

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5924560号
(P5924560)

(45) 発行日 平成28年5月25日 (2016. 5. 25)

(24) 登録日 平成28年4月28日 (2016. 4. 28)

(51) Int. Cl. F I
G06Q 50/10 (2012.01) G06Q 50/10
G06Q 30/02 (2012.01) G06Q 30/02 140

請求項の数 16 (全 18 頁)

(21) 出願番号	特願2014-548854 (P2014-548854)	(73) 特許権者	514160320
(86) (22) 出願日	平成24年12月19日 (2012. 12. 19)		ソハロ インコーポレイテッド
(65) 公表番号	特表2015-507792 (P2015-507792A)		アメリカ合衆国 カリフォルニア州 94
(43) 公表日	平成27年3月12日 (2015. 3. 12)		301 パロ アルト ミドルフィールド
(86) 国際出願番号	PCT/US2012/070723		ロード 575 スイート 170
(87) 国際公開番号	W02013/096506	(74) 代理人	100092093
(87) 国際公開日	平成25年6月27日 (2013. 6. 27)		弁理士 辻居 幸一
審査請求日	平成26年8月25日 (2014. 8. 25)	(74) 代理人	100082005
(31) 優先権主張番号	13/335, 910		弁理士 熊倉 禎男
(32) 優先日	平成23年12月22日 (2011. 12. 22)	(74) 代理人	100067013
(33) 優先権主張国	米国 (US)		弁理士 大塚 文昭
		(74) 代理人	100086771
			弁理士 西島 孝喜
		(74) 代理人	100109070
			弁理士 須田 洋之

最終頁に続く

(54) 【発明の名称】 報酬システム及び方法

(57) 【特許請求の範囲】

【請求項 1】

報酬を受け取ることのできるユーザ、オブジェクト、前記ユーザと報酬及び報酬価値を正当化する、前記ユーザと前記オブジェクトとの間の特定のソーシャルネットワーク接続を特定する、複数の報酬ルールを含むストアと、

前記ユーザのコンピューティング装置と、

前記ユーザのコンピューティング装置と相互作用するコンピュータ実施の報酬稼ぎエンジンと、
を備え、

前記報酬稼ぎエンジンは、該前記報酬稼ぎエンジンとソーシャルネットワークとを相互作用するセマンティックグラフドライバと該ソーシャルネットワークからのユーザ情報を検索して報酬を決定するためのマンティックグラフを有しており、

前記報酬稼ぎエンジンは、前記ストア及び前記セマンティックグラフドライバを用いて前記ソーシャルネットワークから1組のユーザ情報から報酬ルールを検索し、該検索はユーザのコンテキストによって制限されており、かつ前記検索された報酬ルールで特定されたユーザ、オブジェクト、及び特定のソーシャルネットワーク接続と前記1組のユーザ情報との間に一致があるかどうか決定し、そして前記報酬ルールと前記1組のユーザ情報との間に一致がある場合にはユーザの前記コンピューティング装置に報酬を通知するようになっている、報酬システム。

【請求項 2】

前記ストアに結合されたコンピュータ実施の報酬定義エンジンを更に備え、その報酬定義エンジンは、報酬マネージャーが前記ストアに記憶される新たな報酬ルールを生成するのを許す、請求項 1 に記載のシステム。

【請求項 3】

前記セマンティックグラフは、フェースブックであり、前記報酬稼ぎエンジンは、フェースブックアプリケーションである、請求項 1 に記載のシステム。

【請求項 4】

前記ユーザのコンテキストは、特定のユーザ識別子、報酬ルールの範囲、及びユーザに対する最後の問合せのタイムスタンプの 1 つ以上である、請求項 1 に記載のシステム。

【請求項 5】

前記報酬価値は、ポイント及び金額の一方である、請求項 1 に記載のシステム。

【請求項 6】

前記報酬定義エンジンは、ウェブアプリケーションである、請求項 2 に記載のシステム

【請求項 7】

前記報酬稼ぎエンジンは、フェースブックアプリケーションである、請求項 6 に記載のシステム。

【請求項 8】

前記報酬稼ぎエンジンは、報酬に関してユーザに非同期でメッセージを送り、そして次の同期ユーザ相互作用が生じるまで報酬メッセージを記憶する、請求項 1 に記載のシステム。

【請求項 9】

ユーザが報酬を稼ぐためにアクションをとるところの提携サービスを更に備えた、請求項 1 に記載のシステム。

【請求項 10】

前記複数の報酬ルールは、サブルールにおいてアクションをとる 1 組の友人に対してユーザが報酬を稼ぐところのアグリゲートフレンドルールを含む、請求項 1 に記載のシステム。

【請求項 11】

報酬定義エンジン及び報酬稼ぎエンジンを有するコンピュータ実施の報酬システムを使用してユーザに報酬を与える方法において、

報酬稼ぎエンジンによってストアから報酬ルールを検索し、その報酬ルールは、報酬を受け取ることのできるユーザ、オブジェクト、報酬及び報酬価値を正当化する前記ユーザと前記オブジェクトとの間の特定のソーシャルネットワーク接続を特定しており、

ユーザのコンテキストにより検索が制限されるソーシャルネットワークセマンティックグラフから 1 組のユーザ情報を、前記ソーシャルネットワークからのユーザ情報を検索して報酬を決定するために、前記報酬稼ぎエンジンとソーシャルネットワークとの相互作用するセマンティックグラフドライバを有する前記報酬稼ぎエンジンにより、検索し、

前記検索された報酬ルールにおいて特定されたユーザ、オブジェクト及び特定のソーシャルネットワーク接続と前記 1 組のユーザ情報との間に一致があるかどうか前記報酬稼ぎエンジンにより決定し、及び

前記報酬ルールと前記 1 組のユーザ情報との間に一致がある場合には前記報酬稼ぎエンジンにより報酬をユーザに通知する、
ことを含む方法。

【請求項 12】

コンピュータ実施の報酬定義エンジンを使用して、新たな報酬ルールを生成し、そしてその新たな報酬ルールを、前記報酬稼ぎエンジンによりアクセス可能なストアへ記憶することを更に含む、請求項 11 に記載の方法。

【請求項 13】

前記セマンティックグラフは、フェースブックであり、前記報酬稼ぎエンジンは、フェ

10

20

30

40

50

ースブックアプリケーションである、請求項 1 1 に記載の方法。

【請求項 1 4】

前記報酬ルールを検索することは、報酬ルールが繰り返しを制限するかどうか決定することを更に含む、請求項 1 1 に記載の方法。

【請求項 1 5】

前記報酬ルールを検索することは、ユーザがルールの繰り返し限界を越えたかどうか決定することを更に含む、請求項 1 4 に記載の方法。

【請求項 1 6】

前記ユーザのコンテキストは、特定のユーザ識別子、報酬ルールの範囲、及びユーザに対する最後の問合せのタイムスタンプの 1 つ以上である、請求項 1 1 に記載の方法。

10

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本発明は、一般的に、別のソフトウェアアプリケーションに一体化できる報酬システム及び方法に関する。

【0 0 0 2】

アペンディックス：アペンディックス A (4 ページ) は、報酬稼ぎエンジンを実施するのに使用できる P H P コードのセットである。アペンディックス A は、本明細書の一部分を形成し、参考としてここに援用される。

【背景技術】

20

【0 0 0 3】

小さなババママ店から有名ブランドまでのほとんどの企業は、彼等の顧客と連絡を持ちそしてソーシャルメディアを使用して新たな顧客の注目を集めることが課題である。慣習的なマーケティングツールは、急速に増加するソーシャルメディア (主としてフェイスブック) をマーケティングチャンネルとして取り扱うには不十分であり、新たなソーシャルメディア会社は、典型的に、ニッチメディアマネージメント及び分析ツールに焦点を当てている。加えて、商人のロングテールにロイヤルティポイントを配布するための市場は、展開のための時間、カスタマイズ、及び所有権の全体的コストのために、気力がくじかれている。ほとんどの S a a S 会社は、ウェブサイトの助けとなるように、e コマースショップ、C M S、支払又は分析に焦点を当てている。

30

【0 0 0 4】

会社のソーシャルコマース e c o s s y s t e m は、次のような多数のセグメントに分割される。

ゲーム

メディア (C M S)

分析

e コマース

支払

ロイヤルティ

【0 0 0 5】

40

これらのセグメントはある程度オーバーラップするが、ブランドに対する解決策を提供することに直面している会社は、主として、ゲーム、分析及び支払通貨に焦点を当てている。I n v o l v e r、B u d d y M e d i a、及び W i l d f i r e のようなソーシャルプラグイン/メディア会社は、ブログ、ウェブサイト、ツイッター及びファンページからのクロス・プラットフォーム、リッチ・コンテンツマネージメントを管理するツールに焦点を当てている。又、ある事業は、コンテストのための第三者 a p p s 及び解決策を促進するために発展している。

【0 0 0 6】

幾つかの会社は、スケーリング可能なホワイトレベルプラットフォームの形態の報酬振舞いに焦点を当てている。例えば、S h o p k i c k 及び F o u r s q u a r e は、ストア

50

内のジオロケーションサービスを使用するチェックイン又はモバイルクライアント自体からのチェックインのみに報酬を与える彼ら自身のソーシャルネットワーク（ノン・フェースブック）を構築した。しかしながら、これらのノン・フェースブックソーシャルネットワークは、その範囲が限定され、且つ報酬の範囲も限定されている。

【0007】

より広い範囲を有し、特定のソーシャルネットワークに限定されず、複数の報酬ルールを生成することができ、且つソーシャルグラフを聴取してインスタント報酬稼ぎトランザクションを記録できるようにする報酬システムを提供することが要望される。これは、慣習的なマーケティングファンネルを通して顧客を獲得する試みに多額の金銭を消費する多くの企業の実世界での挑戦を解決するものである。この望ましいシステムは、他の有料マーケティング技術（有料調査、広告、等）に比して顧客当たりの獲得コストを下げる

10

と同時に、顧客は、真の商品及びサービスに交換するのに使用できるインスタント報酬を楽しむことができる。

【0008】

ソーシャルゲームはかなり普及しており、ゲームコミュニティは、ゲームのレベルを達成するためほとんどのゲームにおいてポイントを稼ぐが、ピア間のソーシャル状態以外、達成のための有形の真の報酬はない。ゲーム及び他の毎日の活動に対してユーザに報酬を与えるシステムを提供することが要望される。又、Root Musicのようなファンページをベースとする解決策は、バンドに縦断的に焦点が合わせられ、そしてロイヤルティでも報酬でもないイベント駆動のツールを与える。

20

【0009】

従って、現在のシステム及び方法の制約を克服する報酬システム及び方法を提供することが要望され、本発明は、その目的に向けられる。

【図面の簡単な説明】

【0010】

【図1】報酬定義エンジン（RDE）及び報酬稼ぎエンジン（REE）を備えた報酬システムの実施形態を示す図である。

【図2】報酬を決定しそして一致した報酬に作用するプロセスを含む報酬方法を示す図である。

【図3】セマンティックグラフからの情報に対するマッチングルールのプロセスを含む報酬方法を示す図である。

30

【図4】セマンティックグラフ及び報酬エンドユーザの両方と相互作用する報酬システムの別の「非同期」アーキテクチャーを示す図である。

【図5】「提携サービス」の関与を通して報酬をトリガーするための別のメカニズムを有する報酬システムの別の実施形態を示す図である。

【図6】「アグリゲートフレンドルール」として知られているREEルールの一形式の一例を示す図である。

【図7A】報酬システムをサポートするのに使用されるデータベーススキーマの一例を示す図である。

【図7B】報酬システムをサポートするのに使用されるデータベーススキーマの一例を示す図である。

40

【発明を実施するための形態】

【0011】

本開示は、特に、図示して以下に説明するコンピュータで実施される報酬システム及び方法に適用することができ、この分野に関して本発明を説明する。しかしながら、このシステム及び方法は、本開示の範囲内で他の仕方で実施できると共に、本開示の範囲内の他のコンピュータアーキテクチャーを使用しても実施できるので、広い利用性をもつことが明らかである。

【0012】

このシステムは、グローバルな報酬振舞いを指定し且つ顧客をターゲットにし直すこと

50

ができ、他の有料マーケティング技術（有料調査、広告、等）に比して顧客当たりの獲得コストを下げるができる。典型的に、ロイヤルティ通貨価値は、製品価格に割引を与えるコストの一部分であり（即ち、平均エアマイルコスト\$0.001）、顧客は、同時に、真の商品及びサービスに交換するのに使用できるインスタント報酬を楽しむことができる。システムのエンジンは、消費者及びブランドが毎日の活動に基づいて関わり合うための永続的なポイント稼ぎインターフェイスを与えると共に、1つの実施形態ではフェースブックグラフのようなグラフにおいて消費者振舞いのニッチセマンティックパターンに一致させるように我々の報酬振舞いエンジンをカスタマイズすることを希望する多数のブランドに到達すべくホワイトラベル、マルチ顧客能力として規模の経済性を与えることもできる。

10

【0013】

図1は、報酬定義エンジン（RDE）22及び報酬稼ぎエンジン（REE）24を備えた報酬システム20の実施形態を示す。このシステムは、報酬マネージャーコンピューティング装置26を使用してシステム20にアクセスする報酬マネージャー、及び報酬エンドユーザコンピューティング装置28を使用してシステム20にアクセスする報酬エンドユーザにより使用される。エンジン22、24は、その一方又は両方のエンジンをホストするサーバーコンピュータのようなコンピュータの処理ユニットにより実行される複数行のコンピュータコードとして各々実施される。エンジン22、24は、互いにリンクされて互いに通信する同じコンピュータ又は異なるコンピュータにホストされる。コンピューティング装置26、28は、ワイヤード又はワイヤレスリンクのようなリンク、例えば、インターネット、イーサネット（登録商標）、ワイヤレスデジタルデータネットワーク、ワイヤレスネットワーク、コンピュータネットワーク、等を経て、システムにアクセスし、通信する。各コンピューティング装置26、28は、メモリ、持続性ストレージ装置、ディスプレイ、及びシステム20と相互作用できる接続部（ワイヤレス回路、等）を伴う処理ユニットベースの装置である。例えば、各コンピューティング装置は、スマートフォン装置（アップルiPhone、RIMブラックベリー、アンドロイドベースの装置）、タブレットコンピュータ、パーソナルコンピュータ、セルラーホン、等である。

20

【0014】

コンピューティング装置26を使用する報酬マネージャーは、報酬定義エンジン（RDE）22を使用して報酬ルールを定義し、そして報酬稼ぎエンジン（REE）24は、コンピューティング装置28を使用して報酬エンドユーザにそれらのルールを適用し、どのユーザが何の報酬を得るか決定する。1つの実施形態において、RDE22は、報酬を導く条件を定義するフォーム、並びに報酬の種類及び額をユーザに提示するユーザインターフェイスを含むウェブアプリケーションである。従って、RDE22をホストするコンピュータは、ウェブアプリケーションからユーザへウェブページを配給しそしてウェブフォームに対してユーザから情報を受け取るウェブサーバー（ソフトウェア又はハードウェアベースの）を有する。RDE22は、ルールの種類、ルールの各種類のパラメータ、及び特定のセマンティックグラフの用語を決定するセマンティックグラフドライバ（SGD）30を収容する。1つの実施形態において、セマンティックグラフは、フェースブックであり、それ故、SGDは、フェースブックAPIによりサポートされる接続及びオブジェクトタイプ（好み、友情、等）を決定する。別の実施形態では、セマンティックグラフは、ツイッターであり、SGDは、アカウント、フォロー、ツイート、リツイート、等に関する。ユーザの振舞い又はユーザの振舞いの作用をエンコードし、そしてその情報にアクセスするための第三者のAPIを与えるシステムを使用する他の実施形態は、YouTube（登録商標）、Google+、LinkedIn、eBay、及びStack Overflowを含む。又、RDE22は、報酬マネージャーのルールを、報酬ルールストア34のデータベーススキーマへとエンコードするデータベースドライバ32も有する。1つの実施形態において、報酬ルールストア34は、フェースブックユーザ、マーケティングキャンペーン、キャンペーンイベント（ルール）、接続タイプ、及びオブジェクトidを表わすテーブルを伴う関連データベースである（その一例が図7A及び7Bに示さ

30

40

50

れている)。

【0015】

REE24は、データベースドライバ36を使用してストア34からそれらのルールをロードする。特定のユーザが報酬を稼いだかどうか決定するために、REEは、セマンティックグラフドライバ(SGD)38を使用して、報酬ルールに関する情報をセマンティックグラフから検索し、次いで、その情報を報酬ルールに対して直接マッチングさせる。例えば、フェイスブックがそれらのセマンティックグラフにおいて表現し且つそれらのグラフAPIを通して利用できるようにする幾つかの接続がある。

フロム・オブジェクトタイプ	接続	ツー・オブジェクトタイプ
ユーザ	友人	ユーザ
ユーザ	好き	(ファン)ページ
ユーザ	チェックイン	チェックイン・インスタンス (プレイスペース)
ユーザ	聴く	音楽
ユーザ	見る	ビデオ
ユーザ	答える	ポール/クイズ質問
ユーザ	加わる	グループ
ユーザ	読む	ニュース

10

【0016】

従って、各ルールは、現在のエンドユーザが特定のオブジェクトに対して持たねばならない特定の接続を指定し、例えば、ブランドのファンページにリンクした、又は特定ストアのプレイスペースでチェックインを行った、である。各一致に対して、REE24は、ユーザ通知ドライバ40を使用して各報酬エンドユーザ28に通知し、そしてアカウントドライバ44を経て報酬アカウントサービス(RAS)42に通知する。

20

【0017】

システムのセマンティックグラフサービス46は、モバイルApps48、セマンティックグラフ50に焦点を当てたウェブサイト、セマンティックグラフ52に焦点を当てないウェブサイト上のウェブウィジェット、及びセマンティックグラフプラットフォームで実施されるウェブapps54、を使用するユーザを含む種々のソースから各セマンティックグラフの情報を累積する。これら全てのソースは、セマンティックグラフ所有者又は第三者により提供される。

30

【0018】

1つの実施形態において、REE24は、URLにより直接アクセスされるフェイスブック(Facebook(登録商標))アプリケーションの一部分であり、及び/又はフェイスブックページにタブとして実施される。この実施形態におけるセマンティックグラフは、フェイスブックであり、従って、SGD30、38は、フェイスブックグラフAPI(FGA)を使用して、ユーザとフェイスブックオブジェクトとの間の接続に関する情報を検索する。ユーザ通知ドライバ40は、稼いだ報酬についてユーザに通知するようにフェイスブックアプリケーションのユーザインターフェイスを変更する。例えば、報酬を提供するブランドに対するフェイスブックファンページの報酬タブを訪問する際に、ルールが前記SGDからのセマンティックグラフ情報に一致する場合には、ファンページタブのコンテンツは、「おめでとう、あなたは、このファンページをリンクするために10ポイント勝った」を含む。又、このユーザインターフェイスは、過去の報酬に関する情報を(例えば、「報酬履歴」元帳として)、及び他の潜在的な報酬を記述及びハイパーリンクとして含むことができる。1つの実施形態において、報酬は、プライベート又はパブリック通貨でのポイント数又はクレジット数より成る。アカウントドライバ44は、オンラインAPIを使用して報酬アカウントサーバー42に連絡し、ユーザのアカウントに報酬額を振り込むことを要求する。

40

【0019】

50

報酬システムの他の考えられる実施形態は、セマンティックグラフがフェイスブックの他に別のソーシャルネットワークであり、ユーザ通知ドライバが非同期メッセージングを使用してエンドユーザに通知し、報酬アカウントサーバーがローカルにホストされ、セマンティックグラフ情報がローカルにキャッシュされ、そしてアカウントドライバが報酬アカウントサービスへの要求をバッチ処理するような具現化を含む。

【0020】

図2は、報酬を決定しそして一致する報酬に作用するプロセスを含む報酬のための方法60を示す。例えば、図2は、各ルールを処理しそして特定のユーザについて特定のルールに基づき一致した報酬に作用するためのR E E内のロジックを示しているが、この方法は、他のコンピュータシステムで実行することもできる。この方法において、R E Eは、ルールを検索し(62)、そしてそのルールを現在ユーザに適用するかどうか決定するが、これは、そのルールが、それが繰り返される回数を制限するかどうか決定し(64)、もしそうであれば、ユーザの報酬履歴を調べて、その回数が既に満たされたかどうか決定する(66)ことによって行う。繰り返し限界に達していないか、又はユーザによる繰り返し限界に達していない場合には、R E Eは、図3について以下に詳細に述べるように、十分な情報をセマンティックグラフから検索して、ルールが現在ユーザに一致するかどうか決定する(70)。一致がある場合には、R E Eは、エンドユーザに通知し(72)、アカウントサーバーに通知する(74)。任意であるが、セマンティックグラフサービスがそれをサポートする場合には、R E Eは、セマンティックグラフサービスに報酬に関して通知することができる。

【0021】

図2に示すように、繰り返し限界を満足し、ユーザが特定のルールに一致せず、及び/又はルールに一致し及び通知が行われた場合には、R E Eは、特定のユーザに対してチェックすべきルールが更にあるかどうか決定し(78)、次いで、付加的なルールがあるか又はプロセスが完了した場合には新たなルールを検索する(62)。1つの実施形態において、フェイスブックは、セマンティックグラフであり、R E Eは、フェイスブックグラフAPIからPOST、オープングラフ2.0+、通貨及び額より成る「報酬」オブジェクトの「稼ぎ」アクションを使用する。

【0022】

図3は、セマンティックグラフ102からの情報に対してルール100をマッチングさせるプロセスを含む報酬の方法を示す。特に、図1のストア34のルール100(ページ9が好きなユーザが\$20報酬ルールを得る)は、マッチャー106(即ち、R E Eの一部)によりコンテキスト104に基づいてマッチングされる。重要な概念は、R E Eが現在ユーザのコンテキスト104及びルール100のセットをどのように使用して、セマンティックグラフ102の効率的な問合せを行うかである。即ち、1つの戦略は、現在ユーザに関する全ての情報に対してセマンティックグラフに問合せすることであり、これは、各ユーザに関して非常に限定された情報を伴うセマンティックグラフについては合理的な戦略であるが、(フェイスブックのような)大きなセマンティックグラフでは、そのような問合せは、性能が低く、且つ処理すべき結果の禁止的に大きなセットを生じる。ルールは、第1のエンティティ(報酬を稼ぐユーザ)、第2のエンティティ(報酬が許される基礎となるフェイスブックページのようなアイテム/個人、等)、報酬を許可しなければならない場合に決定される第1のエンティティと第2のエンティティとの間の関係(フェイスブックにおける好きのような)、及び報酬額を含む。又、各ルールは、ユーザが特定の報酬を得ることのできる回数を制限する繰り返し限界も有する。

【0023】

R E Eは、セマンティックグラフ問合せを制限するための3つの情報断片の使用を含む。即ち、1)特定のユーザの識別子、2)特定ルールの範囲(当該エンティティのタイプ及び関係)、及び3)(コンテキストとしても知られている)特定のユーザ及びルールに対する最後の問合せのタイムスタンプ。セマンティックグラフによりサポートされる問合せの種類に基づいて、結果は、稼いだ報酬と1-1で対応するか、又は付加的なフィルタ

10

20

30

40

50

リングを要求する。例えば、図3は、2011年11月11日の最後のチェック以来ユーザ12によりリンクされた全てのエンティティに対する問合せの結果を示し、マッチャーは、ページ9のエンティティ2についてそれらの結果を通して見て、2人のユーザ（ユーザ5及びユーザ12）が一致となる。1つの実施形態では、セマンティックグラフは、フェイスブックであり、そして関係は、フェイスブックのソーシャルグラフの「友人」接続、フェイスブックのオープングラフ1.0+の「好き」接続、並びにフェイスブックのオープングラフ2.0+のapps-ネームスペース特有アクションを含む。1つの実施形態において、REEは、MySQLデータベースを使用するPHPクラスであり、そしてフェイスブックグラフAPIは、参考としてここに援用されるアペンディックスAに詳述されている。

10

【0024】

図4は、セマンティックグラフ及び報酬エンドユーザの両方と相互作用する別の「非同期」アーキテクチャーを示す。このアーキテクチャーにおいて、REE24は、セマンティックグラフサービス(SGS)46から、全ての登録されたユーザに対して現在ルールに関する全ての情報への契約を要求することにより、非同期で動作する。従って、関連ユーザ情報が変化するときには、SGS46は、REE24に、ユーザ情報の変化について通知する。そのようなケースでは、REE24は、変化をルールの現在セットと比較し、そして新たな報酬を稼いだかどうか決定し、もしそうであれば、図4に示したように、非同期メッセージングを経てユーザに通知するか、又は稼いだ報酬に関する情報を次の同期ユーザ相互作用のために記憶する（例えば、ユーザがファンページの報酬タブ、特殊なソ

20

【0025】

図5は、1つの実施形態では第三者ウェブサイトである「提携サービス」110の関与を通して報酬をトリガーするための別の実施形態を示す。この考え方は、ユーザが、そのウェブサイトに対して、購入を実行し、旅行を予約し、プロモーションビデオを見る、等のアクションを遂行することにより、報酬を稼げることである。このシステムは、通知APIを経て報酬イベントを遂行するユーザをウェブサイトがREE24に通知するようにさせることでこれを行う。1つの実施形態では、通知APIは、REST(URLベースの)APIを2つのメインエンリポイント、即ちCheckReward(報酬チェック)及びAuthReward(報酬許可)と共に使用する。

30

【0026】

CheckRewardは、報酬がまだ有効で且つアクティブであるかどうか、例えば、報酬ルールがまだ予算計上しているかどうか、又は報酬マネージャーが報酬ルールをターンオフしたかどうかチェックするためにサーバー/ウェブサイトにより呼び出される。例えば、次のURLに要求を発して、

<http://ree.sohalo.com/rest/checkReward.php?fromId=RM02&toId=CUST07&campaignId=CAM13>

40

報酬マネージャーRM02の報酬キャンペーンCAM13に関して有効で且つアクティブなルールがあるかどうか尋ね、もしあれば、報酬額及び通貨を返送する。次いで、ウェブサイトは、報酬を受け取る機会を表示し、そしてどのようにそれを請求するか of インストラクションを与える（1つの実施形態では、ユーザは、フェイスブックapp又はページへのリンクを通してそれを請求することができる）。CheckRewardパラメータは、次のものを含む。

fromId: 報酬寄贈者のID、REEにより決定される値（このパラメータは、オプションではなく、デフォルトがない）

toId: 報酬取得者のID、報酬マネージャーにより決定される値（このパラメータは、オプションではなく、デフォルトがない）

50

`campaignId` : 報酬をトリガーするアクションのID、報酬マネージャーにより決定される値（このパラメータは、オプションではなく、デフォルトがない）
そして非エラー返送値については、

`amount` : 報酬におけるポイント数

`currency` : REEにより定義された3文字頭字語(TLA)としての報酬の通貨

【0027】

ウェブサイト/サービスは、報酬が許可されることをREE 24に知らせるためにAuthReward URLに要求を発生し、従って、REEが報酬を準備する前に必要である。1つの実施形態では、次のURLに要求を発生して、

`http://ree.sohalo.com/rest/checkReward.php?fromId=RM02&toId=CUST07&campaignId=CAM13&amount=2¤cy=FBC&signature=GHY653GF8KWM99`

ユーザCUST07がCAM13により述べられたアクションを遂行するために報酬マネージャーRM02から2フェースブッククレジットの報酬を稼いだことを述べる。`signature`パラメータは、この要求が、認証を受けたエンティティから到来することを確認するための、他のパラメータのデジタルシグネチャーである。AuthRewardパラメータは、次のものを含む。

`fromId` : 報酬寄贈者のID、REEにより決定される値（このパラメータは、オプションではなく、デフォルトがない）

`toId` : 報酬エンドユーザのID、報酬マネージャーにより決定される値（このパラメータは、オプションではなく、デフォルトがない）

`campaignId` : 報酬をトリガーするアクションのID、報酬マネージャーにより決定される値（このパラメータは、オプションではなく、デフォルトがない）

`signature` : 他の5つのパラメータ+報酬マネージャーの機密の連鎖のデジタルシグネチャー

`amount` : 許可するための報酬ポイント数（任意、REEルールで供給されるデフォルト）

`currency` : REEにより定義されたTLAとしての報酬の通貨（任意、REEルールで供給されるデフォルト）

そしてAuthRewardは、推測不能コード又はエラーを返送する。このコードは、要求の受信者として機能し、監査の目的で有用である。

【0028】

図6は、「アグリゲートフレンドルール」120であるREEルールの一形式を示す。即ち、報酬エンドユーザは、サブルール122を遂行したあるソーシャルネットワーク友人をもつために報酬を稼ぐことができる。そのようなルールは、友人の人数及び別のルールへの参照によりRDE 22において定義される。例えば、10人の友人が、特定のフェースブックファンページを好きであるようにし（そのようにするために報酬を稼ぎ）又は5人の友人が特定の位置にモバイルチェックインするようにする。

【0029】

図7A及び7Bは、報酬システムをサポートするのに使用されるデータベーススキーマの一例を示す。データベーススキーマは、キャンペーン運動家が1つの実施形態において`campaignfanpageid`カラムに記憶された所与のフェースブックファンページにおいてプロモーションされるキャンペーンを生成するときにキャンペーンに関する情報を含む`campaign`テーブルを含む。各キャンペーンは、そのキャンペーンをプロモーションし且つマネージするために1つ以上の`campaign_managers`を有する。キャンペーンの期間は、キャンペーンテーブルのカラム`campaign_startdate`及び`campaign_enddate`に値を指定することにより定義される。（開始及び終了日時は、個々のキャンペーンイベントに付加されてもよい。）キャンペーン全体の資金となる預金は、`campaign_event_accountid`カラムにセットされたアカウントから引き出される。

10

20

30

40

50

【0030】

又、データベーススキーマは、キャンペーンマネージャーが報酬定義エンジン(RDE)を使用して、所与のキャンペーンのための1つ以上の報酬ルールを生成するcampaign_eventテーブルも含む。各個々のルールは、campaign_eventテーブルにおいてタプルで表わされる。好ましい実施形態では、endpointfbobjectidカラムに定義されたフェイスブックオブジェクトとの接続を形成するためにユーザに報酬が与えられる。接続タイプは、connection_typeテーブルに定義された1組のタイプからのエレメント(ファンページが好き、ファンページのウォールポストが好き、特定の場所にチェックインする、等)である。キャンペーンマネージャーは、defaultamount及びdefaultcurrencyidカラムに各々記憶される報酬値及び報酬通貨をセットする。又、キャンペーンマネージャーは、eventduplicationmeaningidカラムを経て、所与の接続を形成するためにユーザに報酬が与えられる回数に制約を課する。duplicationmeaningidは、duplicationmeaningテーブルに定義された1組の意味のエレメントである(許可、拒絶又は置き換え、等)。キャンペーン全体の資金となる預金は、campaign_event_accountidカラムに記録されたアカウントから引き出される。所与の報酬(これは先験的に知られている)を与えるに十分な預金残高がないときには、それに関連するcampaign_eventが報酬稼ぎエンジンにより除外される。キャンペーンマネージャーは、親及び/又は子のアカウント間で預金を適宜に自由に移動し、各キャンペーンは、関連するcampaign_eventサブアカウント間で分布及び再分布できる全体的なバジェットを有する。個々のcampaign_eventは、isenabledカラムにおいてブールのフラグをセットすることにより休止及び再開することができる。

10

20

【0031】

又、データベーススキーマは、rewardeeテーブルも含み、キャンペーンファンページへの各訪問者は、rewardeeテーブルにおいてタプルとして表わされる。各報酬取得者は、このキャンペーンに対するユーザへの報酬入金を記録するrewardee_accountidを有する。個々のアカウントトランザクションは、account_action_transactionとして表わされる。

【0032】

又、データベーススキーマは、reward_event_actionテーブルも含み、報酬が報酬取得者に入金されるとき、event_actionテーブルにタプルが追加され、関連タプルがreward_event_actionに追加される。各reward_event_actionは、campaign_eventアカウントからの引き落とし及び報酬取得者のアカウントへの振込みを表わすための1組の関連するaccount_action_transactionを有する。

30

【0033】

以上、本発明の特定の実施形態を説明したが、当業者であれば、本発明の精神及び範囲から逸脱せずにこの実施形態に変更がなされ得ることが明らかであろう。従って、本発明の範囲は、特許請求の範囲において限定される。

40

【0034】

アペンディックスA

APPENDIX A
REE PHP EXEMPLARY CODE

```

<?php
/**
 * RewardVisit: a class for checking for earned rewards when visiting a
 * page
 *
 * @author jay.weber@sohalo.com
 * @copyright Sohalo 2011
 */

require "/var/www/html/core/sql/db.mysql.php";

class RewardVisit
{
    private $dbc = false;
    private $fbuid = 0;
    private $token = "";
    private $rewardeeid = 0;

    function __construct($fbuid, $token, $dbc) {
        $this->dbc = $dbc ? $dbc : new db_c();
        $this->fbuid = $this->dbc.escape_string($fbuid);
        $this->token = $this->dbc.escape_string($token);

        /* create rewardee as necessary using api? ask Stuart */

        $q = "SELECT rewardeeid FROM rewardee WHERE fbuserid={$this->fbuid}";
        if ($res = $this->dbc.query($q)) {
            if ($this->dbc.num_rows($res)) {
                $row = $this->dbc.fetch_array($res);
                $this->rewardeeid = $row[0];
                $r = true;
            }
            else {
                error_log("Rewardee with fbuserid {$this->fbuid} should
                exist but doesn't");
                $r = false;
            }
            $this->dbc.free_result($res);
            return $r;
        }
        else {
            error_log("Query Failed: $q");
            return false;
        }
    }

    /* public methods */

    public function updateRewardEvents() {
        if ($offers = getRelevantOffers()) {
            if ($matches = getMatchesFromFB($offers)) {
                insertNewRewardEvents($matches);
            }
        }
    }

    public function getOpenRewardEvents() {
        // find reward events with dispositions that mean 'earned but not
        redeemed'
        $q = "SELECT
actionvalue,actioncurrencyid,campaigneventid,createdate,campaigneventdes

```

```

cription FROM reward_event_action,event_action,campaign_event WHERE
reward_event_action.eventactionid=event_action.eventactionid AND
campaign_event.campaigneventid=event_action.campaigneventid AND
rewardeeid={$this->rewardeeid} AND dispositionid=1 ORDER BY createdate
DESC";
    if ($res = $this->dbc.query($q)) {
        $events = array();
        while ($a = $this->dbc.fetch_assoc($res)) {
            $events[] = $a;
        }
        $this->dbc.free_result($res);
        return $events;
    }
    else {
        error_log("Query Failed: $q");
        return false;
    }
}

/* private methods */

private function getRelevantOffers() {
    // first, count the number of times offers have already been
redeemed
    $q = "SELECT campaigneventid, COUNT(*) AS count FROM event_action
GROUP BY campaigneventid";
    if ($res = $this->dbc.query($q)) {
        $eventcounts = array();
        while ($a = $this->dbc.fetch_assoc($res)) {
            $eventcounts[$a['campaigneventid']] = $a['count'];
        }
        $this->dbc.free_result($res);

        // now find all the offers that apply to fb actions
        $q = "SELECT campaigneventid, campaigneventdescription,
endpointfbobjectid, connection_type.connectiontypeid, defaultamount,
defaultcurrencyid,eventduplicationmeaningid, connectiontypename FROM
campaign_event,connection_type WHERE isenabled=1 AND endpointfbobjectid
IS NOT NULL AND connectiontypeid IS NOT NULL AND
campaign_event.connectiontypeid=connection_type.connectiontypeid ORDER
BY connectiontypeid,endpointfbobjectid";
        if ($res = $this->dbc.query($q)) {
            $offers = array();
            while ($a = $this->dbc.fetch_assoc($res)) {
                // filter out offers that can't be duplicated
                // todo: generalize to other duplication policies
                if ( ! ($a['eventduplicationmeaningid'] == 3 &&
isset($eventcounts[$a['campaigneventid']])) ) {
                    $offers[] = $a;
                }
            }
            $this->dbc.free_result($res);
            return $offers;
        }
        else {
            error_log("Query Failed: $q");
            return false;
        }
    }
    else {
        error_log("Query Failed: $q");
        return false;
    }
}
}

```

```

private function getRequestGraphApi($connectiontype, $sincecuts) {
    $url = "https://graph.facebook.com/{${this-
>fbuid}}/{${connectiontype}?access_token={${this-
>token}&since={${sincecuts}}";
    $_options = array(
        CURLOPT_URL => $url,
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_HEADER => false);
    $_curl = curl_init();
    curl_setopt_array($_curl, $_options);
    $_response = curl_exec($_curl);
    curl_close($_curl);
    return json_decode($_response, true);
}

private function getMatchesFromFB($offers) {
    // compare the offers to this user's facebook data
    // assuming offers sorted by connectiontypeid
    $matches = array();
    $luts = getLastRewardCheckTimestamp();

    // ask facebook for object changes relevant to $offers since $luts
    $offertype = "";
    $connectedobjs = array();
    foreach ($offers as $offer) {
        // first get the fbdata for this connection (if we don't
already have it)
        if ($offertype != $offer['connectiontypename']) {
            $fbresponse = getRequestGraphApi($offertype, $luts);
            if (isset($fbresponse['data'])) {
                $fbdata = $fbresponse['data'];
            }
            else {
                $fbdata = array();
                error_log("Unexpected response from graph api
".serialize($fbresponse));
            }
            // move $fbdata into an assoc array for quick access (but
use non-numeric keys to avoid huge arrays)
            foreach ($fbdata as $connection) {
                $connectedobjs['id:'.$connection['id']] =
$connection['name'];
            }
            if (isset($connectedobjs['id:'.$offer['endpointfbobjectid']]))
{
                $matches[] = $offer;
            }
        }
        setLastRewardCheckTimestamp();
    }
    return $matches;
}

private function insertNewRewardEvents($events) {
    foreach ($events as $a) {
        $sq = "INSERT INTO event_action SET
campaigneventid={${a['campaigneventid']},actioncategoryid=1,actioncurrenc
yid={${a['defaultcurrencyid']},actionvalue={${a['defaultamount']},createdate=NOW()";
        if (${this->dbc.query($sq)} {
            $sq = "INSERT INTO reward_event_action SET
eventactionid=LAST_INSERT_ID(), rewardeeid={${this->rewardeeid}";
            if (${this->dbc.query($sq)} {

```


【図3】

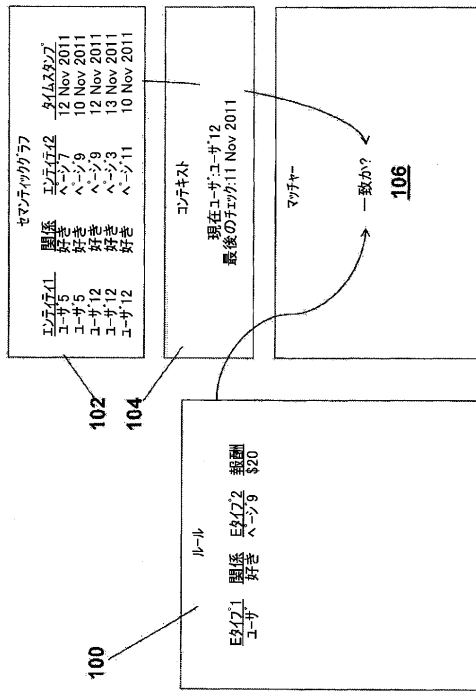


Figure 3

【図4】

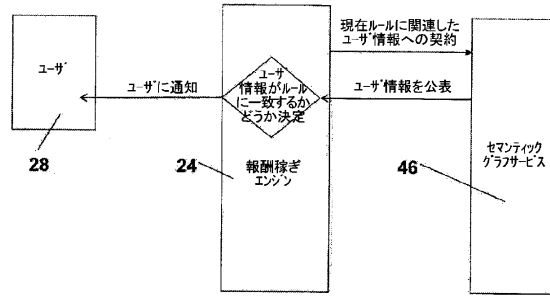


Figure 4

【図5】

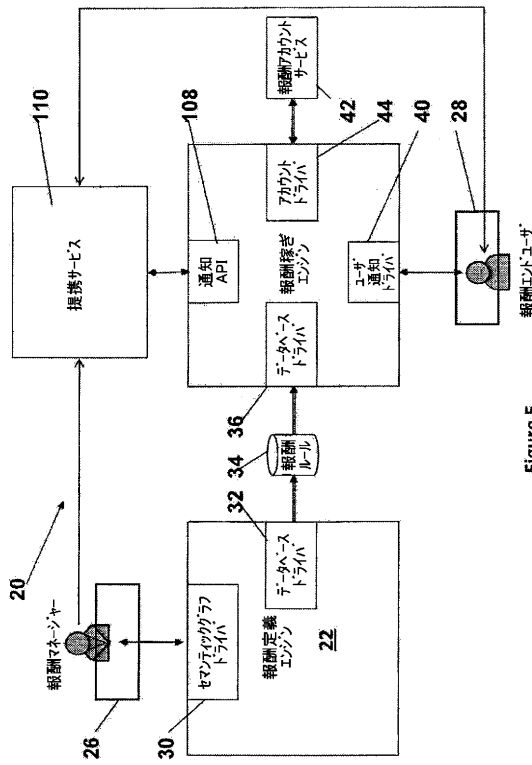


Figure 5

【図6】

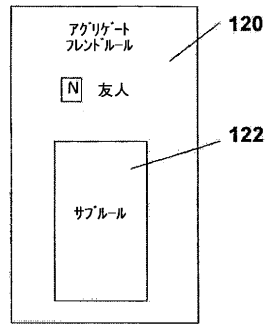


Figure 6

フロントページの続き

- (74)代理人 100109335
弁理士 上杉 浩
- (74)代理人 100120525
弁理士 近藤 直樹
- (72)発明者 ウェバー ジェイ シー
アメリカ合衆国 カリフォルニア州 94304 パロ アルト パンデラ アベニュー 311
0
- (72)発明者 バトラー スチュアート エム
アイルランド カウンティー ダブリン ブラックロック クレイグモア ガーデنز 8
- (72)発明者 ゲラティエー マイケル ピー
アメリカ合衆国 カリフォルニア州 94301 パロ アルト エバレット 535 アパート
メント 612

審査官 貝塚 涼

- (56)参考文献 米国特許出願公開第2010/0250359(US,A1)
米国特許出願公開第2004/0054590(US,A1)
特開2009-145574(JP,A)
特表2010-537323(JP,A)

- (58)調査した分野(Int.Cl., DB名)
G06Q 10/00 - 50/34