在したとをさらに示す。存在するユーザは、それらのユーザのうちの1人だけがマウスまたはタッチパッドで選択を行い、キーボードまたはその他の入力デバイスでテキスト入力するときのように実際に「操作している」可能性があるとしても、コンピュータ202を使用しているものとして特定され得る。このように、コンピュータ202が存在する人を自動的に判定する場合、近くに(すなわち、コンピュータ202と同じ室内に、またはコンピュータ202から特定の距離内、例えば、約4.5メートル(15フィート)内に)いる人は、コンピュータを使用しているものとして含められ得る。その他の場合、存在するユーザは、コンピュータ202によって促されるときか、または自発的にかのどちらかで、コンピュータ202を使用しているものとしてそれらのユーザ自身を断定的に特定し得る。

【0037】

そのような合同作用は、画像ファイルがユーザYおよびユーザZからの入力を用いてユーザXによって生成されたことを示し得る。例えば、ユーザが情報を持続的検索の結果からクリップボードにコピーし、文書を生成し、次いで、検索の結果またはその他のコピーされたデータを文書にペーストした場合、その点に関してさらに強い推定がなされ得る。オブジェクトがユーザX、ユーザY、およびユーザZの間の協力によって生成されたことを示唆するそのような情報も、ボックス230に示されるように、オブジェクト228に関連して記憶され得る。

【0038】

ボックス226は、オブジェクト224に関連して記憶されたコンテキスト情報を示す。名前を付けられたオブジェクトは、オブジェクトが生成された時間を含む。この例において、チャットウィンドウはチャットの相手が誰かを示すことができ、それがオブジェクトを一意に特定する。オブジェクトは、コンピュータでその他のオブジェクトに対して当該オブジェクトを一意に特定する識別番号を含み得る。ボックス226によって示されるように、チャットウィンドウ224が生成されたとき、2つの異なるその他のオブジェクトがアクティブであった。それらのその他のオブジェクトは、画像ファイルと、チャットウィンドウオブジェクトが生成されたときにコンピュータ202上で見られていた別のチャットウィンドウとを含む。

【0039】

本明細書において示される情報を用いて、さまざまなサービスが、コンピュータ202によってユーザに提供され得る。例えば、オブジェクト228がコンピュータ202で次に起動されるとき、コンピュータ202は、情報のボックス230(この図においてはボックス230がそのようなデータを表すが、実際にはコンピュータ202または別のデバイスに記憶されたデータ)を見て、デバイス202のユーザXが、この特定の画像ファイルに対して作業しているときにユーザYおよびユーザZと頻繁に協力すると判定することができる。そのような判定は、適切な状況で、その画像ファイルが開かれるときに、ユーザYおよび/またはユーザZとのチャットウィンドウを自動的に起動するために使用され得る。そのようにして、ユーザ

10

20

30

40

50

Xは、複数の関連が推定されるアプリケーションのうちの1つに関する1つのアイコンのみを選択することによってそれらのアプリケーションがアクティブ化されるようにすることができる。

【0040】

同様に、オブジェクト224が次に起動され、ユーザXとユーザZとの間のチャットを開始するとき、ボックス226の情報を分析することによって、ユーザYとのさらなるチャットウィンドウおよび/または画像ファイルオブジェクト228が起動され得る。

【0041】

別の実装形態において、コンピュータ202は、コンピュータを使用する特定のユーザを特定することができる(例えば、ユーザがそのユーザの識別情報をコンピュータ202に入力するように促されるとき、またはコンピュータ202が近くのもしくはコンピュータ202から特定の距離内の1人または複数のユーザを感知するとき、個人が「コンピュータを使用している」ものとして特定され得る)。例えば、図2Bにかかわるシナリオにおいて、コンピュータは、ユーザYおよび/またはユーザZを伴うユーザXの存在を感知し得る。ボックス230は、オブジェクト228を、一般に、ユーザX、ユーザY、およびユーザZによって協力して作業されるものとして特定するので、コンピュータ202によって検出されるとき、これらの人のうちの1人または複数の存在は、オブジェクト228を自動的にロードするようにコンピュータ202に促すことができる。一部の実施形態においては、オブジェクト228は、そのオペレーティングシステムのオブジェクトが閉じられているときに、オブジェクト228のコンテキストメタデータで特定される人のうちの少なくとも2人の存在が存在するまでコンピュータ202によって自動的にロードされない。オブジェクトのコンテキストメタデータで特定される人のうちの少なくとも1人、または場合によっては少なくとも2人の指示を受信することに応答して、コンピュータ202は、オペレーティングシステムのオブジェクト228をインスタンス化する。

【0042】

異なるコンピュータプログラムおよびアプリケーションにかかわるさまざまなその他の実装形態が使用されてもよい。例えば、ユーザは、ワードプロセッサプログラムを使用しながら音楽を聴くことを好む可能性がある。したがって、コンピュータは、(上述のボックスに関して説明されたメタデータを使用して)そのような傾向に留意し、ユーザがワードプロセッサプログラムを開くときはいつも音楽再生プログラムを自動的にインスタンス化することができる。別のユーザは、ワードプロセッサプログラムで作業しているときに特定のアルバムまたはアーティストを聞くことを好む可能性がある。これらの場合、コンピュータは、プログラムのユーザの過去の使用(または特定のワードプロセッサ文書の過去のアクセス)からそのユーザの好みを認識し、音楽再生プログラムを自動的にインスタンス化し、ユーザが好むアーティスト、アルバム、または曲の再生を自動的に開始することができる。

【0043】

その他の場合、コンピュータは、その他のプログラムまたはオブジェクトがインスタンス化されるときに特定のプログラムまたはオブジェクトを閉じるかまたはシャットダウンする傾向があるというメタデータを記憶することができる。例えば、ユーザは、チャットウィンドウがインスタンス化されるときにはいつもメディアプレーヤーを閉じる傾向がある可能性がある。このシナリオのユーザは、チャットアプリケーションを用いて会話を続けようと試みている間に、メディアプレーヤーで再生されているメディアが気を散らせていることに気づく可能性がある。したがって、ユーザは、チャットウィンドウで会話を始めるとまもなくメディアプレーヤープログラムをシャットダウンする。コンピュータは、(上述のボックスに関して説明されたメタデータを使用して)この傾向に留意し、チャットウィンドウがインスタンス化されるときにはいつもメディアプレーヤー(およびおそらくはその他の同様のメディアプレーヤー)を自動的にシャットダウンすることができる。場合によっては、コンピュータは、任意のチャットウィンドウがインスタンス化されるときにいつもメディアプレーヤーをシャットダウンすることが可能である。その他の例におい

10

20

30

40

50

ては、コンピュータは、インスタンス化されたチャットウィンドウが特定の個人とのチャットを含む場合にメディアプレーヤーをシャットダウンすることが可能である。そのような判定は、概して、チャットアプリケーションに関して記憶されたメタデータ、またはチャットアプリケーションを用いた特定の個人とのインタラクションに関して記憶されたメタデータに基づくことが可能である。ユーザが、そのユーザの個人的な設定または好みを与えてもよい。

【 0 0 4 4 】

図3は、コンピューティングデバイス305のメモリの制御を維持するためのシステム300の概念図である。例示的なシステム300は、デバイス305でのソフトウェアアプリケーション310の実行を制御するオペレーティングシステム330を含み得る。1つの実装形態において、デバイス305は、いくつかのソフトウェアアプリケーションを同時に実行することができるオペレーティングシステム330を含むセルラ電話であってもよい。代替的な実装形態において、コンピューティングデバイス305は、ラップトップ、パーソナルコンピュータ、携帯情報端末、またはその他の適切なコンピューティングデバイスとすることができる。

10

【 0 0 4 5 】

デバイス305が電話である1つの実装形態においては、電話をオンにした後、電話のオペレーティングシステム330が、永続的メモリ320からロードされ得、アイコンの所定の表示を含むグラフィカルユーザインターフェースをユーザに対して表示することができる。各アイコンは、ユーザが利用可能なアプリケーションかまたはアプリケーションの代理かのどちらかである可能性があり、選択されるとき、選択されたアイコンは、必要に応じて、関連するアプリケーションのパラメータおよびメモリ内のファイルの位置をオペレーティングシステム330に渡すことができ、今度は、オペレーティングシステム330がアプリケーション310を実行することができる。それぞれの実行されるアプリケーションは、電話の永続的メモリ320のセグメントを使用し、アプリケーションが実行を続けるとき、そのアプリケーションのメモリの要求が高まる可能性がある。ユーザがより多くのアプリケーション310を実行するか、または実行されているアプリケーション310がさらなるメモリを消費するとき、電話のメモリ320は、最終的に、アプリケーション310のメモリの需要に対応するのに不十分になる可能性がある。

20

【 0 0 4 6 】

例示的な実装形態においては、メモリ管理システム300が、永続的メモリ320が使い果たされたときに1つまたは複数のアプリケーション310を終了させ、ユーザがアプリケーションに戻るときにその終了されたアプリケーションを復元することによってメモリの不足に対応することができる。特定の实装形態においては、終了されたアプリケーションウィンドウが別のアプリケーションウィンドウによって完全にかまたは部分的かのどちらかで覆い隠されている可能性があるため、ユーザは、アプリケーションが終了されたことに気がつかない可能性がある。ユーザがアプリケーションに戻るように切り替えることを選択するとき、アプリケーションが再起動され得、ユーザは、アプリケーションを表示する際のおそらくは鈍い応答を除いてはアプリケーションが一時的に終了されたことが分からない可能性がある。

30

40

【 0 0 4 7 】

例示的な実装形態において、オペレーティングシステム330は、グラフィカルユーザインターフェースとのユーザのインタラクションに従ってアプリケーション310をランク付けすることができ、各アプリケーション310は、オペレーティングシステムからの信号に回答してそのアプリケーション310の現在の状態に関する情報を生成し、保存することができる。アプリケーションは、そのような状態情報を自身で保存することができ、またはその情報をオペレーティングシステム330に与えることができ、ひいては、オペレーティングシステム330がその情報を永続的ストレージ(例えば、フラッシュメモリ)に保存することができる。

【 0 0 4 8 】

50

メモリ320が使い果たされる場合、オペレーティングシステム330は、1つまたは複数のランク付けされたアプリケーション310を終了させ、後で、ユーザの要求に回答してそれらの終了されたアプリケーションを再現することができる。例えば、オペレーティングシステム330がロードされると、ユーザは、文書閲覧アプリケーションを選択して、記憶された文書を読むことができる。その後、ドキュメントビューアがまだ実行されている間に、ユーザがウェブブラウザを開き、インターネットサーフィンを始めることができる。ウェブブラウジングセッション中に、ユーザは、電子メールが到着したとの通知に応じて、電話の電子メールアプリケーションを選択して新着電子メールを調べることができる。ユーザは、電子メールを読んでいるときに、カレンダーアプリケーションを実行して、電子メールで触れられたイベントのリマインダを生成しようと試みることができる。

10

【0049】

例示的な実装形態においては、ユーザが新しいアプリケーションを開くとき、オペレーティングシステム330は、1つまたは複数の動的な基準に従ってアプリケーション310をランク付けすることができる。ここでは、オペレーティングシステム330は、実行されているアプリケーション310を以下のように重要度の降順に、すなわち、電子メールアプリケーション、ウェブブラウザ、およびドキュメントビューアの順にランク付けすることができる。そのような順序づけは、さまざまな方法で行われ得る。例えば、アプリケーションは、必要なビジネスアプリケーション、エンターテインメントアプリケーションなどのさまざまなカテゴリーに分けることができる。一部の实装形態において、オペレーティングシステム330は、特定のアプリケーションがバックグラウンドにあるときはそのアプリケーションが休止状態にあると認識することができ、したがって、オペレーティングシステム330はそのアプリケーションを低優先度に分類することが可能である。しかし、別のアプリケーションは、ネットワーク上の情報に絶えずアクセスしている可能性があり(例えば、メッセージングプログラム)、したがって、より高い優先度のアプリケーションとしてランク付けされ得る。一部の实装形態においては、アプリケーションは、2つのカテゴリー、すなわち、可視のアプリケーションと、不可視のアプリケーションとに分けられる。不可視のアプリケーション(すなわち、ウィンドウがユーザには見えないアプリケーション)は、可視のアプリケーションよりも低くランク付けされる。代替的な実装形態においては、アプリケーションの開発者がアプリケーションを自分で分類することも、またはユーザがアプリケーションを分類する、もしくはその他の方法でランク付けすることも可能であり、そのような分類またはランク付けがオペレーティングシステム330に与えられることもある。

20

30

【0050】

ユーザがアプリケーション310とインタラクションするとき、各アプリケーションは、アプリケーションの現在の状態に関する情報を生成し、保存することができる。例えば、アプリケーションが、オペレーティングシステム330によってそのアプリケーションがいつでも終了され得る(例えば、アプリケーションがもはやユーザに見えない)状態になると、オペレーティングシステム330は、アプリケーションにそのアプリケーションの現在の状態を保存するように指示することができる。

【0051】

例示的な例に戻ると、電話のメモリ320は4つアプリケーションすべてを一度に実行するのに十分でない可能性があるため、オペレーティングシステム330は、メモリを使い果たしたので、カレンダーアプリケーションを開くときに、最も低くランク付けされたアプリケーション(この例においては、ドキュメントビューア)を終了させることを選択することが可能である。代替的な実装形態において、オペレーティングシステム330は、差し迫ったメモリの不足を予測し、メモリが尽きることを防止するために1つまたは複数のアプリケーションを終了させることも可能である。例えば、現在利用可能なメモリの量が所定の閾値未満に落ち込む場合、オペレーティングシステムは、現在利用可能なメモリの量が閾値を超える量にするために、ランク付けの低いアプリケーションを終了させることができる。代替的な実装形態において、オペレーティングシステム330は、アプリケーション310

40

50

による近々のメモリ要求を現在利用可能なメモリの量と比較することができ、要求される量が現在利用可能な量を超える場合、オペレーティングシステムは1つまたは複数のアプリケーションを終了させることができる。

【0052】

そのような状況では、オペレーティングシステム330は、最もランク付けの低い1つのアプリケーションまたは複数のアプリケーションを特定し、それらのアプリケーションを一時的に終了させることが可能である。オペレーティングシステム330は、後で、例えば、メモリが開放されるときに自動的に再起動される必要がある可能性があるアプリケーションを示すものとしてフラグを設定することも可能である。

【0053】

終了させるべきアプリケーションの選択は、その他の技法によって行うことができる。例えば、オペレーティングシステム330は、特定のアプリケーションによって必要とされるメモリの量を判定し、次いで、その量のメモリにオーバーヘッドのメモリの余裕を足した量を開放することができるその他の動作しているアプリケーションを特定することが可能である。一例として、あるアプリケーションが3000Kの追加のメモリを必要とする可能性があり、3つのその他のアプリケーションのそれぞれが、2000K、3000K、および35000Kのメモリをそれぞれ開放することができる可能性がある。オペレーティングシステム330は、初めの2つのプログラムが必要とされるメモリの量に最も近いので、それらの2つのプログラムを終了させることによって、メモリに対する最小の「負担」または利用可能なメモリの最良の使用が実現され得ると判定することができる。代替的に、オペレーティングシステムは、できるだけ少ないアプリケーションを終了させることを優先するようにプログラムされ得る。そのような状況においては、この例では、第3のアプリケーションが終了される。

【0054】

ユーザは、カレンダーアプリケーションの使用を終了すると、文書閲覧アプリケーションに戻ることを選択することが可能である。オペレーティングシステム330は、一時的に終了された文書閲覧アプリケーションに戻ろうとするユーザによる試みを検出するとき、保存された状態情報を用いてアプリケーションを再現することができる。

【0055】

そのようにするために、オペレーティングシステム330は、最初に、文書閲覧アプリケーションにアクセスするコマンドを感知することができ、そのようなアプリケーションがアクティブであったが、一時的に終了されていることを、記憶されたフラグから知ることができる。次に、オペレーティングシステム330は、そのアプリケーションを起動し、そのアプリケーションが一時的に終了されたときにそうであった形態で、または実質的にそうであった形態で再現され得るように、保存された状態情報をそのアプリケーションに渡すことができる。代替的に、アプリケーションが、オペレーティングシステムと協力してそのアプリケーション自体の状態情報を記憶済みであってもよく、自身でそのような情報にアクセスし、そのような情報を組み込んでよい。

【0056】

図4は、コンピュータプロセス間のメッセージの受渡しを伴うスレッドアフィニティを提供するシステムの概念図である。概して、システムは、コンピューティングデバイスで実行されており、プロセス間で情報を受け渡すために互いに通信しているプロセスの対を示す。例えば、1つのプロセスが、メッセージを、第2のプロセスがそのメッセージにどのように応答または反応すべきかに関する情報と一緒に第2のプロセスに渡すことが可能であり、第2のプロセスは、メッセージに応答または反応すると、第1のプロセスに確認を与えることが可能である。図4をより詳しく参照すると、システム400は、第1のプロセス402および第2のプロセス404を含む。プロセス402、404の両方は、単一のコンピューティングデバイス上で時間的に重なるようにして同時にまたは実質的に同時に実行されていてもよい。

【0057】

10

20

30

40

50

第1のプロセス402は、例えば、1つまたは複数の文書412を表示する文書管理プログラムとすることができる。文書管理プログラムは、例えば、ワードプロセッサアプリケーション、またはウェブブラウザでワードプロセッサアプリケーションを実装するウェブアプリケーションであってもよい。特定の場合、ウェブブラウザがオペレーティングシステムの一部であってもよく、ウェブブラウザが、オペレーティングシステムのただ1つのネイティブアプリケーションであり、すべてのその他のアプリケーションは、ウェブブラウザ内でウェブアプリケーションとして動作する。

【0058】

第2のプロセス404は、地図アプリケーション414の実行を示す。例えば、システムで実行されている異なるウェブアプリケーションが、サーバベースの地図サービスにアクセスすることが可能であり、地図化された地理情報のグラフィカルなタイルを使用した地理的位置の表示のために広く知られている方法で情報を提供することが可能である。例えば、プロセス402、404を実行しているデバイスのユーザの位置が、ピンまたはその他のアイコンを用いて地図上に示され得る。また、別々のプロセス402、404は、単一のウェブブラウザの別々のタブとしてデバイス上で表されてもよく、概して、異なるドメイン間の通信がウェブブラウザ内で行われることを避ける実装形態を含めて、互いにサンドボックス化される。

【0059】

プロセス402、404の間を横切る1組の矢印は、プロセスのうちの1つのプロセスから別のプロセスに、およびその逆に伝達され得るメッセージおよび情報を示す。例えば、メッセージ406が、プロセス402からプロセス404に最初に送信され得る。例えば、メッセージは、プロセス404に関する表示を変える情報を示すことができる。この例においては、例えば、メッセージ406は、地図アプリケーション414で表示される地図の異なる地域を表示させるために使用され得る緯度および経度または住所の記述を含み得る。

【0060】

加えて、プロセス402は、メッセージ406に関連する1つのコマンドまたは複数のコマンドがプロセス404によって実行されるべき方法についての情報をプロセス404に渡すことができる。この例において、その情報は、優先度コマンド(priority command) 408の形態である。優先度コマンド408は、プロセス404がメッセージ406に関連するそのプロセス404の実行に与えるべき優先度に関してプロセス404に知らせることができる。例えば、プロセス402が時間的制約が厳しい形態のプロセスではない場合、優先度コマンド408は、メッセージ406に対する応答がプロセス404によって優先度が高いものとして扱われないことを示すことができる。

【0061】

メッセージ406および優先度コマンド408は、特定の状況に応じて別々にまたは一緒に、直接または間接的にプロセス404に渡され得る。例えば、メッセージ406および優先度コマンド408が単一の比較的大きなメッセージと一緒に渡されるとき、プロセス404は、その比較的大きなメッセージを解析して、メッセージのメッセージ部分と、メッセージ内に埋め込まれている優先度コマンドとを特定することができる。その他の例においては、追加的な情報が、メッセージ406とともに含まれてもよく、プロセス402およびプロセス404が動作し、相互運用される方法に応じて特定され、処理され得る。プロセス402およびプロセス404の相互運用性は、アプリケーションプログラミングインターフェース(API)、またはプロセス間の通信の形式を定めるためのその他の同様の規格にそれらの2つのプロセスが準拠することによって維持され得る。

【0062】

特定の実装形態において、プロセス404は、プロセス402にメッセージを渡すことが可能である。1つのそのようなメッセージは、プロセス404が完全にメッセージ406に従って動作したことをプロセス402に示す確認410である。特定の実装形態において、確認410は、単に、プロセス404がメッセージ406の受信に応じてプロセス402に情報を返すことによって行うことができる。この特定の例においては、そのような情報は、プロセス402に

10

20

30

40

50

よって管理されている文書412に自動的に組み込まれ得る、アプリケーション414によって決定された地図化された領域の画像を含み得る。

【 0 0 6 3 】

このようにして、プロセスがウェブブラウザアプリケーションのサンドボックス化された環境内で実行される別々のウィンドウまたはタブの一部であるときなどに1つのプロセスが別のプロセスに追加的な情報を伝達することを可能にすることによって、プロセス間通信が拡張され得る。そのような通信は、第2のプロセスがコードの実行に与える優先度を含む、別のプロセスが特定のコードを実行する方法を第1のプロセスが制御することを可能にすることができる。結果として、プロセスのうちの1つがその他のプロセスのうちの1つを不正に制御するか、またはその他のプロセスのうちの1つに影響を与えることを防止する高レベルのセキュリティを引き続き維持しながら、プロセス間のより緊密な相互運用性が提供され得る。

10

【 0 0 6 4 】

図5は、ステートレス環境で状態情報を提供するシステムの概念図である。概して、システム500は、サーバシステム502を使用して、サーバシステム502の特定のユーザのためにサーバシステム502にログを残されるさまざまなデバイスのブラウザアプリケーションの状態についての最新の状態情報を保存する。例えば、ブラウザがコンピューティングデバイス504などのデバイスのオペレーティングシステムのただ1つのネイティブアプリケーションである場合、状態情報が、継続的にサーバシステム502にアップロードされ、サーバシステム502で更新され得る。そのようにして、ユーザがコンピューティングデバイス504をオフにするか、またはその他の方法で別のコンピューティングデバイスに移り、その他のコンピューティングデバイスからサーバシステム502にログインする場合、ブラウザアプリケーションに関する状態情報が、そのその他のデバイスで実行されているアプリケーションに複製され得、そのアプリケーションは、ウェブブラウザアプリケーションがただ1つのネイティブアプリケーションであるオペレーティングシステムでやはり実行され得、すべてのその他のアプリケーションは、ブラウザアプリケーション内で実行されるウェブアプリケーションである。また、ユーザが1つのデバイスでセッションを閉じ、後で同じデバイスを起動する場合、本明細書において検討される技法は、デバイス自体ではなくサーバシステムに記憶されている状態のデータを用いてセッションを再確立することができる。その結果、システムは、そのような情報をデバイスに記憶するための余分なメモリを必要とせずに動作することができ、状態情報が、デバイス間でより容易に共有され得る。

20

30

【 0 0 6 5 】

この例において、システム500は、インターネットおよび関連する接続されたネットワークなどのネットワーク506を通じてサーバシステム502と通信するコンピューティングデバイス504を含む。サーバシステム502は、さまざまなウェブサービスをシステム500のユーザに提供するシステムのより大きなデータセンターの一部としてホストされ得る。例えば、サーバシステム502は、コンピューティングデバイス504上で表示するための文書を生成するためのHTMLコードを提供することが可能な1つまたは複数のウェブサーバを含み得る。

40

【 0 0 6 6 】

コンピューティングデバイス504は、コンピューティングデバイス504の状態が後でデバイス504でまたはその他のデバイスで複製され得るように、通常はステートレスアプリケーションであるブラウザアプリケーションに関する状態情報が記憶されることを可能にするいくつかの特定のコンポーネントを含む。例えば、ユーザがデバイス504をスリープモードにするか、またはデバイス504をオフにする場合、デバイス504の最新の状態は、(例えば、ユーザがウェブブラウザでそのユーザの最後の動作を行ったときに)既にサーバシステム502に保存され得、またはデバイス504がスリープモードに完全に移行することをオペレーティングシステムが許可する前にサーバシステム502にアップロードされ得る。状態情報は、例えば、デバイス504に関する状態情報に変化があるたびに状態情報の変化の

50

インジケータをアップロードするようにデバイス504がプログラムされている場合、サーバシステム502に既に記憶され得る。状態情報は、デバイスの状態を再現するために必要とされるその他の関連情報に加えて、デバイス上に現在表示されているページのドキュメントオブジェクトモデル(DOM)を含んでいてもよい。

【 0 0 6 7 】

ここで、デバイス504で実施され得る特定のコンポーネントを参照すると、ブラウザアプリケーション510が示されており、デバイス504においてネイティブで実行されるただ1つのアプリケーションとすることができる。別に、HTML508が、ブラウザアプリケーション510によってレンダリングされ得るようにデバイス504に記憶され得る。HTML508は、さまざまな形態を取ってもよく、1つまたは複数の例においては、ドキュメントオブジェクトモデル(DOM)ツリーとして表すことも可能である。また、デバイス504は、JavaScript(登録商標)512およびJavaScript(登録商標)の変数514を記憶し、実施することが可能である。例えば、HTML508がレンダリングされるとき、HTMLは、デバイス504上で実行されるさまざまなJavaScript(登録商標)プログラムへのポインタまたは呼出しを含み得る。これらのプログラムは、実行されるときに、プログラムの動作に必要なとされる変数もしくはその他の情報を取り出す、または生成することができる。

【 0 0 6 8 】

さらに、ユーザインターフェース516がデバイスに記憶され得、デバイス504の視覚的なディスプレイまたはスクリーン上に表示されているものの現在の表現を含め、さまざまなパラメータを示すことができる。例えば、ユーザインターフェース516は、特定の時間にブラウザのどのタブが開いているか、および/またはブラウザの上に表示されるフローティングペイン(floating pane)などのアプリケーションの異なる種類がオペレーティングシステムによって特定の状態としてサポートされるかどうかについての情報、ならびにデバイス504上の表示の現在の状態を示すために使用可能であり、デバイス504上に現在表示されている情報に一致する表示を再構築するためにさらに使用可能であるその他の適切な情報を記憶することができる。

【 0 0 6 9 】

ウェブページ保存アプリケーション518も、デバイス504で実施され得、ブラウザ510およびその他のブラウザまたはデバイス504で実行されているその他のプログラムの現在の状態を追跡することができる。例えば、ウェブページ保存アプリケーション518は、デバイス504の特定のコンポーネントへの呼出しまたは特定のコンポーネントからの呼出しを傍受して、ブラウザ510によって示されるコンテンツまたはその他の情報に関して行われた変更を判定することができる。ウェブページ保存アプリケーション518は、通信が、デバイス504によって生成され、ブラウザ510の現在の状態で行われた変更を示すためにサーバシステム502に振り向けられるようにすることができる。例えば、デバイス504は、ブラウザのタブが開かれたことを示し、さらにそのタブに関するURLを提供する情報をサーバシステム502に与えることができる。

【 0 0 7 0 】

ウェブページ保存アプリケーション518およびデバイス504のその他のコンポーネントは、画像520のデータストアにアクセスできる可能性もある。特定の態様において、画像520は、単に、通常のアイコン、およびデバイス504を実行し、デバイス504の状態を再現するために必要とされるその他の基本的な情報に関する画像などのオペレーティングシステムレベルの画像である。それらの画像は、ウェブページ上の画像またはデバイス504のウェブアプリケーションによって使用される画像であってもよく、アプリケーション518内でもまたはアプリケーション518によって後でアクセスするようにウェブページ保存アプリケーション518のために記憶され得る。

【 0 0 7 1 】

本明細書において一例で説明されるシステム500を用いると、ユーザのデバイスの現在の状態が捕捉され、サーバシステム502にアップロードされ得る。次に、ユーザは、例えば、家のコンピュータからネットワークコンピュータに移動することによって、別のコン

10

20

30

40

50

ピューティングデバイスに移動することができ、ユーザがその別のデバイスをブートし、おそらくは、サーバシステム502に対してユーザ自身を特定するための証明情報を(手動でかまたはその別のデバイス上の保存されたクッキーによってかのどちらかで)提供したとき、デバイス504の完全な状態を職場のコンピュータまたはその他のそのようなデバイスに複製することが可能である。結果として、ユーザは、1つの設定から別の設定に切り替えることができるか、またはそのユーザが離れた場所により簡便に戻ることができ、より効率的にデバイス504でアプリケーションを使用することができ得る。例えば、ユーザがそのユーザのためにアプリケーションまたはブラウザ510で実行されるウェブアプリケーションをセットアップする場合、ユーザは、後でまたは別のデバイスでそれらのウェブアプリケーションを実行し続けることを望む可能性がある。例えば、ユーザは、電子メールのウェブアプリ、文書編集アプリケーション、地図アプリケーション、および通常のウェブブラウジングウィンドウを常に開いておくことを好む可能性があり、状態複製技法は、ユーザが1つのマシンから別のマシンに移動するときにも状態を手動で再現することを必要とせずユーザがそれを行うことを可能にすることができる。

【0072】

図6は、ネットワークを介してコンピューティングデバイスのイメージを取得するシステム600の概念図である(ここで言うイメージとは、写真のようなグラフィカルな画像のことではなくシステムイメージのことである)。概して、システム600は、ホストされるサーバシステム602とインターネットなどのネットワーク606を通じて通信することができるモバイルコンピューティングデバイス604などのさまざまなデバイスを含む。検索エンジンサービス、地図サービス、電子メールサービス、文書管理サービスなどのさまざまなホストされるサービスを提供することに加えて、ホストされるサーバシステム602は、デバイス604のオペレーティングシステムを管理するためにデバイス604のオペレーティングシステムと協力することが可能である。例えば、デバイス604のオペレーティングシステムは、ウェブブラウザの形態の単一のネイティブアプリケーションを有するオペレーティングシステムであってもよく、デバイス604で実行されるその他のアプリケーションは、ウェブブラウザ内でウェブアプリケーションとして実行され得る。そのような手法は、デバイス604に記憶され、デバイス604によって管理される必要があるネイティブアプリケーションの数を最小にすることができる。加えて、本明細書において説明されるように、そのような構成は、デバイス604がホストコンピュータシステム602によって管理され得る方法を簡単にすることができる。

【0073】

この例において、デバイス604は、イメージ608を記憶するものとして示されている。イメージは、どのコンポーネントがデバイス604に永続的に記憶されるかを定義することが可能である。例えば、イメージ608は、特定のプログラムにおいて、デバイス604のファームウェアより上の基本的なオペレーティングシステムのコンポーネントを含んでもよく、単一のプログラムが、ウェブブラウザの形態の単一のネイティブプログラムを有するデバイスのオペレーティングシステムのイメージの一部であってもよい。概して、イメージは、複数のコンピュータが、そのようなコンピュータの運用性を管理し、維持する能力を向上させるためにそれらのコンピュータにコンポーネントの共通の基盤を持つことを保証するために使用され得る。特に、企業は、その企業の従業員が利用可能な特定の数のソフトウェアコンポーネントおよび利用可能でないその他のコンポーネントを欲する可能性があり、したがって、イメージを定義することができ、従業員のマシンが初めて配置されるときにそれらのマシンにそのイメージをインストールすることが可能である。

【0074】

この例の中では、サーバシステム602は、デバイス上のイメージを保守または修復するためにデバイス604、およびシステム602からサービスを受けるその他のデバイスと協力する。図に示されるように、いくつかの特定のコンポーネントが、そのようなサービスを提供する際にサーバ602によって使用され得る。例えば、イメージインターフェース610が、デバイス604のようなりモートのデバイスとインタラクションするために設けられてもよ

10

20

30

40

50

い。イメージインターフェース610は、例えば、デバイスと通信して、デバイスのそれぞれのイメージがまだ正しいことを検証することができる。例えば、ただ1つのネイティブアプリケーションがブラウザであり、追加的なアプリケーションが永続的に記憶されないウェブアプリケーションであるときなど、いかなるアプリケーションもデバイスに追加されることを許されない場合、デバイス604のネイティブコンポーネントからハッシュが構成され得、そのハッシュが記憶され、デバイス604がブートされるたびに計算されるその後のハッシュと比較され得る。ハッシュが一致しない場合、そのことは、デバイス604のオペレーティングシステムの中核的なコンポーネントのセキュリティが損なわれたことを示すことができる。そのような状況において、デバイス604は、イメージインターフェース610にメッセージを送信することができ、そのメッセージにより、ひいては、インターフェース610が、サーバシステム602のその他のコンポーネントに特定の動作を実行させるようにすることができる。

10

【 0 0 7 5 】

例えば、イメージリコンストラクタ616が、デバイス604用の差替えイメージであってもよいイメージに関する特定のコンポーネントを特定し、収集し、組み立てるようにプログラムされ得る。イメージリコンストラクタ616は、最初にイメージインデックス618に目を向けて、デバイス604にインストールされるはずであるイメージの形態を特定する。例えば、オペレーティングシステムに関する特定のリビジョン番号が、デバイス604に与えられてもよい。あるいは、単一の製造業者によって提供される系列のデバイスは共通のイメージを有することもあるが、異なる製造業者またはユーザベースによるデバイスは、それぞれ、カスタムイメージを有することもある。したがって、イメージインデックス618は、デバイスの製造業者およびモデルの識別子、またはデバイスを運用する企業の識別子を受信し、そのようなデバイスからの要求に回答してビルドされるべきイメージを特定することができる(イメージは、システムによって記憶された共通の交換可能なコンポーネントからビルドされ得る)。

20

【 0 0 7 6 】

この例において、イメージは、イメージフラグメントから構築され得る。例えば、オペレーティングシステムの特定のレベルが、イメージの一部であってもよく、モジュール式のオペレーティングシステムの各レベルが、イメージに関する別々のフラグメントであってもよい。また、共通のレベルの異なる特徴が、別々のイメージフラグメントと見なされ、別々のイメージフラグメントとして記憶されてもよい。イメージリコンストラクタ616またはイメージアップデータ614は、例えば、イメージの識別子をイメージ全体を構成するさまざまなフラグメントまたはコンポーネントにマッピングし得るイメージインデックス618からの情報を使用することができる。

30

【 0 0 7 7 】

さらに、ユーザデバイスマップ622が、特定のユーザもしくはユーザグループまたはマシンの特定の種類を特定のイメージにマッピングすることによるなどして、イメージインデックス618に関して上で説明された機能に似た機能を実行することができる。例えば、デバイスのユーザが、システム602にログインし、それらのユーザがそれらのユーザのデバイス604に提供されるようにしたいイメージに関する特定のコンポーネントまたはフラグメントを選択することができる1つまたは複数のウェブページを提供され得る。それらのコンポーネントは、ユーザの識別子に関連して保存され得、したがって、イメージをリビルドしようとするその後の試みは、ユーザによって前に選択されたそれらのコンポーネントを自動的に選択することができる。

40

【 0 0 7 8 】

再構築されたイメージを用いて、イメージアップデータ614は、本明細書において検討されるコンポーネント間の全体的な動作および調整を制御することができる動作インターフェース612と協力して、更新されたイメージをイメージインターフェース610およびネットワーク606を通じてデバイス604に供給することができる。更新は、デバイス604が、誤って、またはサーバシステム602によってネットワーク606を通じて意図的にデータが一掃

50

されたときなどには、デバイス604のためのイメージ全体を含み得る。

【0079】

特定の実装形態においては、デバイス604からイメージをリモートでデータが一掃され得、場合によってはそのイメージを新しいイメージで置き換えるためにサーバシステム602が使用され得る。例えば、デバイス604が盗まれた場合、システム602は、デバイス604のイメージを一掃することができ、デバイス604のファームウェアに組み込まれ得る識別子を用いて、デバイス604が再びイメージを取得することを防ぐことができる。あるいは、デバイス604は、システム602によってリモートでデータが一掃され得、次いで、適切な所有者によって回復させることができ、したがって、サーバシステム602は、その場合、デバイスが適切な者の手に戻っていることを所有者が確認すると、新しい更新されたイメージをデバイス604に提供することができる。

10

【0080】

例えば、デバイス604が最初にブートされているとき、公開鍵がファームウェアの小さな読み出し専用のセグメントに記憶され得、その鍵がデバイス604のカーネルを調べるために使用され得る。その鍵、またはデバイスのオペレーティングシステムのコンポーネントのうちの1つまたは複数のハッシュされたバージョンもしくは(鍵自体がハッシュされたバージョンであるときは)その他のハッシュされたバージョンなどの異なる識別子が、コンポーネントを元の識別子を生成した同じハッシュ関数に通すことによるなどしてブート時にデバイスのコンポーネントと照合され得る。一致しない場合、そのことは、(中核的なコードが変わったはずのないときに変わっている)デバイス604のセキュリティが損なわれており、何者かがデバイス604をポットにすることなどのため、またはその他の目的でデバイス604上のネイティブファイルを変更するように試みたことを示すことができる。

20

【0081】

次いで、動作スタック(operating stack)の低レベルにある通信インターフェースが、ユーザがブートシーケンスに干渉することができるようになる前にサーバシステム602とのネットワーク接続を行うことができ、デバイス604でイメージの破損が生じたことを示すことができる。次いで、サーバシステム602は、上で検討されたコンポーネントを用いてデバイス604のための新しいイメージをビルドし、その新しいイメージをデバイス604に転送することができる。次に、デバイス604は、壊れた可能性があるイメージを新しいイメージで置き換え、セキュリティが損なわれたイメージを完全なものにするなどして新しいイメージで動作することができる。

30

【0082】

デバイス604は、また、セキュリティが損なわれたイメージ、またはセキュリティが損なわれたイメージを特徴付けるデータをサーバシステム602に返すことができる。次いで、サーバシステム602は、何がセキュリティを損なう原因となったかを判定しようとしてセキュリティが損なわれたイメージを分析することができる。例えば、サーバシステム602は、そのセキュリティが損なわれたイメージを、サーバシステム602によって同じ初期イメージを割り振られたその他のデバイスからのセキュリティが損なわれたイメージと比較することができる。いくつかの一致するセキュリティが損なわれたイメージが特定される場合、そのような特定は、調整された攻撃が単一の主体によってデバイスに対して行われたことを示すことができる。

40

【0083】

したがって、本明細書において検討されるようにして、リモートのイメージ再取得システム(remote re-imaging system)600は、便利な方法でデバイス604などのデバイスを更新するための機構を提供することができる。例えば、デバイス604は多量のコードを永続的に記憶しないので、イメージに対する頻繁な更新により、デバイス604のユーザにもたらす中断を最小限にすることができる。また、上で検討されたようにクラウドにポストされたイメージデータを使用すると、特定のクライアントデバイスのデータを一掃し、次にサーバシステム602からそれらのクライアントデバイスへ再びイメージを取得するための便

50

利な方法が可能になる。加えて、サーバシステム602は、すべてのデバイスまたはデバイスの特定の大きなグループに共通の基本構成要素の互いに一意な組合せから特定のイメージを再構築することができるように、単に、イメージ全体と一緒に構成する交換可能なコンポーネントを使用し、次いで、ユーザデバイスマップデータベース622でそれらのコンポーネントの組合せを特定のデバイスにマッピングすることによって、多数の異なるユーザデバイスに関するイメージを記憶することができるようになる。

【0084】

図7は、コンピューティングデバイスのリモートの監視および制御を行うシステム700の概念図である。概して、システム700は、スマートフォンまたはタブレットコンピュータまたはラップトップコンピュータなどのポータブルコンピューティングデバイスの形態であつてもよいコンピューティングデバイス702を含む。コンピューティングデバイス702は、ウェブブラウザの形態のただ1つのネイティブアプリケーションを有し、その他のアプリケーションはウェブブラウザ上で実行されるウェブアプリケーションの形態で提供されるオペレーティングシステムのような非常に軽いオペレーティングシステムを搭載するデバイスであつてもよい。上述のその他のデバイスと同様に、デバイス702は、インターネットなどのネットワーク708を通じてインタラクションして、デスクトップコンピュータ706のサーバシステム704と通信することができる。デスクトップコンピュータ706は、コンピューティングデバイス702と同じ人によって所有されているコンピュータとすることができる。例えば、ユーザは、職場または家にデスクトップコンピュータを有する可能性があり、路上でポータブルコンピューティングデバイス702を使用する可能性がある。本明細書に記載のシステム700は、詐欺師などがデバイス702のオペレーティングシステムに干渉し、悪意のあるコードまたはコンテンツを調べる機会を与えることをより難しくすることによって、コンピューティングデバイスにより高いセキュリティを提供することを対象とすることができる。

【0085】

ブートプロセス中の早い段階でセキュリティ侵害を特定するための1つの例示的な機構が、図の例示的なスタックによって示される。この例のスタックは、比較的小さくまとめられており、ファームウェアで実装され得るBIOS716を最も低いレベルに含む。BIOSは、BIOSに関する通常の動作を実行することができ、サーバシステム704との限られたネットワーク接続を確立するためのコードも含み得る。この例における限られた接続は、(ブートプロセスが容易にハッキングされ得るレベルに達してしまう前に)デバイス702からサーバシステム704に、またはサーバシステム704からデバイス702に不正な活動をレポートするために使用される。一例として、ユーザは、デバイス702が盗まれたことを発見することができ、コンピュータ706を使用してコンピュータシステム704にログオンすることができる。次いで、ユーザは、デバイス702のデータを一扫し、シャットダウンすること、またはデバイス702が現在のユーザの写真を撮り、デバイス702の現在位置をレポートすること、またはその他の適切な動作を要求することができる。デバイス702が次にブートされる時、BIOS716は、ワイヤレス通信を実行するための機構を含み得るネットワークハードウェア718によって送信されるメッセージを通じてサーバシステム704に知らせることができる。サーバシステム704は、ステータスメッセージで応答を返すことができる。通常は、ステータスメッセージは、何も問題がないことを示すことができる。しかし、ユーザがログインし、デバイス702がなくなったことをレポートしたので、サーバシステム704は、この例においては、問題が起こったことをデバイス702に示す。

【0086】

サーバシステム704は、ネットワーク708とデバイス702との間の往復の矢印によって示されるように、デバイス702のその後の動作に関する命令を与えることができる。例えば、サーバシステム704は、何者かがデバイス702でタイピングしていることをデバイス702が感知するときなどに、デバイス702のユーザのデジタル画像を撮るようにデバイス702に指示することができる。ブートが完了したデバイス702は、自動的に、現在のユーザ(おそらく泥棒)に気付かれずに写真をアップロードすることができる。加えて、そのような画

10

20

30

40

50

像は、デバイス702の現在位置を示すGPSデータを伴う可能性がある。このようにして、警察がデバイス702を取り戻すためにどこに行くべきかを知ることができ、デバイス702が盗まれている間に特定のユーザがデバイス702を使用していたという証拠を持つこともできる。

【 0 0 8 7 】

デバイス702をブートするためのスタックの残りも、比較的小さくまとまっている。例えば、BIOSのコード716を用いて基本的なセキュリティチェックが行われた後、オペレーティングシステム714のコードが実行され得、オペレーティングシステムの基本的なコンポーネントが起動され得る。オペレーティングシステムの起動の一部として、または同様であるが別の動作の一部として、次に、サービス712が開かれ、デバイス702上で利用できるようにされ得、オペレーティングシステム714の専用のネイティブウェブブラウザアプリケーションなどのアプリケーション710が起動され得る。

10

【 0 0 8 8 】

また、セキュリティチェックは、完全なブートが行われた後を含め自動的かつ周期的に行うことができる。例えば、ウォッチドッグタイマー720が、デバイス702で実行され得、デバイス702でセキュリティが周期的にチェックされるようにすることができる。例えば、本明細書において示されるスタックは、独自のシステムパーティション内であってもよく、スタックが更新されるとき、ハッシュ値を生成するためにそのスタックのコードに関数を適用することができる。ウォッチドッグタイマー720がトリガされるとき、オペレーティングシステムがこれらのコンポーネントのそれぞれが常にスタック内にあるオペレーティングシステムである場合、同様のハッシュが、そのときにスタックが含むものすべてに対して実行され得、更新中を除いて変わらない。次に、新しいハッシュが、スタックに関する記憶されたハッシュと比較され得る(ハッシュは、スタック内のすべてのコンポーネントよりも少ないコンポーネントに対して行うことができる)。値が変わった場合、サーバシステム704は通知され得、その後、デバイスをロックすること、デバイス702のストレージのデータを一扫すること、デバイスのストレージ媒体を再フォーマットすること、またはその他のそのような動作を実行することなどのためにデバイス702に信号を送出することが可能である。サーバシステム704は、職場の電子メールまたはテキストメッセージの通知などの予備の伝達手段によってデバイス702の正規ユーザに知らせることもできる。

20

30

【 0 0 8 9 】

デバイス702は、サーバシステム704に問題をレポートする際、問題を発見すると直ちにレポートすることも、または後でレポートすることも可能である。例えば、BIOS716は、デバイス702の問題を示す識別子を生成することができる。次いで、BIOS716は、デバイス702が完全な機能を有するようにするためにスタックのその他のコンポーネントが実行されることを可能にし得る。デバイス702が完全にブートされると、サーバシステム704が、デバイス702に問題があることを知り、理解するように、かつサーバシステム704が、リブートし、特定の1つのプロセスもしくは複数のプロセスをリブートし、デバイス702のストレージ媒体を消去するための、またはデバイス702のストレージ媒体を再フォーマットするためのメッセージなどの適切なメッセージをデバイス702に送信することができるように、ネットワークハードウェア718が、サーバシステム704に識別子を提供することにもなり得る。

40

【 0 0 9 0 】

図8は、ホストされるコンピュータシステムに集中的に記憶されるデータのコンピューティングデバイスへのキャッシュを行うためのシステムの概念図である。概して、システム800は、サーバシステム804のストレージ810のキャッシュとして働くようにコンピューティングデバイス802のストレージ808を使用する。このようにして、処理を高速化するために、キャッシュされたデータ808が、多くの場合、デバイス802にとどまり、比較的少ない場合にネットワーク806を介した往復を必要とすることを可能にすることによって、デバイス802の動作が、より効率化され、高速化され得る。本明細書におけるデバイス802は

50

、上で検討されたその他のデバイスと同様に、ウェブブラウザがただ1つのアプリケーションであり、その他のアプリケーションはウェブブラウザのウェブアプリケーションとして動作するオペレーティングシステムを有するものとして実装され得る。また、図7に関して検討されたスタックが、デバイス802に実装され得る。

【0091】

図において、デバイス802は、3つの特定の値、A、B、およびCを記憶しているところを示されている。同様に、サーバシステム804は、対応する方法で、それらの値のうちの2つをAおよびBの形態で記憶している。サーバシステム804は、Dの値も記憶している。ストレージ810内のAおよびBの値は、それらの値がデバイス802によってストレージ808内で変更されたので、それらの項目が「ダーティ(dirty)」であり、ストレージ810へのアクセスを試みるその他のデバイスによって信用され得ないことを示すために括弧で囲んで示されている。基本的に、サーバシステム804は、デバイス802がそれらの値をチェックアウトしたかのように動作するが、ただし、別のデバイスによってそれらの値の要求が行われる場合、サーバシステム804は、それらの最新の値を得るためにデバイス802に問合せを行い、それらの値の制御をデバイス802から取り戻すことができる。また、図に示されるように、デバイス802において、項目Bのみが、ストレージ808内の項目Bの上のアポストロフィーによって示されるように、サーバシステム804から取得されたその項目のバージョンから変更されている。

10

【0092】

動作の間、デバイス802は、項目A、B、および/またはCの値が変更される結果となるさまざまな動作を実行することができる。適切な時点で、デバイス802は、そのような変更をサーバシステム804に送ることができ、続いて、サーバシステム804が、ストレージ810内のその項目の値を更新することができる。デバイス802は、それらの項目を用いてそれが行われることを示すことができ、それに応じて、サーバシステム804は、システム800内の制御された種類の値であるという項目のマークを解除することができる。このようにして、データが、デバイス802で実行されるウェブアプリケーションとサーバシステム804のためのデータストアとの間で便利にキャッシュされ得る。

20

【0093】

図9は、コンピューティングデバイスの遅らせたロックを行うためのプロセスの流れ図である。概して、プロセスは、クラムシェルデバイスのカバーを閉じること、またはタブレットデバイスを保護されたスリープモードにするためにタブレットデバイスのボタンもしくはその他のコンポーネントを押すことによってユーザがデバイスを閉じるときに、アクティブモードからスリープモードなどの保護されたまたはロックされたモードに移るのを遅らせるように構成されているコンピューティングデバイスで実行され得る。この例において、保護されたスリープモードとは、デバイスを再び使用するために、ユーザがパスワードを入力するか、または別の同様の動作を行ってデバイスのロックを解除する必要があるスリープモードである。概して、そのようなデバイスは、単にデバイスを再びオンにすることに加えて、再アクティブ化するために有効な入力が必要とするので、ロックされていると言われる。

30

【0094】

本明細書に記載のプロセスによって意図的に引き起こされる遅延は、ユーザが、デバイスがスリープまたはロックされたモードに移行することを望んでいることを示した後、デバイスを完全にスリープモードにする前にすぐに考えを変え、デバイスを再び使用する場合に、ユーザが短時間のうちの再びデバイスを使い始めることを可能にすることができる。遅延が適切に時間を調整されており、短い場合、そのような遅延は、適切な元のユーザが遅延期間の間ずっとデバイスの近くに留まる可能性が高く、したがって、デバイスがロックされたモードに遷移する前に侵入者がデバイスを奪うことができないという点で、プロセスにもたらすセキュリティ上の負担が最小限である可能性がある。

40

【0095】

プロセスは、デバイスが特定の動作を実行すべきであることをユーザが示したかどうか

50

を判定するためにコンピューティングデバイスを監視することを開始するボックス902で始まる。ボックス904において、プロセスは、ロックする時間に関するユーザ入力を受信する。例えば、ユーザは、デバイスを特定のユーザに関して変わり得るスリープモードに変える前に遅延を有するようにそれらのユーザのコンピューティングデバイスを構成することができる。例えば、この場合、ユーザは、ユーザが考えを変え、デバイスのアクティブ状態を再確立するための適当な時間を与えるために9秒の遅延を特定することができる。デバイスをロックするための遅延時間に関するユーザ入力に応じて、ボックス906において、プロセスは、デバイスに関するロックの時間パラメータを調整する。そのようなパラメータは、デバイスに永続的に記憶され得、したがって、セッションが異なっても、デバイスをロックするための遅延期間は同じであり、ユーザは、時間が経過する前に素早くデバイスを再開することができることを快適に感じ始める可能性がある。

10

【0096】

ボックス908において、プロセスは、ユーザがデバイスのシェルカバーを閉じてデバイスの電源スイッチを押すこと、またはユーザが直接スイッチを押してデバイスをオフにするかもしくはデバイスをスリープモードに移行させることなどによって、デバイスが開いた構成から閉じた構成に移行させられたことを特定する。本明細書において検討される関連する遷移は、デバイスのアクティブな動作モードに戻るために実質的なユーザ入力が必要とする遷移であり、したがって、遷移は、スクリーンをオフにするが、モード変更に応じてデバイスがロックされず、かなりのコンポーネントが停止されていないという点で容易に元に戻すことができ得る、単にデバイスを閉じた構成から開いた構成に移すことだけでは済まない。とはいえ、ユーザの入力が受信されたというフィードバックをユーザに与えるために、元に戻るために実質的なユーザ入力が必要とするスリープモードにデバイスを移す際の遅延が続いているとしても、デバイスが開いた構成から閉じた構成に移った後に直ちにデバイスのスクリーンを消すこともできる。

20

【0097】

ボックス910において、プロセスは、時間パラメータが切れるときにデバイスをロックすることに関連付けられるシャットダウンタイマーを開始する。この場合の時間パラメータは、ボックス904においてユーザにより選択され、ボックス906においてデバイスにより適用されたパラメータである。例えば、デバイスは、閉じられるとすぐにスクリーンを消し、9秒のシャットダウンタイマーを開始することができるが、タイマーが切れるまで異なるモードに移行しなくてもよい。したがって、912において、プロセスは、設定された時間が過ぎたかどうかを判定するためのチェックを繰り返し行う。時間が過ぎていない場合、プロセスは、ボックス914において、デバイスが開かれたかどうかを調べる。デバイスが開かれた場合、プロセスは、ボックス902のコンピューティングデバイスの監視に戻る。その後、ユーザは、ロックのタイマーに関する新しい時間を設定することができ、または再びデバイスを開いた構成から閉じた構成に移し、したがって、ここで検討された動作の一部もしくはすべてを繰り返すことができる。

30

【0098】

タイマーが切れると、プロセスはボックス916においてデバイスをロックする。デバイスのそのようなロックは、デバイスをアクティブ状態に戻すためにユーザのロック解除の入力を必要とすることに加えて、デバイスの特定のプロセッサの動作を遅くすることもしくはそれらのプロセッサへの電力を完全に取り除くこと、ディスプレイスクリーンをオフにすること、換気ファンおよび電力を使うその他のものをオフにすること、ならびにユーザがデバイスを再アクティブ化し、ロックを解除するのを待つことも含み得る。

40

【0099】

図10Aは、オペレーティングシステムでコンテキストオブジェクトを管理するためのプロセスの流れ図である。概して、プロセスは、オペレーティングシステム内のオブジェクトが生成されたとき、および/またはそれらのオブジェクトが変更されたか、もしくはその他の方法でデバイス上で操作されたときに、それらのオブジェクトを、それらのオブジェクトを取り巻くコンテキストを記述するメタデータと関連付けることを含む。

50

【0100】

プロセスは、オペレーティングシステムにおいてオブジェクトがインスタンス化される1002で始まる。オブジェクトは、さまざまな形態を取ってもよく、ここでは例示を目的として、ワードプロセッサアプリケーションで編集され得るワードプロセッサ文書である。ボックス1004において、プロセスは、開始時のその他の開いているオブジェクトの状態を定義するコンテキストメタデータを特定する。例えば、ユーザはウェブブラウザで特定のURLを開くことができ、プロセスは、ウェブブラウザに関するおよびそのURLに関する識別子を記憶することができる。そのようなコンテキストメタデータは、そのメタデータが、ユーザがウェブページを見ており、次いで、ワードプロセッサプログラムでそのウェブページについて書き留めようと判断したことを示す場合、重要である可能性がある。したがって、ブラウザアプリケーションのウェブページのURLとワードプロセッサアプリケーションのワードプロセッサ文書との間の相関が生成され、記憶されることが有益である可能性がある。結果として、コンテキストメタデータがボックス1006において記憶され、インスタンス化されたオブジェクトに関する情報がボックス1008において記憶される。

10

【0101】

ボックス1010において、ユーザが情報をワードプロセッサ文書に打ち込み、その文書の編集を終えた後など、ある時間の後に、オブジェクトが閉じられることもあり、その後、ユーザがもう少し文書を編集したいときなどに、オブジェクトを開く要求が受信され得る。オブジェクトが再び開かれるとき、さまざまなその他のアプリケーションがそのとき実行され得、それらのアプリケーションは、やはり、この文書がユーザによって開かれた理由に関連し得る。この場合も、例えば、ブラウザは、ユーザにとって重要な、ひいては、オブジェクトであるその文書にとって重要なトピックを開いていることがある。

20

【0102】

結果として、そのような情報を捕捉するために、ボックス1012において、プロセスは、コンテキストメタデータを、インスタンス化されたオブジェクト以外の開いているオブジェクトについての情報を用いて更新する。したがって、オブジェクト自体を表すドメインファイル(domain file)の一部であってもよい、オブジェクトに関するメタデータリポジトリが、ユーザがオブジェクトを継続的に開き、閉じ、操作するとき生成され、追加され、更新され得、メタデータリポジトリに追加されるデータは、同時に実行されているその他のアプリケーションによって定義されるときを含むそれぞれの状況においてオブジェクトが存在したコンテキストと、それらのその他のアプリケーションが何をしていたのかについての情報とを表す。したがって、ボックス1014において、ユーザは、オブジェクトを閉じて開くその他のサイクルを通り抜け、メタデータが、更新される。

30

【0103】

そのようなコンテキストメタデータは、最終的に、ユーザの意図を十分に示すと判定されるときに、自動的に動作を実行するために使用され得る。例えば、ユーザが文書を開くとき、システムは、コンテキストメタデータの分析が、ユーザが、文書を開くと、通常、検索を実行するというを示す場合、同時に、ブラウザで検索エンジンに対して検索を実行することができ、文書とともに検索結果を表示することができる(例えば、文書が企業の株価を追跡する場合、システムは、企業に関する現在の価格に関する検索を実行することができ、したがって、ユーザはそれを文書にカットアンドペーストすることができる)。

40

【0104】

図10Bは、オペレーティングシステムでコンテキストオブジェクトを管理するためのプロセスの流れ図である。図10Aに示されたプロセスと同様に、プロセスは、オペレーティングシステム内のオブジェクトが生成されたとき、および/またはそれらのオブジェクトが変更されたか、もしくはその他の方法でデバイス上で操作されたときに、それらのオブジェクトを、それらのオブジェクトを取り巻くコンテキストを記述するメタデータと関連付けることを含む。

【0105】

50

プロセスは、コンピューティングデバイス上で開いている第1のオペレーティングシステムのオブジェクトに対する第1の事前に定義された動作の第1の発生が記録される1022で始まる。ボックス1024において、第1のオペレーティングシステムのオブジェクトと同時にコンピューティングデバイス上で開いている第2のオペレーティングシステムのオブジェクトに対する第2の事前に定義された動作の第1の発生が、記録される。オブジェクトは、ワードプロセッサ文書、画像ファイル、特定のURLを表示しているウェブブラウザなどのさまざまな形態を取ってもよい。

【0106】

次に、それぞれ、第1のオペレーティングシステムのオブジェクトおよび第2のオペレーティングシステムのオブジェクトに関連する第1の事前に定義された動作および第2の事前に定義された動作が、ボックス1026において相互に関連付けられる。第1のおよび第2のオペレーティングシステムのオブジェクトに関連する相互に関連付けられた動作を含む相互に関連付けられた状態が、ボックス1028において記憶される。相互に関連付けられた状態を記憶するデータベースが、保守される。これらの状態を記憶することは、特定の時間にさまざまなオブジェクトに関して取られた動作の履歴を提供し、したがって、オブジェクトの関連の履歴を記憶する。そのようにすることは、コンピュータがデータベースにアクセスして特定のオブジェクトにかかわるあらゆる相互に関連付けられた状態を検索することを可能にし、したがって、さまざまなオブジェクトの間の関係が判定され得る。

【0107】

ボックス1030において、第1のオペレーティングシステムのオブジェクトに対する第1の事前に定義された動作の第2の発生が特定される。ボックス1032において、次に、第2の発生が、第1のオペレーティングシステムのオブジェクトに対する第1の事前に定義された動作を含むデータベース内の相互に関連付けられた状態と比較される。比較は、ボックス1034に示されるように、第2のオペレーティングシステムのオブジェクトに対する第2の事前に定義された動作が第1のオペレーティングシステムのオブジェクトに対する第1の事前に定義された動作に関連するという判定をもたらす。その結果、ボックス1036において、第2のオペレーティングシステムのオブジェクトがインスタンス化される。相関は、任意の所与の時間にコンピュータ上で開いているオブジェクトの「スナップショット」を取得することによって確立され得る。相関は、個々のオブジェクトに関連するメタデータを記憶することによって決定され得、メタ状態は、図2Aおよび2Bに関して上で説明されたように、その他のオブジェクトまたはユーザについての情報を含む。

【0108】

図11は、コンピューティングデバイスでメモリの制御を維持するためのプロセスの流れ図である。プロセス1100は、例えば、システム400などのシステムによって実行され得る。しかし、別のシステム、またはシステムの組合せが、プロセス1100を実行するために使用されてもよい。

【0109】

図11を参照すると、プロセス1100は、コンピューティングデバイスで潜在的なメモリの不足を管理するための例示的な方法におけるアプリケーションのライフサイクルを示す。この方法は、アプリケーションを実行するステップ1105で始まる。例えば、オペレーティングシステムが、ユーザの要求に回答して、または別のアプリケーションの要請で永続的メモリからアプリケーションをロードすることができる。次に、ステップ1110において、システム400は、アプリケーションの状態が変わったかどうかを判定する。一部の実装形態においては、アプリケーションモニタ220が、いつアプリケーションがフォーカスを取得するか、いつ新しいアプリケーションが生成されたか、またはいつアプリケーションが終了されたかに注目することができる。代替的な実施形態においては、上述の変化に加えて、または上述の変化の代わりに、アプリケーションマネージャは、システムへのユーザ入力(例えば、キーの押下、マウスのクリック、スタイラスまたは指のタップなど)を監視して、いつアプリケーションが状態を変えた可能性があるのかを判定することができる。

【0110】

10

20

30

40

50

状態の変化が検出されない場合、方法は、システムがメモリの不足が存在するかどうかを判定するステップ1115に進む。システムが、メモリの不足が確かに存在すると判定する場合、方法は、以下で検討されるステップ1130に進む。しかし、メモリの不足が存在しない場合、ステップ1110が繰り返される。

【0111】

状態の変化が検出されると、方法はステップ1120に進む。ステップ1120において、状態情報が生成され、記憶される。特定の状態のアプリケーションはいつでも終了され得るので、アプリケーションモニタは、状態の変化を検出した後、アプリケーションに、状態情報を生成し、永続的メモリに記憶するように指示することができる。例示的な実装形態において、状態情報は、アプリケーションが終了される前に存在していたようにアプリケーションを再現するために使用される情報を含み得る。例えば、状態情報は、ディスプレイ上のアプリケーションのウィンドウの位置、ユーザによって行われたアプリケーションのファイルへのすべての変更、およびユーザが好む表示モードを含み得るがこれらに限定されない。状態情報が生成されると、アプリケーションは、その状態情報を永続的メモリに記憶することができる。

10

【0112】

状態情報が生成され、保存された後、方法は、システムがメモリの不足が存在するかどうかを判定するステップ1115に進む。システムがメモリが底をついたと判定する場合、カーネルが、アプリケーションターミネータに、1つまたは複数のアプリケーションを終了させることによってメモリを利用可能にするように指示することができる。代替的な実装形態においては、カーネルは、メモリが不足しつつあるかどうかを判定することができる。カーネルは、アプリケーションからのメモリ要求を現在利用可能なメモリと比較することによって差し迫ったメモリの不足を検出することができる。カーネルは、利用可能なメモリの量が所定の閾値未満であるかどうかを判定することも可能である。不足が検出されるとき、カーネルは、選択されたアプリケーションが通常の動作を継続するのに十分なメモリを有することを保証するために十分なメモリを解放するためのステップを踏むことができる。

20

【0113】

システムがメモリを必要すると判定した場合、方法は、アプリケーションが最も低くランク付けされたアプリケーションであるかどうか、すなわち、アプリケーションがアプリケーションの階層の一番下にあるかどうかをアプリケーションターミネータが判定するステップ1130に進む。アプリケーションが階層224の一番下でない場合、方法は、システムが状態の変化に関してアプリケーションを監視するステップ1110に戻る。アプリケーションが階層の一番下にある場合、方法は、アプリケーションが終了されるステップ1140に進む。一部の实装形態において、アプリケーションターミネータは、アプリケーションがプログラムスタック内の最後のアプリケーションであるかどうかを判定する。そうである場合、そのアプリケーションおよびすべての関連するスレッドが、終了される。

30

【0114】

ステップ1150において、システムは、アプリケーションが復元されるべきかどうかを判定する。例示的な実装形態においては、ユーザが終了されたアプリケーションに戻ろうと試みることをシステムが検出する場合、方法は、記憶された状態情報を用いてアプリケーションが復元されるステップ1105に戻る。例えば、システムは、終了されたアプリケーションの上に被さっているウィンドウを閉じるかまたは最小化しようとするユーザによる試みを検出することができる。それに応じて、システム400は、アプリケーションに関する状態情報を永続的メモリからロードし、記憶された情報を用いてアプリケーションを復元する。

40

【0115】

システム400が終了されたアプリケーションとインタラクションしようとする試みを検出しない場合、方法は、システム400がアプリケーションを復元すべきかどうかを再び判定するステップ1150に戻る。

50

【 0 1 1 6 】

図12は、コンピュータプロセス間のメッセージの受渡しを伴うスレッドアフィニティを提供するためのプロセスの流れ図である。概して、本明細書において、プロセスは、単一のコンピュータで実行されているプロセス間で行われるものとして示される。例えば、2つのプロセスが、デバイスで実行されており、他方のアプリケーションに対してデバイス41のメモリを保護するために互いにサンドボックス化されるように構成されている2つの異なるアプリケーションを表すことができる。プロセスは、プロセスのそれぞれがインスタンス化されるボックス1202で始まる。

【 0 1 1 7 】

ボックス106において、プロセスがある時間の間実行された後、プロセスAは、プロセスAがプロセスBによる動作を必要とするか、またはプロセスBから返された情報を受信する必要があると判定することが可能である。結果として、ボックス106において、プロセスAは、そのプロセスBがプロセスAのためにメッセージをどのくらい速く処理する必要があるかを定義する情報を伴うメッセージを渡す。したがって、ボックスPaul Blakeにおいて、プロセスBが、メッセージを受信し、ボックス110において、優先度情報(priority information)からメッセージの本体を分ける。次に、プロセスBは、優先度情報、またはメッセージによってプロセスAから渡され得るその他の情報を用いて、メッセージをどのように処理すべきかを決定する。例えば、プロセスAは、その処理を完了するためにコンピュータデバイスの現在の状態に関するレポートを必要とする可能性があり、それを行うための情報を待つことがかなり遅く、そのことにより、プロセスに、ユーザによって直ちに必要とされていると伝えさせる場合、プロセスAは、プロセスAの動作をすべてのその他の動作よりも優先させるようにプロセスBに伝える情報を与えることができる。したがって、ボックスPaulハンドレッド12において、プロセスは、優先度情報に基づいて焦点を変えることができる。例えば、プロセスBが、前に、長い時間の制約が厳しくないプロセスに対処していた場合、プロセスBは、そのような活動に関する状態情報を保存することができ、プロセスBでのその処理をひとまず置いておくことができる。このようにして、プロセスは、便利な方法で、それらのプロセスを通じて行われる要求に関するより集中的な応答を提供するように互いに通信し合うことができる。

【 0 1 1 8 】

図13は、ステートレス環境で状態情報を提供するためのプロセスの流れ図である。概して、プロセスは、ウェブブラウザのペインまたはウェブブラウザで実行されている1つもしくは複数のウェブアプリケーションなどのアプリケーションの状態の変化を追跡することと、それらの変化についての情報を記憶するサーバシステムにそのような情報を渡すことを伴う。同じユーザに関してログインされる後のコンピュータは、そのとき、サーバシステムから状態情報を取得することができ、したがって、最後に保存された状態を再現することができる。

【 0 1 1 9 】

プロセスは、ユーザが通常の方法でコンピューティングデバイスを動作させるボックス1302で始まる。コンピューティングデバイスは、ウェブブラウザの形態で単一のネイティブアプリケーションを実行するオペレーティングシステムを実行することができ、その他のアプリケーションは、ウェブアプリケーションのようにウェブブラウザ内で実行される。ブラウザの各ペインおよびウェブアプリケーションのそれぞれは、セキュリティのために独自の隔離されたプロセスでサンドボックス化され得る。

【 0 1 2 0 】

ボックス1304において、プロセスは、周期的に、またはデバイスの状態の変化に応じて、デバイス上の1つまたは複数のアクティブなDOMのイメージを保存する。そのような活動は、ユーザがウェブページ上のアクティブコンテンツ(例えば、JavaScript(登録商標))によって生成されるコンテンツ)とインタラクションしたかどうかなど、関連する状態がウェブページの現在の状態であるときに行われ得る。デバイスの、およびデバイス内のさまざまなプロセスの現在の状態を完全に捕捉するために、DOM情報の代替として、または

10

20

30

40

50

その情報に加えて、その他の状態情報も保存され得る。例えば、デバイス上のアクティブなプロセスのそれぞれのリストが、保有され、更新され得、それらのプロセスの現在の状態を定義する特定のパラメータも、更新され得る。そのような情報は、コンピューティングデバイスに情報を提供しているサーバに同時にまたは後でアップロードされ得、サーバシステムは、デバイスの最後に保存された状態が、後で再構築され得るようにその情報を体系化することができる。

【0121】

ボックス1308において、そのような再構築の要求が発生する。例えば、同じユーザによってサーバシステムを介して第1のデバイスとして登録されるデバイスのブートプロセスは、自動的にサーバシステムに働きかけて、登録されたデバイスのユーザアカウントに関する最新の状態情報を取得することができる。そのとき、サーバシステムは、特定のデバイスに関する記憶されたイメージにアクセスすることができる(ユーザまたはユーザのグループは、デバイスがブートされるときにデバイスが自動的に取得するべきそれらのデバイスに関するイメージを定義することができる)。また、デバイスは、前のコンピュータと後のコンピュータとが互いに同じであるのかまたは異なるのかにかかわらず、イメージのコンポーネントに関連するデータを用いて、ウェブページおよびその他のオブジェクトを、サーバシステムのユーザのアカウントに登録されたコンピュータをユーザが最後に使ったときのそれらの状態に再構築することができる。

10

【0122】

ボックス1314において、プロセスは、ページおよびその他のオブジェクトをそれらが前に保存された状態のまま表示する。

20

【0123】

図14は、ネットワークを介してコンピューティングデバイスのイメージを取得するプロセスの流れ図である。概して、本明細書において、プロセスは、ホストされるサーバシステムと通信する1つまたは複数のクライアントデバイスで行われるものとして示される。この例において示される処理の特定の分割は、例示のみを目的として与えられており、異なる実装形態においては、その他の動作が、同様のコンポーネントによって、または異なるコンポーネントによって行われてもよい。概して、ここで示されるプロセスは、クライアントデバイスがブートされるときにどのように動作するかをそれぞれが定義するイメージを構築するためのイメージデータを記憶することを含む。イメージは、集中的なホストされるサーバシステムで記憶され、構築され、イメージデータは、ブート時にクライアントデバイスに与えられる。

30

【0124】

プロセスは、複数の異なるクライアントデバイスがデバイスのユーザによって通常の方法で動作させられるボックス1402で始まる。ユーザのそれぞれは、ボックス1404のように、それらのユーザの特定のデバイスのイメージに関するパラメータを定めることができ、ボックス1406でイメージを受信することができる、ホストされるサーバシステムにデバイスのイメージを送ることができる。例えば、ユーザは、それらのユーザのデバイスをブートするときに(ウェブアプリケーションなどの)特定のアプリケーションがロードされることを望む可能性があり、特定の 방법으로設定されたデバイスの設定を望む可能性がある。

40

【0125】

ボックス1408において、サーバシステムは、特定のデバイスに関するイメージを、サーバシステムの記憶されたデータと比較する。例えば、サーバシステムは、さまざまな異なるイメージと一緒に構成するコンポーネントを記憶することができ、単純に、各コンポーネントの1つのコピーと、それらのコンポーネントをクライアントデバイスに関するイメージのそれぞれにマッピングする基本的なテキストのまたは同様のファイルとを記憶することができる。例えば、2進の数字のリストが、イメージ全体を特定するために使用されてもよく、リストのそれぞれの位置が、特定のイメージに存在することも、または存在しないこともある特定のコンポーネントを特定することができ、特定のデバイスに関してその位置に1があることは、デバイスがそのイメージ内にその特定のコンポーネントを有す

50

ることをサーバシステムに示すことができる。したがって、例えば、ボックス1410において、プロセスは、デバイスのイメージの一部をより小さなシノニム(synonym)で置き換える。説明されたばかりの例において、シノニムは、単純なビット値とすることができるが、その他の実装形態においては、シノニムは、システム内の特定のコンポーネントに対して一意である英数字の識別子とすることもできる。

【0126】

ボックス1412において、サーバシステムは、特定のイメージに属するデバイスを指し示すデバイスの識別子を有するサイズを削減されたイメージを保存する。例えば、システムは、1つの位置に特定のコンポーネントに関するコードを記憶することができ、上で検討された2進値のストリームと一緒に、その他の位置に、デバイスに対して一意であるデバイスの識別子を記憶することができる。

10

【0127】

後日、特定のデバイスが、ボックス1414によって示されるように、イメージを要求することができ、サーバシステムが、ボックス1416において、そのデバイスに関するイメージにアクセスし、そのイメージを展開することができる。一例として、デバイスは、ブートされるたびにイメージを要求することが可能であり、オフであるときにローカルにそのデバイスのイメージを記憶しないこともできる。そのような展開は、サーバシステムによって、上で検討された2進値のリストをステップスルーし、リストの中の各値が0ではなく1であるコンポーネントを集めることを含み得る。コンポーネントを収集するか、またはそれ以外の方法でデバイスに関するイメージをビルドするためのその他の技法も、使用され得る。

20

【0128】

ボックス1418において、サーバシステムは、そのサーバシステムがビルドしたイメージをクライアントデバイスに送り返し、ボックス1420において、クライアントデバイスが、イメージをロードし、そのデバイスがデバイスのユーザによって完全に動作させられることを可能にする。本明細書において検討される技法を使用して、イメージデータは、特に、システムに関して多数のデバイスが動作するときに、システム全体でそのイメージデータのサイズが削減される方法で、ホストされるシステムに記憶され得る。

【0129】

図15は、コンピューティングデバイスのリモートの監視および制御を行うためのプロセスの流れ図である。概して、プロセスは、チェックが行われているレベルよりも低いレベルのスタックに入り込むハッカーの能力を少なくするために、オペレーティングシステムのスタックの低いレベルでセキュリティチェックを行うことによってクライアントコンピューティングデバイスにセキュリティを提供する。

30

【0130】

プロセスは、デバイスがユーザによってオンにされるボックス1502で始まる。ボックス1504において、デバイスは、そのデバイスのブートファームウェアにアクセスし、同様の方法でブートプロセスを開始する。ファームウェアは、BIOSまたはデバイスのその他の構造の一部であってもよい。ファームウェアは、デバイスとホストされるサーバシステムとの間のワイヤレス送信のための機構も含み得、ボックス1506において、プロセスは、そのような機能を用いてホストされるサーバシステムにメッセージを送信する。メッセージは、デバイスがホストされるサーバシステムに最後に連絡して以降のデバイスの状況の変化を特定する要求として扱われ得る。ボックス1508において、サーバシステムは、メッセージを受信し、デバイスおよびデバイスのパラメータを特定する。例えば、ホストされるサーバシステムは、デバイスの識別子を用いて、デバイスが盗まれ、ロックされるか、またはデータが一掃されるか、または再フォーマットされるべきであることをデバイスのユーザがレポートすることなど、デバイスに関連する何らかのイベントが起こったかどうかを判定することができる。加えて、デバイスの識別情報は、図14に関して検討されたように、デバイスのためのイメージを特定し、デバイスのさらなるブートのためにデバイスにそのイメージを返すためにコンポーネントの収集を開始するために使用され得る。

40

50

【0131】

ボックス1510において、サーバシステムは、デバイスのユーザによるレポートに関してつい先ほど検討された情報およびデバイスが盗まれたかどうかなどのデバイスについての記憶された情報にアクセスする。ボックス1512において、サーバシステムは、デバイスで実行されるべき動作のためのデータを生成し、送信する。図14を用いて検討されたように、そのような動作は、デバイスによってブートの際に実行されるべきデバイスのためのイメージを提供することを含み得る。そのような動作は、デバイスに関して実施されるべきセキュリティにも関連することができる。例えば、ブートファームウェアによって送信されたメッセージが、デバイスのスタックのその部分が最新の更新以降変更されたことを示す場合、デバイスで実行されるべき動作は、デバイスのデータを一扫すること、またはデバイスのフラッシュメモリもしくはハードドライブなどのストレージ構造を再フォーマットすることを含み得る。同様の動作が、デバイスが盗まれたという、デバイスのユーザからの外部の指示に回答して実行され得る。したがって、デバイスにおける動作は、デバイスからブート時に受信された情報に回答するか、またはデバイスから受信されないが、レポート内のユーザの呼出しなどの外部ソースから受信されるその他の情報に回答することができる。

10

【0132】

ボックス1514において、クライアントデバイスは、データおよび命令を受信し、ボックス1516において、クライアントデバイスは、ブートファームウェアを用いて、受信されたデータを処理し、および/または受信された命令を実行する。受信され得るその他の種類の命令は、デバイスが回復され得るようにデバイスについての情報をレポートして返す命令を含む。例えば、デバイスは、上で検討されたように、デバイスを使っている泥棒の画像をキャプチャするように試みて、搭載されたカメラを用いてデジタル写真を撮影することができ、その写真を、デバイスの位置を示すGPSデータと一緒に送り返すことができる。

20

【0133】

図16は、ホストされるコンピュータシステムに集中的に記憶されるデータのコンピューティングデバイスへのキャッシュを行うためのプロセスの流れ図である。概して、プロセスは、どのようにしてウェブベースのコンピューティングデバイスのための特定のデータがホストされるサーバシステムに通常記憶され得、ただし、ホストされるサーバシステムと通信し、ウェブブラウザのウェブアプリケーションを含むデバイス上のアプリケーションを動作させるためにホストされるサーバシステムに頼るクライアントデバイスにキャッシュされる可能性もあるのかを示す。

30

【0134】

プロセスは、クライアントデバイスとホストされるサーバデバイスとがデバイスに関するアカウントに基づく情報を記憶するボックス1602および1604で始まる。例えば、それぞれのデバイスまたはシステムは、デバイスをホストされるサーバシステムの特定のユーザアカウントに相互に関連付けるデバイスに関するユーザIDを記憶することができる。ボックス1606において、クライアントデバイスは、サーバ側の情報を要求し、ボックス1608において、ホストされるサーバシステムは、要求された情報を取得し、その情報をクライアントデバイスに送信する。ホストされるサーバシステムは、将来、要求の情報に対してなされる変更を特定するための情報も記憶することができる。特定の実装形態において、ホストされるサーバシステムは、編集のためにクライアントデバイスに提供される任意の情報が編集されることを想定することができる。したがって、ホストされるサーバシステムは、そのようなデータがクライアントデバイスに渡されるときにそのようなデータをデータであるものとしてマークするか、または少なくとも暫定的にマークすることができる。ボックス1612において、クライアントデバイスは、要求された情報を受信し、クライアントデバイスとのユーザのインタラクションに回答して、要求された情報を修正することができる。適切なときに、修正された情報は、クライアントデバイスからホストされるサーバシステムに送り返される。そのような適切なときは、周期的にサーバシステムに情報を

40

50

返すクロックによって、またはアプリケーションにおいて定義された制御をユーザが選択することなどの特定のイベントによって決定され得る。

【0135】

ボックス1616において、サーバシステムは、修正された情報を受信し、ボックス1618において、記憶された情報を用いて要求された情報に対する変更を特定する。例えば、システムは、クライアントデバイスから返ってきて受信される情報を、前にダーティであるとマークされた情報と比較して、クライアントデバイスに与えられた情報に対して何らかの関連する変更が行われたかどうかを判定することができる。ボックス1620において、サーバシステムは、情報のサブセットを更新されたサブセットで置き換えることができる。したがって、例えば、システムが、システム上のダーティなデータとクライアントデバイス

10

【0136】

図17は、本明細書に記載の技法で使用され得る汎用的なコンピュータデバイス1700および汎用的なモバイルコンピュータデバイス1750の例を示す。コンピューティングデバイス1700は、ラップトップ、デスクトップ、ワークステーション、携帯情報端末、サーバ、ブレードサーバ、メインフレーム、およびその他の適切なコンピュータなどのさまざまな形態のデジタルコンピュータを表すように意図される。コンピューティングデバイス1750は、携帯情報端末、セルラ電話、スマートフォン、およびその他の同様のコンピューティングデバイスなどのさまざまな形態のモバイルデバイスを表すように意図される。本明細書

20

【0137】

コンピューティングデバイス1700は、プロセッサ1702と、メモリ1704と、ストレージデバイス1706と、メモリ1704および高速拡張ポート1710に接続する高速インターフェース1708と、低速バス1714およびストレージデバイス1706に接続する低速インターフェース1712とを含む。コンポーネント1702、1704、1706、1708、1710、および1712のそれぞれは、さまざまなバスを用いて相互接続されており、共通のマザーボードに搭載されるか、または

30

【0138】

メモリ1704は、コンピューティングデバイス1700内で情報を記憶する。1つの実装形態

40

【0139】

ストレージデバイス1706は、コンピューティングデバイス1700に大容量記憶を提供することができる。1つの実装形態において、ストレージデバイス1706は、フロッピー（登録商標）ディスクデバイス、ハードディスクデバイス、光ディスクデバイス、またはテープデバイス、フラッシュメモリもしくはその他の同様のソリッドステートメモリデバイス、またはストレージエリアネットワークもしくはその他の構成内のデバイスを含む一群のデ

50

バイスなどのコンピュータ可読媒体であるか、またはそのようなコンピュータ可読媒体を含んでもよい。コンピュータプログラム製品が、情報担体で有形で具現化され得る。コンピュータプログラム製品は、実行されるときに上述の方法などの1つまたは複数の方法を実行する命令を含み得る。情報担体は、メモリ1704、ストレージデバイス1706、プロセッサ1702上のメモリ、または伝播信号などのコンピュータ可読媒体または機械可読媒体である。

【0140】

高速コントローラ1708が、コンピューティングデバイス1700に関する帯域を大量に消費する動作を管理する一方、低速コントローラ1712は、帯域をそれほど消費しない動作を管理する。そのような機能の割当ては、例示に過ぎない。1つの実装形態において、高速コントローラ1708は、メモリ1704と、(例えば、グラフィックプロセッサまたはグラフィックアクセラレータを介して)ディスプレイ1716と、さまざまな拡張カード(図示せず)を受け入れることができる高速拡張ポート1710とに接続される。その実装形態において、低速コントローラ1712は、ストレージデバイス1706および低速拡張ポート1714に接続される。さまざまな通信ポート(例えば、USB、Bluetooth(登録商標)、イーサネット(登録商標)、ワイヤレスイーサネット(登録商標))を含み得る低速拡張ポートは、キーボード、ポインティングデバイス、スキャナなどの1つまたは複数の入力/出力デバイスと、または例えばネットワークアダプタを介してスイッチもしくはルータなどのネットワークデバイスと接続され得る。

【0141】

コンピューティングデバイス1700は、図に示されるように、多くの異なる形態で実装され得る。例えば、コンピューティングデバイス1700は、1つの標準的なサーバ1720として実装されるか、または一群のそのようなサーバ内で複数回実装され得る。コンピューティングデバイス1700は、ラックサーバシステム1724の一部として実装され得る。さらに、コンピューティングデバイス1700は、ラップトップコンピュータ1722などのパーソナルコンピュータで実装され得る。代替的に、コンピューティングデバイス1700のコンポーネントは、デバイス1750などのモバイルデバイスのその他のコンポーネント(図示せず)と組み合わされてもよい。そのようなデバイスのそれぞれは、コンピューティングデバイス1700、1750のうちの1つまたは複数を含んでもよく、システム全体が、互いに通信する複数のコンピューティングデバイス1700、1750によって構成されていてもよい。

【0142】

コンピューティングデバイス1750は、コンポーネントの中でもとりわけ、プロセッサ1752、メモリ1764、ディスプレイ1754などの入力/出力デバイス、通信インターフェース1766、およびトランシーバ1768を含む。デバイス1750は、追加的なストレージを提供するために、マイクロドライブまたはその他のデバイスなどのストレージデバイスを備え得る。コンポーネント1750、1752、1764、1754、1766、および1768のそれぞれは、さまざまなバスを用いて相互接続されており、それらのコンポーネントのうちのいくつかは、共通のマザーボードに搭載されるか、または適宜その他の方法で搭載され得る。

【0143】

プロセッサ1752は、メモリ1764に記憶された命令を含むコンピューティングデバイス1750内の命令を実行することができる。プロセッサは、別々の複数のアナログおよびデジタルプロセッサを含むチップのチップセットとして実装され得る。プロセッサは、例えば、ユーザインターフェース、デバイス1750によって実行されるアプリケーション、およびデバイス1750によるワイヤレス通信の制御などの、デバイス1750のその他のコンポーネントの調整を行うことができる。

【0144】

プロセッサ1752は、ディスプレイ1754に接続された制御インターフェース1758およびディスプレイインターフェース1756を介して、ユーザとコミュニケーションすることができる。ディスプレイ1754は、例えば、TFT LCD(薄膜トランジスタ液晶ディスプレイ)もしくはOLED(有機発光ダイオード)ディスプレイ、またはその他の適切なディスプレイテクノ

10

20

30

40

50

ロジーであってもよい。ディスプレイインターフェース1756は、ユーザに対してグラフィカルな情報およびその他の情報を提示するようにディスプレイ1754を駆動するための適切な回路を含むことができる。制御インターフェース1758は、ユーザからコマンドを受信し、そのコマンドを、プロセッサ1752に送るために変換することができる。さらに、デバイス1750とその他のデバイスとの近距離通信を可能にするために、プロセッサ1752と通信する外部インターフェース1762が設けられ得る。外部インターフェース1762は、例えば、一部の実装形態においては有線通信を、またはその他の実装形態においてはワイヤレス通信を提供することができ、複数のインターフェースが使用され得る。

【0145】

メモリ1764は、コンピューティングデバイス1750内で情報を記憶する。メモリ1764は、1つのコンピュータ可読媒体もしくは複数のコンピュータ可読媒体、1つの揮発性メモリユニットもしくは複数の揮発性メモリユニット、または1つの不揮発性メモリユニットもしくは複数の不揮発性メモリユニットのうちの1つまたは複数として実装され得る。また、拡張メモリ1774が設けられ、例えば、SIMM(シングルインラインメモリモジュール(Single In Line Memory Module))カードインターフェースを含み得る拡張インターフェース1772を介してデバイス1750に接続され得る。そのような拡張メモリ1774は、デバイス1750に追加的な記憶空間を提供することができ、またはデバイス1750に関するアプリケーションまたはその他の情報を記憶することができる。具体的には、拡張メモリ1774は、上述のプロセスを実行または補足する命令を含むことができ、安全な情報を含むこともできる。したがって、例えば、拡張メモリ1774は、デバイス1750のセキュリティモジュールとして設けられてもよく、デバイス1750の安全な使用を可能にする命令でプログラムされ得る。さらに、ハッキングできない方法でSIMMカードに識別情報を置くなど、追加的な情報とともに、安全なアプリケーションがSIMMカードによって提供され得る。

【0146】

メモリは、例えば、以下で検討されるように、フラッシュメモリおよび/またはNVRAMメモリを含み得る。1つの実装形態においては、コンピュータプログラム製品が、情報担体で有形で具現化される。コンピュータプログラム製品は、実行されるときに上述の方法などの1つまたは複数の方法を実行する命令を含む。情報担体は、メモリ1764、拡張メモリ1774、プロセッサ1752上のメモリ、または例えばトランシーバ1768もしくは外部インターフェース1762を介して受信され得る伝播信号などのコンピュータ可読媒体または機械可読媒体である。

【0147】

デバイス1750は、必要に応じてデジタル信号処理回路を含み得る通信インターフェース1766を介してワイヤレス通信することができる。通信インターフェース1766は、とりわけ、GSM(登録商標)音声電話、SMS、EMS、またはMMSメッセージ通信、CDMA、TDMA、PDC、WCDMA(登録商標)、CDMA2000、またはGPRSなどのさまざまな方式またはプロトコルでの通信を行うことができる。そのような通信は、例えば、無線周波数トランシーバ1768を介して行われ得る。さらに、近距離通信が、Bluetooth(登録商標)、WiFi、またはその他のそのようなトランシーバ(図示せず)を用いるなどして行われ得る。加えて、GPS(全地球測位システム)受信機モジュール1770が、デバイス1750で実行されているアプリケーションによって適宜使用され得るさらなるナビゲーションおよび位置に関連するワイヤレスデータをデバイス1750に提供することができる。

【0148】

デバイス1750は、ユーザから発話された情報を受信し、その情報を使用可能なデジタル情報に変換することができる音声コーデック1760を用いて音声通信することもできる。同じく、音声コーデック1760は、例えば、デバイス1750のハンドセットのスピーカを介するなどして、ユーザのための聞くことができる音声を生成することができる。そのような音声は、音声電話通話の音声を含んでも、記録された音声(例えば、ボイスメッセージ、音楽ファイルなど)を含んでも、デバイス1750上で動作しているアプリケーションによって生成された音声も含んでもよい。

【0149】

コンピューティングデバイス1750は、図に示されるように、多くの異なる形態で実装され得る。例えば、コンピューティングデバイス1750は、セルラ電話1780として実装され得る。また、コンピューティングデバイス1750は、スマートフォン1782、携帯情報端末、またはその他の同様のモバイルデバイスの一部として実装され得る。

【0150】

本明細書に記載のシステムおよび技法のさまざまな実装形態は、デジタル電子回路、集積回路、特別に設計されたASIC(特定用途向け集積回路)、コンピュータハードウェア、ファームウェア、ソフトウェア、および/またはそれらの組合せで実現され得る。これらのさまざまな実装形態は、ストレージシステム、少なくとも1つの入力デバイス、および少なくとも1つの出力デバイスからデータおよび命令を受信し、それらにデータおよび命令を送信するために接続された、専用であっても、または汎用であってもよい少なくとも1つのプログラム可能なプロセッサを含むプログラム可能なシステム上の、実行可能および/または解釈可能な1つまたは複数のコンピュータプログラムでの実装形態を含み得る。

【0151】

これらのコンピュータプログラム(プログラム、ソフトウェア、ソフトウェアアプリケーション、またはコードとしても知られる)は、プログラム可能なプロセッサ用の機械命令を含み、高級手続き型プログラミング言語および/もしくはオブジェクト指向プログラミング言語、ならびに/またはアセンブリ/機械言語で実装され得る。本明細書で使用される場合、用語「機械可読媒体」、「コンピュータ可読媒体」とは、機械命令を機械可読信号として受信する機械可読媒体を含む、プログラム可能なプロセッサに機械命令および/またはデータを提供するために使用される任意のコンピュータプログラム製品、装置、および/またはデバイス(例えば、磁気ディスク、光ディスク、メモリ、プログラマブルロジックデバイス(PLD))を指す。用語「機械可読信号」とは、プログラム可能なプロセッサに機械命令および/またはデータを提供するために使用される任意の信号を指す。

【0152】

ユーザとのインタラクションを行うために、本明細書に記載のシステムおよび技法は、ユーザに対して情報を表示するためのディスプレイデバイス(例えば、CRT(ブラウン管)またはLCD(液晶ディスプレイ)モニタ)、ならびにユーザがコンピュータに入力を与えることができるキーボードおよびポインティングデバイス(例えば、マウスまたはトラックボール)を有するコンピュータ上に実装され得る。その他の種類のデバイスが、ユーザとのインタラクションを行うためにさらに使用され得、例えば、ユーザに提供されるフィードバックは、任意の形態の感覚フィードバック(例えば、視覚的フィードバック、聴覚的フィードバック、または触覚的フィードバック)であってもよく、ユーザからの入力、音響、発話、または触覚による入力を含む任意の形態で受け取ることができる。

【0153】

本明細書に記載のシステムおよび技法は、バックエンドコンポーネントを(例えば、データサーバとして)含むか、またはミドルウェアコンポーネント(例えば、アプリケーションサーバ)を含むか、またはフロントエンドコンポーネント(例えば、ユーザが本明細書に記載のシステムおよび技法の実装形態とインタラクションすることができるグラフィカルユーザインターフェースまたはウェブブラウザを有するクライアントコンピュータ)を含むか、またはそのようなバックエンドコンポーネント、ミドルウェアコンポーネント、もしくはフロントエンドコンポーネントの任意の組合せを含むコンピューティングシステムで実装され得る。システムのコンポーネントは、任意の形態または媒体のデジタルデータ通信(例えば、通信ネットワーク)によって相互接続され得る。通信ネットワークの例には、ローカルエリアネットワーク(「LAN」)、広域ネットワーク(「WAN」)、およびインターネットが挙げられる。

【0154】

コンピューティングシステムは、クライアントおよびサーバを含み得る。クライアントおよびサーバは、概して互いに離れており、通常は通信ネットワークを介してインタラク

10

20

30

40

50

ションする。クライアントとサーバとの関係は、それぞれのコンピュータ上で実行されており、互いにクライアント-サーバの関係にあるコンピュータプログラムによって生じる。

【0155】

いくつかの実施形態が説明されてきた。しかしながら、本発明の精神および範囲から逸脱することなくさまざまな修正がなされ得ることが理解されるであろう。例えば、本明細書の多くの部分は、テレビ広告に関して説明されたが、ラジオ広告およびオンラインビデオ広告などのその他の形態の将来的な視聴者に基づく広告も、対応され得る。

【0156】

加えて、図面に示された論理フローは、所望の結果を達成するために、示された特定の順序、または逐次の順序である必要はない。さらに、その他のステップが設けられても、またはステップが説明されたフローから削除されてもよく、その他のコンポーネントが説明されたシステムに追加されても、または説明されたシステムから削除されてもよい。したがって、その他の実施形態は、添付の特許請求の範囲の範囲内にある。

10

【符号の説明】

【0157】

- 202 コンピュータ
- 204 アイコン、オブジェクト
- 206 ボックス
- 208 アイコン、オブジェクト、文書
- 210 ボックス
- 212 ウィジェット、ガジェット
- 214 ニュースフィールド
- 220 アプリケーションモニタ
- 224 アイコン、オブジェクト、チャットウィンドウ、階層
- 226 ボックス
- 228 アイコン、オブジェクト、画像ファイル
- 230 ボックス

20

【 図 1 】

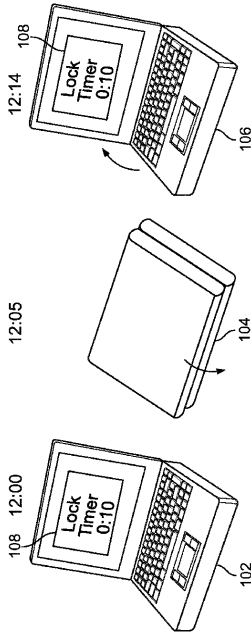


FIG. 1

【 図 2 A 】

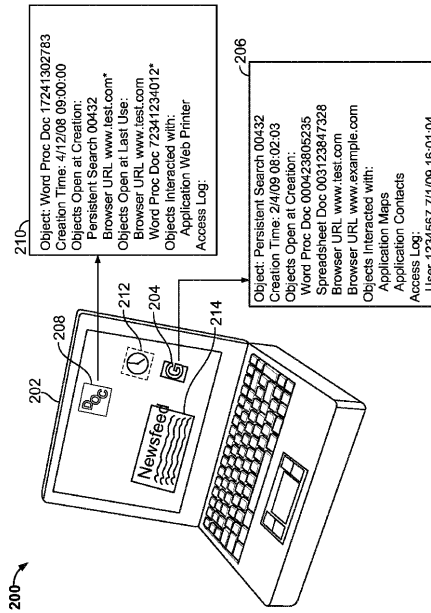


FIG. 2A

【 図 2 B 】

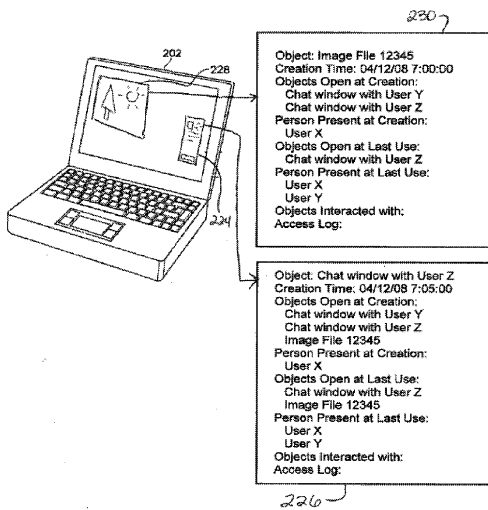
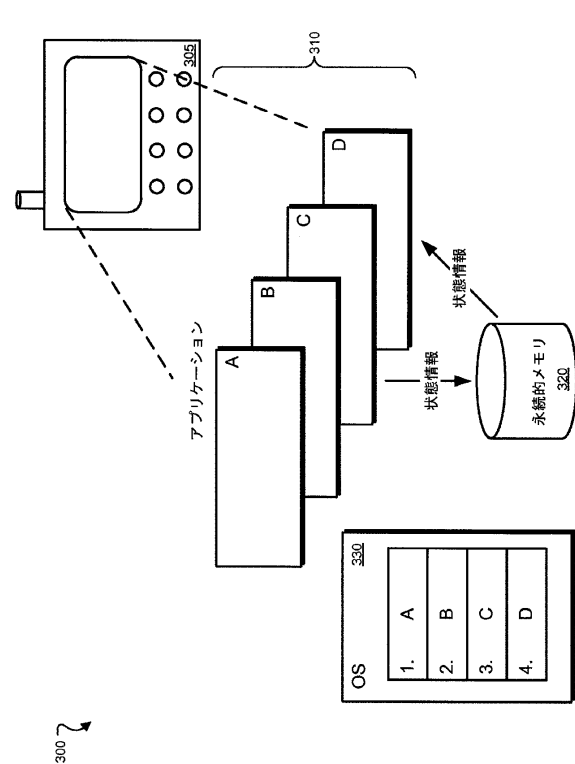
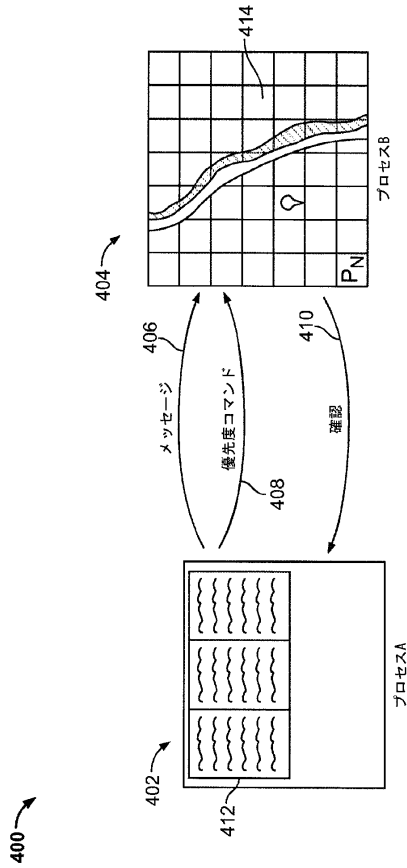


FIG. 2B

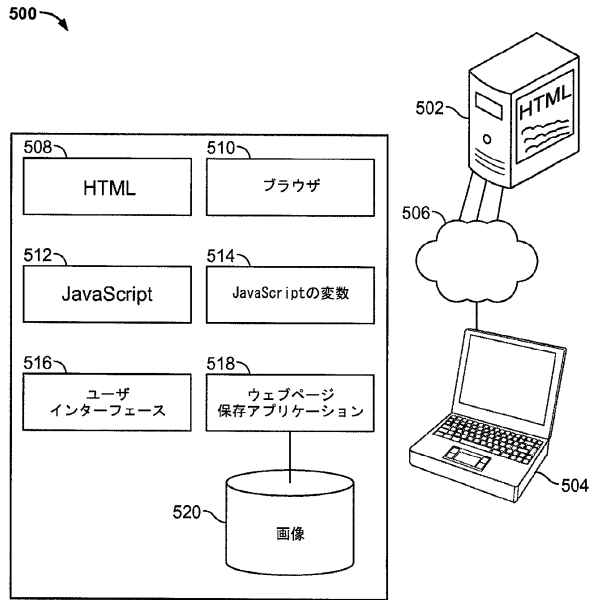
【 図 3 】



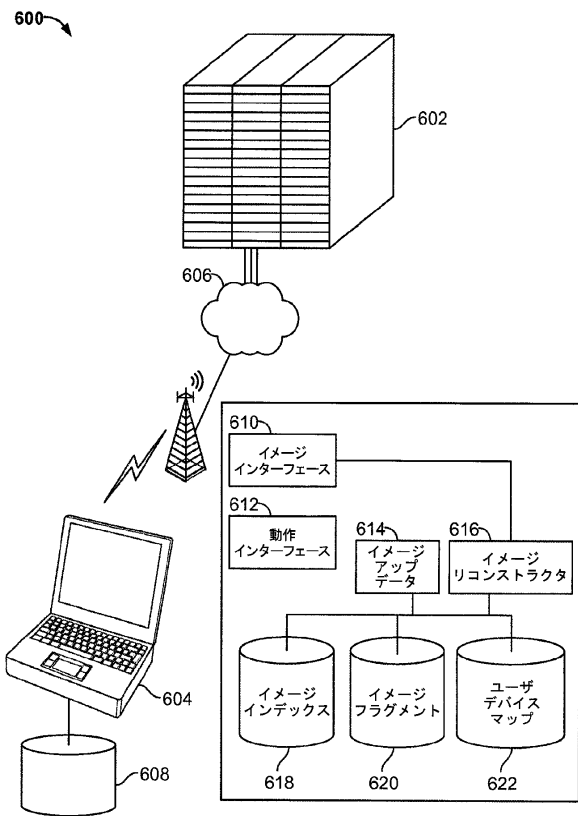
【図4】



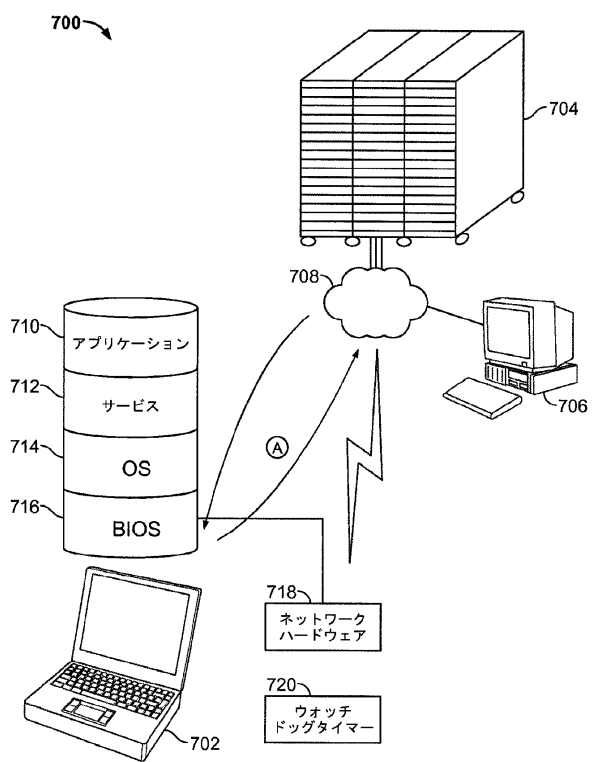
【図5】



【図6】



【図7】



【図8】

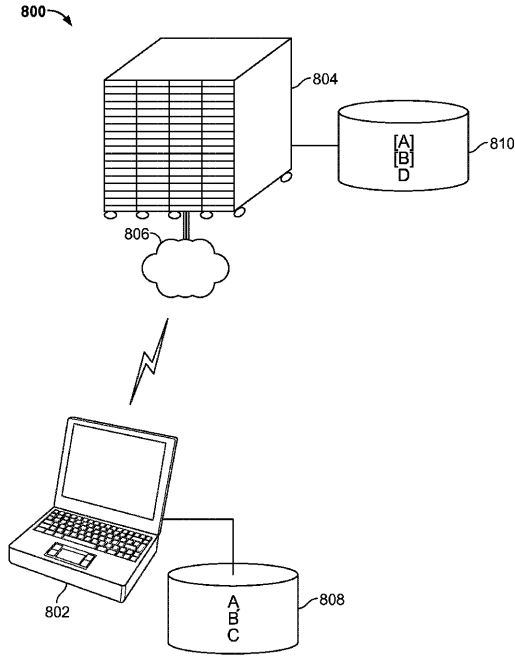
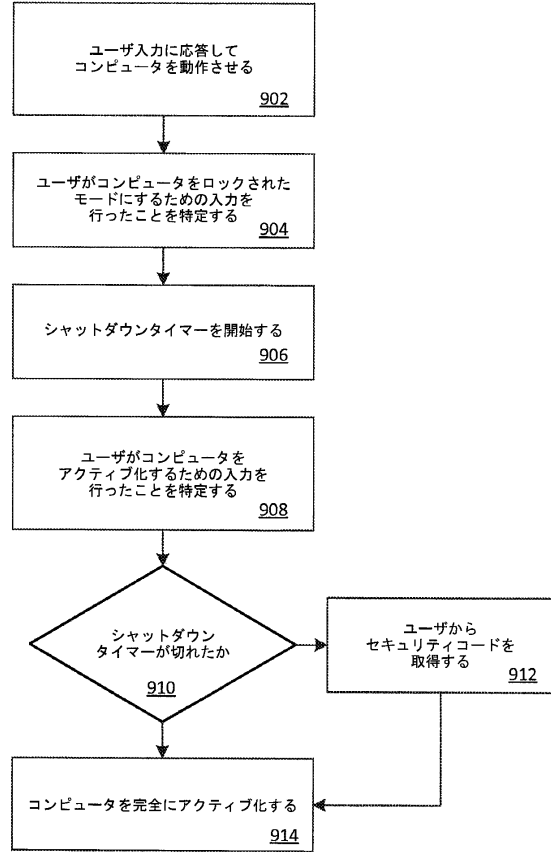
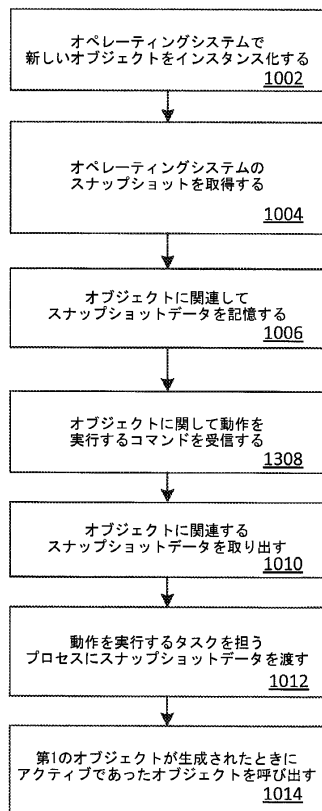


FIG. 8

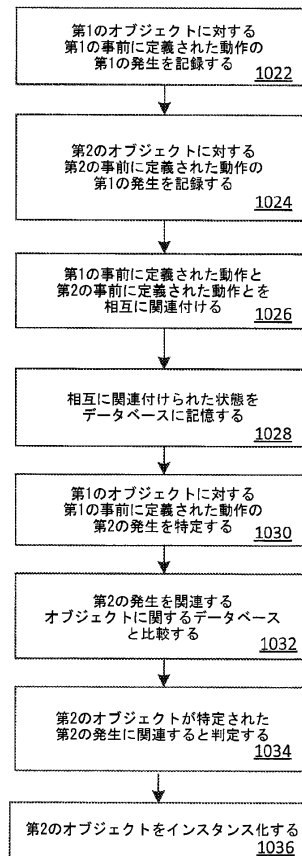
【図9】



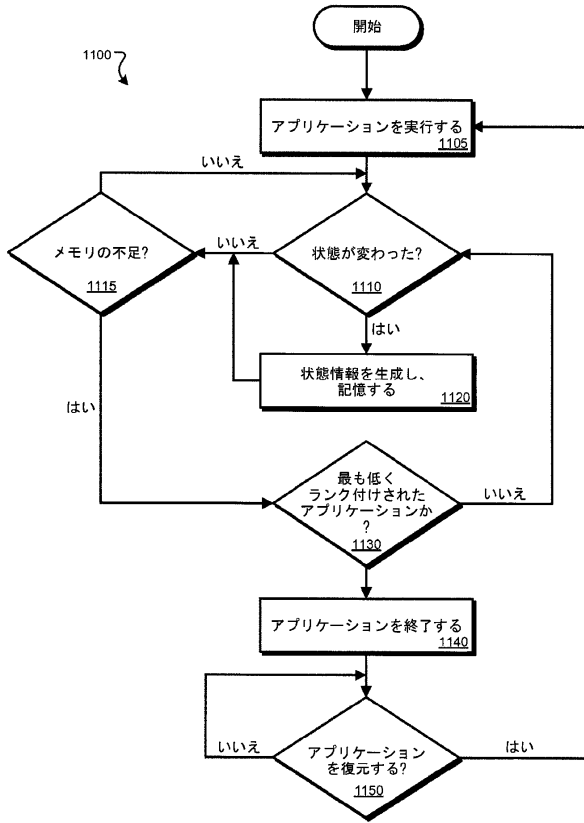
【図10A】



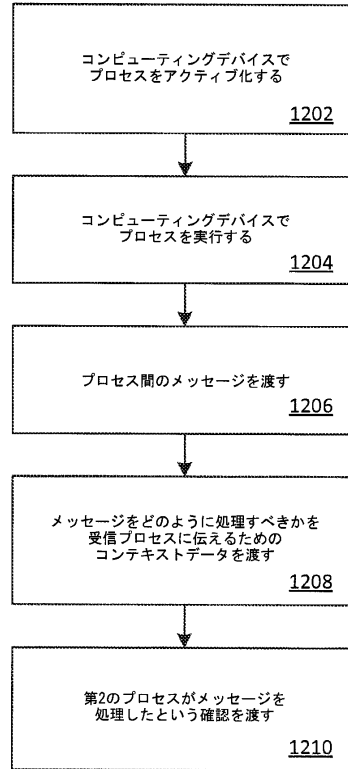
【図10B】



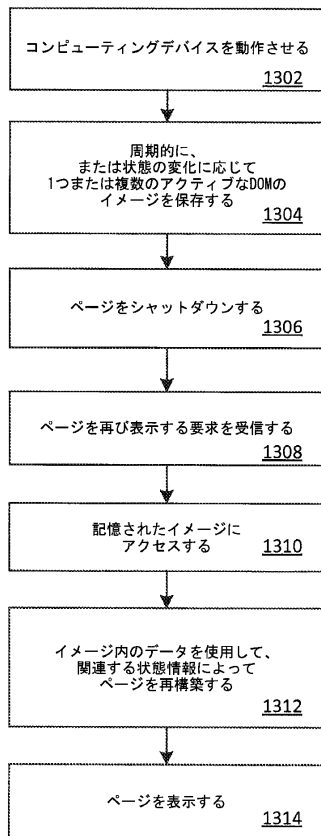
【図11】



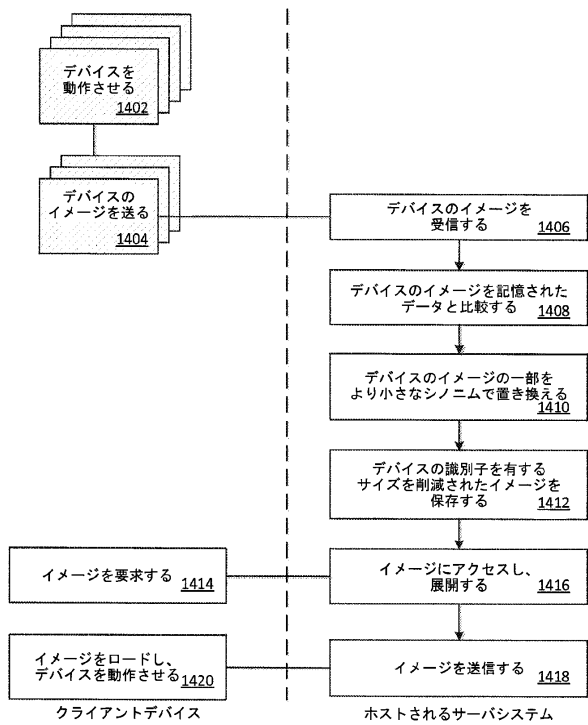
【図12】



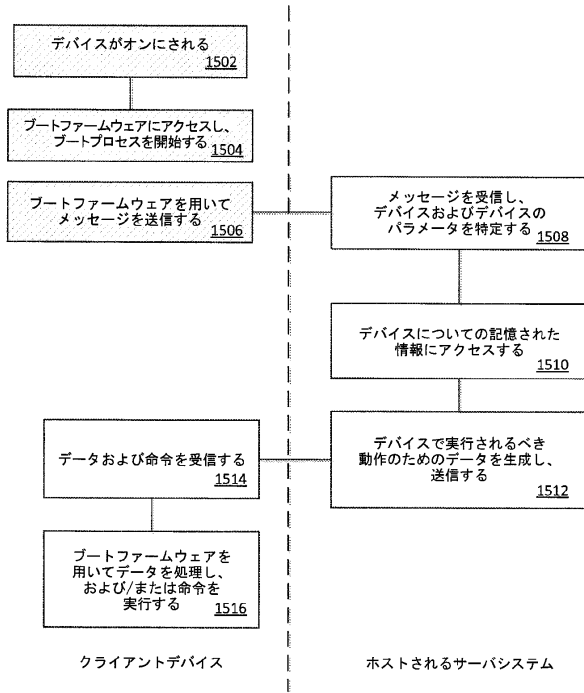
【図13】



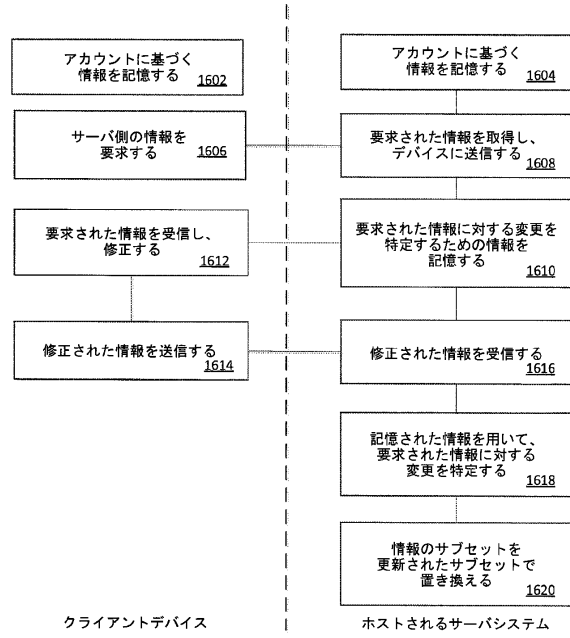
【図14】



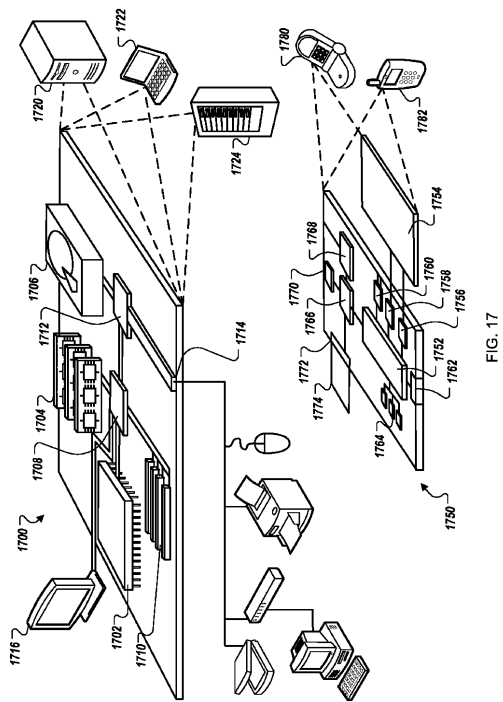
【図15】



【図16】



【図17】



フロントページの続き

(72)発明者 カン・リウ

アメリカ合衆国・カリフォルニア・94040・マウンテン・ビュー・リンコン・ストリート・9
27

審査官 坂庭 剛史

(56)参考文献 特開2005-227951(JP,A)

特開2000-099076(JP,A)

特開2006-287458(JP,A)

特開2005-005909(JP,A)

特開平10-333926(JP,A)

特開平09-006582(JP,A)

特開平08-314679(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/445