

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4536817号
(P4536817)

(45) 発行日 平成22年9月1日(2010.9.1)

(24) 登録日 平成22年6月25日(2010.6.25)

(51) Int. Cl. F I
G06Q 50/00 (2006.01) G06F 17/60 132
G06Q 10/00 (2006.01) G06F 17/60 162C

請求項の数 8 (全 17 頁)

(21) 出願番号	特願2008-548548 (P2008-548548)	(73) 特許権者	500046438
(86) (22) 出願日	平成18年12月7日 (2006.12.7)		マイクロソフト コーポレーション
(65) 公表番号	特表2009-522648 (P2009-522648A)		アメリカ合衆国 ワシントン州 9805
(43) 公表日	平成21年6月11日 (2009.6.11)		2-6399 レッドモンド ワン マイ
(86) 国際出願番号	PCT/US2006/047240		クロソフト ウェイ
(87) 国際公開番号	W02007/078673	(74) 代理人	100140109
(87) 国際公開日	平成19年7月12日 (2007.7.12)		弁理士 小野 新次郎
審査請求日	平成21年11月4日 (2009.11.4)	(74) 代理人	100089705
(31) 優先権主張番号	11/321,777		弁理士 社本 一夫
(32) 優先日	平成17年12月29日 (2005.12.29)	(74) 代理人	100075270
(33) 優先権主張国	米国 (US)		弁理士 小林 泰
早期審査対象出願		(74) 代理人	100080137
			弁理士 千葉 昭男
		(74) 代理人	100096013
			弁理士 富田 博行

最終頁に続く

(54) 【発明の名称】 エンドユーザによるワークフローの動的な再配置

(57) 【特許請求の範囲】

【請求項 1】

コンピュータ実行可能なコンポーネントを格納するコンピュータ可読記憶媒体を備え、
前記コンピュータ実行可能なコンポーネントは、

複数のワークフロー状態及び前記複数のワークフロー状態に対応する複数のウェブページを含むワークフローを提供するワークフロープロバイダであって、前記複数のウェブページは、

前記複数のワークフロー状態のうち第1のワークフロー状態に対応する第1のウェブページと、

前記複数のワークフロー状態のうち前記ワークフローにおいて前記第1のワークフロー状態より後に位置する第2のワークフロー状態に対応し、前記ワークフローにおいて前記第1のウェブページより後に位置する第2のウェブページと、

前記複数のワークフロー状態のうち前記ワークフローにおいて前記第1のワークフロー状態より後であって前記第2のワークフロー状態より前に位置する少なくとも1つの中間ワークフロー状態に対応し、前記ワークフローにおいて前記第1のウェブページより後であって前記第2のウェブページより前に位置する少なくとも1つの中間ウェブページと

を含む、ワークフロープロバイダと、

前記ワークフローのエンジンの一部としての動的状態マネージャコンポーネントとを含み、前記動的状態マネージャコンポーネントは、前記第2のウェブページから前記第1のウ

10

20

ウェブページへ戻るブラウザのブラウジングに応答して、前記ワークフローにおけるランタイムでのアドホック状態の変更を可能にするため、前記ワークフローの一部としてモデル化されていない前記複数のワークフロー状態間の1つ以上の関係を作成し、

前記第2のウェブページから前記第1のウェブページへ戻るブラウザのブラウジングに応答して作成される前記1つ以上の関係は、

前記第1のウェブページの提出に応答して、前記ワークフローに、前記第1のウェブページから前記第2のウェブページへナビゲートさせ且つ前記少なくとも1つの中間ウェブページを飛ばさせる、前記第1のウェブページと前記第2のウェブページとの間の関係を含む、コンピュータによって実装されるシステム。

【請求項2】

前記コンピュータ実行可能なコンポーネントは、前記ワークフローを特定の点で停止し、前記複数のワークフロー状態を駆動するホストアプリケーションからの入力を待つ対話型アクティビティ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1に記載のコンピュータによって実装されるシステム。

【請求項3】

前記ワークフローは、ブラウザに表示されるアプリケーションナビゲーションコントロールに関連付けられることを特徴とする請求項2に記載のコンピュータによって実装されるシステム。

【請求項4】

各表示と各状態との間に1対1の関連付けが存在することを特徴とする請求項3に記載のコンピュータによって実装されるシステム。

【請求項5】

前記コンピュータ実行可能なコンポーネントは、前記ホストによって供給されるナビゲーション経路をさらに備えることを特徴とする請求項4に記載のコンピュータによって実装されるシステム。

【請求項6】

前記ナビゲーション経路は、「次ページ (go - to)」の形態を有することを特徴とする請求項5に記載のコンピュータによって実装されるシステム。

【請求項7】

前記イベントはユーザインタフェースに対応することを特徴とする請求項5に記載のコンピュータによって実装されるシステム。

【請求項8】

前記イベントは、前記ワークフローのロジックの一部としてモデル化されていないアドホックイベントであることを特徴とする請求項5に記載のコンピュータによって実装されるシステム。

【発明の詳細な説明】

【背景技術】

【0001】

高コスト、低性能のデータ処理システムから、低コスト、高性能の通信システム、問題解決システム、および娯楽システムへのコンピュータおよびネットワーク技術の発展により、通信、請求の支払い、買い物、予算編成、情報収集等の日常作業を行う負担を軽減する、費用対効果が大きく、時間を節約する手段がもたらされた。現在、有線または無線技術を介してインターネットにつながれたコンピューティングシステムは、世界中に存在する多数のウェブサイトやサーバにある豊富な情報に容易にほぼ瞬時にアクセスする経路をユーザに提供することができる。

【0002】

例えば、そのような「ワールドワイドウェブ」ネットワークでは、ユーザは、自身が「行きたい」サイトまたは目的ページのアドレスを指定することができる。ユーザは次いで、選択したサイトに入り、すると目的ページあるいはサイトのホームページがユーザの表示画面に表示される。通例、ウェブサイトおよびサーバを介して入手できる情報は、ウェブ

10

20

30

40

50

ブクライアント（例えばコンピュータ）で実行されるウェブブラウザを介してアクセスされる。例えば、ウェブユーザは、ウェブブラウザを配備し、ウェブサイトのユニフォームリソースロケータ（URL）（例えばウェブアドレスおよび/またはインターネットアドレス）をウェブブラウザのアドレスバーに入力し、キーボードのEnterキーを押すか、マウスで「go」ボタンをクリックすることにより、そのウェブサイトにアクセスすることができる。目標の画面が表示された時、その画面は、予期された情報あるいは求められる情報を提示しない場合が多く、その場合、ユーザは、アクセスする別の目標アドレスを指定しなければならない。

【0003】

ページナビゲーションに関わるプロセスは、アプリケーション間の通信に依拠して自動化されることができ、一方、他のプロセスは、当該プロセスの開始や、そのプロセスが使用する文書の承認、発生する例外的な状況の解決等を人に依拠する。いずれの場合も、ユーザと当該プロセスに関係するソフトウェアのアクティビティを記述したワークフロー（例えばページフローコントロール）と称される個別の一続きの段階を規定するのが一般的である。そのようなワークフローが定義されると、その定義を中心としてアプリケーションが構築されて、業務プロセスを支援することができる。

10

【0004】

現在、コンピュータシステムとそれに関連するソフトウェアは、企業およびその他の組織が各自のワークフローを改良できるツールを提供している。ソフトウェアツールを使用して、業務ワークフローのプロセスやスケジュールをモデル化し、非効率な点や可能な改良点を明らかにすることができる。また、プロセスが、人、部門、工場、さらには別個の企業間のデータ交換を伴う場合には、コンピュータシステムとネットワークが使用されてそのような交換を実施することができる。そのようなシステムとソフトウェアツールは、さらに、ビジネスに関連する情報に一般的に伴う大規模な計算や他のデータまたは情報の処理を実施することができる。

20

【0005】

したがって、ワークフローの管理には、組織の業務プロセスにおける情報の流れと制御の効果的な管理が含まれ、そのような情報処理の自動化は、現在のビジネス界に多くの効率面の改良をもたらしている。さらに、そのようなワークフロー管理の自動化により、現在、企業とその他の組織は、インターネット等の世界規模のコンピュータネットワークを含むコンピュータシステムでワークフロー処理を実行することにより、さらに業績を向上させることができるようになっている。

30

【0006】

ワークフローに基づく典型的なアプリケーションは、しばしば、満たされるべき複数の条件を必要とする。そのような条件の1つは、例えば、業務規則に基づいて意思決定する能力である。そのような規則には、単純な規則（例えば信用調査の結果に基づく肯定または否定の決定など）と、より複雑な規則（例えば最初の引き受けの決定を行うために評価されなければならない、可能性としては大きなセット）が含まれる。別の要件は、当該ワークフローの外部にある他のソフトウェアおよび他のシステムとの通信である。例えば、アプリケーションの一部分から最初の要求が受け取られる可能性があり、一方、一部の側面（例えばクレジットサービスとの接触）が、他のウェブサービスあるいは技術を使用した通信を必要とする場合がある。満たされるべきさらなる条件は、ユーザとワークフローの適切な対話である。例えば、ワークフローは、通例は、ユーザインタフェース自体を表示すること、または他のソフトウェアを通じて人間と対話することが可能でなくてはならない。さらに、ワークフローの全存続期間にわたって状態を維持する能力が、満たされる必要がある別の条件である。したがって、ソフトウェアでワークフローを作成し、実行することは、固有の課題を呈する。

40

【0007】

現在、ワークフローシステムをページフロー制御と併合するための解決法が複数ある。そのような解決法には、例えばモデルビューコントローラ（MVC）、ページコントロー

50

ラ、フロントコントローラのパターンを用いて、ウェブページナビゲーションを調整する解決法が含まれる。そうした解決法では、ユーザは、ハードコードされた関係を用いて、あらかじめ定義された一連のページ間を移動することができる。それでも、すでに処理されたページをサブミットするには、関係が前もって定義されることが必要となる。

【発明の開示】

【発明が解決しようとする課題】

【0008】

したがって、任意のページセット間の関係のすべての可能な組み合わせをモデル化することは困難になる可能性がある。そのような複雑性のため、開発者と分析者は、最も一般的で、生じる可能性の高いページ間の関係、例えば「happy path」をモデル化することを余儀なくされる。そのため、開発者は、サブミットされた最後のページを常に把握することを必要とする再サブミットロジックを、開発するページの内部に構築せざるを得ない可能性がある。一般に、ページフローの定義を可能にする機構は存在せず、ページフローの定義が可能な場合、エンドユーザは、ページ間の形式上の関係が定義されることなくフローを再配置することができる。

10

【0009】

したがって、従来のシステムおよび機器に伴う上述した典型的な欠陥を克服する必要がある。

【課題を解決するための手段】

【0010】

以下に、クレームされた主題のいくつかの態様の基本的な理解を提供するために、簡略化した要約を提供する。この要約は、広範囲にわたる概要ではない。主要/不可欠な要素を明らかにするものでも、クレームされた主題の範囲を詳述するものでもない。その唯一の目的は、その後提示されるより詳細な説明の前置きとして、簡略化した形でいくつかの概念を示すことである。

20

【0011】

本発明は、ワークフローの外部にある動的状態マネージャを用いることによりワークフロー状態を動的にリセットし、また、ワークフローインスタンスとホストの対話を可能にするシステムおよび方法を提供する。そのような動的状態マネージャは、ワークフローの外側から（例えばワークフロー自体の一部として構築されずに）ワークフローの状態を管理することができる。そのため、ワークフローの一回限りの状態変化をモデル化する必要性が軽減される。したがって、イベント駆動型のワークフローが供給されることができ、一回限りのイベントが用いられて、ワークフロー自体の一部としてモデル化されていない状態にワークフローをリセットおよび操作し、また、実行時に動的に関係を作成する。ホストは、ワークフロープロバイダを呼び出し、設定された状態を（例えば特定の状態とUIとともに）渡し、次いでワークフローランタイムが選択される。動的状態マネージャは、設定状態を受け取って、ワークフローインスタンスをリセットすることによりその状態を駆動する。ワークフロー状態は、一時停止（例えばイベントを待つ）に関する通知をワークフローランタイムに送信することもでき、通知はワークフロープロバイダに戻り、その後ホストに通信される。

30

40

【0012】

関連する態様で、対話型アクティビティコンポーネントは、特定の点でワークフローを停止し（例えばワークフローランタイムをアイドルに設定する）、ワークフローを再開するためのホストからの入力を待つことができる。ホストは、ワークフローインスタンスをロードするためにワークフロープロバイダを呼び出すことができる。その後、イベントが設定されることができ、このイベントは、ユーザインタフェース（例えばワークフローのある状態に対応するウェブページ）に対応することができる。したがって、ホストは、ナビゲーション経路に関する情報を供給することができる（例えば「次ページ」の形で）。

【0013】

次いで、動的状態マネージャにより、そのワークフローのある状態の利用可能性に関して

50

確認が提供されることができる。その状態が存在しない場合は、ユーザにエラーが表示されることができる。そうでなくその状態が存在する場合は、動的状態マネージャは、ワークフロー中で実行するためにその状態をアクティブに設定することができる。その後実行されると、ワークフローは、イベントが新しい状態に遷移するのを待つことができ、ホストから遷移イベントが受領されると、ワークフローは、新しい状態に移ることができる。次いで、確認が行われて、その状態がワークフローインスタンス中で最後の状態であるかどうかを調べる。最後の状態である場合、方法論は終了する。最後の状態でない場合、方法論は、戻って別の状態を実行する。本発明は、上記に限定されず、ウェブページ以外にも、本発明は、ウェブサービス、システムサービスの実行等に適用できることを理解されたい。

10

【0014】

本発明の特定の方法によれば、初めに、アプリケーションナビゲーションコントロール（例えばページフロー制御フロー）の一部である表示（例えばページ）がブラウザ内に表示され、ユーザは、サブミットするためにそのページ内の情報を完成させる。その後、その表示に関連付けられたワークフローインスタンスがロードされ、ワークフローの状態が、その表示に関連付けられた状態に設定される。通例は、表示と状態の間に1対1の関連付けが存在する。状態が設定されると、表示によって生成されたイベントは、消費のためにワークフローインスタンスに渡される。イベントは、ワークフローインスタンスの現在の状態に渡される。状態は、そのイベントが消費されることができるかどうかを判定する。イベントが現在の状態によって消費される場合は、関連付けられたハンドラが、状態遷移が必要とされるかどうかを判定することができる。そうでなくイベントが現在の状態に消費されない場合は、そのイベントは次いで、消費のために、その状態を含んでいる状態に渡される。

20

【0015】

上述の目的およびそれに関連する目的を達成するために、下記の説明および添付図面と関連して、クレームされた主題の特定の例示的態様が本明細書に記載される。それらの態様は、主題が実施されることが可能な各種方式を示し、それら方式はすべて、クレームされた主題の範囲内にあるものとする。その他の効果および新規の特徴は、図面と併せて以下の詳細な説明を検討することから明らかになる。

【発明を実施するための最良の形態】

30

【0016】

次いで添付図面を参照して本発明の各種態様が説明されるが、すべての図面を通じて、同様の参照符号は、同様の要素または対応する要素を指す。ただし、図面とそれに関連する詳細な説明は、クレームされた主題をここに開示される特定の形態に制限するものではないことを理解されたい。その意図するところは、クレームされた主題の主旨および範囲に該当するすべての変形形態、均等形態、および代替形態を包含することである。

【0017】

本発明明細書で使用される場合、用語「コンポーネント」、「システム」、「サービス」、「マネージャ」等は、ハードウェア、ハードウェアとソフトウェアの組み合わせ、ソフトウェア、実行中のソフトウェアを問わず、コンピュータに関連するエンティティを指すものとする。例えば、コンポーネントは、これらに限定されないが、プロセッサで実行されるプロセス、プロセッサ、オブジェクト、実行ファイル、実行のスレッド、プログラム、および/またはコンピュータ等である。例示として、コンピュータで実行中のアプリケーションとコンピュータの両方がコンポーネントとなることができる。実行のプロセスおよび/またはスレッド内に1つまたは複数のコンポーネントが存在することができ、1つのコンポーネントが1台のコンピュータに局所化されても、かつ/または2つ以上のコンピュータ間に分散されてもよい。

40

【0018】

本明細書では、単語「例示的な」を例、事例、あるいは例示の役割を果たすものとして使用する。本明細書で「例示的な」と記される態様や設計は、必ずしも、他の態様や設計

50

より好ましい、あるいは有利であるとは解釈されるべきでない。

【0019】

さらに、ここに開示される主題は、標準的なプログラミング技術および/またはエンジニアリング技術を使用して、システム、方法、装置、または製造物として実装されて、ここに詳細に述べられる態様を実装するようにコンピュータまたはプロセッサに基づく機器を制御するソフトウェア、ファームウェア、ハードウェア、またはそれらの任意の組み合わせを作り出すことができる。本明細書で使用される用語「コンピュータプログラム」は、任意のコンピュータ可読機器、担体、または媒体からアクセスすることが可能なコンピュータプログラムを包含するものとする。例えば、コンピュータ可読媒体には、これらに限定しないが、磁気記憶装置（例えばハードディスク、フロッピー（登録商標）ディスク、磁気ストリップ）、光ディスク（例えばコンパクトディスク（CD）、デジタル多用途ディスク（DVD））、スマートカード、およびフラッシュメモリ機器（例えばカード、スティック）が含まれる。また、電子メールの送受信や、インターネットあるいはローカルエリアネットワーク（LAN）等のネットワークへのアクセスで使用されるものなど、コンピュータ可読の電子データを搬送するために搬送波が使用されることができると理解されたい。無論、当業者は、クレームされた主題の範囲および主旨から逸脱することなく、この構成に多くの変更が加えられてよいことを理解されよう。

10

【0020】

初めに図1を参照すると、システム101のブロック図が示され、システム101は、ワークフローの外部にある動的状態マネージャ105を用いることによりワークフロー状態102を動的にリセットし、また、ホスト120がワークフローインスタンス102と対話することを可能にする。システム101は、動的状態マネージャ105を用いて、システム101に関連付けられたワークフローの一回限りの状態変化をモデル化する必要性を軽減する。したがって、イベント駆動型のワークフローが供給されることができ、一回限りのイベントが用いられて、当該ワークフローの一部としてモデル化されていない状態にワークフローをリセットおよび操作し、また、実行時に動的に関係を作成する。

20

【0021】

そのようなワークフローは、アクティビティのマップとして定義された人間のプロセスまたはシステムプロセスをモデル化することができる。アクティビティは、ワークフロー中のステップであり、ワークフローの実行、再使用、比較の単位である。アクティビティのマップは、規則、アクション、状態、およびそれらの関係を表す。通例、ワークフローは、ワークフローエンジン/ランタイム110を介して実行され、ワークフローランタイムは、ホスト120で示すように、外部アプリケーションがいくつかの規則に従って当該ランタイムをホストすることを必要とする。ホスト120とワークフローシステム100との間のデータ交換は、ビジネス文書および他のオブジェクトに関連するか、かつ/またはそれらを記述することができ、他のオブジェクトは、ワークフローアクティビティのために用いられるか、またはそれを容易にし、またはホストアプリケーション120との間でデータを通信するために使用されるワークフローのローカル変数間のフローのために用いられるか、またはそれを容易にするために使用される。

30

【0022】

例えば、ホストアプリケーション120は、Workflow Provider ベースクラス117を通じて、ワークフローシステム100に関連付けられたワークフローロバイダ115と対話することができる。また、ホスト120は、1つまたは複数のプロセスの作成や、ワークフローの適正な実行のために必要とされる各種コンポーネント間の呼び出しの整理、分離機構のセットアップ等、いくつかの追加的で不可欠な態様を担うことができる。さらに、ホスト120は、拡張性の理由からマシン内の複数の中央演算処理装置（CPU）を利用するために、または多数のワークフローインスタンスをマシン群で実行するために、複数のプロセスを作成することができる。ホスト120はさらに、ワークフローが長い待ちをしなければならない時に適用すべきポリシーを制御し、特定のイベントをリッスンし、それらのイベントをユーザまたは管理者に伝え、各ワークフローのタ

40

50

イムアウトおよび再試行を設定し、パフォーマンスカウンタを公開し、デバッグと診断のためにログ情報を書くことができる。

【 0 0 2 3 】

ワークフローシステム 1 0 1 に関連付けられたワークフローは、特にその目的のために確立されたサービスを通じて外界と通信することができ、そのようなサービスは、当該ワークフロー内部のイベント駆動型アクティビティが中継する (hook up) イベントを発生させる。同様に、このサービスは、ワークフローがホスト 1 2 0 を呼び出し、データを送信するためのパブリックメソッドを公開する。ワークフローは、コンピュータシステムにおける実行のスケジュールの形で定義されることができる。スケジュールは、当該アクションに関連付けられた指定された並行性属性、依存性属性、およびトランザクション属性を有するアクションのセットを含むことができる。各スケジュールは、関連付けられたスケジュール状態を有することができ、スケジュール状態は、当該スケジュールの定義、スケジュール内における現在の位置、ならびに当該スケジュールに関連付けられたアクティブな、あるいは生きているデータおよびオブジェクトを含む。スケジュール内で、アクションのグループ化に基づいてトランザクションの境界が存在することができる。これに関して、トランザクションは、個々のアクション、トランザクション、またはそれらのグループを包含することができる。以下でさらに述べるように、アクションはグループ化して、順次実行されるシーケンスにされることができ、また、それらアクションが同時に実行されるタスクにされることもできる。したがって、グループ化に基づいて、スケジュール内のアクションとトランザクションについての並行性属性が解決されることが

10

20

【 0 0 2 4 】

ホスト 1 2 0 は、ワークフロープロバイダ 1 1 5 を呼び出し、設定された状態 (例えば特定の状態と UI を伴う) を渡すことができ、その後、ワークフローランタイムが選択される。動的状態マネージャ 1 0 5 は、設定状態を受け取って、ワークフローインスタンス 1 1 1 をリセットすることによりその状態を駆動する。ワークフロー状態 1 0 2 は、一時停止 (例えばイベントを待つ等) に関してワークフローランタイム 1 1 0 に通知を送ることもでき、その通知は、ワークフロープロバイダ 1 1 5 に送られ、次いでホスト 1 2 0 に通信される。

【 0 0 2 5 】

以下に示すのは、動的状態マネージャの例示的定義であり、ここでは、状態機械ワークフロー内の各状態は、あるウェブページに対応するものとする。

30

【 0 0 2 6 】

【表 1】

```

//Retrieve information from the page to send to the workflow
Data pageInfo = Page.GetFormData();
//Retrieve the name associated with the page.
String pageAlias = Page.GetCurrentPageAlias();
//Retrieves the workflow instance of the state machine workflow
StateMachineWorkflow StateMachineInstance =
WorkflowRuntime.GetWorkflow(workflowID);
//Retrieves the current state of the state machine
State currentState = StateMachineInstance.GetCurrentState();
//If the current state of the statemachine doesn't match the page
name
//then we need to reset the state, otherwise, continue and submit
the page
//to the current state.
If (currentState.Name != pageAlias)
{
    StateMachineWorkflow.SetState(pageName);
}
StateMachineWorkflow.Submit(pageInfo);
//Waits for the state transition to take place before it updates
the page
State nextState = StateMachine.GetCurrentState();
//Retrieves the mapping between the state of the workflow and the
page.
//Remember there is a one to one mapping.
String nextPage = Page.LookUpAlias(nextState.Name);
Page.DisplayPage(nextPage);

```

10

20

【 0 0 2 7 】

図 2 は、本発明の一態様による方法論 2 0 0 を説明する。ここでは各種のイベントおよび/または動作を表す一連のブロックとして例示的な方法が図示され、説明されるが、本発明は、それらブロックの図示される順序によって限定されない。例えば、一部の動作またはイベントは、本発明により、本明細書で図示される順序とは異なる順序で、かつ/または他の動作またはイベントと同時に進行してよい。また、図示されるブロック、イベント、または動作のすべてが、本発明による方法論を実装するために必要とは限らない。さらに、本発明による例示的方法および他の方法は、本明細書に図示および記載される方法との関連で、ならびに図示または記載されない他のシステムおよび装置との関連で実装されてよいことが理解されよう。2 1 0 で、ホストは、ワークフローインスタンスをロードするためにワークフロープロバイダを呼び出すことができる。例えば、ワークフローのインスタンスをロードするために、ワークフローインスタンス表現を格納した対応する永続ストアへのアクセスが提供されることができ、次いでその表現がワークフローインスタンスに変換され、ホストアプリケーションに提供される。その後、2 2 0 で、イベントが設定されることができ、イベントは、ユーザインタフェース（例えばワークフローのある状態に対応するウェブページ）に対応することができる。したがって、ホストは、ナビゲーション経路に関する情報を供給することができる（例えば「次ページ」の形で）。次いで 2 3 0 で、動的状態マネージャがそのワークフローの状態の利用可能性を確認することができる。その状態が存在しない場合は、ユーザにエラーが表示されることができ、方法論は 2 3 5 で停止する。

30

40

【 0 0 2 8 】

そうでなく、その状態が存在する場合は、動的状態マネージャは、ワークフロー中で実行するために 2 4 0 でその状態をアクティブに設定することができる。その後 2 5 0 で実行されると、ワークフローは、イベントが新しい状態 2 6 0 に遷移するのを待つことができ、ホストから遷移イベントが受け取られると、ワークフローは 2 7 0 で新しい状態に移ることができる。次いで 2 8 0 で、確認が行われて、その状態がワークフローインスタンス中で最後の状態であるかどうかを調べる。最後の状態である場合、方法論は 2 9 5 で終了

50

する。最後の状態でない場合、方法論は別の状態を実行するために動作 250 に戻る。

【0029】

図3および4は、本発明の一態様による動的状態マネージャの特定の実装例300および400を示す。従来のシステム（ユーザが、各種のページ/ステップ間の複雑なロジックのモデル化を介してページナビゲーションを定義するワークフロータイプシステムと対話しなければならない）と異なり、本発明は、関連するロジックを簡略化する、状態管理の動的な考え方を可能にする。例えば、ワークフローの状態310、410は、当該イベントについての情報がワークフローによって処理される前にリセットされることができる。

【0030】

図3および4に示すように、2つのバージョンのワークフローが存在し、すなわち、状態機械ワークフロータイプ315、415と、順次ワークフロータイプ317、417である。通例は、どのワークフローもいくつかのアクティビティを含んでおり、各アクティビティは、そのワークフローの機能の何らかの態様を行う。ワークフローは、それらアクティビティのコンテナとして機能し、アクティビティの存続期間と実行の順序を制御する手段を提供する。システムワークフローは、アクティビティを明確に定義された予測可能な方式で実行する傾向があるのに対し、人間のワークフローはそうではない。

【0031】

こうした両方の必要性を扱うために、Windows（登録商標）Workflow Foundationは、2つの組み込みワークフロータイプを提供する。すなわち、あらかじめ定義されたパターンでアクティビティを実行することが可能な順次ワークフロー317、417と、発生次第外部のイベントに対応することが可能な状態機械ワークフロー315、415である。両タイプとも同じ実行時環境に依拠し、両タイプとも同じカスタムアクティビティを使用することができる。順次方式は、システムワークフローに適し、一方、状態機械は、システムワークフローよりも大まかに定義された人間ワークフローの性質をモデル化する手段を提供する。単一のワークフローが両方のスタイルの要素を組み合わせることができ、2つを組み合わせるものを可能にする。さらに、開発者がカスタムのワークフロータイプを作成することもできる。

【0032】

ワークフローランタイム311、411は、ワークフローインスタンスを、アクティブサーバページ（ASP.netアプリケーション）の内部でワークフローランタイムとしてプログラマ的に管理する。さらに、対話型アクティビティコンポーネント330、430は、特定の時点でワークフローの実行を停止し（例えばワークフローランタイムをアイドルに設定する）、ワークフローを再開するホストからの入力を待つことができる。ホスト（例えばウェブページ）は、ワークフローを別の状態に移すイベントを送信することができる。例えば、ホストは、ワークフローを再開して次の状態にする再開イベントを発生させる（fire）ことができる。図4はさらに、状態機械450内で再開と一時停止に分割された対話型アクティビティコンポーネントを示す。

【0033】

対話型アクティビティコンポーネント330、440は、ユーザ入力を要求する各種の対話性の断絶時に、ユーザ入力とそれに関連する対話をワークフローに供給することができる。それにより、ワークフローと、そのワークフローに関連付けられたホストアプリケーションとの間で制御された/同期したデータ交換が可能になる。したがって、ワークフローの実行中の定義された時間に、ユーザコンテキストがワークフローインスタンスに関連付けられることができる。さらに、ワークフローは、一時停止中に、スイッチユーザコンポーネント332、432を介して、ワークフローのユーザに関連する情報等、ワークフローの状態に基づいてユーザインタフェース（UI）を駆動することができる。さらに、イベントハンドラ334、434は、現在の状態によって処理されないイベントを消費して、ある状態について定義された現在の制御フローが無効にされることを可能にする。したがって、ワークフローの状態がワークフローの外部から管理されることができ（例えばワークフロー自体の一部として構築されることなく）、ワークフローの一回限りの状態

10

20

30

40

50

変化をモデル化する必要性が軽減される。

【0034】

図5および6は、例示的なフローチャートを示し、ここでは、特定の点まで実行され、ワークフローを再開するためのホストからの入力待ちは対話型ワークフローが供給され、そのようなイベントは、ワークフローを別の状態に移すことができる。方法論500に示すように、ワークフロープロセスは505で実行され、プロセス中の次の動作に進む。ワークフローの実行は、510で、実行すべき動作がワークフローの終了を示すかどうかを確認することができる。終了を示す場合、ワークフローの実行は520で終了する。そうでない場合、方法論は動作530に進み、ワークフローによって実行されるべき動作が調べられて、それが対話型のアクティビティであるかどうかを確認する。対話型のアクティビティである場合、540でワークフローが一時停止されて、上記で詳細に述べたように、ワークフローの状態を動的にリセットするためにホストアプリケーションとの対話を可能にする。そうでない場合、方法論は動作505に戻り、ワークフローが現在の動作を実行し、次の動作に進む。

10

【0035】

図5に示すように、540におけるワークフローの一時停止の後、実行のフローは、550でワークフローを再開するためのホストからの入力待ち、そのようなイベントはワークフローを別の状態に移すことができる。その後、図6の660で、イベントが設定されることができ、そのイベントは、そのような状態が実際に存在するものと想定して、ワークフローのある状態に対応するユーザインタフェース（例えばウェブページ）に対応することができる。したがって、ホストは、670で例えば「次ページ」の形でナビゲーション経路に関する情報を供給することができる。そのため、動的状態マネージャは、680で、ワークフロー中で実行するためにその状態をアクティブに設定することができる。その後、状態が実行されると、ワークフローは、イベントが新しい状態に遷移するのを待つことができ、ホストからの遷移イベントを受信すると、ワークフローは690でその新しい状態に移ることができる。したがって、ワークフローの状態がワークフローの外部から（例えばワークフロー自体の一部として構築されずに）管理されることができ、ワークフローの一回限りの状態変化をモデル化する必要性が軽減される。

20

【0036】

図7および8は、イベント駆動型のワークフローの一例を示し、ここでは、一回限りのイベントが用いられて、ワークフロー自体の一部としてモデル化されている状態にワークフローをリセットおよび操作して、実行時に動的に関係を作成する。ウェブページアプリケーションは、ワークフロープロバイダを呼び出し、設定された状態（例えば特定の状態およびUIとともに）を渡すことができ、次いでワークフローランタイムが選択される。動的状態マネージャは、設定状態を受け取って、ワークフローインスタンスをリセットすることにより、その状態を駆動することができる。最初に、ページフロー制御フローの一部であるページがブラウザに表示され、ユーザはそのページをサブミットするためにページ内の情報を完成させる。その後、そのページに関連付けられたワークフローインスタンスがロードされ、ワークフローの状態が、そのページに関連付けられた状態に設定される。通例、ページと状態との間には1対1の関連付けがある。状態が設定されると、ページによって生成されるイベントが消費のためにワークフローインスタンスに渡される。イベントは、ワークフローインスタンスの現在の状態に渡される。状態は、その状態がそのイベントを消費することができるかどうかを判定する。イベントが現在の状態によって消費される場合は、関連付けられたハンドラが、状態遷移が必要であるかどうかを判定することができる。そうでなく、イベントが現在の状態に消費されない場合は、そのようなイベントは、消費のために、その状態を含んでいる状態に渡される。

30

40

【0037】

例えば、選択された機能を備える電話機を購入するシナリオでは、開発者は、ユーザがウェブページを使用して情報を入力することを要求するワークフローアプリケーションを開発する場合がある。ウェブページは、あらかじめ定義されており、状態機械ワークフロ

50

一中の状態に対応付けられる。開発者の目標は、状態機械ワークフローを使用してアプリケーションの制御フローをモデル化することである。ページ内で定義されたイベントは、ある状態における入力としてモデル化されることができる。イベントのタイプに基づいて、状態機械は、完了するまで状態を変え、実行することができる。状態機械における状態の変化は、ページフローアプリケーションの制御フローを定義する。

【0038】

イベントがページからワークフローにサブミットされる前に、状態機械ワークフローの現在の状態が、表示されているページに関連付けられた状態に位置づけられることができる。それにより、ユーザがhappy pathにあり、そのページにいたことが予想されたかのように、イベントが操作されることが可能になる。このように、ワークフローの最後の設定状態とワークフローの新たに選択された状態との間に動的な関係が（再配置を介して）作成されることができる。携帯電話を購入する際のステップを図7のチャート700に示す。ユーザが状態機械ワークフローの現在の状態に対応するページを操作している場合は、通例はその状態を変える、または再配置する必要はないことを理解されたい。一般に、状態は、ページと現在の状態が一致しない場合にのみリセットされる。

10

【0039】

次いで図8を参照すると、順序800が示され、ここではエンドユーザがステップ1、2、3、4、5をサブミットした後に購入すべきプランのタイプを再考することができる。本発明では、エンドユーザは、ブラウザのBackボタンを押して、ステップ2である820に位置を変え、プリペイドのプランを選択し、その情報をサブミットし、ステップ5である840に移ることができる。ステップ5と2の関係は、事前に定義されておらず、そのため一回限りの関係として存在できることを理解されたい。また、キャンセルイベント860は、メインの状態機械ワークフローについてのイベントハンドラとしてモデル化されることができる。したがって、本発明は、関係が実行時に動的に作成されることができるフレームワークを提供する。それにより、解析者と開発者は、ページ間の最も一般的な実行パスだけをモデル化すれば済み、ワークフローモデルに対して定義された制御フローが、エンドユーザの要求に合わせて順応することが可能になる。

20

【0040】

図9を参照して、本発明の各種態様を実装するための例示的環境910を説明する。例示的環境910は、コンピュータ912を含む。コンピュータ912は、処理装置914、システムメモリ916、およびシステムバス918を含む。システムバス918は、これらに限定されないが、システムメモリ916を含むシステムコンポーネントを処理装置914に結合する。処理装置914は、各種の利用可能なプロセッサのいずれでもよい。デュアルマイクロプロセッサおよび他のマルチプロセッサアーキテクチャも、処理装置914として用いられることができる。

30

【0041】

システムバス918は、これらに限定しないが、11ビットバス、ISA (Industrial Standard Architecture)、MSA (Micro-Channel Architecture)、EISA (Extended ISA)、IDE (Intelligent Drive Electronics)、VLB (VESAローカルバス)、PCI (Peripheral Component Interconnect)、USB (Universal Serial Bus)、AGP (Advanced Graphics Port)、PCMCIA (Personal Computer Memory Card International Association バス)、およびSCSI (Small Computer Systems Interface) を含む利用可能な各種のバスアーキテクチャを使用した、メモリバスまたはメモリコントローラ、ペリフェラルバスまたは外部バス、および/またはローカルバスを含む数種のバス構造のいずれでもよい。

40

【0042】

システムメモリ916は、揮発性メモリ920と不揮発性メモリ922を含む。起動時

50

などにコンピュータ912内の要素間に情報を転送する基本ルーチンを含んだ基本入出力システム(BIOS)は、不揮発性メモリ922に格納される。限定ではなく例示として、不揮発性メモリ922には、ROM(読み取り専用メモリ)、PROM(プログラム可能ROM)、EPROM(電氣的にプログラム可能なROM)、EEPROM(電氣的に消去可能なROM)、またはフラッシュメモリが含まれる。揮発性メモリ920には、外部キャッシュメモリとして機能するランダムアクセスメモリ(RAM)が含まれる。限定ではなく例示として、RAMは、SRAM(シンクロナスRAM)、DRAM(ダイナミックRAM)、SDRAM(シンクロナスDRAM)、DDR SDRAM(ダブルデータレートSDRAM)、ESDRAM(エンハンスドSDRAM)、SLDRAM(SynchLink DRAM)、およびDRAM(ダイレクトRambus RAM)等の多数の形態で入手することができる。

10

【0043】

コンピュータ912は、取り外し可能/取り外し不能、揮発性/不揮発性のコンピュータ記憶媒体も含む。例えば、図9にはディスク記憶924を示す。ディスク記憶924には、これらに限定しないが、磁気ディスクドライブ、フロッピー(登録商標)ディスクドライブ、テープドライブ、Jazドライブ、Zipドライブ、LS-100ドライブ、フラッシュメモリカード、メモリスティック等の機器が含まれる。また、ディスク記憶924は、記憶媒体を単独で、またはこれらに限定しないがCD-ROM(コンパクトディスクROMデバイス)、CD-Rドライブ(CD記録可能ドライブ)、CD-RWドライブ(CD書き換え可能ドライブ)、またはDVD-ROM(デジタル多用途ディスクROMドライブ)を含む他の記憶媒体と組み合わせて含むことができる。システムバス918へのディスク記憶装置924の接続を容易にするために、通例はインタフェース926などの取り外し可能または取り外し不能インタフェースが使用される。

20

【0044】

図9には、ユーザと、適切な動作環境910内に記載される基本的なコンピュータ資源との間の仲介役として機能するソフトウェアを記載していることを理解されたい。そのようなソフトウェアには、オペレーティングシステム928が含まれる。オペレーティングシステム928は、ディスク記憶924に記憶されることができ、コンピュータシステム912の資源を制御し、割り当てるように働く。システムアプリケーション930は、システムメモリ916またはディスク記憶924のいずれかに記憶されたプログラムモジュール932とプログラムデータ934を通じて、オペレーティングシステム928による資源の管理を利用する。本明細書に記載される各種コンポーネントは、各種のオペレーティングシステムまたはオペレーティングシステムの組み合わせで実装されることが可能であることを理解されたい。

30

【0045】

ユーザは、入力装置936を通じてコンピュータ912にコマンドまたは情報を入力する。入力装置936には、これらに限定しないが、マウス、トラックボール、スタイラス、タッチパッド、キーボード等のポインティングデバイス、マイクロフォン、ジョイスティック、ゲームパッド、衛星受信アンテナ、スキャナ、TVチューナカード、デジタルカメラ、デジタルビデオカメラ、ウェブカメラ等が含まれる。上記およびその他の入力装置は、インタフェースポート938を介してシステムバス918を通じて処理装置914に接続する。インタフェースポート938には、例えばシリアルポート、パラレルポート、ゲームポート、ユニバーサルシリアルバス(USB)が含まれる。出力装置940は、入力装置936と同タイプのポートのいくつかを使用する。したがって、例えば、USBポートを使用してコンピュータ912に入力を提供し、コンピュータ912から出力装置940に情報を出力することができる。出力アダプタ942は、出力装置940の中にはモニタ、スピーカ、プリンタなど、特殊なアダプタを必要とするものがあることを説明するために提供されている。限定ではなく例示として、出力アダプタ942には、出力装置940とシステムバス918間の接続手段を提供するビデオカードおよび音声カードが含まれる。リモートコンピュータ944など、他の装置および/または装置のシステムが、入

40

50

力と出力両方の機能を提供することに留意されたい。

【0046】

コンピュータ912は、リモートコンピュータ944などの1つまたは複数のリモートコンピュータと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944は、パーソナルコンピュータ、サーバ、ルータ、ネットワークPC、ネットワークノード等であり、通例は、コンピュータ912に関連して記載される要素の多くまたはすべてを備える。図を簡潔にするために、リモートコンピュータ944にはメモリ記憶装置946のみを図示する。リモートコンピュータ944は、ネットワークインタフェース948を通じて論理的にコンピュータ912に接続され、次いで通信接続950を介して物理的に接続される。ネットワークインタフェース948は、ローカルエリアネットワーク(LAN)やワイドエリアネットワーク(WAN)などの通信ネットワークを包含する。LAN技術には、FDDI(ファイバ分散データインタフェース)、CDDI(銅線分散データインタフェース)、Ethernet(登録商標)/IEEE802.3、Token Ring/IEEE802.5等がある。WAN技術には、これらに限定されないが、2地点間リンク、ISDN(総合サービスデジタル網)などの回路交換網およびその変種、パケット交換網、およびデジタル加入者回線(DSL)がある。

10

【0047】

通信接続950は、ネットワークインタフェース948をバス918に接続するために用いられるハードウェア/ソフトウェアを指す。通信接続950は、図示を分かりやすくするためにコンピュータ912の内部に示しているが、コンピュータ912の外部にあってもよい。ネットワークインタフェース948に接続するために必要なハードウェア/ソフトウェアには、単に例示の目的で、通常の電話グレードモデム、ケーブルモデム、DSLモデム等のモデム、ISDNアダプタ、Ethernet(登録商標)カード等の内部および外部技術が含まれる。

20

【0048】

図10は、本発明のワークフロー実装を組み込むために用いられることができるコンピューティング環境例1000の概略ブロック図である。システム1000は、1つまたは複数のクライアント1010を含む。クライアント1010は、ハードウェアおよび/またはソフトウェア(例えばスレッド、プロセス、コンピューティングデバイス)とすることができる。システム1000は、1つまたは複数のサーバ1030も含む。サーバ1030もハードウェアおよび/またはソフトウェア(例えばスレッド、プロセス、コンピューティングデバイス)とすることができる。サーバ1030は、例えば、本明細書に記載されるコンポーネントを用いることによってトランザクションを行うスレッドを収容することができる。クライアント1010とサーバ1030間で可能な通信の1つは、2つ以上のコンピュータプロセス間で送信されるように適合されたデータパケットの形をとることが可能である。システム1000は、クライアント1010とサーバ1030間の通信を容易にするために用いられることができる通信フレームワーク1050を含む。クライアント1010は、クライアント1010にとってローカルな情報を格納するために用いられることができる1つまたは複数のクライアントデータストア1060に動作的に接続されている。同様に、サーバ1030は、サーバ1030にとってローカルな情報を格納するために用いられることができる1つまたは複数のサーバデータストア1040に動作的に接続されている。

30

40

【0049】

上記の内容は各種の例示的態様を含む。言うまでもなく、それらの態様を説明するために構成要素や方法論のあらゆる着想可能な組み合わせを記載することは可能でなく、当業者は、多数のさらなる組み合わせおよび置き換えが可能であることを認識されよう。したがって、本明細書に記載される態様は、頭記の特許請求の範囲の主旨および範囲に該当するそのような改変、変更、および変形をすべて包含するものとする。さらに、用語「~を含む」が詳細な説明または特許請求の範囲で使用される限りでは、この用語は、「~を備える」が請求項中で移行語として用いられる場合の解釈と同様に包含的な意味とする。

50

【図面の簡単な説明】

【0050】

【図1】ワークフロー状態をリセットする動的状態マネージャを有するワークフローの例示的システム図である。

【図2】本発明の例示的態様によるワークフロー状態をリセットする例示的方法論を説明する図である。

【図3】本発明の態様によるASP（アクティブサーバページ）ウェブ開発技術の特定のシステム図である。

【図4】本発明のさらなる態様によるASPウェブ開発技術のさらなるシステム図である。

【図5】ワークフローを別の状態に移すためのホストからの入力を待つ対話型ワークフローが供給される例示的方法論を説明する図である。

【図6】ワークフローの状態をリセットするさらなる方法論を説明する図である。

【図7】本発明の態様によるイベント駆動型のワークフローに関連するステップの図である。

【図8】ワークフロー自体の一部としてモデル化されていない状態にワークフローをリセットし、操作するために用いられる一連の動作を説明する図である。

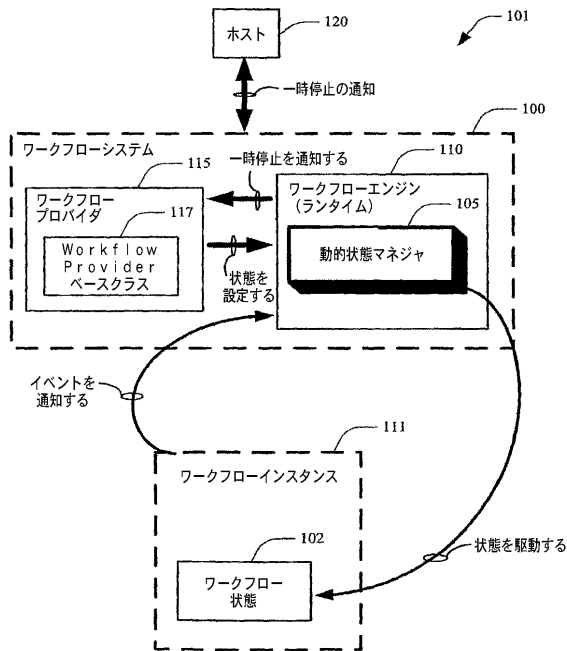
【図9】本発明の各種態様を実装するための例示的環境を説明する図である。

【図10】本発明のルーティング永続サービスと併せてワークフローを実装するために用いられることができる追加的なコンピューティング環境の概略ブロック図である。

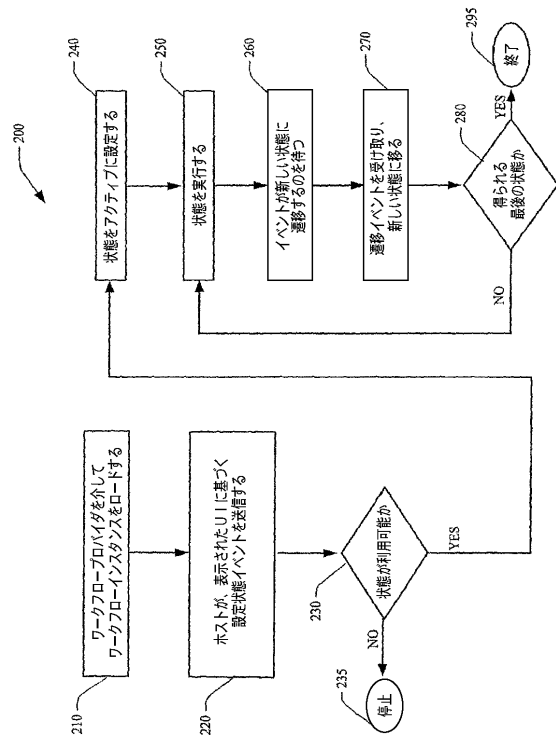
10

20

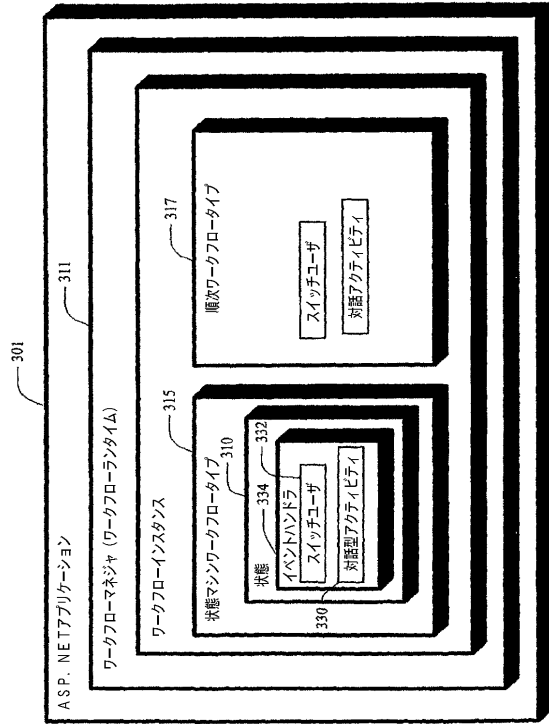
【図1】



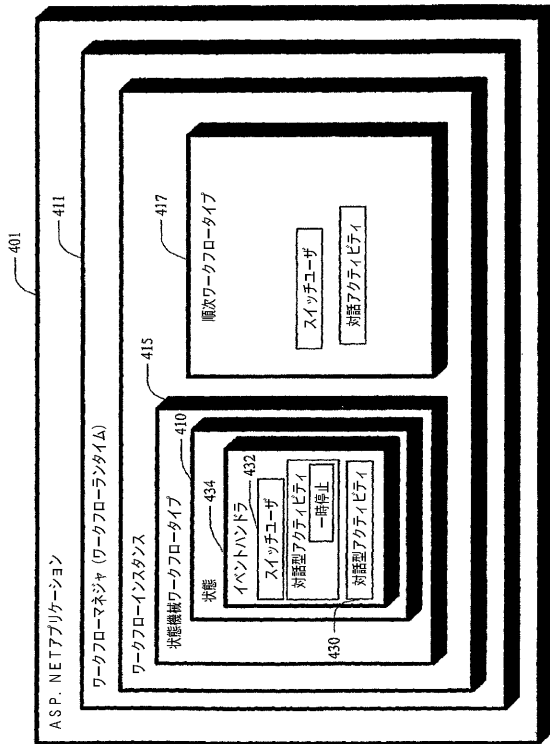
【図2】



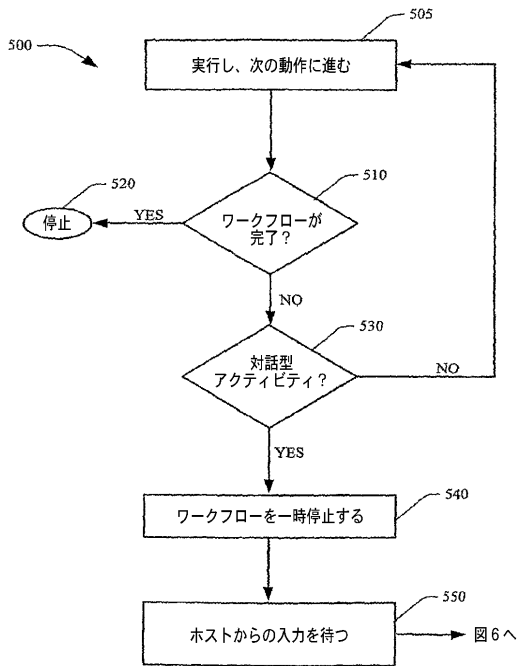
【 図 3 】



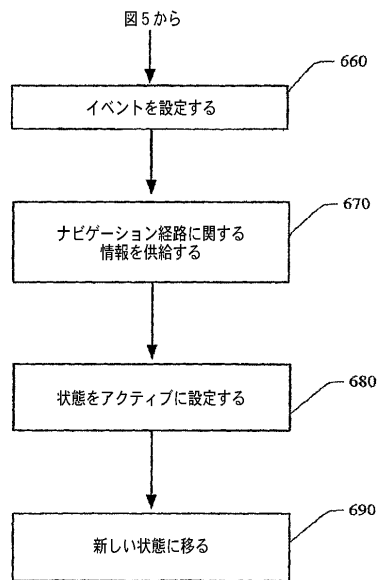
【 図 4 】



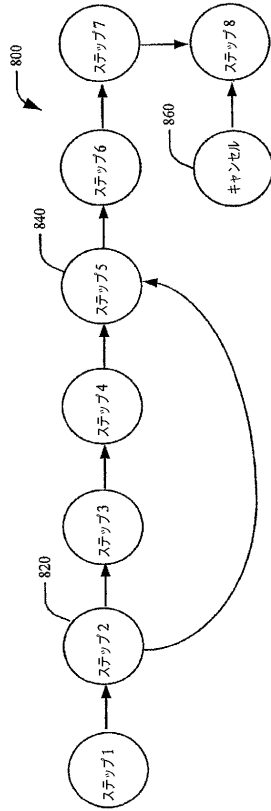
【 図 5 】



【 図 6 】



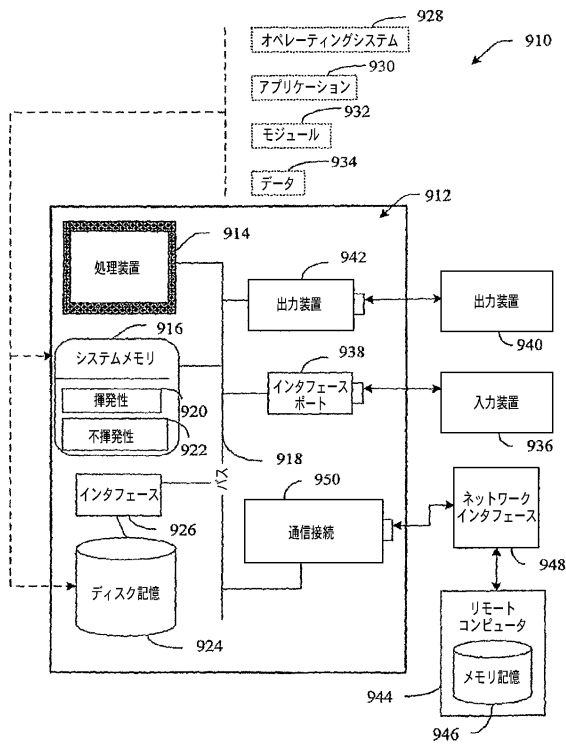
【図7】



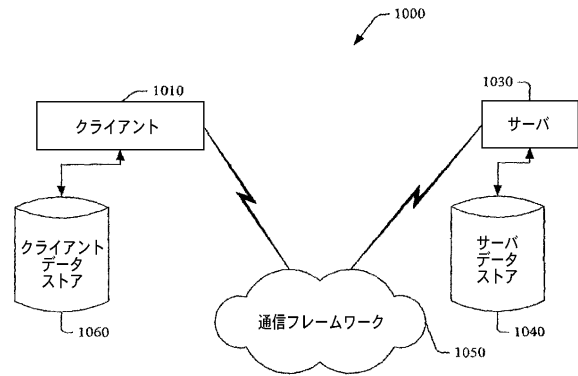
【図8】

ステップ	説明	次のステップ
1	ZIPコードを入力する	2
2	プランを選択する	3または5 (プリペイドプランの場合)
3	電話番号を選択する	4
4	付属品を選択する	5
5	支払い情報を入力する	6
6	配送情報を入力する	7
7	注文を確認する	8
8	領収書をプリントする	

【図9】



【図10】



フロントページの続き

- (74)代理人 100091063
弁理士 田中 英夫
- (74)代理人 100096068
弁理士 大塚 住江
- (74)代理人 100153028
弁理士 上田 忠
- (74)代理人 100120112
弁理士 中西 基晴
- (74)代理人 100113974
弁理士 田中 拓人
- (74)代理人 100147991
弁理士 鳥居 健一
- (72)発明者 イスラエル ヒレリオ
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション インターナショナル パテント内
- (72)発明者 アンドレス サナブリア
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション インターナショナル パテント内
- (72)発明者 コンスタンティン ミハイ
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション インターナショナル パテント内
- (72)発明者 ニキル コタリ
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション インターナショナル パテント内
- (72)発明者 プラビン インダーカル
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション インターナショナル パテント内
- (72)発明者 マルセロ アール・ウエムラ
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション インターナショナル パテント内
- (72)発明者 マイケル ハーダー
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション インターナショナル パテント内

審査官 宮地 匡人

- (56)参考文献 特開2002-123417(JP,A)
国際公開第2004/057483(WO,A1)
特開2001-306507(JP,A)
特開2005-107989(JP,A)
高橋 信頼, フレームワークで生産性を上げる Webアプリの“煩雑さ”は隠せる, 環境が整いノウハウも見えてきた, 日経オープンシステム, 2002年 2月15日, No.107, p.107-120

(58)調査した分野(Int.Cl., DB名)

G06Q 10/00-50/00