

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4522734号
(P4522734)

(45) 発行日 平成22年8月11日 (2010. 8. 11)

(24) 登録日 平成22年6月4日 (2010. 6. 4)

(51) Int. Cl.

F I

G 0 6 F 9 / 5 4 (2006. 01)

G 0 6 F 9 / 4 6 4 8 0 C

請求項の数 48 (全 29 頁)

(21) 出願番号 特願2004-93773 (P2004-93773)
 (22) 出願日 平成16年3月26日 (2004. 3. 26)
 (65) 公開番号 特開2004-295899 (P2004-295899A)
 (43) 公開日 平成16年10月21日 (2004. 10. 21)
 審査請求日 平成19年3月23日 (2007. 3. 23)
 (31) 優先権主張番号 10/402, 179
 (32) 優先日 平成15年3月26日 (2003. 3. 26)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 500046438
 マイクロソフト コーポレーション
 アメリカ合衆国 ワシントン州 9805
 2-6399 レッドモンド ワン マイ
 クロソフト ウェイ
 (74) 代理人 100077481
 弁理士 谷 義一
 (74) 代理人 100088915
 弁理士 阿部 和夫
 (72) 発明者 ティモシー ピー. マッキー
 アメリカ合衆国 98112 ワシントン
 州 シアトル 35 アベニュー イース
 ト 165

最終頁に続く

(54) 【発明の名称】 テスト通知を利用するシステムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

ユーザに通知を配信するコンピュータシステムにおいて、テスト通知を利用するコンピ
ュータ実施方法であって、

実際の通知をシミュレートするテスト通知を、アプリケーションプログラムによって呼
び出されたテスト通知インターフェースから受け取るステップと、

前記テスト通知を評価するが、前記テスト通知を前記アプリケーションプログラムのユ
ーザに配信しないステップと、

前記テスト通知を評価した結果についての指示を前記アプリケーションプログラムに戻
すステップと

を含むことを特徴とする方法。

【請求項 2】

前記シミュレーションの結果は、前記実際の通知が前記アプリケーションプログラムの
前記ユーザに配信されるか否かを指示することを特徴とする請求項 1 に記載の方法。

【請求項 3】

前記テスト通知を評価するステップは、ユーザコンテキストに従って行われることを特
徴とする請求項 1 に記載の方法。

【請求項 4】

前記ユーザコンテキストは、真または偽である条件と、前記条件が真である場合に実行
されるべき命令とを含むことを特徴とする請求項 3 に記載の方法。

【請求項 5】

前記ユーザコンテキストの条件は、通知を受け取ることについての前記アプリケーションプログラムの前記ユーザの連絡可能性であり、前記命令は、前記アプリケーションプログラムの前記ユーザに実際の通知を配信すべきか否かであることを特徴とする請求項 4 に記載の方法。

【請求項 6】

前記テスト通知を評価するステップは、指定された特定の内容を含むテスト通知の対処法を指示するユーザ規則に従って行われることを特徴とする請求項 1 に記載の方法。

【請求項 7】

追加的なテスト通知を利用して前記コンピュータシステムを再ボールし、ユーザコンテキストについて、指定された変更がいつ生じたかを判定するように、前記追加的なテスト通知をポーリングするステップをさらに含むことを特徴とする請求項 1 に記載の方法。

10

【請求項 8】

前記コンピュータシステムが、ユーザコンテキストについて変化が生じた場合に前記アプリケーションプログラムに指示を自動的に提供するように、ユーザコンテキストについての変化をサブスクライブするステップをさらに含むことを特徴とする請求項 1 に記載の方法。

【請求項 9】

前記テスト通知を評価した結果についての前記アプリケーションプログラムへの前記指示は、配信の指示を含むことを特徴とする請求項 1 に記載の方法。

20

【請求項 10】

前記配信の指示は、真または偽であることを特徴とする請求項 9 に記載の方法。

【請求項 11】

前記テスト通知を評価した結果についての前記アプリケーションプログラムへの前記指示は、転送または延期の指示のうちの 1 つをさらに含むことを特徴とする請求項 9 に記載の方法。

【請求項 12】

前記テスト通知を評価した結果についての前記アプリケーションプログラムへの前記指示は、侵入度の指示をさらに含むことを特徴とする請求項 9 に記載の方法。

【請求項 13】

前記テスト通知を評価した結果についての前記アプリケーションプログラムへの前記指示は、音量の指示をさらに含むことを特徴とする請求項 9 に記載の方法。

30

【請求項 14】

テスト通知を利用するためのコンピュータプログラムを記憶したコンピュータ可読媒体であって、前記コンピュータプログラムがコンピュータによって実行されると実施される方法は、

実際の通知をシミュレートするテスト通知を、アプリケーションプログラムによって呼び出されたテスト通知インターフェースから受け取るステップと、

前記テスト通知を評価するが、前記テスト通知を前記アプリケーションプログラムのユーザに配信しないステップと、

40

前記テスト通知を評価した結果についての指示を前記アプリケーションプログラムに戻すステップと

を含むことを特徴とするコンピュータ可読媒体。

【請求項 15】

前記シミュレーションの結果は、前記実際の通知が前記アプリケーションプログラムの前記ユーザに配信されるか否かを指示することを特徴とする請求項 14 に記載のコンピュータ可読媒体。

【請求項 16】

前記テスト通知を評価するステップは、条件と、前記条件に応じて実行されるべき命令とを含むユーザコンテキストに従って、行われることを特徴とする請求項 14 に記載のコ

50

ンピュータ可読媒体。

【請求項 17】

前記条件は真または偽であり、前記命令は前記条件が真である場合に実行されることを特徴とする請求項 16 に記載のコンピュータ可読媒体。

【請求項 18】

前記ユーザコンテキストの条件は、通知を受け取ることについての、前記アプリケーションプログラムの前記ユーザの連絡可能性であり、前記命令は、前記アプリケーションプログラムの前記ユーザに実際の通知を配信すべきか否かであることを特徴とする請求項 17 に記載のコンピュータ可読媒体。

【請求項 19】

前記テスト通知を評価するステップは、指定された特定の内容を含むテスト通知の対処法を指示するユーザ規則に従って行われることを特徴とする請求項 14 に記載のコンピュータ可読媒体。

【請求項 20】

追加的なテスト通知を利用して前記コンピュータ可読媒体を再ポーリングし、ユーザコンテキストについて、指定された変更がいつ生じたかを判定するように、前記追加的なテスト通知をポーリングするステップを実行するコンポーネントをさらに備えることを特徴とする請求項 14 に記載のコンピュータ可読媒体。

【請求項 21】

前記コンピュータ可読媒体が、ユーザコンテキストについて変化が生じた場合に前記アプリケーションプログラムに指示を自動的に提供するように、ユーザコンテキストについての変化をサブスクライブするステップを実行するコンポーネントをさらに備えることを特徴とする請求項 14 に記載のコンピュータ可読媒体。

【請求項 22】

前記テスト通知を評価した結果についての前記アプリケーションプログラムへの前記指示は、配信の指示、転送の指示、または延期の指示のうちの少なくとも1つを含むことを特徴とする請求項 14 に記載のコンピュータ可読媒体。

【請求項 23】

ユーザに通知を配信するコンピュータシステムにおいて、テスト通知を利用するコンピュータ実施方法であって、

実際の通知をシミュレートするテスト通知を、アプリケーションプログラムによって呼び出されたテスト通知インターフェースから受け取るステップと、

前記テスト通知を評価し、前記実際の通知が前記アプリケーションプログラムのユーザに配信されるか否かについて、前記アプリケーションプログラムに指示を提供するステップと

を含むことを特徴とする方法。

【請求項 24】

前記テスト通知を評価するステップは、ユーザコンテキストに従って行われることを特徴とする請求項 23 に記載の方法。

【請求項 25】

前記ユーザコンテキストは、真または偽である条件と、前記条件が真である場合に実行されるべき命令とを含むことを特徴とする請求項 24 に記載の方法。

【請求項 26】

前記ユーザコンテキストの条件は、通知を受け取ることについての前記アプリケーションプログラムの前記ユーザの連絡可能性であり、前記命令は、前記アプリケーションプログラムの前記ユーザに実際の通知を配信すべきか否かであることを特徴とする請求項 25 に記載の方法。

【請求項 27】

前記テスト通知を評価するステップは、指定された特定の内容を含む実際の通知の対処法を指示するユーザ規則に従って行われることを特徴とする請求項 23 に記載の方法。

【請求項 28】

追加的なテスト通知を利用して前記コンピュータシステムを再ポールし、ユーザコンテキストについて、指定された変更がいつ生じたかを判定するように、前記追加的なテスト通知をポーリングするステップをさらに含むことを特徴とする請求項 23 に記載の方法。

【請求項 29】

前記コンピュータシステムが、ユーザコンテキストについて変化が生じた場合に前記アプリケーションプログラムに指示を自動的に提供するように、ユーザコンテキストについての変化をサブスクライブするステップをさらに含むことを特徴とする請求項 23 に記載の方法。

【請求項 30】

前記実際の通知が前記アプリケーションプログラムの前記ユーザに配信されるか否かについての前記アプリケーションプログラムへの前記指示は、前記実際の通知の対処法についての追加的な指示を含むことを特徴とする請求項 23 に記載の方法。

【請求項 31】

テスト通知を利用するためのコンピュータプログラムを記憶したコンピュータ可読媒体であって、前記コンピュータプログラムがコンピュータによって実行されると実施される方法は、

実際の通知をシミュレートするテスト通知を、アプリケーションプログラムによって呼び出されたテスト通知インターフェースから受け取るステップと、

前記テスト通知を評価し、前記実際の通知が前記アプリケーションプログラムのユーザに配信されるか否かについての指示を前記アプリケーションプログラムに提供するステップと

を含むことを特徴とするコンピュータ可読媒体。

【請求項 32】

前記テスト通知を評価するステップは、ユーザコンテキストに従って行われることを特徴とする請求項 31 に記載のコンピュータ可読媒体。

【請求項 33】

前記ユーザコンテキストは、真または偽である条件と、前記条件が真である場合に実行されるべき命令とを含むことを特徴とする請求項 32 に記載のコンピュータ可読媒体。

【請求項 34】

前記ユーザコンテキストの条件は、通知を受け取ることについての前記アプリケーションプログラムの前記ユーザの連絡可能性であり、前記命令は、前記アプリケーションプログラムの前記ユーザに実際の通知を配信すべきか否かであることを特徴とする請求項 33 に記載のコンピュータ可読媒体。

【請求項 35】

前記テスト通知を評価するステップは、指定された特定の内容を含む実際の通知の対処法を指示するユーザ規則に従って行われることを特徴とする請求項 31 に記載のコンピュータ可読媒体。

【請求項 36】

追加的なテスト通知を利用して前記コンピュータ可読媒体を再ポールし、ユーザコンテキストについて、指定された変更がいつ生じたかを判定するように、前記追加的なテスト通知をポーリングするステップを実行するコンポーネントをさらに備えることを特徴とする請求項 31 に記載のコンピュータ可読媒体。

【請求項 37】

前記コンピュータ可読媒体が、ユーザコンテキストについて変化が生じた場合に前記アプリケーションプログラムに指示を自動的に提供するように、ユーザコンテキストについての変化をサブスクライブするステップを実行するコンポーネントをさらに備えることを特徴とする請求項 31 に記載のコンピュータ可読媒体。

【請求項 38】

前記実際の通知が前記アプリケーションプログラムの前記ユーザに配信されるか否かに

10

20

30

40

50

ついでの前記アプリケーションプログラムへの前記指示は、前記実際の通知の対処法についての追加的な指示をさらに含むことを特徴とする請求項 3 1 に記載のコンピュータ可読媒体。

【請求項 3 9】

1 つ以上のプロセッサと、
前記プロセッサによってアクセス可能なメモリと、
前記プロセッサによってアクセス可能な 1 つ以上の揮発性及び不揮発性記憶装置と、
通知を提供する通知ツールと
を備えるシステムであって、前記通知ツールは、
実際の通知をシミュレートするテスト通知を、アプリケーションプログラムによって呼び出されたテスト通知インターフェースから受け取る手段と、
前記テスト通知を評価するが、前記テスト通知を前記アプリケーションプログラムのユーザに配信しない手段と、
前記テスト通知を評価した結果についての指示を前記アプリケーションプログラムに戻す手段と
を含むことを特徴とするシステム。 10

【請求項 4 0】

前記通知ツールは、ユーザコンテキストに従って、前記テスト通知を評価する手段をさらに含むことを特徴とする請求項 3 9 に記載のシステム。

【請求項 4 1】

前記通知ツールは、ユーザ規則に従って、前記テスト通知を評価する手段をさらに含むことを特徴とする請求項 3 9 に記載のシステム。 20

【請求項 4 2】

前記通知ツールは、追加的なテスト通知をポーリングする手段をさらに備え、前記追加的なテスト通知を利用することにより前記システムを再ポーリングして、指定された変化が生じたかを判定することを特徴とする請求項 3 9 に記載のシステム。

【請求項 4 3】

前記通知ツールは、ユーザコンテキストについての変化をサブスクライブする手段をさらに備え、前記システムは、変化が生じた場合に前記アプリケーションプログラムに指示を自動的に提供することを特徴とする請求項 3 9 に記載のシステム。 30

【請求項 4 4】

1 つ以上のプロセッサと、
前記プロセッサによってアクセス可能なメモリと、
前記プロセッサによってアクセス可能な 1 つ以上の揮発性及び不揮発性記憶装置と、
通知を提供する通知ツールと
を備えるシステムであって、前記通知ツールは、
実際の通知をシミュレートするテスト通知を、アプリケーションプログラムによって呼び出されたテスト通知インターフェースから受け取る手段と、
前記テスト通知を評価し、前記実際の通知が前記アプリケーションプログラムのユーザに配信されるか否かについての指示を前記アプリケーションプログラムに提供する手段と
を含むことを特徴とするシステム。 40

【請求項 4 5】

前記通知ツールは、ユーザコンテキストに従って、前記テスト通知を評価する手段をさらに含むことを特徴とする請求項 4 4 に記載のシステム。

【請求項 4 6】

前記通知ツールは、ユーザ規則に従って、前記テスト通知を評価する手段をさらに含むことを特徴とする請求項 4 4 に記載のシステム。

【請求項 4 7】

前記通知ツールは、追加的なテスト通知をポーリングする手段をさらに備え、前記追加的なテスト通知を利用することにより前記システムを再ポーリングして、指定された変化がい 50

つ生じたかを判定することを特徴とする請求項 4 4 に記載のシステム。

【請求項 4 8】

前記通知ツールは、ユーザコンテキストについての変化をサブスクライブする手段をさらに備え、前記システムは、変化が生じた場合に前記アプリケーションプログラムに指示を自動的に提供することを特徴とする請求項 4 4 に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コンピューティングシステムにおける通知(notification)に関し、より詳細には、テスト通知(test notification)を利用するシステムおよび方法に関する。

10

【背景技術】

【0002】

コンピュータシステムでは、通知は、指定されたイベント(event)が発生したことをユーザに知らせるプログラムからの信号の形態をとることができる。そのような通知は、各種のテキスト、サウンド、およびグラフィックの要素を含むことができる。通知と共に、優先度(priority)、通知の送信者(電子メールまたはインスタントメッセージなどのチャネルの場合)、および通知の有効期限などの他のプロパティ(property)も含まれる可能性がある。通知は、何らかのコードの要素も含むことができ、ユーザは通知と対話し、任意のコードを起動(launch)することができる(例えば通知内のボタンまたはテキストをクリックすることにより新しいプログラムを起動させる、あるいは現在実行中のプログラムでアクションを行わせることができるなど)。

20

【発明の開示】

【発明が解決しようとする課題】

【0003】

オペレーティングシステムは、ユーザにネットワークの接続性と更新について知らせる通知(notification)を作成することができる。「連絡先リスト(contact list)」を使用するインスタントメッセージングプログラム(instant messaging program)は、画面上に通知をドロー(draw)して、連絡先リストの現在の状況、あるいは連絡先がいつインスタントメッセージの会話を開始したかをユーザに知らせることができる。他のプログラムは、ディスプレイの同様の領域にドローされる同様の通知を提供することができる。この種の通知に伴う問題の 1 つは、このような通知は一般に他の通知を認識せず、そのため他の通知の上に通知がドローされる場合があることである。

30

【0004】

既存の通知システムに伴う別の問題は、不適切な形で、あるいは不適切な時に通知を配信(deliver)する場合があることである。例えば、フルスクリーンの画面表示(presentation)を提供しているユーザにとって、その画面表示中に他のプログラムが画面に通知をドローするのは不適切である可能性がある。そのような不適切な通知をドローする可能性があるプログラムの一例はインスタントメッセージングプログラムであり、このプログラムは、オペレーティングシステムのバックグラウンドで実行され、連絡先リスト(contact list)中の連絡先がサインオンするか、またはインスタントメッセージ(instant message)を開始するとそのような通知をドローする。画面表示中のこの種の「割り込み(interruption)」は、ユーザにとって望ましくない可能性がある。

40

【0005】

本発明は、前述およびその他の不都合点を克服するシステムおよび方法を提供することを目的とする。より詳細には、本発明は、テスト通知を利用するシステムおよび方法を提供することを目的とする。

【課題を解決するための手段】

【0006】

本発明は、テスト通知を利用するシステムおよび方法を提供する。本発明の一態様によれば、呼び出しプログラム(calling program)は、実際の通知(actual notification)に似

50

たテスト通知(test notification)を構築する。実際の通知との大きな違いは、テスト通知は実際にはユーザに配信(deliver)されないことである。

【 0 0 0 7 】

本発明の別の態様によれば、テスト通知は、真または偽の指示(indication of true or false)を戻す(return)。真の指示(indication of true)は、実際の通知が現在ドロウされていたはずであることを意味し、偽の指示(indication of false)は、実際の通知が現在ドロウできなかったはずであることを意味する。別の実施形態では、テスト通知(test notification)によって、よりリッチな戻り値(richer return value)を戻す(return)ことができる。よりリッチな戻り値の例としては、通知(notification)が延期(defer)または経路指定(route)されたはずであったかどうか、通知が再生(play)されときのサウンドレベル(sound level)などが含まれる。

10

【 0 0 0 8 】

本発明の別の態様によれば、ポーリング法(polling method)が利用される。この形態において、呼び出しプログラムは周期的に再ポーリング(re-poll) (テスト通知を送信)することにより、ユーザの現在のコンテキスト(context)、および、呼び出しプログラム(calling program)によって送信される同報通信データ(broadcast data)をどのように変更すべきか、を判定(determine)する。同報通信データの一例は、インスタントメッセージングプログラムがビジーな状態またはフリーな状態を連絡先に同報通信するものである。別の形態では、サブスクリプションコールバック(subscription callback)の方法が利用される。この形態では、呼び出しプログラムは、コンテキスト変化(context change)の受け取りをサブスクライブ(subscribe)する。すなわち、ユーザのコンテキストは同報通信される情報のタイプに関連する形で変化するので、コンテキストエンジン(context engine)は、周期的に再ポーリングするのではなく、それらの更新(update)と共に呼び出しプログラムにコールバックする。この後者の形態には、コンテキストの変化と同報通信される内容に時間差がないという利点がある。他の事例では、1回限りのイベント(one-time event)などの場合にポーリング法がより適する場合がある。

20

【 0 0 0 9 】

本発明のこのテストシステムおよび方法により、ユーザにいつ通知(notificatin)を送信(send)するのが適切であるかをオペレーティングシステムおよび任意のプログラムが判定できることが理解されよう。テスト通知を利用する利点の1つは、プログラムがユーザコンテキスト(user's context)のよりリッチなビュー(richer view)を得られるようにすることにより、ユーザが通知を受け取れる状態になるまで不要な実際の通知(unwanted actual notification)の生成を防止できることである。また、テスト通知を利用することによりユーザコンテキストについての情報を得られるので、通知の配信(delivery of notification)に同じユーザインターフェースを利用しないプログラムに対してより高い柔軟性が与えられる。一例として、将来開発されるプログラムは、現在のユーザコンテキストシステム(user context system)ではサポートされない特殊化されたグラフィックを用いた通知を開発する可能性があるが、そのような将来のプログラムもテスト通知から得られる情報を利用して、現在の通知を送信または変更することが適切であるかどうかを判断することができる。

30

40

【 0 0 1 0 】

本発明の別の態様によれば、テスト通知はユーザコンテキストシステム中で利用される。ユーザコンテキストシステムは、複数ソースからの通知の配信(delivery of notification)を仲介(broker)し、順番(serialize)にする。また、各通知に適した対処法を判断するために、共有(share)されるユーザコンテキストの概念が提供される。これらの態様によれば、ユーザコンテキストシステムによって配信される通知は、ユーザがよりそれらを受け取れる状態にある時に送信されるのでより価値が高いと考えることができる。これらの態様は、ユーザが不要な通知が生成されないようにするのを助ける共通の規則も提供する。一形態では、ユーザコンテキストシステムにおけるテスト通知の使用は、基本的に、集約されたユーザコンテキスト(user's aggregated context)をカプセル化し、それを任

50

意のプロセスに公開する機能を持つシステムツールまたはシステムによって提供される機構を提供する。

【 0 0 1 1 】

本発明の別の態様によれば、ユーザコンテキストシステムでは、コンテキストはオペレーティングシステムおよび任意のプログラムによって宣言(declare)される。一形態では、ユーザコンテキストは、真(true)または偽(false)とすることができる条件(condition)と、その条件が真である場合に従うべき命令(instruction)とからなる。例えば、条件は「ユーザが音楽を聴いている時」であり、それについての命令は「通知は画面に配信するが、サウンドは生成しない」等である。一般に、ユーザコンテキストの条件は、通信を配信するのにユーザを何らかの形で連絡不可能にするとシステムがみなす状態、または通知が配信される方式を、その通知をスタートしたプログラムによって送信された方式とは異なるものにする状態と考えることができる。ユーザは「連絡不可能(unavailable)」とみなされる(deemed)状態にある可能性があり、その場合通知は配信されないか、またはユーザが「連絡可能(available)」になるまで保留(hold)にされる。例えばユーザがフルスクリーンアプリケーション(full screen application)を実行中で、そのアプリケーションが表示画面の全領域を使用しているか、全領域に表示されている場合、そのユーザは連絡不可能とみなされる場合がある。あるいは、ユーザは「連絡可能」であるが、ユーザに適した形に通知を変更する必要があるような状態にある場合もある。

10

【 0 0 1 2 】

本発明の別の態様によれば、ユーザコンテキストシステム(user context system)では、オペレーティングシステムがコンテキストを宣言(declare)するのに加えて、プログラムが、そのプログラムが提供するコンテキストとコンテキストの通知に対するその影響(画面へのドロワーが適切かどうか、画面へのドロワーに適した侵入度(invasiveness)のレベル、サウンドが適切な否か、あるいはサウンド再生の際の相対的な音量などに関して)をシステムに登録し、宣言し、そしてそのコンテキストが真であるか偽であることをシステムに知らせる。一形態では、コンテキストは、通知を配信しようとする時に真か偽かを評価(evaluate)されることもできる。一形態では、システムは呼び出しプログラムのプロセス(process of the calling program)を追跡(track)することもでき、プロセスがもはや存在しない場合はコンテキストが偽(false)にリセットされることができる。プロセスを追跡することにより望ましくない状況(undesirable situation)が回避(avoid)されることができ、例えば、アプリケーションがユーザがビジーであると宣言し、その後クラッシュ(crash)し、通知の受け取りができないと宣言された状態にユーザを置いたままにするとといった状況が回避される。

20

30

【 0 0 1 3 】

本発明の別の態様によれば、ユーザコンテキストシステムでは、通知のドロワーイング(drawing of notification)について指定された異なるレベルの侵入度(invasiveness)があってよい。すなわち、ユーザコンテキストに基づいて、通知をドロワーするための段階があってよく、ドロワーされる通知の形態に様々なレベルの侵入度があってよい。例えば、通常の通知(normal notification)では自由にクライアント領域(client area)中にドロワーされ、一時的にウィンドウを覆い隠してよい。ユーザがわずかに制限的なコンテキスト(a slightly restrictive context)にある場合は、通知は自由に表示(free to show)してよいが、別のウィンドウの上にドロワーすることは許されないなど、侵入度がより低い形でのみ表示される。別の例として、ユーザが最大化されたアプリケーション(maximized application)を実行中の場合は、ユーザが現在のアプリケーションにクライアント領域全体を使用したいと明確に指示していることから、ユーザコンテキストがわずかに制限されているという設定になることが考えられる。この状況では、通知のドロワーはなお許可されるが、サイドバー(sidebar)にのみ表示させることができる。この種の通知ドロワー形態の侵入性の低下(reduced invasiveness in the notification drawing form)は、通知の影響(impact of the notification)を小さくし、認識上の負荷(cognitive load)を小さくする。

40

【 0 0 1 4 】

50

本発明の別の態様によれば、ユーザコンテキストシステムでは、提供されたコンテキストはユーザに公開されることができ、オフにされることができる（例えばユーザがプログラムによるコンテキストの査定に同意しない場合など）か、または配信の影響の大きさの点から変更されることができる。

【 0 0 1 5 】

本発明の別の態様によれば、ユーザコンテキストシステムでは、ユーザは、指定された要素を含む通知をどのように配信すべきかを指示する規則を定義することができる。例えば、ユーザ規則(user rule)は、「J o h n D o e」から受信した通知で、件名に「緊急(urgent)」とある通知は直ちに配信することを指示することができる。一形態では、そのようなユーザ規則は、ユーザコンテキストに優先される。ユーザ規則は、ユーザがそのユーザの嗜好に応じて修正(modification)することができる。

10

【 0 0 1 6 】

前述の本発明の態様およびそれに伴う利点の多くは、添付の図面と併せて以下の詳細な説明を参照することによりそれらをより深く理解されるとより容易に理解されよう。

【発明を実施するための最良の形態】

【 0 0 1 7 】

本発明は、テスト通知を利用するシステムおよび方法を対象とする。一実施形態では、テスト通知は、ユーザコンテキストシステムによって処理されることができる。ユーザコンテキストシステムは、通知の配信を制御する。

【 0 0 1 8 】

20

従来のシステムでは、通常は、ユーザへの通知の送信を求める多数の競合する要素があり、それぞれの要素がそのような通知を送信する各自の方式を設計していた。一般には、競合する要素は他の通知を認識せず、そのため互いの通知および互いのアプリケーションの上にドロウする傾向があり、その結果、各要素が各自の通知の指示を同時にレンダリングすることを選択した時に衝突が生じる場合があった。また、共有されるユーザコンテキストの概念がなく、その結果一部の通知が不適切な形で、あるいは不適切な時に配信されていた。このユーザコンテキストシステムは、オペレーティングシステムのリッチな一部分として通知を構築することによりこれらの問題に対処し、システムがそれらの画面上でのレンダリングを適切に仲介し、順番にするので、ユーザコンテキストシステムによって提供される通知のユーザインターフェースが似たものになり、そのためインターフェース同士の衝突がなくなる。また、ユーザコンテキストシステムによって提供される通知は、ユーザが通知をより受け付けられる状況にある時に配信されるのでより価値が高くなると考えることができ、さらに、共通の規則を使用することによりユーザが不要な通知を解消するのを助ける。

30

【 0 0 1 9 】

上述のように、ユーザコンテキストシステムを利用して本発明のテスト通知を処理することができる。本発明のテスト通知は、現在の状況下で実際の通知が配信されるかどうかを判定するために使用される。すなわち、テスト通知は、ユーザが現在通知を受け取れる状態にあるかどうかの指示を提供するものである。この指示は、例えば、プログラムが現時点で実際の通知を送信しようとするのを妨げるか、あるいはプログラムに通知を部分的に変更させるために使用されることができる。

40

【 0 0 2 0 】

図 1 および以下の説明は、本発明を実施することが可能な適切なコンピューティング環境の簡単で概略的な説明を提供するものである。これは必須ではないが、本発明は、パーソナルコンピュータによって実行されるプログラムモジュールなどのコンピュータ実行可能命令の一般的なコンテキストで説明される。一般に、プログラムモジュールには、特定のタスクを行うか、特定の抽象データ型を実装するルーチン、プログラム、文字、コンポーネント、データ構造などが含まれる。当業者には理解されるように、本発明は、ハンドヘルド機器、マルチプロセッサシステム、マイクロプロセッサを使用した、あるいはプログラム可能な消費者家電製品、ネットワーク P C、ミニコンピュータ、メインフレームコ

50

ンピュータなどを含む他のコンピュータシステム構成で実施することが可能である。本発明は、通信ネットワークを通じてリンクされた遠隔の処理デバイスによってタスクが行われる分散コンピューティング環境で実施されることもできる。分散コンピューティング環境では、プログラムモジュールは、ローカルおよびリモート両方のメモリ記憶装置に配置されることができる。

【0021】

図1を参照すると、本発明を実施するための例示的システムは、従来のパーソナルコンピュータ20の形態の汎用コンピューティングデバイスを含み、これは、処理ユニット21、システムメモリ22、システムメモリ22を含む各種のシステム構成要素を処理ユニット21に結合するシステムバス23を含む。システムバス23は、各種のバスアーキテクチャの任意のものを使用したメモリバスまたはメモリコントローラ、周辺機器バス、およびローカルバスを含む数種のバス構造のいずれでもよい。システムメモリは、読み取り専用メモリ(ROM)24およびランダムアクセスメモリ(RAM)25を含む。起動時などにパーソナルコンピュータ20内の要素間の情報転送を助ける基本ルーチンを含む基本入出力システム(BIOS)26は、ROM24に記憶される。パーソナルコンピュータ20はさらに、ハードディスク39の読み書きを行うハードディスクドライブ27、取り外し可能磁気ディスク29の読み書きを行う磁気ディスクドライブ28、およびCD-ROMまたは他の光学媒体などの取り外し可能光ディスク31の読み書きを行う光ディスクドライブ30を含む。ハードディスクドライブ27、磁気ディスクドライブ28、および光ディスクドライブ30は、それぞれハードディスクドライブインターフェース32、磁気ディスクドライブインターフェース33、および光ドライブインターフェース34によってシステムバス23に接続される。これらのドライブとそれに関連付けられたコンピュータ可読媒体は、パーソナルコンピュータ20のコンピュータ可読命令、データ構造、プログラムモジュール、およびその他のデータの不揮発性の記憶を提供する。ここで説明する例示的環境は、ハードディスク39、取り外し可能磁気ディスク29、および取り外し可能光ディスク31を用いるが、当業者には理解されるように、磁気カセット、フラッシュメモリカード、デジタルビデオディスク、ベルヌーイカートリッジ、ランダムアクセスメモリ(RAM)、読み取り専用メモリ(ROM)など、コンピュータからアクセス可能なデータを格納することができる他の種のコンピュータ可読媒体もこの例示的動作環境で使用されることができる。

【0022】

ハードディスク39、磁気ディスク29、光ディスク31、ROM24またはRAM25にはいくつかのプログラムモジュールが記憶されることができ、これにはオペレーティングシステム35、1つまたは複数のアプリケーションプログラム36、他のプログラムモジュール37、およびプログラムデータ38が含まれる。ユーザは、キーボード40およびポインティングデバイス42などの入力装置を通じてパーソナルコンピュータ20にコマンドと情報を入力することができる。他の入力装置(図示せず)には、マイクロフォン、ジョイスティック、ゲームパッド、衛星パラボラアンテナ、スキャナなどが含まれる。これらおよびその他の入力装置は、システムバス23に結合されたシリアルポートインターフェース46を通じて処理ユニット21に接続されることが多いが、パラレルポート、ゲームポート、またはユニバーサルシリアルバス(USB)などの他のインターフェースで接続されることがもある。モニタ47の形態のディスプレイも、ビデオカードまたはアダプタ48などのインターフェースを介してシステムバス23に接続される。1つまたは複数のスピーカ57も、オーディオアダプタ56などのインターフェースを介してシステムバス23に接続されることができる。ディスプレイおよびスピーカに加えて、パーソナルコンピュータは通常、プリンタなどの他の周辺出力装置(図示せず)を含む。

【0023】

パーソナルコンピュータ20は、リモートコンピュータ49など1つまたは複数のパーソナルコンピュータへの論理接続を使用するネットワーク環境で動作することができる。リモートコンピュータ49は、別のパーソナルコンピュータ、サーバ、ルータ、ネットワ

ークPC、ピアデバイス、あるいはその他の一般的なネットワークノードでよく、通常は、パーソナルコンピュータ20との関係で上述した要素の多くまたはすべてを含む。図1に示す論理接続は、ローカルエリアネットワーク(LAN)51およびワイドエリアネットワーク(WAN)52を含む。このようなネットワーキング環境は、オフィス、企業内のコンピュータネットワーク、イントラネット、およびインターネットなどに一般的に見られる。

【0024】

LANネットワーキング環境で使用される場合、パーソナルコンピュータ20は、ネットワークインターフェースまたはアダプタ53を通じてローカルエリアネットワーク51に接続される。WANネットワーキング環境で使用される場合、パーソナルコンピュータ20は、通常は、インターネットなどのワイドエリアネットワーク52を通じて通信を確立するためのモデム54または他の手段を含む。モデム54は、内蔵型でも外付け型でもよく、シリアルポートインターフェース46を介してシステムバス23に接続される。ネットワーク環境では、パーソナルコンピュータ20との関係で図示したプログラムモジュール、またはその一部は、リモートのメモリ記憶装置に記憶されることができる。図のネットワーク接続は例示的なものであり、コンピュータ間に通信リンクを確立する他の手段を使用してよいことは理解されよう。

【0025】

本発明は、図1に示すタイプのシステムで実施することができる。上述のように、本発明のテスト通知は、ユーザコンテキストシステムによって評価されることができる。テスト通知を評価するユーザコンテキストシステムについて、以下の図2～図10について中心的に説明する。テスト通知の評価について、その後の項で図11～14との関係でより詳細に説明する。

【0026】

図2～図10において、複数ソースからの通知の配信を制御するユーザコンテキストシステムが提供される。一実施形態では、ユーザコンテキストシステムは、通知の処理法を決定するために比較される3つの要素からなることができる。第1の要素は、ユーザのコンテキストである(オペレーティングシステムと、オペレーティングシステムを拡張した任意のプログラムから提供されることができる)。第2の要素は、ユーザの規則と嗜好である。第3の要素は、通知自体(ユーザの規則に合致することが可能なデータおよびプロパティなどの要素を含む)である。

【0027】

下記でより詳細に説明するように、ユーザコンテキストシステムは、オペレーティングシステムおよび他のプログラムがユーザのコンテキストを宣言することによって動作し、コンテキストの宣言後は、システムがユーザのコンテキストと規則を仲介する。通知は、システム中に呼び出される他のプログラムによって生成される。ユーザのコンテキスト、規則、および通知の要素が比較され、通知の対処法に関する判定が行われる。通知の対処法についての各種のオプションの例には、拒否(通知がドローされる、あるいは音を立てることが許されない場合。その場合通知はユーザに見られない)、延期(ユーザのコンテキストが変化するか、ユーザの規則が、配信が後に適切になることを指示するまで通知が保留にされる)配信(ユーザのコンテキストおよび規則に従って通知の配信が許可される)、および転送(通知が現在のシステムでの配信も許可されるかどうかに関係なく、ユーザの規則が通知を別のシステムに渡すことを指示する)すること、が含まれる。

【0028】

以下、通知を配信する各種ルーチンについてより詳細に説明する。一般に、ユーザは「連絡不可能」と考えられる状態にある可能性があり、その場合通知は配信されないか、ユーザが「連絡可能」になるまで保留にされる。例えば、ユーザがフルスクリーンアプリケーションを実行中の場合、そのユーザは連絡不可能とみなされる可能性がある。あるいは、そのユーザは「連絡可能」であるが、ユーザに適した形に通知を修正する必要があるような状態にあるとみなされる場合もある。例えば、ユーザが音楽を聴いているか会合中の

場合、ユーザは、通知をユーザの画面に配信するが、通知に伴うサウンドはより小さくするか、全く生じさせないように指示している場合がある。

【 0 0 2 9 】

上記のように、ユーザコンテキストが、通知がユーザの画面に表示されるかどうかを部分的に決定する。通知が表示される場合は、ユーザコンテキスト中のある段階(gradient)に基づいて表示されることができる。すなわち、ドロ―される通知の形態の侵入度には、指定することが可能な様々なレベルがある。例えば、通常のお知らせは、いつでもクライアント領域中に表示され、一時的にウィンドウを覆うことができる。ユーザがわずかに制限的なコンテキストにある場合、通知は自由に表示してよいが、別のウィンドウの上にドロ―することはできないなどより侵入性の低い形でのみ表示することができる。別の例として、ユーザが最大化されたアプリケーションを実行中の一実施形態におけるデフォルトの設定は、例えば、アプリケーションが最大化されていることはコンテキストがわずかに制約されていることを意味し、ユーザがそのアプリケーションにクライアント領域全域を使用したいことを明確に表明しているという設定にすることができる。この設定では、通知のドロ―はなお許可されることができるが、サイドバー中のみに表示されるように行われることができる。すなわち、このタイプの通知のドロ―形態の侵入性の低下は、通知の影響を小さくし、全体として認知的な負荷を小さくする。

10

【 0 0 3 0 】

図 2 は、通知を処理するルーチン 2 0 0 を説明する流れ図である。ブロック 2 0 2 で、オペレーティングシステムまたは任意のプログラムが、通知のアプリケーションプログラムインターフェース (A P I) を呼び出す。ブロック 2 0 4 で、オペレーティングシステムおよび任意のプログラムによって設定され、さらにユーザによって承認または変更されたユーザコンテキストと、ユーザによって設定されたユーザ規則について、通知の要素が評価される。ブロック 2 0 6 で、ユーザコンテキストおよびユーザ規則に応じて通知が配信、延期、拒否、転送、またはその他の形で処理される。

20

【 0 0 3 1 】

ユーザコンテキストおよびユーザ規則について下記でより詳細に説明する。一実施形態では、ユーザコンテキストは、真または偽のどちらかである条件と、その条件が真である時の通知の対処法を決定する命令とからなる。一般に、ユーザコンテキストの条件とは、通信を配信するのにユーザを何らかの形で連絡不可能にするとシステムが見なす状態、または通知が配信される方式を、その通知をスタートしたプログラムによって送信された方式とは異なるものにする状態と考えることができる。すなわち、一実施形態では、ユーザコンテキストは「条件 X が真である時には、これが着信する通知に行う操作である」という文と考えることができる。一例は「音楽プレーヤが音楽を再生している時は、着信する通知はサウンドなしで画面に表示する」である。別の例は「フルスクリーンモードで実行されているアプリケーションがある時は、着信する通知は後まで延期する」である。

30

【 0 0 3 2 】

そのようなユーザコンテキストについて、一実施形態では、ユーザは着信する通知に対処する特別な規則を定義することもでき、したがって、ユーザコンテキストの命令に特別な例外を提供することができる。一例として、ユーザ規則は、「私が「 J o h n D o e 」から新しい電子メールを受信し、そのテキスト中に「緊急」とあり、「高優先度」とマークされている場合は、他のユーザコンテキストに関係なくその電子メールを配信する」と指示することができる。すなわち、この例では、このユーザ規則は、それ以外の場合は着信電子メールについての通知をその時に配信することは不適切であると指示するユーザコンテキストに例外を提供する。ユーザ規則がそれとの関係で評価される通知の要素に関して、その要素は、テキスト、サウンド、グラフィックなどの事物、および優先度、通知の送信者 (電子メールやインスタントメッセージなどのチャネルの場合) 、通知の有効期限などの他のプロパティ、および何らかのコード要素を含むことができ、ユーザは通知と対話し、任意のコードを起動することができる (例えば、通知中のボタンやテキストをクリックすることにより、新しいプログラムを起動させる、あるいは現在実行中のプログラ

40

50

ムで動作（電子メールの削除など）を行わせることができる）。

【0033】

図3は、オペレーティングシステムまたは任意のプログラムがユーザコンテキストを宣言するルーチン220を説明する流れ図である。ブロック222で、オペレーティングシステムまたはプログラムが、デフォルトのコンテキストと、ユーザのビジー状態に対するその影響を宣言する。すなわち、プログラムは、通知に対するその影響（例えば、画面へのドローが適切であるかどうか、およびサウンドが適切であるか否か、あるいはどれほどの相対的な音量でサウンドを再生すべきかなど）を含めたユーザコンテキストをシステムに登録し、提供する。一例として、音楽プレーヤプログラムは、「音楽プレーヤがユーザのために音楽を再生している時は、着信する通知は画面に表示するが、サウンドは再生しない」と指示するデフォルトコンテキストを宣言することができる。別の例として、オペレーティングシステムは、「フルスクリーンモードで実行中のアプリケーションがある時は、着信する通知は後の時間まで延期する」と指示するコンテキストを宣言することができる。

10

【0034】

再び図3に戻る。ブロック224において、オペレーティングシステムまたはプログラムが宣言したコンテキストを真または偽として設定する。例えば、「音楽プレーヤが音楽を再生している時は、着信した通知は画面に表示するが、サウンドは再生しない」というコンテキストを宣言する音楽プレーヤの場合、その音楽プレーヤプログラムは、宣言したそのコンテキストが現在真であるか偽であるかも設定する。すなわち、音楽プレーヤプログラムは、音楽プレーヤが現在音楽を再生中であるということが真であるか偽であるかを指示する。下記でより詳細に説明するように、一実施形態では、コンテキストが真か偽かの決定は、通知APIが呼び出される時に評価されても、ユーザ規則および例外が再度評価される時に評価されてもよい。追加的な機能として、システムは、呼び出しプログラムのプロセスハンドルを追跡することもでき、プロセスが最初にコンテキスト値をそのデフォルトの「偽」の値に再設定せずに終了した場合、システムは、最初のプロセスがもう存在しないことを検出すると直ちにそのコンテキスト値を再設定する（一実施形態では、プロセスハンドルの状態は、プロセスが終了した時にそれを伝えるように設定され、その状態の変化は、プロセスハンドルを監視するシステムによって認識(pick up)される）。これにより、プロセスが予期せずに終了した場合、あるいはプロセスがコンテキストの再設定を忘れた場合に、それ以降の通知の配信が必要以上に影響を受けないことが保証される。例えば、上述の例で音楽プレーヤプログラムが終了され、プロセスがもはや存在しない場合は、コンテキストが自動的に偽に再設定されることが可能である。別の例として、あるプログラムが最初にユーザがビジーであると宣言したが、その後そのプログラムがクラッシュしてプロセスがもはや存在しなくなった場合、通知が受信されない状態にユーザを置いたままにするのではなく、コンテキストが自動的に偽に設定されることができる。いずれにせよ、一実施形態では、コンテキストがアクティブに設定されているか、または関数として評価されているか否かに関係なく、コンテキストは一般には真または偽のいずれかに解決することができる。

20

30

【0035】

図3のブロック226において、システムに記憶されたユーザコンテキストにコンテキスト情報が追加される。このプロセスは、コンテキストを宣言する追加的なプログラムによって繰り返される。また、上述のように、すでに宣言されたコンテキストが真であるか偽であるかの状態は、ユーザが様々なプログラムを起動および終了し、様々なタスクをスタートするのに従って時間の経過と共に変化する。

40

【0036】

上述のように、一実施形態では、コンテキストの登録は宣言的なプロセスである。下記でより詳細に説明するように、ユーザコンテキストシステムでは、ユーザコンテキストに登録することによりコンテキストのリストをユーザに提示することができ、ユーザはあるコンテキストを容認しないことを選択するか、ユーザがコンテキストパラメータに同意し

50

ない場合はコンテキストの意味を変更することができる。上述のように、一実施形態では、コンテキストは、真または偽である条件と、その条件が真である時の通知の処理法の命令とからなることができる。これに関して、ユーザコンテキストは、人間が読むことのできる文字列（エンドユーザにその内容を知らせる）、一意の識別子（グローバル一意識別子（GUID：globally unique identifier））などの特定のプログラミング要素を含むことができ、プログラムは、そのコンテキストがいつ真であるか否かと、画面への通知の表示に関してそのコンテキストが何を意味するかを示す文を含むことができる命令（侵入度のレベル、サウンド、および音量を含むことができる）とをオペレーティングシステムに知らせることができる。下記でより詳細に説明するように、コンテキストは動的であってもよい。

10

【0037】

図4は、通知APIが呼び出される時にコンテキストが真であるか偽であるかを評価するルーチン230を説明する流れ図である。判定ブロック232で、通知APIが呼び出される時にユーザコンテキストを評価すべきかどうかの判定が行われる。ユーザコンテキストを評価する場合は、ルーチンはブロック234に進む。通知APIが呼び出される時にユーザコンテキストを評価しない場合、ルーチンは終了する。ブロック234で、ユーザコンテキストが真であるか偽であるかが評価される。

【0038】

図3および図4に示し、また先に述べたように、コンテキストは前もって設定されても、または関連する時に評価される関数であってもよい。一例として、プログラムは、ユーザが音楽を聴いていることをアクティブに認識することができる。別の例として、通知が評価される時にプログラムは自身のコールバックを登録している可能性があり、通知が評価される時にそのコンテキストが真であるかどうかシステムからプログラムに問い合わせられる。この2番目のプロセスが特に重要である事例の一例は、ユーザコンテキストがユーザ規則と組み合わせられて動的なコンテキストを形成する場合である。（ユーザ規則については下記でより詳細に説明する。）ユーザ規則と組み合わせられたユーザコンテキストの具体的な一例は、ユーザが「現在自分が会っている人々は、自分がビジーである状態に関係なく常に私に通知を送信することができる」と指示する規則を設定している場合である。この場合、「ユーザが会合中の時」というユーザコンテキストをさらにユーザがその会合で一緒である者の点から評価しなければならない。この例では、会合を扱うプログラムはこれを動的なコンテキストとして登録することができ、通知が評価される時にはその通知を送信した者がそのコンテキストに照らして評価される（会合の出席者は時間の経過と共に変わる可能性があるので、このコンテキストは動的にする以外に事前に真または偽と宣言することはできない）。すなわち、この特定の例では、他の人々のコンテキストに依存するユーザ規則に照らしてユーザのコンテキストを評価することが必要になる。

20

30

【0039】

図5は、ユーザがコンテキストを調整し、新しい規則を作成するためのルーチン240を説明する流れ図である。ブロック242において、ユーザがコンテキストを調整したいか否かが判定される。ユーザがコンテキストの調整を希望しない場合、ルーチンは判定ブロック246に進む。このブロックについては下記でより詳細に説明する。ユーザがコンテキストの調整を希望する場合、ルーチンはブロック244に進み、ユーザがコンテキストの変更を行う。

40

【0040】

一実施形態では、提供されたコンテキストは、ユーザがそのコンテキストをオフにすることができる（例えばユーザがプログラムによるそのコンテキストの査定に同意しない場合）か、または通知の配信に対する影響の点からコンテキストを変更できる形でユーザに公開されることができる。より具体的な例として、ユーザコンテキストは、「フルスクリーンモードで実行中のアプリケーションがある時」、「自分が音楽またはビデオを再生している時」、「会合マネージャが自分が会合中であると示す時」、あるいは「不在時アシスタントがオンになっている時」などの事柄を含むことができる。こうした事柄それぞれに

50

ついて、与えられた条件が真である時には、着信する通知は選択された手順に従うという命令を指定する選択をユーザが行えるようにすることができる。この命令は、通知を画面にドロウすることの可否またはその方式、および通知に伴うサウンドまたは音量といった事柄を指定することができる。音量について、ユーザは、所与の条件下で希望する音量のパーセンテージを指定することができる。通知を画面にドロウするオプションについては、通知を一切ドロウしない、あるいは指定された外部ディスプレイのみに通知をドロウする、あるいは現在の画面に通知をドロウするなどの選択肢がユーザに提供されることができる。通知のドロウについては様々なレベルの侵入性を指定することができる。例えば、ユーザが最大化されたアプリケーションを実行中で、コンテキストがわずかに制約されている場合、侵入度の設定は、通知のドロウは可能であるがサイドバー中のみに表示するという設定にすることができる。

10

【 0 0 4 1 】

図5のブロック246において、ユーザが新しいユーザ規則を作成することを希望するか否かが判定される。ユーザが新しいユーザ規則の作成を希望しない場合、ルーチンは判定ブロック250に進み、このブロックについては下記でより詳細に説明する。ユーザが新しいユーザ規則の作成を希望する場合、ルーチンはブロック248に進み、新しい規則が作成される。一般に、ユーザ規則は、指定された要素を含む通知をどのように扱うべきかを指示する。例えば、規則は、指定された者からの通知は常に直ちに配信すると指示することができる。その規則は、その通知が指定された者からであれば、通知を開始したプログラムに関係なくすべての通知に適用されることができる。より具体的な例として、他のユーザ規則は、「ワシントン州ブレマートンについてのMSNオートの交通警報」や「John Doeからの重要な電子メール」などの事柄を対象とすることができる。John Doeからの重要な電子メールについてのユーザ規則の一例として、その規則は、John Doeから到着する電子メールで、テキスト中に「緊急」とあり、「高優先度」とマークされた電子メールは指定された処理条件に従うように指示することができる。その処理条件は、通知を直ちに配信し、通知を受信したことを相手に知らせることをユーザに要求することを指定することができる。一般に、通知を受信したことを相手に知らせることをユーザに要求することは、ユーザが明確にその通知を消すまで通知が画面に表示され続けるという点で、通知の侵入度の形態にわずかな増大があることを意味する。一実施形態では、ユーザによる受信通知の要求は、ユーザ規則を介してのみ設定することができる。別の例として、規則は、通知のために指定された音量で再生されるカスタムサウンドを指定することもでき、特別な通知が到着したというアラートをユーザに提供することができる。ユーザのコンテキストによって決定されるユーザの「通常」および「ビジー」の状態中の通知の対処法については、様々な設定が選択されることもできる。この対処命令は、「John Doeからの通知はページャに配信する」など通知の転送オプションのような事柄も含むことができる。一実施形態では、コンテキストが評価される時には、最も制限的な現在真のコンテキストが適用されるコンテキストとなる。ユーザ規則が評価されるということは、特定の通知がユーザの作成した規則と合致したことを意味し、その場合は通知と合致したユーザ規則のうち最も侵入度の高い設定が適用される。すなわち、ユーザ規則で、ユーザは重要とする何らかの事柄を指定しており、この手順はユーザの嗜好に従うものとする。2つの規則に矛盾がある場合は、最も侵入度の高い規則が適用される。

20

30

40

【 0 0 4 2 】

一実施形態では、ユーザ規則は、特定の通知サービスからの通知の配信を制御することも対象とすることができる。例えば、通知サービスに従って通知を提供するオペレーティングシステムは、通知の配信方式を変更する方法をユーザに提供することができる。例えば、指定された通知サービスが「シアトルの交通警報」を提供することができ、ユーザは、そのような通知が受信された時にはシステムが「通知を表示し、サウンドを再生する」ように配信を編集することができる。

【 0 0 4 3 】

50

図5の判定ブロック250において、ユーザが既存のユーザ規則の調整を希望するか否かが判定される。ユーザが規則の調整を希望しない場合、ルーチンは終了する。ユーザがユーザ規則の調整を希望する場合、ルーチンはブロック252に進み、ユーザが規則の変更を行う。

【0044】

先に図3～図5に関して述べたように、ユーザコンテキストおよびユーザ規則は、オペレーティングシステム、プログラム、およびユーザによって設定される。システムは、ユーザの嗜好に従って通知の配信を適切に仲介し、順番（直列化：serialize）にする。ユーザコンテキストおよびユーザ規則はユーザに公開されることができ、ユーザは各種のコンテキストおよび規則を変更または調整するか、または新しい規則を作成することができる。これにより、通知の対処方式についての嗜好を管理する集中化された方法がユーザに提供される。これにより、ユーザは、ユーザへの通知の送信を求める可能性があるコンピューティングシステム中の多数の競合する要素を効果的に管理できることは理解されよう。

10

【0045】

図6は、ユーザコンテキストおよびユーザ規則に従って通知を処理するルーチン300を説明する流れ図である。ブロック302において、オペレーティングシステムまたは任意のプログラムが通知APIを呼び出す。判定ブロック304において、その通知を通知の履歴に記録すべきか否かの判定が行われる。通知を記録すべき場合は、ルーチンはブロック306に進み、通知が履歴に記録される。通知を記録すべきでない場合は、ルーチンは判定ブロック310に進む。

20

【0046】

判定ブロック310において、通知がいずれかのユーザ規則に合致するか否かが判定される。通知が合致する規則がある場合、ルーチンはブロック312に進み、（通知の内容とユーザコンテキストに基づいて）そのユーザ規則に従い、ルーチンは図7に続く点Aに進む。判定ブロック310で通知がどのユーザ規則にも合致しない場合、ルーチンは判定ブロック320に進む。

【0047】

一実施形態では、ユーザ規則は常に現在のユーザコンテキストよりも優先される。上述のように、ユーザ規則は、任意の通知の要素に基づくことができる。例えば、通知を開始した者の評価に基づく規則は、その規則が基づく者からの通知であればどのプログラムが通知を開始したのかに関係なくすべての通知に適用されることができる（例えば「John Doe」は常に私と連絡をとることができるなど）。また、通知は、本来はその通知をドロー(draw)させないコンテキスト中であっても画面にドローすることができる（例えば、ユーザコンテキストが、一般にはユーザは会合中には通知を受け取らないことを指示していても、「私と一緒に会合に出ている人は常に私に通知を送信できる」など）。

30

【0048】

図6の判定ブロック320において、現時点で(at the present time)通知をドローできるか否かが（ユーザコンテキストのみに基づいて）判定される。現時点で通知をドローできる場合は、ルーチンはブロック322に進み、通知がドローされ、適切なサウンドと音量が提供される。現時点では通知をドローすることが適切でない場合、ルーチンは判定ブロック330に進む。

40

【0049】

判定ブロック330において、通知の有効期限が切れているか否かが判定される。通知の有効期限が切れている場合、ルーチンはブロック332に進み、通知が破棄される。通知の有効期限が切れていない場合、ルーチンはブロック334に進み、通知が延期され、ルーチンは図7に続く点Bに進む。

【0050】

図7は、指定されたユーザ規則に従って通知を処理するルーチン350を説明する流れ図である。このルーチンは、上記のように図6の点Aから続いている。図7に示すように

50

、判定ブロック352で、通知を転送すべきか否かが判定される。通知を転送すべきでない場合、ルーチンは判定ブロック362に進み、このブロックについては下記でより詳細に説明する。通知を転送すべき場合ルーチンはブロック354に進み、指定に従って通知が転送される。通知が転送される時は、その通知が、その通知を別のシステムに渡すことを必要とするユーザの規則と合致する要素を含んでいることを意味する。これは、例えばユーザがビジューである場合に行われる可能性があり、あるいは、ユーザが連絡不可能であるか否かに関係なくユーザの規則で指定される基準に合致するすべての通知で行われる可能性がある。一例として、その中に「緊急」という単語が含まれる通知は常にユーザのページに転送されるのに対し、その他の通知はユーザの規則とコンテキストの組合せに基づいてのみ転送されるようにすることができる。

10

【0051】

転送命令(routing instruction)のいくつかの例には、「その通知をある電子メールアドレスに転送する」、「その通知を別のPCに転送する」、「その通知をページに転送する」、「その通知を携帯電話に転送する」、あるいは「その通知を電子メールサーバに転送する」などがある。下記でより詳細に説明するように、通知が転送される場合、その通知は配信され、画面にドローされることもできる。また、通知が転送されるデバイスはこれと同じコンテキストシステムを実装することができ、そのデバイスにはユーザのコンテキストについての追加的または異なる知識があってよく、そのデバイスのコンテキストシステムは、通知に対して異なる動作を行うことを選択することができる。

【0052】

20

図7の判定ブロック362において、通知を拒否するか否かが判定される。通知を拒否しない場合、ルーチンは判定ブロック366に進み、このブロックについては下記でより詳細に説明する。通知を拒否する場合、ルーチンはブロック364に進み、通知が破棄され、その通知はユーザによって見られない。すなわち、拒否される通知は、ドローする、または音を立てることが許されない。通知の拒否は、例えば、特定の通知を拒否すると指示するユーザ規則に基づいて行われる場合があり、または上記で図6のブロック332を参照して述べたように通知の有効期限が切れている場合に行われる可能性がある。

【0053】

図7の判定ブロック366において、通知を延期すべきか否かが判定される。通知を延期すべきでない場合、ルーチンは判定ブロック370に進み、このブロックについては下記でより詳細に説明する。通知を延期すべき場合、ルーチンはブロック368に進み、ユーザコンテキストが変化するまで通知が保留にされ、ルーチンは図8に続く点Bに進む。一般に、通知を延期するということは、通知の配信はいずれは許可されるが、ユーザの現在のコンテキストまたは規則がその時に通知を配信するには適当でないといみなされるものであることを意味する。後に図8を参照してより詳細に述べるように、ユーザのコンテキストが変化するか、あるいはユーザの規則が後に配信が適切になると示すと、ユーザ規則およびユーザコンテキストの指示に従って通知がユーザの画面に配信され、ドロー(draw)および/またはそのサウンド(sound)を生成することが許可される。

30

【0054】

図7の判定ブロック370において、通知を配信すべきか否かが判定される。通知を配信すべきでない場合、ルーチンは終了する。通知を配信すべき場合、ルーチンはブロック372に進み、適切な侵入度のレベルに従って通知がドローされ、適切なサウンドと音量が提供される。すなわち、通知の配信は許可されるが、ユーザのコンテキストおよび規則に従って配信される(例えば通知をドローしてもよいが、サウンドなしでドローすることが要求されるなど)。

40

【0055】

図8は、通知の配信を延期するルーチン380を説明する流れ図である。上記のように、このルーチンは図6または図7の点Bから続いている。図8に示すように、ブロック382で通知が保留にされる。ブロック384において、システムが、真または偽であるとして宣言されたコンテキストへの変更があるか否か、または通知の配信が現在適切である

50

と示すユーザ規則があるか否かを調べる。判定ブロック 386 において、ユーザコンテキストが変化しているか否か、またはユーザ規則が通知を配信するのが現在適切であると示すか否かが判定される。ユーザコンテキストが変化しておらず、他に指示をするユーザ規則がない場合、ルーチンはブロック 382 に戻り、通知が引き続き保留にされる。ユーザコンテキストが変化しているか、通知の配信が現在適切である可能性があるというユーザ規則が示す場合、ルーチンは図 6 に続く点 C に進む。図 6 の点 C は判定ブロック 304 に戻り、ここから通知を評価するプロセスが再びスタートする。

【0056】

図 9 は、各種の制約に応じて通知のドロウを決定するルーチン 400 を説明する流れ図である。このルーチンは、図 6 のブロック 322 または図 7 のブロック 372 など通知処理の一部として実施してよいことを理解されよう。一般に、通知がシステムに入ると、その時点で真であるすべてのコンテキストについて評価が行われ、その通知についての最も制限的な設定がその時点のユーザの状態に応じて適用される。図 9 に示すように、判定ブロック 402 において、通知を一切ドロウすべきでないか否かが判定される。通知を一切ドロウしない場合、ルーチンはブロック 404 に進み、どのディスプレイにもドロウされないように通知が設定される。通知をドロウすべき場合、ルーチンは判定ブロック 406 に進む。

【0057】

判定ブロック 406 において、通知を外部でのみドロウすべきか否かが判定される。通知を外部でのみドロウすべき場合ルーチンはブロック 408 に進み、外部のハードウェアディスプレイのみに通知がドロウされる。通知を外部のハードウェアディスプレイにドロウすべきでない場合、ルーチンは判定ブロック 410 に進む。

【0058】

判定ブロック 410 において、通知を現在のディスプレイでドロウすべきか否かが判定される。通知を現在のディスプレイでドロウすべき場合、ルーチンはブロック 412 に進み、該当する侵入度レベルに応じて現在のディスプレイに通知がドロウされる。通知を現在のディスプレイにドロウすべきでない場合、ルーチンは終了する。

【0059】

図 10 は、各種の制約に従って通知のサウンドが再生される音量を決定するルーチン 420 を説明する流れ図である。図 9 との関連で上述したように、このルーチンは、図 6 のブロック 322 または図 7 のブロック 372 など通知の処理の一部として実施してよいことは理解されよう。通知がシステムに入ると、その時点で真であるすべてのコンテキストの評価が行われ、その時点のユーザの状態に応じて最も制限的な設定が通知に適用される。図 10 に示されるように、判定ブロック 422 において、通知を消音にすべきか否かが判定される。通知を消音にすべき場合、ルーチンはブロック 424 に進み、その通知には音量が提供されない。通知を消音にすべきでない場合、ルーチンは判定ブロック 426 に進む。

【0060】

判定ブロック 426 において、最大のボリュームより小さいいくらかのパーセンテージの音量で通知を提供すべきか否かが判定される。いくらかのパーセンテージの音量を提供すべき場合、ルーチンはブロック 428 に進み、指定されたパーセンテージの音量で通知が再生される。指定されたパーセンテージの音量を提供すべきでない場合、ルーチンは判定ブロック 430 に進む。

【0061】

判定ブロック 430 において、その通知に最大の音量を提供すべきか否かが判定される。最大の音量を提供すべき場合、ルーチンはブロック 432 に進み、最大の音量レベルで通知が再生される。最大の音量を提供すべきでない場合、ルーチンは終了する。一実施形態では、通知に異なる音量レベルを提供するのに加えて、ユーザのコンテキストおよび規則に従って通知に異なるサウンドを選択することも可能である。

【0062】

図１～図１０に関して上記で説明したユーザコンテキストシステムは、システムが各自の画面へのレンダリングを適切に仲介し、順番（直列化）にすることで通知が互いと衝突しなくなるような形で各種ソースからの通知の配信を制御することは理解されよう。また、ユーザコンテキストシステムによって処理される通知は、ユーザがより通知を受け付けられる状況にある時に配信されるのでより価値が高くなると考えることができ、また、共通の規則を使用することによりユーザが不要な通知を解消するのを助ける。

【００６３】

図１１～図１４は、テスト通知の評価(evaluation of test notification)を対象とする。下記でより詳細に説明するように、本発明の一態様によれば、テスト通知は、現在のユーザコンテキストの状態についての情報を得るために任意のプログラムによって利用されることができる。この態様の１つの利点は、そのプログラムがシステム中にすでに構築されているこのサービスを使用することを意図するか否か、あるいはそのプログラムが、通知の表示方式とその配信方式の独自の解釈を展開(roll)することによりこのサービスを拡張するか否かに関係なく、どのプログラムもユーザコンテキスト情報を得られることである。すなわち、このシステムによって提供されるレンダリングで制限されるように設計されていないより高度な通知を備える将来のプログラムも、テスト通知を利用してユーザの現在のコンテキストについての情報を得ることができる。そのようなより高度な通知は、通知方式のリッチさが成長と変化を続けるのに伴って、そして通知のための新たなユーザインターフェースが発展し続けるのに伴って出現することが考えられる。

【００６４】

一例として、将来のユーザインターフェースは、ユーザが「ビジー」でない時にのみドロールされるリッチなフルスクリーン動画を提供する可能性がある。例えば、ＣＤ－ＲＯＭドライブにＣＤを入れると、ＣＤ－ＲＯＭが回転速度を上げる間に画面にＣＤの動画が提示される（技術的な制約のために、ＣＤがドライブ中にあることが分かっているにもかかわらずＣＤが最初に挿入されてからＣＤを読み出せるようになるまでいくらかの時間があり、この時間中に動画を使用してシステムがＣＤを認識しているが、単にまだ読み出しが可能でないことをユーザに知らせることができる）。本発明のテスト通知を使用することにより、動画プログラムは、ユーザの現在のコンテキストについて知ることができ、ユーザがその時点で通知を受け取れない状況である場合は画面への表示を行わないことを選択することができる。

【００６５】

別の例として、将来のインスタントメッセージングプログラムは、現在の通知エンジンでは行うことができない新規の通知のユーザインターフェースを開発する可能性がある。そのような新しいユーザインターフェースの開発は、競争の激しい現在の市場で求められる柔軟性の一環として必要とされる。本発明によれば、テスト通知は、ユーザの現在のコンテキストに応じてそのようなインスタントメッセージングプログラムが備えるより高度な通知を表示すべきか／表示すべきでないかを判定するために、将来のインスタントメッセージングプログラムによって利用され続けることができる。

【００６６】

本発明の別の態様によれば、テスト通知は、不要な通知の生成を防止するために利用されることができる。この態様は、システムに通知を送信しようとする任意のプログラムに適用されることができる。すなわち、プログラムがよりリッチなユーザコンテキストのビューを得られるようにすることにより、不要な通知がプログラムによって生成されないようにすることができ、したがってユーザが通知を受け取れる状態になるまでその種の通知の生成を時間的に先んじて停止させることができる。次の例は、本発明のこの態様を例証する助けとなる。

【００６７】

一例として、インスタントメッセージングプログラムは連絡先のリストを提供することができる。テスト通知は、連絡先単位でコンテキストシステムを利用する(tap into)ことができる（例えば「トムが今インスタントメッセージを送信した場合、それは表示される

か？」と「クリスが今インスタントメッセージを送信した場合、それは表示されるか？」等)。その情報に基づいて、インスタントメッセージングプログラムは、明確なビジーな状態または暇な状態を個々の連絡先に同報通信し始めることができる。この技術を使用して、単に受信された後に抑制するのではなく、不必要な通知の生成を時間的に先んじてやめさせることができる。

【 0 0 6 8 】

別の例として、ユーザがビジーである場合、メールプログラムはテスト通知を使用して、送信者に自動化された返信を提供することができる(「直属の部下」や「部長」などユーザが提供した規則に基づくすべての送信者に)。この自動化された返信は、「今は忙しいが機会がきたら返事をする」ことを伝えることができる。一般に、このシステムの通信は概して、ユーザのコンテキストを任意のプログラムに公開することによって向上させることができる。

10

【 0 0 6 9 】

上述のように、本発明によれば、アプリケーションは、テスト通知を構築し、実際の通知が現時点で画面にドロウされるか否かを明確に受け取ることができる。先に述べたように、これにより、通知のための新しいユーザインターフェースが開発された後でも、プログラムがこのユーザコンテキストシステムを引き続き使用することが可能になる。また、他のプログラムのためにこうした新しいよりリッチなシナリオを可能にすることにより、ユーザの振る舞いおよび嗜好についての情報へのアクセスをより多く得られる点で、このシステムを利用するすべてのプログラムはよりリッチであり、よりインテリジェントであると考えることができる。

20

【 0 0 7 0 】

図 1 1 は、本発明によりテスト通知を処理する一般的なルーチン 5 0 0 を説明する流れ図である。このルーチンは、実際の通知を処理する図 2 のルーチンと同様である。図 1 1 のブロック 5 0 2 において、通知テストの A P I が呼び出される。ブロック 5 0 4 において、オペレーティングシステムおよび任意のプログラムによって設定され、さらにユーザによって承認または変更されたユーザコンテキストと、ユーザによって設定されたユーザ規則とについて、テスト通知の要素が評価される。ブロック 5 0 6 において、テスト通知の評価に従ってテスト通知の対処法に関する指示が提供される。この指示(indication)は、次いで、呼び出しアプリケーション(calling application)に戻される。

30

【 0 0 7 1 】

一実施形態では、オペレーティングシステムまたは任意のプログラムが現在ユーザがどれほどビジーであるか理解する必要があると判断した時に通知テスト A P I が呼び出される。これが行われる時の一例は、画面に通知をドロウするかしないかの判定ポイントがある場合である。別の例は、このデータを使用して、プログラムがユーザのために行いたい動作を通知するものである。

【 0 0 7 2 】

通知テストの A P I が呼び出されると、呼び出しプログラムは、実際の通知を送信するためにユーザコンテキストシステムの通知法を使用した場合に送信する通知に近い通知を構築し、最適なテスト方法を使用する(それにより結果が返され、またその特定の通知が画面に表示されないことも保証される)。このプロセスの一例は、インスタントメッセージングプログラムが、現在のユーザのコンテキストに基づいて該当する暇な状態またはビジーの状態を各連絡先に同報通信することを求める場合である。インスタントメッセージングプログラムは、各連絡先についてのテスト通知を作成し、戻り値に基づいて異なる暇な状態またはビジーの状態を連絡先単位で同報通信する。別の例は、ユーザのコンテキストに基づく動画を表示しようとするプログラムである(例えば上述の C D - R O M 動画の例)。動画を表示したいコードは、通知を構築し(この場合、通知はその通知をドロウするか否かについてのテストなので、内容は単に画像または動画のシーケンスを備えた単純な通知である)、そしてこのテスト方法を使用し、返された結果は今動画を再生すべきか否かについての指針として使用されることができる。一実施形態では、一般には、呼び出

40

50

し元のコードは少なくとも、テスト通知として可能な最も一般的な通知を生成する。ユーザは戻される結果に影響する可能性がある個人単位のカスタムユーザ規則を有する可能性がある。よりリッチなデータ（連絡先リストの連絡先など）を利用できる場合は、その情報を含めることによりテスト通知がより正確になる。

【0073】

通知テストAPIに利用することが可能な実施の1つは、ポーリングの実施である。上述のインスタントメッセージングプログラムの例では、ポーリングの実施のために、インスタントメッセージングプログラムは周期的に通知テストAPIを再ポーリングして、同報通信データをどのように変更するかを決定する。通知テストAPIに利用することができる別の実施は、サブスクリプションコールバックの実施である。この実施では、インスタントメッセージングプログラムはコンテキストの変化を「サブスクライブ」する。そして、ユーザのコンテキストはインスタントメッセージングプログラムが同報通信できる内容を変える形で変化するので、コンテキストエンジンは、周期的に再ポーリングするのではなく、適切な更新と共にインスタントメッセージングプログラムにコールバックすることができる。シナリオによっては、コールバックは、コンテキストの変化と同報通信される内容の間に時間差がない点で有利である（対してポーリングの実施では同報通信の状態が現在のユーザコンテキストと合致しない時がある傾向がある）。他のシナリオでは、ポーリングの実施がより適切な場合がある（それらは、CDがCD-ROMに挿入されるなどの1回限りの事象に対する応答であるため）。

【0074】

図12は、テスト通知を処理し、そのテスト通知を現時点ではドロースるかしないかについて真または偽の指示を戻すルーチン520を説明する流れ図である。ブロック522において、通知テストAPIが呼び出される。判定ブロック530において、そのテスト通知と合致するユーザ規則があるか否かが判定される。テスト通知がどのユーザ規則とも合致しない場合、ルーチンは判定ブロック550に進み、このブロックについては下記でより詳細に説明する。テスト通知が合致するユーザ規則がある場合は、ルーチンは判定ブロック540に進む。

【0075】

判定ブロック540において、ユーザ規則がそのテスト通知が現時点ではドロースることを指示するか否かが判定される。テスト通知が現時点ではドロースる場合、ルーチンはブロック542に進み、真の指示が提供される。テスト通知が現時点ではドロースられない場合、ルーチンはブロック544に進み、偽の指示が提供される。

【0076】

判定ブロック550において、テスト通知を現時点でドロースることが可能であるか否かが判定される（ユーザコンテキストとの関連でのみ）。現時点でテスト通知をドロースることが可能な場合、ルーチンはブロック552に進み、真の指示が提供される。現時点では通知をドロースすることが可能でない場合、ルーチンはブロック554に進み、偽の指示が戻される。ルーチンは、ブロック542、544、552、および554から、指定された指示と共に呼び出し元のアプリケーションに戻る。

【0077】

図13は、通知を処理し、詳細な指示を戻す(returning detailed indication)ルーチン600を説明する流れ図である。上述のように、図12のルーチン520は、通知を現時点ではドロースるか否かに関して真または偽の指示を与える戻り値を提供するだけである。下記でより詳細に説明するように、図13のルーチン600は、よりリッチな戻り値を戻す（例えば通知は今すぐにはドロースせず、ユーザのコンテキストが変化したら直ちにドロースする、あるいは別のデバイスに転送するなど）。これにより呼び出しコード(calling code)に、よりリッチなロジック(richer logic)が提供される。これにより、そのようなよりリッチな戻り値(richer return value)を利用することが可能なプログラムでは高度な機能が可能になる。

【0078】

また、関数呼び出し(function call)の一部として戻り値(return value)を説明しているが、別の実施形態ではこのデータはコールバックの一部として渡されてもよい。すなわち、呼び出し元のアプリケーションは、通知の「サブスクリプション(subscription)」をセットアップすることができ、ユーザのコンテキストが後に変化すると(呼び出しアプリケーション(calling application)からの通知の配信に影響する可能性がある)、呼び出しアプリケーションにその変化が通知される。これにはポーリングは必要とされず、したがって事例によってはシステムの堅牢性とパフォーマンスのためにより適当である。

【0079】

図13のブロック602において、通知テストAPIが呼び出されるか、またはサブスクリプションが登録される(それぞれ上述のポーリングの実施形態とサブスクリプションの実施形態との関連で)。判定ブロック610において、テスト通知が合致するユーザ規則があるか否かが判定される。テスト通知がどのユーザ規則とも合致しない場合、ルーチンは判定ブロック620に進み、このブロックについては下記でより詳細に説明する。通知が合致するユーザ規則がある場合、ルーチンはブロック612に進む。ブロック612において、ユーザ規則に従ってテスト通知が(テスト通知の内容とユーザコンテキストに基づいて)評価され、ルーチンは図14に続く点Dに進み、これについては下記でより詳細に説明する。

【0080】

判定ブロック620において、テスト通知を現時点でドロウすることが可能であるか否かが(ユーザのコンテキストのみに基づいて)判定される。現時点ではテスト通知をドロウすることが可能でない場合、ルーチンは判定ブロック630に進み、このブロックについては下記でより詳細に説明する。現時点でテスト通知をドロウすることが可能な場合、ルーチンはブロック622に進む。ブロック622において、ルーチンは、ユーザコンテキストに応じて適切と思われるサウンドレベルを決定する。ブロック624において、通知がドロウされる旨の指示が提供され、またこの指示はその通知に適切と思われるパーセンテージのサウンドレベルを含む。

【0081】

判定ブロック630において、テスト通知を後に配信するために保留にされるか否かが判定される(テスト通知の内容とユーザ規則に基づいて)。テスト通知を後まで保留する場合は、ルーチンはブロック632に進み、延期の指示が提供される。テスト通知を後に配信するために保留しない場合、ルーチンはブロック634に進み、拒否の指示が提供される。ブロック624、632、および634から、ルーチンは指定された指示と共に呼び出しアプリケーション(calling application)に戻る。

【0082】

図14は、ユーザ規則(user rule)に従ってテスト通知を評価するルーチン650を説明する流れ図である。このルーチンは図13の点Dから続いている。図14に示すように、判定ブロック652において、テスト通知を転送すべきか否かが判定される。テスト通知を転送すべき場合、ルーチンはブロック654に進み、転送の指示が提供され、ルーチンは判定ブロック662に進む。テスト通知を転送すべきでない場合もルーチンは判定ブロック662に進む。

【0083】

判定ブロック662において、テスト通知が拒否されるか否かが判定される。テスト通知が拒否される場合、ルーチンはブロック664に進み、拒否の指示が提供される。テスト通知を拒否しない場合、ルーチンは判定ブロック666に進む。

【0084】

判定ブロック666において、テスト通知が延期されるか否かが判定される。テスト通知が延期される場合、ルーチンはブロック668に進み、延期の指示が提供される。テスト通知が延期されない場合、ルーチンは判定ブロック670に進む。

【0085】

判定ブロック670において、テスト通知が配信されるか否かが判定される。テスト通

10

20

30

40

50

知が配信される場合、ルーチンはブロック 672 に進み、配信の指示が提供される。一実施形態では、配信の指示は、指定された侵入度の指示とサウンドおよび音量の指示も含むことができる。テスト通知を配信しない場合は、ルーチンは呼び出しアプリケーションに戻る。ブロック 664、668、672 から、ルーチンは、指定された指示と共に呼び出しアプリケーションに戻る。

【0086】

図 1 ~ 図 14 に関して上記で説明した本発明は、プログラムがユーザの連絡可能性についての指示を得ることができるテスト通知を利用するシステムおよび方法を提供することは理解されよう。プログラムがよりリッチなユーザコンテキストのビューを得られるようにすることにより、不要な通知の生成をソースで阻止することができ、したがってユーザが通知を受け取れる状態にある時にのみ通知を生成することができる。また、プログラムは、一般にはそのプログラムからの通知に異なるユーザインターフェースを利用しているが、このテスト通知を利用してユーザのコンテキストを判断することができる。これらの態様により、ユーザコンテキストシステムの可能な使用法により高い柔軟性が得られる。これらの態様は、他のプログラムのためのよりリッチな新しいシナリオも可能にし、システム全体がユーザの振る舞いと嗜好に基づいてよりリッチかつインテリジェントになることができる。

【0087】

本発明の好ましい実施形態を図示し、説明したが、本発明の精神および範囲から逸脱せずに各種の変更を行えることは理解されよう。

【図面の簡単な説明】

【0088】

【図 1】本発明を実施するのに適した汎用コンピュータシステムを示すブロック図である。

【図 2】本実施形態により通知を処理する一般的なルーチンを説明する流れ図である。

【図 3】オペレーティングシステムまたは任意のプログラムがユーザコンテキストを宣言するルーチンを説明する流れ図である。

【図 4】通知 API が呼び出される時にユーザコンテキストが真であるか偽であるかを評価するルーチンを説明する流れ図である。

【図 5】ユーザコンテキストを調整し、新しいユーザ規則を作成するルーチンを説明する流れ図である。

【図 6】ユーザコンテキストおよびユーザ規則に従って通知を処理するルーチンを説明する流れ図である。

【図 7】通知の内容およびユーザコンテキストに基づいてユーザ規則を実行するルーチンを説明する流れ図である。

【図 8】通知の配信を延期するルーチンを説明する流れ図である。

【図 9】各種の制約的な設定に従って通知のドロー方式を決定するルーチンを説明する流れ図である。

【図 10】各種の制約的な設定に従って通知の音量レベルを決定するルーチンを説明する流れ図である。

【図 11】本発明によりテスト通知を処理する一般的なルーチンを説明する流れ図である。

【図 12】テスト通知を処理し、真または偽の指示を戻すルーチンを説明する流れ図である。

【図 13】テスト通知を処理し、十分な詳細事項を含む指示を戻すルーチンを説明する流れ図である。

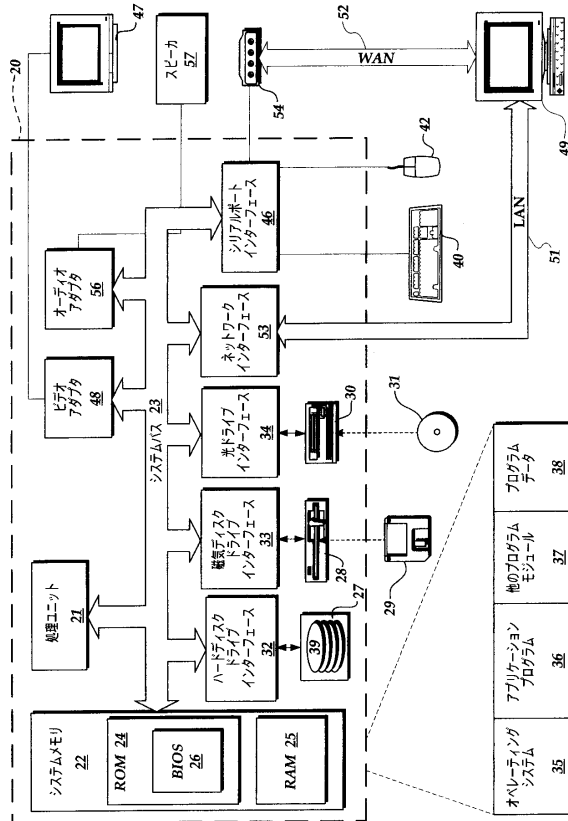
【図 14】ユーザ規則を使用して、テスト通知の内容と現在のユーザコンテキストに基づいてテスト通知を処理するルーチンを説明する流れ図である。

【符号の説明】

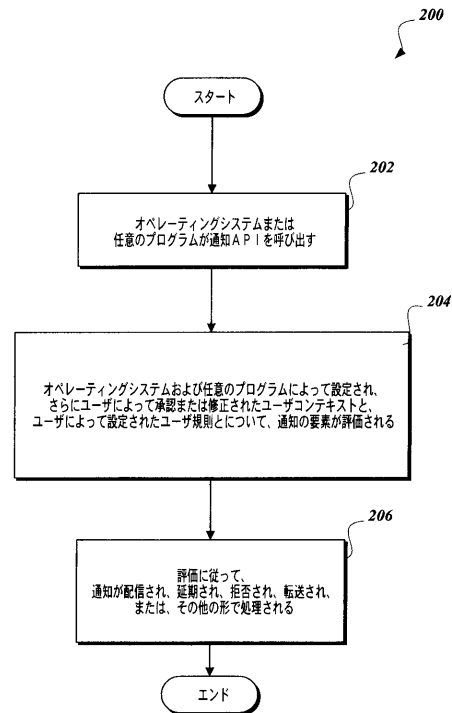
【0089】

2 0	パーソナルコンピュータ	
2 1	処理ユニット	
2 2	システムメモリ	
2 3	システムバス	
2 4	R O M	
2 5	R A M	
2 6	B I O S	
2 7	ハードディスクドライブ	
2 8	磁気ディスクドライブ	
2 9	磁気ディスク	10
3 0	光ディスクドライブ	
3 1	光ディスク	
3 2	ハードディスクドライブインターフェース	
3 3	磁気ディスクドライブインターフェース	
3 4	光ドライブインターフェース	
3 5	オペレーティングシステム	
3 6	アプリケーションプログラム	
3 7	プログラムモジュール	
3 8	プログラムデータ	
3 9	ハードディスク	20
4 0	キーボード	
4 2	ポインティングデバイス	
4 6	シリアルポートインターフェース	
4 7	モニタ	
4 8、5 3	アダプタ	
4 9	リモートコンピュータ	
5 1	L A N	
5 2	W A N	
5 4	モデム	
5 6	オーディオアダプタ	30
5 7	スピーカ	

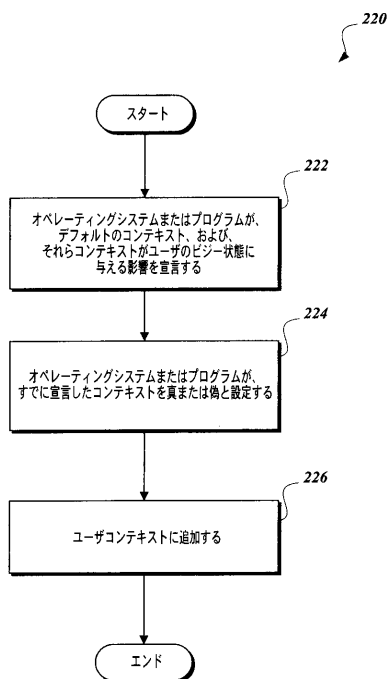
【 図 1 】



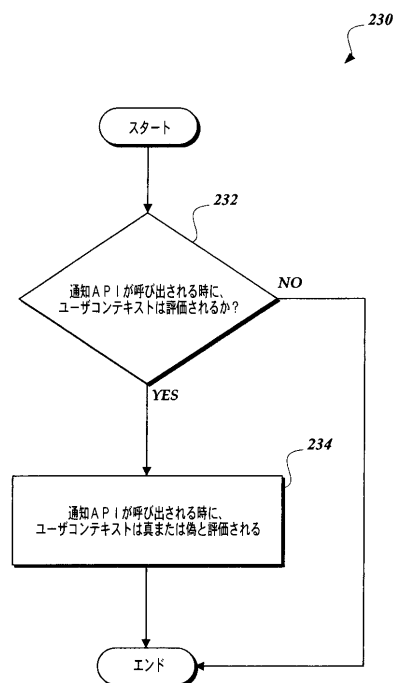
【 図 2 】



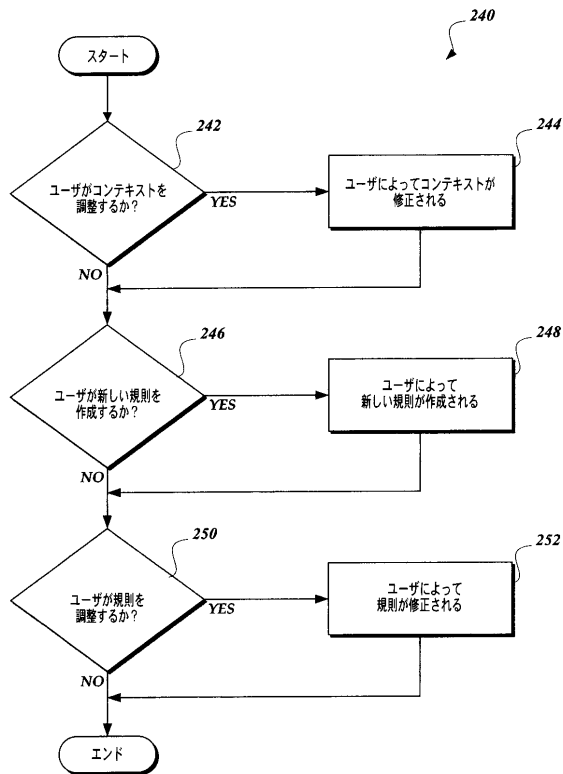
【圖 3】



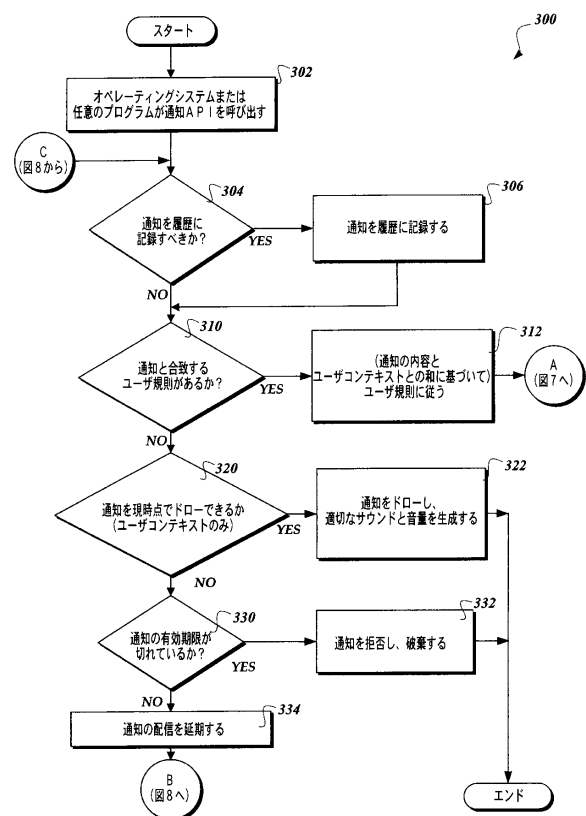
【 図 4 】



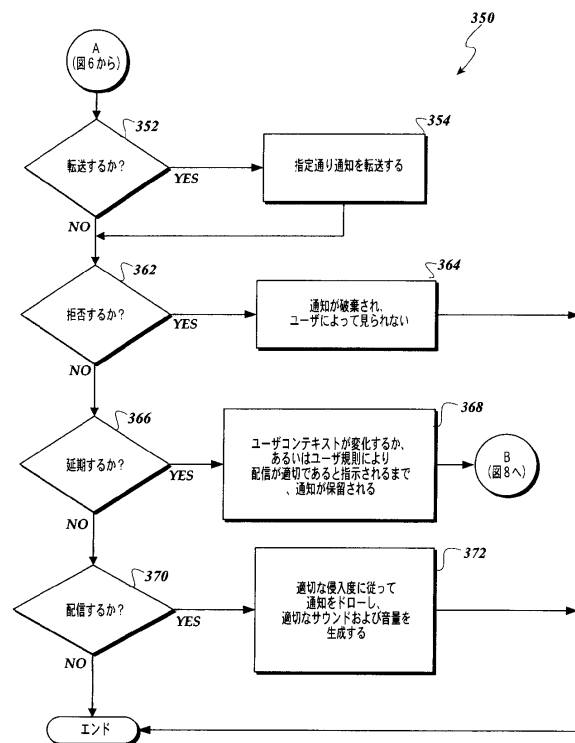
【図 5】



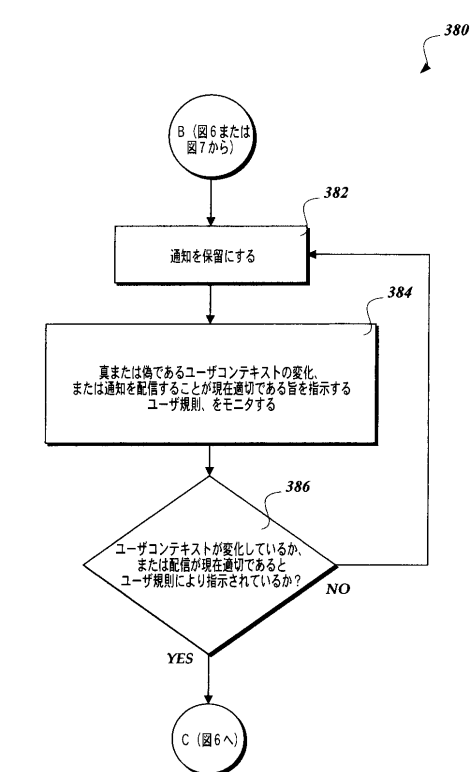
【図 6】



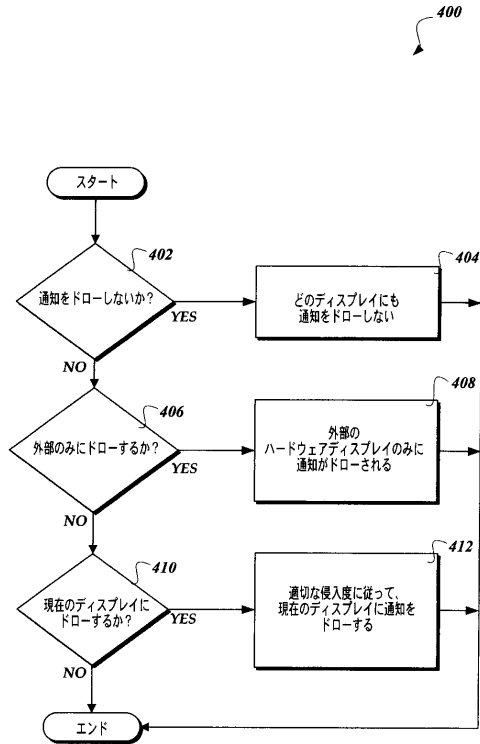
【図 7】



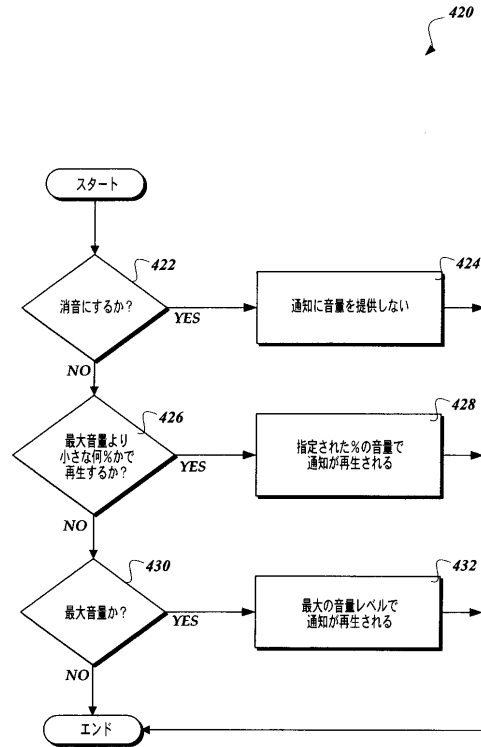
【図 8】



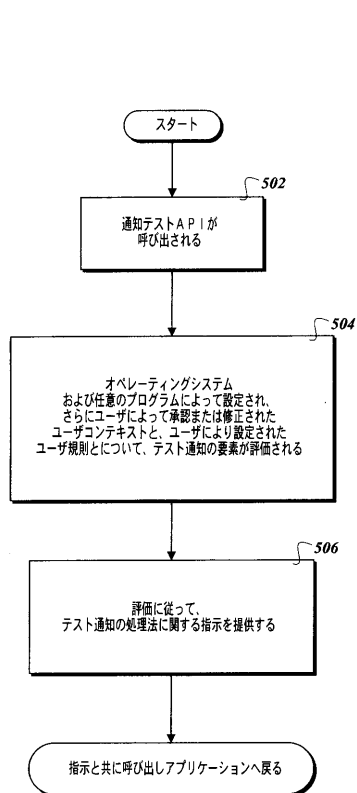
【図 9】



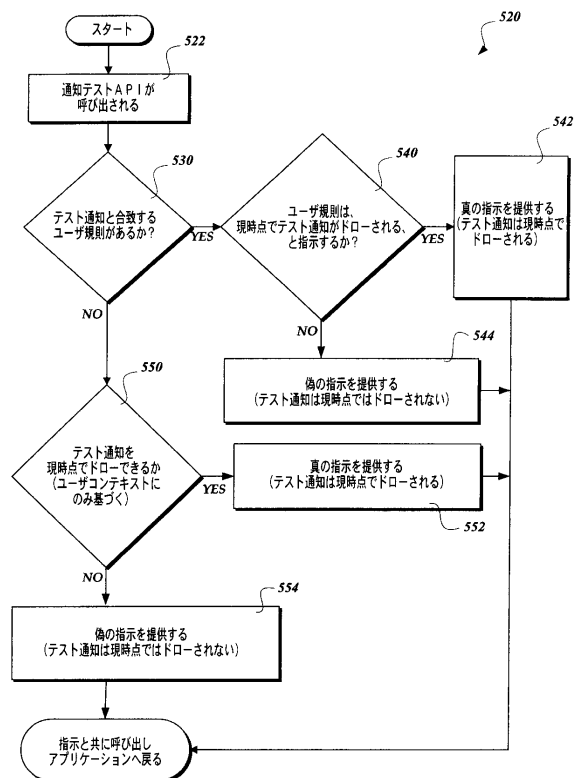
【図 10】



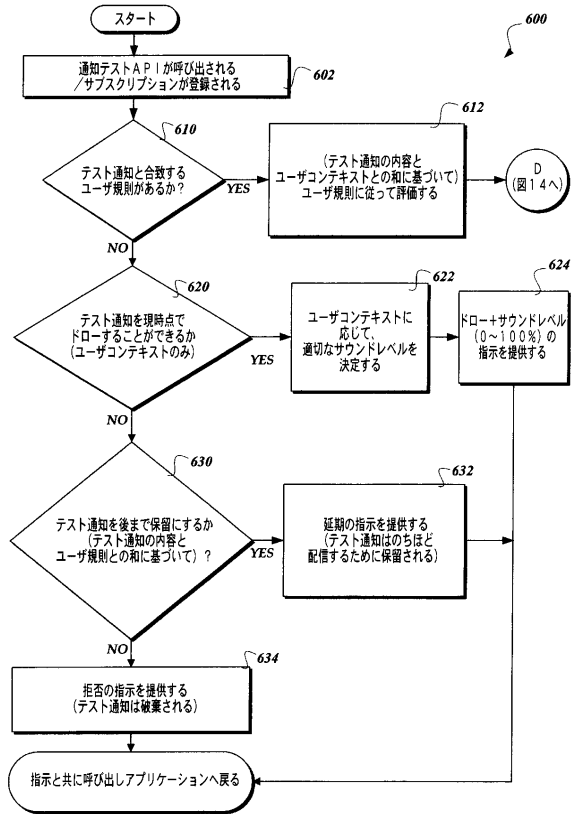
【図 11】



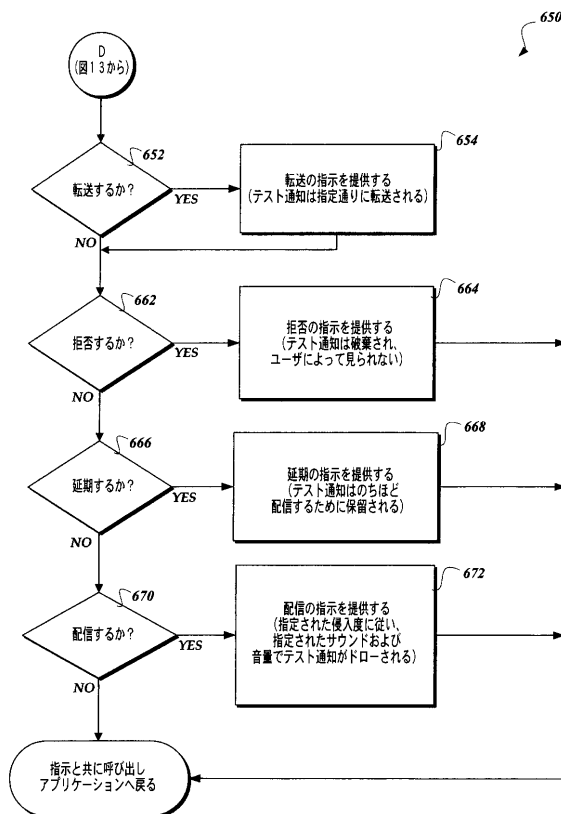
【図 12】



【図 13】



【図 14】



フロントページの続き

(72)発明者 ファブリス エー・デブリー
アメリカ合衆国 98005 ワシントン州 ベルビュー ノースイースト 7 ストリート 1
3616 アpartment イー - 5

審査官 はま 中 信行

(56)参考文献 特開平09 - 062523 (JP, A)
特開2000 - 284972 (JP, A)
特開2001 - 014179 (JP, A)
特開2003 - 067356 (JP, A)
特表2004 - 532479 (JP, A)
特表平07 - 508116 (JP, A)
国際公開第01 / 009755 (WO, A1)

(58)調査した分野(Int.Cl., DB名)
G06F 9 / 54
G06F 13 / 00