

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6523425号  
(P6523425)

(45) 発行日 令和1年5月29日 (2019.5.29)

(24) 登録日 令和1年5月10日 (2019.5.10)

(51) Int. Cl. F I  
G O 6 F 9/54 (2006.01) G O 6 F 9/54 C

請求項の数 10 (全 30 頁)

(21) 出願番号	特願2017-500935 (P2017-500935)	(73) 特許権者	314015767
(86) (22) 出願日	平成27年7月6日 (2015.7.6)		マイクロソフト テクノロジー ライセン
(65) 公表番号	特表2017-524184 (P2017-524184A)		シング, エルエルシー
(43) 公表日	平成29年8月24日 (2017.8.24)		アメリカ合衆国 ワシントン州 9805
(86) 国際出願番号	PCT/US2015/039170		2 レッドモンド ワン マイクロソフト
(87) 国際公開番号	W02016/007399		ウェイ
(87) 国際公開日	平成28年1月14日 (2016.1.14)	(74) 代理人	100140109
審査請求日	平成30年7月3日 (2018.7.3)		弁理士 小野 新次郎
(31) 優先権主張番号	14/326, 239	(74) 代理人	100118902
(32) 優先日	平成26年7月8日 (2014.7.8)		弁理士 山本 修
(33) 優先権主張国	米国 (US)	(74) 代理人	100106208
			弁理士 宮前 徹
		(74) 代理人	100120112
			弁理士 中西 基晴

最終頁に続く

(54) 【発明の名称】 仮想ストリームおよび処理エージェントを利用するストリーム・ベース・イベントの処理

(57) 【特許請求の範囲】

【請求項 1】

イベントを配信するためのコンピューター実装方法であって、  
 イベント待ち行列においてイベントの発行を受けるステップであって、前記イベントが  
 特定のタイプのイベントである、ステップと、

複数の異なる処理エージェントのうちの処理エージェントに前記イベントをマッピング  
 するステップであって、前記処理エージェントが前記特定のタイプのイベントを受けるこ  
 とになり、前記処理エージェントが前記イベントを受けることを明示的に登録申込みして  
 いない、ステップと、

前記処理エージェントが有効化されていないことを決定するステップと、

前記処理エージェントが有効化されていないと決定したことに応答して、前記処理エー  
 ジェントを有効化するステップであって、前記処理エージェントがアプリケーションに対  
 し機能を提供するために利用可能となるようにする、ステップと、

前記イベントを前記処理エージェントに配信するステップと、

前記処理エージェントから登録申込みを受けるステップと、

前記イベント待ち行列において1つ以上の追加のイベントの1つ以上の追加の発行を受  
 けるステップであって、前記1つ以上の追加のイベントが前記特定のタイプのイベントの  
 関連し、前記1つ以上の追加のイベントが、前記処理エージェントから前記登録申込みを  
 受けた後に発生する、ステップと、

前記登録申込みに少なくとも部分的に基づいて、前記1つ以上の追加のイベントを前記

10

20

処理エージェントに配信するステップと、

1つ以上の過去のイベントが配信されることを前記登録申込みが要求していると決定するステップであって、前記登録申込みを前記処理エージェントから受ける前に前記1つ以上の過去のイベントが前記イベント待ち行列において発行されている、ステップと、

前記1つ以上の過去のイベントが配信されることを前記登録申込みが要求していると決定したことに応答して、前記1つ以上の過去のイベントを前記処理エージェントに配信するステップと、

を含む、方法。

【請求項2】

請求項1記載の方法において、前記イベント待ち行列において前記イベントの発行を受けるステップが、

前記イベントをイベント待ち行列において発行する要求を受けるステップと、

前記イベントに対応する仮想ストリームを有効化するステップと、

前記仮想ストリームを介して前記イベント待ち行列において前記イベントを発行するステップと、

を含む、方法。

【請求項3】

請求項1記載の方法において、前記1つ以上のイベントに対する前記登録申込みが仮想ストリームを定め、前記登録申込みが、前記1つ以上の追加のイベントに対応する前記仮想ストリームに対する登録申込みである、方法。

【請求項4】

請求項3記載の方法において、前記処理エージェントを有効化するステップが、システム・リソースを前記処理エージェントに割り当てるステップを含み、前記仮想ストリームに対する前記登録申込みが前記仮想ストリームを有効化し、システム・リソースを前記仮想ストリームに割り当てさせる、方法。

【請求項5】

請求項4記載の方法であって、更に、

一定の時間期間にわたって前記仮想ストリームにおいてイベントが受けられないとき、前記仮想ストリームに割り当てられたシステム・リソースを取り戻すことによって、前記仮想ストリームを無効化するステップと、

前記処理エージェントが所与の時間期間にわたってアイドルになったとき、前記処理エージェントに割り当てられたシステム・リソースを取り戻すことによって、前記処理エージェントを無効化するステップと、

を含む、方法。

【請求項6】

請求項1記載の方法において、前記処理エージェントの構成が、プログラミングによって得られるか、または前記処理エージェントの宣言的定義から決定される、方法。

【請求項7】

請求項1記載の方法において、前記処理エージェントが有効化されているか否かに関係なく、前記イベントを配信するために前記処理エージェントがアドレス可能である、方法

【請求項8】

請求項1記載の方法において、前記特定のタイプのイベントが、ビデオ・ゲームの特性に関連付けられ、前記1つ以上の追加のイベント及び前記1つ以上の過去のイベントを含む前記イベント待ち行列において発行された前記特定のタイプのイベントの全てのイベントを前記登録申込みが要求する、方法。

【請求項9】

ランタイムを実行する1つ以上の計算デバイスを含むシステムであって、前記ランタイムが、

イベントの発行を受けて永続化するための待ち行列を管理し、

仮想ストリームにおいて特定のタイプのイベントの発行を受け、前記イベントの発行を前記待ち行列に転送し、

複数の異なる処理エージェントのうちの処理エージェントに、前記特定のタイプの前記イベントをマッピングし、前記処理エージェントが、前記特定のタイプの前記イベントを受けることを明示的に登録申込みしておらず、前記処理エージェントが有効化されているかまたは有効化されていないかには関係なく、前記特定のタイプの前記イベントを配信するために前記処理エージェントがアドレス可能であり、

前記処理エージェントが有効化されていないことを決定し、

前記処理エージェントが有効化されていないと決定したことに応答して、前記処理エージェントを有効化して、前記処理エージェントがアプリケーションに対し機能を提供するために利用可能となるようにし、

10

前記特定のタイプの前記イベントを前記処理エージェントに配信し、

前記特定のタイプの前記イベントが前記処理エージェントに配信された後に、前記処理エージェントから前記仮想ストリームに対する明示的登録申込みを受け、前記明示的登録申込みが、前記待ち行列において発行された前記特定のタイプのすべてのイベントを要求し、

前記明示的登録申込みを受けた後に1つ以上の追加のイベントの1つ以上の追加の発行を前記仮想ストリームにおいて受け、前記1つ以上の追加のイベントの前記1つ以上の追加の発行を前記待ち行列に転送し、

前記明示的登録申込みに少なくとも部分的に基づいて、前記1つ以上の追加のイベントを前記処理エージェントに配信し、

20

前記明示的登録申込みが受けられる前に、前記待ち行列において前記特定のタイプの過去のイベントが発行されていることを決定し、

前記待ち行列において発行された前記特定のタイプの前記過去のイベントを引き出し、前記特定のタイプの前記過去のイベントを前記処理エージェントに配信する、  
ように構成される、システム。

#### 【請求項10】

請求項9記載のシステムにおいて、前記処理エージェントを有効化することが、システム・リソースを前記処理エージェントに割り当てることを含み、前記仮想ストリームに対する前記明示的登録申込みが、前記仮想ストリームにシステム・リソースを割り当てさせる、システム。

30

#### 【発明の詳細な説明】

#### 【背景技術】

#### 【0001】

[0001] ストリーム処理システムでは、ストリーム処理ユニットが、処理するイベントを受けるために、他の処理ユニットと通信する場合がある。これは、処理ユニット同士の直接通信によって、または出版購読型モデル(publication-subscription model)を介して行うことができる。出版購読型モデルでは、処理ユニットが一定のイベントに登録申込み(subscribe)し、これらが発行(publish)されたときにイベントを受ける。

#### 【0002】

40

[0002] ストリーム処理システムの規模が大きくなるに連れ、例えば、数百万台の処理ユニットが関与する場合、処理ユニット間において直接通信技法を利用することは非実用的となる場合もある。出版購読型モデルは実現可能に思われるかもしれないが、従前からの出版購読型モデルは、処理ユニットがイベントに関する何らかの情報を受ける前に、イベント登録申込みを明示的に提出しなければならない。これは、場合によっては、処理ユニットが、一定のイベントが発生する前に登録申込みすべきイベントを推定することは不可能である場合もあるので、問題となるおそれがある。例えば、ゲーム・アプリケーションのマップ上で発生するイベントを処理するように構成された処理ユニットは、そのマップ上でゲームが開始されるまでは、どのゲームに登録申込みすべきかわからない。このような状況では、処理ユニットに関心があるイベントを受けるためには、処理ユニットは全

50

ての可能なイベントに登録申込みしなければならず、この場合も大規模なストリーム処理システムでは、非実用的および非効率的になる。

【 0 0 0 3 】

[0003] 本明細書において本開示を紹介するのは、これらおよび他の検討事項(considerations)に関してである。

【 発明の概要 】

【 0 0 0 4 】

[0004] 本明細書では、仮想処理エージェントを利用するイベント配信およびストリーム処理のための技術について記載する。1つ以上の仮想処理エージェントを作成し、ストリーム・イベントを受けて処理するように構成することができる。ストリーム処理システムにおいて種々のエンティティからイベント発行を受けるためには、待ち行列のようなメカニズムを利用するおよび/または構成するのでもよい。また、発行されたイベントは、必要になったときに後に引き出すおよび配信するために、永続化(persist)するまたは格納することもできる。イベント発行を受けると、ランタイム・システムがこの発行されたイベントを受け関心があるかもしれないが、明示的にこの発行されたイベントに登録申込みしていない1つ以上の仮想処理エージェントを識別することができる。次いで、発行されたイベントのイベント情報を、識別された仮想処理エージェントに配信することができる。イベントまたはイベント情報の実際の配信に先立って、ランタイム・システムは、更に、仮想処理エージェントが有効化(activate)されているか否か判定することもでき、処理エージェントが有効化されていない場合、ランタイム・システムは処理エージェントを有効化することができる。次いで、イベントまたはその中に含まれるイベント情報を、有効化された処理エージェントに配信することができる。

【 0 0 0 5 】

[0005] 受けたイベントまたはイベント情報に基づいて、仮想処理エージェントは、待ち行列から更に多くのイベントを受けるために、明示的に登録申込みを提出することを決定することができる。次いで、明示的な登録申込みは、登録申込みされたイベントを処理エージェントに配信するきっかけとなることができ、この明示的な登録申込みが提出される前に待ち行列において発行された過去のイベントを含むこともできる。本明細書において説明する技法を利用することによって、仮想処理エージェントは、待ち行列においてイベントが発行される前には処理エージェントが他の方法では知ることができないイベントに登録申込みし、受け、処理することが可能になる。仮想処理エージェントを利用してイベントに登録申込みし、受け、処理することによって、計算リソースのスケラビリティ、信頼性、および柔軟性を向上させることができる。

【 0 0 0 6 】

[0006] 更に、以上で説明した主題は、コンピューター制御型装置、コンピューター・プロセス、計算システムとして、またはコンピューター読み取り可能記憶媒体のような製品としても実現できることは認められてしかるべきである。これらおよび種々の他の特徴は、以下の詳細な説明を読み、添付図面を検討することによって明白になるであろう。

【 0 0 0 7 】

[0007] この摘要は、詳細な説明において以下で更に説明する概念から選択したものを、簡略化した形態で紹介するために設けられている。この摘要は、特許請求する主題の主要な特徴や必須の特徴を特定することを意図するのではなく、特許請求する主題の範囲を限定するために使用されることを意図するのでもない。更に、特許請求する主題は、本開示の任意の部分に記される任意の欠点を解決する実現例にも、全ての欠点を解決する実現例にも限定されない。

【 図面の簡単な説明 】

【 0 0 0 8 】

【 図 1 】 図 1 は、仮想処理エージェントを利用するストリーム処理のために本明細書において開示される1つのメカニズムの全体的な説明を示すコンピューター・システム図である。

【図 2】図 2 は、待ち行列および仮想ストリームの形態(aspect)を示すブロック図である。

【図 3】図 3 は、イベントを仮想処理エージェントに配信するための 1 つの例示的なプロセスの更に他の形態を示す流れ図である。

【図 4】図 4 は、仮想処理エージェントによるイベント登録申込みおよび処理のための 1 つの例示的なプロセスの形態を示す流れ図である。

【図 5 A】図 5 A は、異なる時点において待ち行列において発行されたイベントに応答する仮想処理エージェント例による動作を示す。

【図 5 B】図 5 B は、異なる時点において待ち行列において発行されたイベントに応答する仮想処理エージェント例による動作を示す。

【図 5 C】図 5 C は、異なる時点において待ち行列において発行されたイベントに応答する仮想処理エージェント例による動作を示す。

【図 5 D】図 5 D は、異なる時点において待ち行列において発行されたイベントに応答する仮想処理エージェント例による動作を示す。

【図 6】図 6 は、本明細書において提示した技術を実現することができる計算システムの例示的なコンピューター・ハードウェア・アーキテクチャを示すコンピューター・アーキテクチャ図である。

【発明を実施するための形態】

【0009】

[0014] 本明細書では、仮想処理エージェントを利用するイベント配信およびストリーム処理のための技術について記載する。本明細書において開示する技術では、本明細書では「処理エージェント」または「PA」と呼ぶこともある、1 つ以上の仮想処理エージェントを、ストリーミング処理のための処理ユニットとして採用することができる。これらの仮想処理エージェントは 1 つ以上のサーバーにおいて実行することができ、処理エージェントが有効化されているかまたは有効化されていないかには関係なく、動作および相互作用のためにアドレス可能にすることができる。仮想処理エージェントの各々は、種々のイベントを発行する、受ける、および / または処理するように構成することができる。イベントは、待ち行列において発行することができ、更に、後に引き出すために、待ち行列内に永続化するまたは格納することもできる。

【0010】

[0015] 一旦イベントの発行が待ち行列において受けられたなら、発行されたイベントを受ける 1 つ以上の処理エージェントを識別することができる。識別される処理エージェントは、発行されたイベントを受けることを明示的に登録申込みしていない処理エージェントを含むことができ、ここでは「暗示サブスクライバー処理エージェント」(implicit subscriber processing agent)または「暗示サブスクライバー」(implicit subscriber)と呼ぶ。暗示サブスクライバーの識別は、仮想処理エージェントの設定または構成に基づいて行うことができ、仮想処理エージェントによって実行されるアクション、仮想処理エージェントの入力および出力、仮想処理エージェントのステータス、ならびに仮想処理エージェントの可能な他の構成を含むが、これらに限定されるのではない。仮想処理エージェントの構成は、処理エージェントの宣言的定義(declarative definition)から導くことができ、またはプログラムによって得ることができる。

【0011】

[0016] 次いで、発行イベント、または発行イベント内に含まれるまたはこれに関連する情報を、識別された暗示サブスクライバー処理エージェントに配信することができる。この情報をここでは「イベント情報」と呼ぶ。実際の配信に先立って、仮想処理エージェントをサポートおよび管理するランタイム・システムは、暗示サブスクライバー処理エージェントが有効化されているか否か判定することができ、処理エージェントが有効化されていない場合、処理エージェントを有効化することができる。次いで、イベントまたはイベント情報を、有効化された処理エージェントに配信することができ、ここでイベントまたはイベント情報を更に処理することができる。同様に、イベント発行が待ち行列に受け

10

20

30

40

50

入れられたとき、発行されたイベントを受けることを明示的に登録申込みした1つ以上の処理エージェントも識別し、必要であれば有効化することができる。次いで、発行されたイベントを明示サブスクライバー処理エージェントに処理のために配信することができる。

【0012】

[0017] 受けたイベントまたはイベント情報に基づいて、暗示サブスクライバー処理エージェントは、処理のために更に他のイベントが必要とされると判断することもあり、これらのイベントが待ち行列において発行されるときに受けるために明示登録申込みを提出することができる。処理エージェントの構成に依存して、明示登録申込みにおいて指定されるイベントは、既に発生し待ち行列において発行されている過去のイベントも含むことができる。このようなシナリオでは、待ち行列において発行されるイベントが永続化されるまたは格納されることもあるので、待ち行列は、これら登録申込みされた過去のイベントを引き出し、処理エージェントに配信することができる。更に、他のイベントが待ち行列において発行されるに連れて、待ち行列は登録申込みイベントを処理エージェントに配信することができる。

10

【0013】

[0018] 更に他の形態によれば、待ち行列内にあるイベントを1つ以上の仮想ストリームにマッピングまたは抽象化することができ、仮想ストリームの各々が、待ち行列において発行されているイベントの部分集合を含むこともできる。処理エージェントおよび/または他のエンティティによって実行されるイベント発行およびイベント登録申込みは、待ち行列ではなく、仮想ストリームに導くことができる。仮想ストリームの各々は、その中で発行されるイベントによって、および/または処理エージェントが受けることに関心があるイベントのタイプによって定めることができる。したがって、仮想ストリームは動的であってもよく、仮想ストリームの存在は、発行されるイベントおよび/または配信されるイベントに依存してもよい。本明細書において提示する技術のこれらおよび他の形態に関する追加の詳細は、図1～図6に関して以下に示す。

20

【0014】

[0019] 本明細書において説明する主題は、コンピューター・システムにおけるオペレーティング・システムおよびアプリケーション・プログラムの実行と併せて実行するプログラム・モジュールという一般的なコンテキストで紹介するが、他のタイプのプログラム・モジュールとの組み合わせで他の実現例も実行できることは、当業者には認められよう。一般に、プログラム・モジュールは、ルーチン、プログラム、コンポーネント、データ構造、および他のタイプの構造を含み、特定のタスクを実行するか、または特定の抽象データ型を実装する。更に、本明細書において説明する主題は、他のコンピューター・システム構成でも実施できることは、当業者には認められよう。他のコンピューター・システム構成には、ハンドヘルド・デバイス、マルチプロセッサ・システム、マイクロプロセッサ・ベースまたはプログラマブル消費者用電子機器、ミニコンピューター、メインフレーム・コンピューター等が含まれる。

30

【0015】

[0020] 以下の詳細な説明では、本明細書の一部を形成する添付図面を参照する。添付図面は、例示、具体的な形態、または例として示される。これより図面を参照して、イベントを登録申込みし、受け、そして処理する計算システムおよび方法について説明する。図面において、同様の番号は様々な図にわたって同様のエレメントを表すものとする。

40

【0016】

[0021] これより図1に移り、本明細書において紹介する形態によるコンピューター・システムおよびその中に含まれる様々なコンポーネントに関する詳細を示す。具体的には、図1は、仮想処理エージェントを利用するイベント配信およびストリーム処理のためのストリーム処理システム100の形態を示す。図1に示すように、ストリーム処理システム100は、ランタイム・システム108を含むことができる。ランタイム・システム108は、ストリーム処理システム100における種々のエンティティ間の通信をサポート

50

しそしてやり易くする。種々のエンティティには、仮想処理エージェント 102A ~ 102N (ここでは個々に仮想処理エージェント 102 と呼ぶことも、または纏めて仮想処理エージェント 102 と呼ぶこともある)、待ち行列システム 104、クライアント計算デバイス 116、およびサーバー計算デバイス 118 が含まれるが、これらに限定されるのではない。

#### 【0017】

[0022] ランタイム・システム 108 は、一群の分散型サーバー (図 1 には示されていない) を含むことができる。分散型サーバーは、「仮想化分散型コンポーネント」(ここでは「仮想コンポーネント」と呼ぶこともある) を纏めて提供しサポートするランタイム・ベース・レイヤー 122 を実行するように構成することができる。本明細書において使用する場合、「コンポーネント」とは、ソフトウェア・オブジェクトのようなソフトウェア・コンポーネント、または分散型オブジェクト、エージェント、アクター(actor) 等のような、他の種類の個々にアドレス可能な分離エンティティ(isolated entity) を指す。ランタイム・システム 108 によってサポートされる仮想化分散型コンポーネントは、そのメモリー内インスタンス化の内任意のものの寿命(lifetime)とも独立した、したがってランタイム・システム 108 における任意の特定のサーバーの寿命とも独立した存在(existence)を有する。仮想コンポーネントのメモリー内インスタンスがない場合、コンポーネントに送られるメッセージが、ランタイム・ベース・レイヤー 122 によって利用可能なサーバーにおいて新たなインスタンスが自動的に作成させられる。利用可能なサーバーとは、ランタイム・システム 108 における任意の利用可能なサーバーでよい。したがって、仮想コンポーネントは、仮想コンポーネントがシステムにおいて有効化されているか否かには関係なく、相互作用および/または動作のために、仮想コンポーネントの識別によって常にアドレス可能である。

#### 【0018】

[0023] また、ランタイム・ベース・レイヤー 122 は、仮想コンポーネントの有効化および無効化を管理するように構成することもできる。更に、ランタイム・ベース・レイヤー 122 は、仮想コンポーネントを使用するプログラムには透過的に、仮想コンポーネントへの通信を管理するように構成することもできる。これを達成するために、ランタイム・ベース・レイヤー 122 は、仮想コンポーネントの任意の有効化されたインスタンスのために 1 つ以上のサーバー位置を示す情報を維持し、1 つ以上の通信を有効化された仮想コンポーネント・インスタンスに送り、または無効化状態にある仮想コンポーネントへの通信を、この仮想コンポーネントのインスタンスが、それへの通信を受信するために、有効化(activation)に有効化されるまで保存することができる。このように、ランタイム・ベース・レイヤー 122 は、開発者に、システムにおいて任意の可能な仮想コンポーネントを、これがあたかも有効化されているかのように、即ち、メモリー内に存在するかのように、呼び出すことを可能にする仮想「コンポーネント空間」を与えることができる。

#### 【0019】

[0024] 仮想コンポーネントが有効化されているサーバーがクラッシュした場合、ランタイム・システム 108 における他のサーバーにおいて仮想コンポーネントの他の有効化を再有効化する(reactivating another activation)することによって、この仮想コンポーネントを復元することができる。したがって、仮想コンポーネントを監視する必要がなく、および/またはランタイム・ベース・レイヤー 122 に基づいて開発されたアプリケーションによって明示的に作成し直す必要もない。

#### 【0020】

[0025] コンポーネントの仮想化は、仮想コンポーネントから、現在実行している物理インスタンス化/有効化にマッピングする間接参照(indirection)のレベルに基づく。この間接参照のレベルは、ランタイム・ベース・レイヤー 122 に、仮想コンポーネント配置および負荷均衡化、使用されていない仮想コンポの無効化、ならびにサーバー障害後における仮想コンポーネントの復元というような、他の場合ではアプリケーション開発者が取り組む必要がある多くの問題に対処する機会を与えることができる。つまり、ランタイム

ム・ベース・レイヤー 122 によって提供される仮想コンポーネントは、負荷を均衡化し透過的に障害から復元するときにランタイムの柔軟性を許容しつつ、プログラミング・モデルを簡略化することができる。ランタイム・ベース・レイヤー 122 についての更なる詳細は、2014 年 3 月 26 に出願され、"Virtualized Components in Computing System" (計算システムにおける仮想化コンポーネント) と題する同時係属中の米国特許出願第 14/228,129 号において見ることができる。

【0021】

[0026] 本明細書において紹介する技術の形態によれば、ストリーム処理システム 100 における仮想処理エージェント 102 は、ランタイム・システム 108 のランタイム・ベース・レイヤー 122 によって提供およびサポートされる仮想コンポーネントとして構成することができる。このような構成に基づいて、仮想処理エージェント 102 は、常に、仮想処理エージェント 102 が有効化されているか否かには関係なく、相互作用および/または動作のためにアドレス可能であることができる。仮想処理エージェント 102 は、ランタイム・システム 108 における他のサーバーにおいて仮想処理エージェント 102 を再有効化することによって、仮想処理エージェント 102 が有効化されたサーバーの障害後復元することができる。

【0022】

[0027] ある形態では、仮想処理エージェント 102 が、各々、一定のアクションを実施するように構成されてもよく、そして大規模なソフトウェア・アプリケーションのための機能を集散的に提供することもできる。例えば、コンピューター・ゲーム・アプリケーションでは、仮想処理エージェント 102 は、ゲーム・アプリケーションにおけるゲーム毎に作成され、各ゲームを管理するように構成することができる。また、仮想処理エージェント 102 は、ゲーム・アプリケーションにおいてゲームをホストするマップ毎に作成することもできる。更に、別々の仮想処理エージェント 102 を作成し、ゲーム・アプリケーションの種々のゲームに参加する個々のプレイヤーを管理するように構成することができる。仮想処理エージェント 102 によって実行されるアクション、仮想処理エージェント 102 の入力および出力、および/または他の構成を含む仮想処理エージェント 102 の構成は、仮想処理エージェント 102 の宣言的定義によって指定することができ、あるいはプログラミングによって得ることができる。

【0023】

[0028] ストリーム処理のためのコヒーレントなシステムを提供するために、ストリーム処理システム 100 における種々の仮想処理エージェント 102 が互いに通信しなければならないことがある。例えば、仮想処理エージェント 102 のステータスを更新するため、他の仮想処理エージェント 102 を有効化するため、および/または他の仮想処理エージェント 102 によって一定のアクションが実行されるきっかけを作るために、メッセージが 1 つの仮想処理エージェント 102 から他に伝達されることがある。本明細書において説明するある形態によれば、他の仮想処理エージェント 102 および/またはシステムにおける他のエンティティに伝達されることが意図された仮想処理エージェント 102 からのメッセージは、待ち行列システム 104 によって実装された待ち行列 106 内においてイベント 112 として発行することができる。

【0024】

[0029] 待ち行列システム 104 は、ストリーム処理システム 100 における種々のエンティティからのイベント発行を受けて永続化するための待ち行列 106 を実装する 1 つ以上のサーバーを含むことができる。尚、待ち行列システム 104 およびランタイム・システム 108 におけるサーバーは、同じまたは異なるサーバーを含んでもよく、待ち行列システム 104 およびランタイム・システム 108 におけるサーバーは、ウェブ・サーバー、アプリケーション・サーバー、ネットワーク・アプライアンス、専用コンピューター・ハードウェア・デバイス、パーソナル・コンピューター(「PC」)、あるいはこれらおよび/または当技術分野において周知である他の計算デバイスの任意の組み合わせであってもよいことは理解されてしかるべきである。これらのサーバーはネットワーク(図 1

10

20

30

40

50



には示されていない)を介して接続することができる。ネットワークは、ローカル・エリア・ネットワーク(「LAN」)、ワイド・エリア・ネットワーク(「WAN」)、インターネット、または計算デバイスを接続する当技術分野において周知である任意の他のネットワーク・トポロジであってもよい。

#### 【0025】

[0030] 待ち行列106において発行されるイベント112は、イベント送出エンティティにおいて起こったまたは観察されたイベントに関連する種々の情報を含むことができる。イベント送出エンティティは、仮想処理エージェント102、クライアント計算デバイス116、サーバー計算デバイス118、および/またはストリーム処理システム100において可能な他のエンティティであってもよい。イベント112に含まれる情報は、イベントが発生したエンティティの識別、イベントの時刻、イベントの場所、イベントに関与するエンティティの識別、および/またはイベントを記述するために使用することができる他の情報を含むことができるが、これらに限定されるのではない。イベント112は、イベント発行110をランタイム・システム108を介して送ることによって、待ち行列106において発行することができる。イベント発行110は、対応するイベント112において含まれる全ての情報を、イベントが発行のためであることの指示、イベントをどこにそしていつ発行するか指定するデーター等のような追加情報と共に含むことができる。更に他の形態では、待ち行列システム104は発行されたイベント112を待ち行列106の中に、あるいは待ち行列システム104および/またはランタイム・システム108にアクセス可能な他の記憶位置に永続化するまたは格納することもできる。

#### 【0026】

[0031] 他の仮想処理エージェント102、および/またはストリーム処理システム100における他のイベント発生エンティティから処理のためにイベントを受けるために、仮想処理エージェント102は、待ち行列106において発行されるイベントを受けることを登録申込みするために、1つ以上のイベント登録申込み(event subscription)を送ることができる。このような仮想処理エージェント102をここでは「明示サブスクライバー処理エージェント102」またはこれらのイベントの「明示サブスクライバー102」と呼ぶこともある。一例として、そして限定ではなく、イベント登録申込み114は、登録申込みする仮想処理エージェント102が受けることに興味があるイベントのタイプまたはコンテンツを指定するデーターを含むことができる。イベント登録申込み114は、登録申込みされたイベントが起こる時間期間を指定することができる。更に、イベント登録申込み114は、仮想処理エージェント102に配信されるイベントをフィルタリングするために利用できる1つ以上の述部(predicate)または規則も指定することができる。尚、仮想処理エージェント102が受けることに興味があるイベントを記述する種々の他のデーターも、イベント登録申込み114に含んでもよいことは理解されてしかるべきである。

#### 【0027】

[0032] 以下で更に詳しく説明するが、ランタイム・システム108は、ストリーム処理レイヤー124を実行するように構成することもできる。ストリーム処理レイヤー124を、ここでは、「ストリーム処理レイヤー124」または「ランタイム・ストリーム処理レイヤー124」と呼ぶこともある。ランタイム・ストリーム処理レイヤー124は、ストリーム処理システム100におけるエンティティ間のイベント通信を管理するように構成することができ、待ち行列106においてイベント発行110を管理すること、仮想処理エージェント102および/または他のイベント・サブスクライバーへのイベント112の配信を調整すること、イベント登録申込み114を受けて処理すること、ならびに他の可能な動作を含むがこれらに限定されるのではない。

#### 【0028】

[0033] ランタイム・ストリーム処理レイヤー124は、イベント登録申込み114を1つ以上の明示サブスクライバー処理エージェント102から受け、登録申込みされたイベント112が待ち行列106において発行されたときに、登録申込みされたイベント1

12を対応する明示サブスクライバー処理エージェント102に送ることができる。更  
他の形態によれば、ランタイム・ストリーム処理レイヤー124は、更に、イベント11  
2に含まれる発行イベント112またはイベント情報を、明示的にイベント112を登録  
申込みしていないが発行イベント112に含まれる情報を受けることに関心があるかもし  
れない1つ以上の仮想処理エージェント102に送ることもできる。この仮想処理エー  
ジェント102を、ここでは、発行イベント112の「暗示サブスクライバー処理エー  
ジェント102」または「暗示サブスクライバー102」とも呼ぶ。

【0029】

[0034] 暗示サブスクライバー処理エージェント102は、仮想処理エージェント10  
2によって実行されるアクション、仮想処理エージェント102の入力および出力、仮想  
処理エージェント102のステータス、および/または仮想処理エージェント102の他  
の構成というような、仮想処理エージェント102の構成または設定を調べることによ  
って識別することができる。例えば、仮想処理エージェント102によって実行されるアク  
ションが、開始された、述べられた、またはそれ以外で発行イベント112に関連付けら  
れたオブジェクトあるいはエンティティを伴う場合、この仮想処理エージェント102を  
暗示サブスクライバー処理エージェント102として識別することができる。また、発行  
イベント112に含まれるデータが仮想処理エージェント102によって消費または処理  
される場合、この仮想処理エージェント102を暗示サブスクライバー処理エー  
ジェント102として識別することができる。尚、これらの例は単なる例示に過ぎず、限定とし  
て解釈してはならないことは理解されてしかるべきである。暗示サブスクライバー処理エ  
ジェント102を識別する他の方法も利用することができる。

【0030】

[0035] 一旦ランタイム・ストリーム処理レイヤー124が暗示サブスクライバー処理  
エージェント102を識別したなら、ランタイム・ベース・レイヤー122を介して、こ  
の暗示サブスクライバー処理エージェント102にイベント112を配信することができ  
る。先に説明したように、ストリーム処理システム100における仮想処理エージェント  
102は、ランタイム・システム108のランタイム・ベース・レイヤー122によって  
サポートされる仮想コンポーネントとして実装することができる。ランタイム・ストリー  
ム処理レイヤー124の観点からは、仮想処理エージェント102がシステムにおいて有  
効化されているか否かには関係なく、イベント112の配信のために常にアドレス可能で  
あることができる。したがって、ランタイム・ストリーム処理レイヤー124は、イベン  
ト112を配信のためにランタイム・ベース・レイヤー122に渡すことができる。

【0031】

[0036] イベント112を受けると、ランタイム・ベース・レイヤー122は、暗示サ  
ブスクライバー処理エージェント102が有効化されているか否、即ち、暗示サブスクラ  
イバー処理エージェント102がメモリーに既にロードされており動作する準備ができて  
いるか否か判定することができる。暗示サブスクライバー処理エージェント102が未だ  
有効化されていない場合、ランタイム・ベース・レイヤー122は暗示サブスクライバー  
処理エージェント102を有効化し、次いで有効化した暗示サブスクライバー処理エー  
ジェント102にイベント112を配信することができる。同様に、イベント112を明示  
サブスクライバー処理エージェント102に配信するときも、ランタイム・ベース・レイ  
ヤー122は、この明示サブスクライバー処理エージェント102が有効化されているか  
否判定し、必要であれば、イベント112の配信前に、明示サブスクライバー処理エー  
ジェント102を有効化することができる。

【0032】

[0037] 仮想処理エージェント102の実現例に依存して、そして有効化されたときの  
仮想処理エージェント102のステータスに依存して、有効化された暗示サブスクライバ  
ー処理エージェント102の内一部は、未だイベント112を受け入れて処理するように  
構成されていない場合がある。例えば、暗示サブスクライバー処理エージェント102は  
、有効化されたときに、未だイベント・ハンドラーが構築されていないこともあり得る。

このようなシナリオでは、イベント 1 1 2 に含まれるデータの一部または全部を含むイベント情報を編成し直して(rearranged)、暗示サブスクライバー処理エージェント 1 0 2 に受け入れ可能なフォーマットで、暗示サブスクライバー処理エージェント 1 0 2 に配信することができる。他のシナリオでは、イベント 1 1 2 に含まれる詳細データを送ることなく、イベント 1 1 2 の発生を示すイベント情報を仮想処理エージェント 1 0 2 に送れば十分な場合もある。

#### 【 0 0 3 3 】

[0038] 受けたイベント 1 1 2 またはイベント情報に基づいて、一部の暗示サブスクライバー処理エージェント 1 0 2 は、待ち行列 1 0 6 において発行される更に多くのイベントを処理のために受けるために、1 つ以上のイベント登録申込み 1 1 4 をランタイム・ストリーム処理レイヤー 1 2 4 に提出することを決定することができる。先に端的に述べたように、暗示サブスクライバー処理エージェント 1 0 2 によって提出された明示イベント登録申込み 1 1 4 に含まれるイベントは、イベント登録申込み 1 1 4 が受けられて処理される前に待ち行列 1 0 6 において既に発行されている過去のイベントを含む場合がある。ランタイム・ストリーム処理レイヤー 1 2 4 は、対応するイベントが永続化されている待ち行列または他のストレージから過去のイベントを引き出し、引き出したイベントを暗示サブスクライバー処理エージェント 1 0 2 に配信することができる。尚、暗示サブスクライバー処理エージェント 1 0 2 がイベント登録申込み 1 1 4 を提出するとき、暗示サブスクライバー処理エージェント 1 0 2 は、登録申込みしたイベントに対しては、明示サブスクライバー処理エージェント 1 0 2 に変わることができ、明示サブスクライバー処理エージェント 1 0 2 を扱うことについて先に説明したのと同様に処理されればよい。

#### 【 0 0 3 4 】

[0039] 更に、以上では、仮想処理エージェント 1 0 2 がイベント発行 1 1 0 を送り、イベント登録申込み 1 1 4 を提出し、および / またはイベント 1 1 2 を受けることができると主に説明したが、ストリーム処理システム 1 0 0 における種々の他のエンティティもこれらの動作を実行できることは理解されてしかるべきである。例えば、コンピュータ・ゲーム・アプリケーションのユーザーが、そのゲームのリーダー・ボード情報のような、ゲームに関する情報を受けることに関心がある場合がある。このユーザーは、このような情報をクライアント計算デバイス 1 1 6 を介して得るために、イベント登録申込み 1 1 4 をランタイム・システム 1 0 8 に送ることができる。クライアント計算デバイス 1 1 6 は、パーソナル・コンピュータ(「PC」)、ラップトップ、ノートブック、パーソナル・デジタル・アシスタント(「PDA」)、ゲーム・コンソール、セットトップ・ボックス、電子書籍、消費者用電子デバイス、スマートフォン、タブレット計算デバイス、サーバー・コンピュータ、またはランタイム・システム 1 0 8 と通信することができる任意の他の計算デバイスであってもよい。

#### 【 0 0 3 5 】

[0040] 同様に、サーバー計算デバイス 1 1 8 も、サーバー・コンピュータ、ウェブ・サーバー、アプリケーション・サーバー、ネットワーク・アプライアンス、専用コンピュータ・ハードウェア・デバイス、PC、あるいはこれらおよび / または当技術分野において周知の他の計算デバイスの任意の組み合わせであってもよく、そこでゲームを実行することができる。サーバー計算デバイス 1 1 8 は、待ち行列 1 0 6 においてゲーム開始イベント 1 1 2 を発行するためにイベント発行 1 1 0 を送ることができる。また、サーバー計算デバイス 1 1 8 が処理することに関心があるイベント 1 1 2 を受けるために、サーバー計算デバイス 1 1 8 は 1 つ以上のイベント登録申込み 1 1 4 を送ることもできる。また、ストリーム処理システム 1 0 0 における他のエンティティも、以上で説明したのと同様にイベント 1 1 2 を発行し、受け、および / または登録申込みするように構成することができる。

#### 【 0 0 3 6 】

[0041] また、図 1 は、主に、ストリーム処理システム 1 0 0 における種々のエンティティからイベント発行 1 1 0 を受けて永続化するための待ち行列 1 0 6 の使用を開示した

が、種々の他のメカニズムも同様に利用できることは認められてしかるべきである。実際に、後の処理のためにデータを格納および保持することができる事実上あらゆるデータ構造を、待ち行列 106 に関して以上で説明したのと同様に利用することができる。本願は、イベント発行 110 を保持し格納するための全てのこのようなデータ構造を含むことを意図している。

【0037】

[0042] これより図2を参照して、待ち行列106およびランタイム・ストリーム処理レイヤー124の更に他の形態を示すブロック図について説明する。先に端的に説明したように、ランタイム・ストリーム処理レイヤー124は、イベント発行110およびイベント登録申込み114を処理するために、仮想ストリームを提供しサポートすることができる。仮想ストリームを利用することによって、仮想処理エージェント102および/または他のエンティティによって提出されたイベント発行110およびイベント登録申込み114を、待ち行列106ではなく、仮想ストリームに導くことができる。その結果、仮想処理エージェント102は、待ち行列106の存在を知る必要がなくなることが可能である。図2は、ランタイム・ストリーム処理レイヤー124によって抽象化された種々の仮想ストリーム、ならびに仮想ストリーム、待ち行列106、および仮想処理エージェント102間における相互作用を示す。

【0038】

[0043] 図2に示すように、ランタイム・ストリーム処理レイヤー124は、仮想ストリームに関連する機能を提供しサポートするため、そして待ち行列106および仮想ストリーム間におけるイベントのマッピングを維持するために、仮想ストリーム・モジュール202を含むことができる。仮想ストリーム・モジュール202によって提供される仮想ストリームは、イベント発行仮想ストリーム204A~204C(ここでは、個々にイベント発行仮想ストリーム204と呼ぶことも、または纏めてイベント発行仮想ストリーム204と呼ぶこともある)と、イベント登録申込み仮想ストリーム206A~206D(ここでは、個々にイベント登録申込み仮想ストリーム206と呼ぶことも、または纏めてイベント登録申込み仮想ストリーム206と呼ぶこともある)とを含む。

【0039】

[0044] イベント発行仮想ストリーム204の各々は、その中で発行されるイベントによって定めることができ、イベント登録申込み仮想ストリーム206の各々は、登録申込みエンティティが受けることに関心があるイベントによって定めることができる。例えば、仮想処理エージェント102Aが、イベント112を発行するために、イベント発行110をランタイム・ストリーム処理レイヤー124に送るとしてもよい。イベント発行110は、イベント発行仮想ストリーム204Cを介して、ランタイム・ストリーム処理レイヤー124に送ることができる。イベント発行110がランタイム・ストリーム処理レイヤー124によって受けられる前には、イベント発行仮想ストリーム204Cが存在しないこともあり得るので、イベント発行110の到達が、イベント発行仮想ストリーム204Cを作成させてもよい。言い換えると、イベント発行110またはその中に含まれるイベントが、イベント発行110が送られるイベント発行仮想ストリーム204を定め、これを存在させるのでもよい。同様に、他の仮想処理エージェント102も、イベントを発行するために、イベント発行110をそれらの対応する発行仮想ストリーム204に提出することができる。イベント発行仮想ストリーム204を介してイベント発行110を受けた後、仮想ストリーム・モジュール202は、イベント発行110において指定されたイベントを、発行のために、待ち行列106に送ることができる。

【0040】

[0045] 同様に、仮想処理エージェント102Pが、仮想処理エージェント102Pが受けることに関心があるイベントを示すイベント登録申込み114を提出すると、このようなイベント登録申込み114が、イベント登録申込み114によって指定されるイベントに対応するイベント登録申込み仮想ストリーム206Dを定めることができる。その結果、イベント登録申込み114を提出することによって、仮想処理エージェント102P

10

20

30

40

50

は、イベント登録申込み114によって定められたイベント登録申込み仮想ストリーム206Dに登録申込みしているのと同等となる。尚、仮想処理エージェント102は、1つまたは複数のイベント登録申込み114をランタイム・ストリーム処理レイヤー124に提出することによって、複数のイベント登録申込み仮想ストリーム206に登録申込みできることは理解されてしかるべきである。一旦ランタイム・ストリーム処理レイヤー124がイベント登録申込み114を受けたなら、イベント登録申込み仮想ストリーム206Dを作成し、存在させることができる。イベント登録申込み114に基づいて、仮想ストリーム・モジュール202は、待ち行列106において発行されたイベント112を引き出し、このイベント112をイベント登録申込み仮想ストリーム206Dにプッシュすることができ、次いで、イベント登録申込み仮想ストリーム206Dを仮想処理エージェント102Pに配信することができる。

10

#### 【0041】

[0046] 例えば、仮想処理エージェント102Aは、ゲームを処理するように構成することができ、このゲームにおいて発生したイベント112を発行するために、イベント発行110をランタイム・ストリーム処理レイヤー124に送ることができる。提出されたイベント発行110は、ゲーム・イベント・ストリーム204Cを作成させて、発行のためにイベント112を受けることができる。ゲーム・イベント・ストリーム204Cを介して受けられたイベント112は、更に、発行および格納のために、待ち行列106に送ることができる。その間に、ランタイム・ストリーム処理レイヤー124が、ゲームにおける全てのキル・イベント(kill event)を受けられることを登録申込みするために、仮想処理エージェント102Pからイベント登録申込み114を受けることができた。このようなイベント登録申込み114が、キル・イベント・ストリーム206Dを作成させることができた。

20

#### 【0042】

[0047] ゲーム・イベント・ストリーム204Cにおいてイベント発行110を受けたときに、仮想ストリーム・モジュール202は、ゲーム・イベント・ストリーム204Cを介して発行されたキル・イベントが1つでもある場合には、仮想処理エージェント102Pへの配信のために、それをキル・イベント・ストリーム206Dに転送すべきことを検出することができる。この検出は、例えば、先に説明したように、待ち行列106におけるイベントおよび仮想ストリーム・モジュール202によって維持されている仮想ストリームにおけるイベントのマッピングを利用することによって実行することができる。その後、ゲーム・イベント・ストリーム204Cを介して発行されたキル・イベント112毎に、仮想ストリーム・モジュール202はキル・イベント112を待ち行列106から引き出し、仮想処理エージェント102Pへの配信のためにそれをキル・イベント・ストリーム206Dにプッシュすることができる。尚、以上の例は単なる例示に過ぎず、種々の他のイベント112、イベント発行110、イベント登録申込み114、および/または他のデータも仮想処理エージェント102と待ち行列106との間で、仮想ストリーム204および206、および/またはランタイム・ストリーム処理レイヤー124を介して通信できることは認められてしかるべきである。

30

#### 【0043】

[0048] 更に他の形態によれば、仮想処理エージェント102によって提出されたイベント登録申込み114は、対応するイベント登録申込み仮想ストリーム206を更に定めるために利用することができる1つ以上の述部を更に指定することもできる。述部は、イベント登録申込み仮想ストリーム206にプッシュされるイベントをフィルタリングするために利用することができる1つ以上のフィルタを作成するというように、イベント登録申込み仮想ストリーム206をリファインするために、適正な時点において、ランタイム・ストリーム処理レイヤー124によって処理することができる。述部を利用すると、仮想処理エージェント102に送られるイベントを更にリファインし、不要なイベントの仮想処理エージェント102への配信を排除することができる。

40

#### 【0044】

50

[0049] 以上からわかるように、仮想ストリーム 204 および 206 は非常に動的で柔軟性があり、必要なときにはいつでも作成することができ、これによってストリーム処理システム 100 のスケーラビリティを著しく高めることができる。更に他の形態によれば、仮想ストリーム 204 および 206 は論理的に常時存在することができるが、処理すべきイベントが仮想ストリーム 204 および 206 内にあるときにだけ、システム・リソースがこれらに割り当てられる。一定の時間期間にわたって仮想ストリーム 204 および 206 においてイベントが受けられないとき、それらのリソースを取り戻すことによって、仮想ストリーム 204 および 206 を無効化することができる。この時間期間は予め決められていても、システム・リソースの使用度のような、システムのステータスにしたがって動的に調節されてもよい。このような実現例は、仮想ストリームが細かい粒度で組み立てられるシナリオでは、有用となることができる。例えば、アプリケーション・システムにおける仮想ストリームは、ユーザー毎、デバイス毎、および/またはセッション毎に組み立てることができる。大きなアプリケーション・システムでは、ユーザー数、デバイス数、および/またはセッション数、つまり仮想ストリーム数も、億単位または 10 億単位となることもあり得る。これらの仮想ストリームに関連するイベントは、しかしながら、希にしか発生しない、および/または纏まって(in burst)発生する場合もあり、しかも一度に仮想ストリームの小さな部分だけがアクティブであればよい。このようなシステムでは、処理すべきイベントが仮想ストリーム 204 および 206 にあるときにだけリソースを仮想ストリームに割り当てることによって、システムの効率およびスケーラビリティを著しく高めることができる。

10

20

#### 【0045】

[0050] その上、仮想ストリーム・モジュール 202 は、更に、待ち行列 106 と相互作用する前に、仮想処理エージェント 102 への通信および仮想処理エージェント 102 からの通信を集計するように構成することもできる。例えば、仮想ストリーム・モジュール 202 は、複数のイベント発行 110 を複数の仮想処理エージェント 102 から受けることもある。これら複数のイベント発行 110 を、例えば、多重化によって集計し、1 回の通信接続によって複数のイベント 112 を発行するために、待ち行列 106 に伝達することができる。同様に、複数の仮想処理エージェント 102 から受けた複数のイベント登録申込み 114 を処理して集計することができ、同様に待ち行列 106 との 1 回の通信接続によって、複数の仮想処理エージェント 102 に送るべきイベント 112 を待ち行列 106 から引き出すことができる。

30

#### 【0046】

[0051] 尚、以上の例では、複数の通信が集計され、1 回の通信接続によって待ち行列 106 に伝達されると説明したが、1 回よりも多い通信接続が、伝達のために、ランタイム・ストリーム処理レイヤー 124 と待ち行列 106 との間に作られてもよいことは理解されてしかるべきである。作られる通信接続の回数は、しかしながら、イベント発行 110 および/またはイベント登録申込み 114 のような、仮想処理エージェント 102 からの通信要求の回数よりもはるかに少なくすることができる。通信集計の利点は、ストリーム処理システム 100 の規模が大きくなる程、増々重要になるであろう。例えば、数百万個の仮想処理エージェント 102 があるストリーム処理システム 100 では、仮想処理エージェント 102 からの通信要求は、数千万回の規模になる可能性があり、通例、仮想処理エージェント 102 の各々と、このような大多数の直接 TCP/IP 接続を作ることは、待ち行列 106 にとって非実用的である。通信の集計によって、仮想ストリーム・モジュール 202 は、待ち行列 106 に対する直接 TCP/IP 接続の回数を数千回に低減し、これによって通信リソースの消費を著しく減らすことができる。

40

#### 【0047】

[0052] 更に、図 1 および図 2 は、1 つの待ち行列 106 がイベント 112 を発行し永続化するために利用されることを例示するが、複数の待ち行列 106 を採用してもよいことも認められてしかるべきである。このような実現例では、仮想ストリーム・モジュール 202 は、更に、複数の待ち行列 106 を管理し、これらの待ち行列 106 と種々の仮想

50

ストリーム 204 および 206 との間におけるイベントのマッピングを維持するように構成することができる。また、クライアント計算デバイスおよびサーバー計算デバイスのような、ストリーム処理システム 100 における他のエンティティも、イベント発行 110 によってイベントを発行し、イベント登録申込み 114 を提出し、および/または仮想ストリーム 204 および 206 を介してイベント 112 を受けることができることも理解されてしかるべきである。

#### 【0048】

【0053】 図 2 は、更に、処理エージェント・コンテナ 208A ~ 208B（ここでは、個々に処理エージェント・コンテナ 208 と呼ぶこともあり、または纏めて処理エージェント・コンテナ 208 と呼ぶこともある）を利用して仮想処理エージェント 102 を編成 (organize) および管理する形態を示す。処理エージェント・コンテナ 208 の各々は、他の処理エージェント・コンテナ 208 から分けられ孤立された仮想処理エージェント 102 に、独立した計算環境を提供することができる。仮想処理エージェント 102 をホストするために、仮想機械のような種々の計算リソースを処理エージェント・コンテナ 208 に割り当てることができる。また、各処理エージェント・コンテナ 208 は、他の処理エージェント・コンテナ 208 とは独立して、スケーリングする (scale) ことができる。尚、イベント 112 を発行する仮想処理エージェント 102、およびイベント 112 に登録申込みしこれを受ける仮想処理エージェント 102 は、1 つの同じ処理エージェント・コンテナ 208 内にあっても、または 2 つの異なる処理エージェント・コンテナ 208 内にあってもよいことは理解されてしかるべきである。更に、仮想処理エージェント 102 は、処理エージェント・コンテナ 208 の使用を伴わない種々の他の方法で編成および管理されてもよいことも理解されてしかるべきである。

#### 【0049】

【0054】 図 3 は、イベントを仮想処理エージェント 102 に配信するための 1 つの例示的なルーチン 300 の形態を示す流れ図である。ある実現例では、ルーチン 300 は、図 1 に関して先に説明したランタイム・システム 108 のランタイム・ストリーム処理レイヤー 124 によって実行される。しかしながら、ルーチン 300 は、ランタイム・システム 108 の他のモジュールおよび/またはコンポーネントによって、あるいは図 1 に示したストリーム処理システム 100 における他のエンティティのモジュールおよび/またはコンポーネントによって実行されてもよいことは認められてしかるべきである。

#### 【0050】

【0055】 尚、図 3 および他の図に関して本明細書において説明する論理動作は、(1) 計算システムにおいて実行するコンピューター実装アクトまたはプログラム・モジュールのシーケンスとして、および/または (2) 計算システム内部で相互接続された機械論理回路または回路モジュールとして実現されることは認められてしかるべきである。実現例は、計算システムの性能および他の要件に依存する選択事項である。したがって、本明細書において説明する論理動作は、様々に、状態、動作、構造的デバイス、アクト、またはモジュールと呼ばれる。これらの状態、動作、構造的デバイス、アクト、およびモジュールは、ソフトウェア、ファームウェア、特殊目的デジタル・ロジック、およびこれらの任意の組み合わせで実現することができる。また、図に示しここで説明するよりも多い動作または少ない動作が実行されてもよいことも認められてしかるべきである。これらの動作は、ここで説明する順序とは異なる順序で実行することもできる。また、図示する方法の各々は任意の時点で終了することができ、その全体を実行する必要がないことも理解されてしかるべきである。

#### 【0051】

【0056】 以上の方法の動作、および/または実質的に等価な動作の一部または全部は、以下に定めるような、コンピューター記憶媒体に含まれるコンピューター読み取り可能命令の実行によって実行することができる。「コンピューター読み取り可能命令」という用語およびその異形は、この説明および特許請求の範囲において使用される場合、本明細書では、ルーチン、アプリケーション、アプリケーション・モジュール、プログラム・モジ

ジュール、プログラム、コンポーネント、データ構造、アルゴリズム等を含むように広く使用される。コンピュータ読み取り可能命令は、種々のシステム構成において実装することができ、単一プロセッサまたはマルチプロセッサ・システム、ミニコンピュータ、メインフレーム・コンピュータ、パーソナル・コンピュータ、ハンドヘルド計算デバイス、マイクロプロセッサ・ベース、プログラマブル消費者用電子機器、これらの組み合わせ等が含まれる。

#### 【 0 0 5 2 】

[0057] ルーチン 3 0 0 は、動作 3 0 2 において開始し、ランタイム・システム 1 0 8 のストリーム処理レイヤー 1 2 4 が対応するイベント発行仮想ストリーム 2 0 4 においてイベント発行 1 1 0 を受けることができる。先に説明したように、受けたイベント発行 1 1 0 に基づいてイベント発行仮想ストリーム 2 0 4 を動的に作成することができる。次いで、ルーチン 3 0 0 は動作 3 0 4 に進み、ランタイム・ストリーム処理レイヤー 1 2 4 は、イベント発行 1 1 0 において指定されたイベント 1 1 2 を、発行のために待ち行列 1 0 6 に送る。更に、ランタイム・ストリーム処理レイヤー 1 2 4 は、イベント 1 1 2 が後に引き出せるように、イベント 1 1 2 を待ち行列 1 0 6 に永続化するまたは格納するように、待ち行列 1 0 6 に命令することができる。ある実現例では、イベント 1 1 2 は、待ち行列 1 0 6 および / またはランタイム・ストリーム処理レイヤー 1 2 4 にアクセス可能である記憶デバイスに永続化するまたは格納することもできる。

#### 【 0 0 5 3 】

[0058] 動作 3 0 4 から、ルーチン 3 0 0 は動作 3 0 6 に進み、ランタイム・ストリーム処理レイヤー 1 2 4 はイベント 1 1 2 を待ち行列 1 0 6 から引き出すことができ、ルーチン 3 0 0 は更に動作 3 0 8 に進み、イベント 1 1 2 に基づいて、暗示サブスクライバ処理エージェント 1 0 2 を識別することができる。先に注記したように、暗示サブスクライバ処理エージェント 1 0 2 とは、イベント 1 1 2 を明示的に登録申込みしていないが、発行されたイベント 1 1 2 に含まれる情報を受けることに関心があるかもしれない仮想処理エージェント 1 0 2 である。暗示サブスクライバ処理エージェント 1 0 2 は、仮想処理エージェント 1 0 2 によって実行されるアクション、仮想処理エージェント 1 0 2 の入力および出力、仮想処理エージェント 1 0 2 のステータス、および / または仮想処理エージェント 1 0 2 の他の構成というような、仮想処理エージェント 1 0 2 の構成または設定を調べることによって識別することができる。例えば、仮想処理エージェント 1 0 2 によって実行されるアクションが、開始した、述べられた、またはそれ以外で発行されたイベント 1 1 2 に関連付けられたオブジェクトまたはエンティティを伴う場合、仮想処理エージェント 1 0 2 を暗示サブスクライバ処理エージェント 1 0 2 として識別することができる。また、発行されたイベント 1 1 2 に含まれるデータが仮想処理エージェント 1 0 2 によって消費または処理される場合、仮想処理エージェント 1 0 2 を暗示サブスクライバ処理エージェント 1 0 2 として識別することができる。

#### 【 0 0 5 4 】

[0059] ある実現例では、ランタイム・ストリーム処理レイヤー 1 2 4 が、異なるタイプのイベントまたは対応するイベント登録申込み仮想ストリーム 2 0 6 を 1 つ以上の暗示サブスクライバ処理エージェント 1 0 2 にマッピングする暗示有効化テーブル(implicit activation table)を維持することもできる。暗示有効化テーブルの一例を図 6 に示し、後に説明する。暗示有効化テーブルは、予め生成され、必要なときにランタイム・ストリーム処理レイヤー 1 2 4 が参照することができる。また、暗示有効化テーブルは、ランタイム・ストリーム処理レイヤー 1 2 4 において実行して、所与のイベントまたはイベント情報に対して 1 つ以上の暗示サブスクライバ処理エージェント 1 0 2 を出力することができる機能またはモジュールによって実現することもできる。尚、暗示有効化テーブルは、以上で説明した同じ原理に従うことによって、他の方法で実現されてもよいことは注記してしかるべきである。

#### 【 0 0 5 5 】

[0060] 動作 3 0 8 から、ルーチン 3 0 0 は動作 3 1 0 に進み、例えば、イベント 1 1

10

20

30

40

50



2 またはイベント 1 1 2 のイベント情報を暗示サブスクライバー処理エージェント 1 0 2 に送ることによって、動作 3 0 8 において識別された暗示サブスクライバー処理エージェント 1 0 2 にイベント 1 1 2 を知らせることができる。先に述べたように、イベント情報は、イベント 1 1 2 内に含まれる情報、またはイベント 1 1 2 に関連する情報を含むことができる。暗示サブスクライバー処理エージェント 1 0 2 の性質に依存して、ある暗示サブスクライバー処理エージェント 1 0 2 に送られるイベント情報は、イベント 1 1 2 に含まれるデータの一部分または全部を含んでもよい。他の暗示サブスクライバー処理エージェント 1 0 2 については、イベント情報がイベント 1 1 2 の存在の指示だけを含むのでもよい。図 4 に関して詳しく説明するが、受けたイベント 1 1 2 またはイベント情報に基づいて、暗示サブスクライバー処理エージェント 1 0 2 は、更に多くのイベントを受けること、および / または更に他の動作を実行することに登録申込みするか否か決定することができる。

10

#### 【 0 0 5 6 】

[0061] 動作 3 1 0 から、ルーチン 3 0 0 は動作 3 1 2 に進み、ランタイム・ストリーム処理レイヤー 1 2 4 は、対応するイベント登録申込み仮想ストリーム 2 0 6 に登録申込みするために、暗示サブスクライバー処理エージェント 1 0 2 の内 1 つ以上からのイベント登録申込み 1 1 4 を受けることができる。次いで、ルーチン 3 0 0 は動作 3 1 4 に進み、ランタイム・ストリーム処理レイヤー 1 2 4 は、明示イベント登録申込み 1 1 4 において指定されたイベントが、既に発生し待ち行列 1 0 6 において発行された過去のイベントを含むか否か判定することができる。明示イベント登録申込み 1 1 4 が過去のイベントを含むと判定した場合、ルーチン 3 0 0 は動作 3 1 6 に進み、ランタイム・ストリーム処理レイヤー 1 2 4 は待ち行列 1 0 6 および / または過去のイベントを格納する他の記憶デバイスから過去のイベントを引き出し、引き出した過去のイベントに対応する仮想処理エージェント 1 0 2 に配信することができる。

20

#### 【 0 0 5 7 】

[0062] 動作 3 1 6 から、または動作 3 1 4 において明示イベント登録申込み 1 1 4 が過去のイベントを含まないと判定した場合、ルーチン 3 0 0 は動作 3 1 8 に進み、ランタイム・ストリーム処理レイヤー 1 2 4 は、それが維持する明示登録申込みを更新する、および / または受けた明示イベント登録申込み 1 1 4 に基づいて、イベント発行仮想ストリーム 2 0 4、イベント登録申込み仮想ストリーム 2 0 6、および待ち行列 1 0 6 におけるイベントのマッピングを更新するというような、他の動作を実行することができる。

30

#### 【 0 0 5 8 】

[0063] 動作 3 0 6 から、ルーチン 3 0 0 は動作 3 2 0 にも進むことができる。動作 3 2 0 において、ランタイム・ストリーム処理レイヤー 1 2 4 は、以前に仮想処理エージェント 1 0 2 によって提出されたイベント登録申込み 1 1 4 に基づいて、1 つ以上の明示サブスクライバー処理エージェント 1 0 2 を識別することができる。更に、ルーチン 3 0 0 は動作 3 2 2 に進むことができ、明示サブスクライバー処理エージェント 1 0 2 にイベント 1 1 2 を配信することができる。動作 3 1 8 または動作 3 2 2 から、ルーチン 3 0 0 は動作 3 2 4 に進み、ランタイム・ストリーム処理レイヤー 1 2 4 が更に多くのイベントを配信する必要があるか否か判定することができる。ランタイム・ストリーム処理レイヤー 1 2 4 が更に多くのイベントを配信する必要がある場合、ルーチン 3 0 0 は動作 3 0 2 に戻り、処理および配信のために更に多くのイベント発行 1 1 0 を受ける。それ以外の場合、ルーチン 3 0 0 は動作 3 2 6 に進み、ルーチン 3 0 0 は終了する。

40

#### 【 0 0 5 9 】

[0064] 尚、以上で説明したイベント配信プロセスにおいて、イベント 1 1 2 を明示サブスクライバー処理エージェント 1 0 2 に配信するとき、および / またはイベント 1 1 2 またはイベント情報を暗示サブスクライバー処理エージェント 1 0 2 に送るとき、ランタイム・ベース・レイヤー 1 2 2 は、最初に、対応する処理エージェント 1 0 2 が有効化されているか否か判定してもよいことは認められてしかるべきである。有効化されていない処理エージェント 1 0 2 がある場合は、ランタイム・ベース・レイヤー 1 2 2 はその処理

50

エージェント 102 を有効化し、次いで有効化した処理エージェント 102 にイベント 112 またはイベント情報を配信することができる。

【0060】

[0065] これに関して、暗示サブスクライバー処理エージェント 102 のような処理エージェント 102 を動的に有効化することができ、処理エージェント 102 が処理することを責務とするストリーム 206 における第イベント 112 の到達時にシステム・リソースを割り当てることを含む。つまり、第 1 イベント 112 が到達する前に、メモリおよび CPU のようなシステム・リソースを処理エージェント 102 に割り当てる必要性をなくすることができる。同様に、例えば、処理エージェント 102 が登録申込みしたストリーム 206 から処理のためにイベント 112 を受けないために、処理エージェントがある所与の時間期間アイドルになるとき、そのシステム・リソースを引き取るために処理エージェント 102 を透過的に無効化することもできる。処理エージェント 102 を無効化するための所与の時間期間は、予め決められていてもよく、またはシステム・リソースの使用度のような、システムのステータスにしたがって動的に調節されてもよい。

【0061】

[0066] 図 4 は、イベント処理および登録申込みのための 1 つの例示的なルーチン 400 の形態を示す流れ図である。ある実現例では、ルーチン 400 は、図 1 および図 2 に関して先に説明した仮想処理エージェント 102 によって実行される。しかしながら、ルーチン 400 は、図 1 に示したストリーム処理システム 100 における他のエンティティによって実行されてもよいことは認められてしかるべきである。

【0062】

[0067] ルーチン 400 は動作 402 において開始し、暗示サブスクライバー処理エージェント 102 がイベント 112 またはイベント 112 のイベント情報をランタイム・ストリーム処理レイヤー 124 から受け、更に受けたイベント 112 またはイベント情報を処理することができる。動作 402 から、ルーチン 400 は更に動作 404 に進み、暗示サブスクライバー処理エージェント 102 は、受けたイベント 112 またはイベント情報に基づいて、更に多くのイベント 112 に登録申込みする必要があるか否か判定することができる。暗示サブスクライバー処理エージェント 102 が更に多くのイベントに登録申込みする必要があると判定した場合、ルーチンは動作 406 に進み、暗示サブスクライバー処理エージェント 102 は、暗示サブスクライバー処理エージェント 102 がイベントを受けることに関心がある 1 つ以上のイベント登録申込み仮想ストリーム 206 を識別することができる。動作 408 において、暗示サブスクライバー処理エージェント 102 は、イベント登録申込み 114 をランタイム・ストリーム処理レイヤー 124 に提出することによって、識別されたイベント登録申込み仮想ストリーム 206 に登録申込みすることができる。イベント登録申込み 114 を提出することによって、暗示サブスクライバー処理エージェント 102 は、提出したイベント登録申込み 114 に関して、明示サブスクライバー処理エージェント 102 に変わることができる。

【0063】

[0068] 動作 408 から、ルーチン 400 は動作 410 に進み、明示サブスクライバー処理エージェント 102 は、待ち行列 106 において発行された、登録申込みイベントを受けることができる。先に説明したように、動作 408 において提出されたイベント登録申込み 114 が過去のイベントを指定する場合、動作 410 において受けられたイベントはこれら過去のイベントを含む可能性がある。次いで、ルーチン 400 は動作 412 に進み、受けたイベントを処理することができる。更に、ルーチン 400 は動作 414 に進み、明示サブスクライバー処理エージェント 102 が更に多くのイベントを処理する必要があるか否か判定することができる。更に多くのイベントを必要とすると判定した場合、ルーチン 400 は動作 410 に戻り、明示サブスクライバー処理エージェント 102 は処理のために更に多くのイベント 112 を受けることができる。動作 414 において、それ以上のイベントは必要ないと判定した場合、または動作 404 において暗示サブスクライバー処理エージェント 102 が更に多くのイベントに登録申込みする必要があると判定した

場合、ルーチン 4 0 0 は動作 4 1 6 に進み、ルーチン 4 0 0 は終了する。

【 0 0 6 4 】

[0069] 図 5 A ~ 図 5 D は、本明細書において紹介する 1 つ以上の形態にしたがって、異なる時点において待ち行列 1 0 6 において発行されたイベント 1 1 2 に応答した、仮想処理エージェント例 1 0 2 による動作を示す。図 5 A に示すように、ランタイム・システム 1 0 8 のストリーム処理レイヤー 1 2 4 は、暗示有効化テーブル 5 0 6 を維持する、またそうでなければ生成することができる。暗示有効化テーブル 5 0 6 は、異なるタイプのイベント 1 1 2 またはそれらの対応するイベント登録申込み仮想ストリーム 2 0 6 を暗示サブスクライバー処理エージェント 1 0 2 にマッピングする。図 5 A に示す暗示有効化テーブル例 5 0 6 によれば、マップ処理エージェント、ゲーム処理エージェント、リーダー・ボード処理エージェント、および/またはエリート・プレーヤー処理エージェントが、ゲーム・ストリーム 5 0 4 の暗示サブスクライバー処理エージェント 1 0 2 として識別されればよい。同様に、プレーヤー・ストリーム 5 2 0 では、可能な暗示サブスクライバー処理エージェント 1 0 2 は、プレーヤー処理エージェント、ゲーム処理エージェント、および/またはエリート・プレーヤー処理エージェントを含むことができる。暗示有効化テーブル 5 0 6 に基づいて、ランタイム・ストリーム処理レイヤー 1 2 4 は、イベント 1 1 2 が受けられ待ち行列 1 0 6 において発行されるに連れて、1 つ以上の暗示サブスクライバー処理エージェント 1 0 2 を識別し有効化することができる。

10

【 0 0 6 5 】

[0070] 時点 T<sub>0</sub> において、例えば、イベント発行仮想ストリーム 2 0 4 を介して、ゲーム開始イベント 5 0 2 が待ち行列 1 0 6 において発行される。ゲーム開始イベント 5 0 2 に含まれるデータは、ゲーム X がマップ H 上において、現在ゲーム X にいるプレーヤー A ~ F と開始したことを示す。ランタイム・ストリーム処理レイヤー 1 2 4 は、ゲーム開始イベント 5 0 2 を待ち行列 1 0 6 から引き出し、ゲーム開始イベント 5 0 2 に対応するゲーム・ストリーム 5 0 4 の暗示サブスクライバー処理エージェント 1 0 2 を識別することができる。暗示有効化テーブル 5 0 6 に基づいて、ランタイム・ストリーム処理レイヤー 1 2 4 は、ゲーム・ストリーム 5 0 4 の暗示サブスクライバー処理エージェント 1 0 2 が、マップ G 上におけるキル(kill)の数を計算するマップ処理エージェント 5 0 8、マップ H 上におけるキルの数を計算するマップ処理エージェント 5 1 0、およびエリート・プレーヤー C、M、および S に対するキルの数を計算するエリート・プレー処理エージェント 5 1 2 を含むことができる。次いで、ランタイム・ストリーム処理レイヤー 1 2 4 は、ゲーム開始イベント 5 0 2 に基づいて、ゲーム開始イベント情報 5 1 4 を生成し、識別した暗示サブスクライバー処理エージェント 5 0 8、5 1 0、および 5 1 2 への配信のために、それをゲーム・ストリーム 5 0 4 にプッシュすることができる。

20

30

【 0 0 6 6 】

[0071] 図 5 B に示す時点 T<sub>1</sub> において、ゲーム開始イベント情報 5 1 4 が、暗示サブスクライバー処理エージェント 5 0 8、5 1 0、および 5 1 2 によって配信され処理されていけばよい。処理エージェント 5 0 8 において、ゲーム開始イベント情報 5 1 4 において示されたゲーム X に関連するマップ H が、マップ処理エージェント 5 0 8 に関心のあるマップ G とは異なることを判定することができる。その結果、処理エージェント 5 0 8 は、受けたゲーム開始イベント情報 5 1 4 に関して何も行わなくてもよい。マップ処理エージェント 5 1 0 において、しかしながら、ゲーム開始イベント情報 5 1 4 に含まれるマップ H が、マップ処理エージェント 5 1 0 に関心のあるものであると判定することができ、したがって、マップ処理エージェント 5 1 0 は次いで、ゲーム X における全てのキル・イベントを受けるためにゲーム X キル・ストリーム 5 3 0 に登録申込みするために、登録申込み 5 2 0 を送ることができる。

40

【 0 0 6 7 】

[0072] 同様に、エリート・プレー処理エージェント 5 1 2 は、ゲーム X が、それが処理しているエリート・プレーヤーの内 1 つを含むと判定することができ、したがってゲーム X プレーヤー C キル・ストリーム 5 3 2 に登録申込みしてゲーム X におけるプレーヤー

50

Cによる全てのキル・イベントを受けるために、登録申込み524を提出することができる。尚、ゲーム開始イベント情報504が配信され処理エージェント508、510、および512において処理されている間に、プレイヤーCによるキル・イベント514、プレイヤーAに報いるための報償イベント516、およびプレイヤーFによるキル・イベント518のような、更に多くのイベントが待ち行列106において発行される可能性があることは注記してしかるべきである。

【0068】

[0073] 図5Cに示す時点 $T_2$ において、ランタイム・ストリーム処理レイヤー124は、マップ処理エージェント510およびエリート・プレイヤー処理エージェント512によって提出されたイベント登録申込みを処理することができる。登録申込み522はゲームXの開始以来全てのキル・イベントを要求するので、ランタイム・ストリーム処理レイヤー124は、このような登録申込みが、登録申込み522が受けられる前に発行された過去のキル・イベント514および518を含むと判定することができる。同様に、ランタイム・ストリーム処理レイヤー124は、登録申込み524が過去のキル・イベント514を含むと判定することもできる。したがって、ランタイム・ストリーム処理レイヤー124は、キル・イベント514および518を待ち行列106から引き出し、キル・イベント514および518をマップ処理エージェント510に、ゲームXキル・ストリーム530を介して送り、更にキル・イベント514をエリート・プレイヤー処理エージェント512に、ゲームXプレイヤーCキル・ストリーム532を介して送ることができる。

【0069】

[0074] 図5Dに示すように、時点 $T_3$ において、新たなキル・イベント528が待ち行列106において発行される。ランタイム・ストリーム処理レイヤー124は、イベント528を待ち行列106から引き出し、この新たなキル・イベント528がゲームXキル・ストリーム530に属すると判定することができる。次いで、キル・イベント528をゲームXキル・ストリーム530にプッシュし、マップ処理エージェント510に配信することができる。ランタイム・ストリーム処理レイヤー124は、更に多くのイベントが待ち行列106において同様に発行されるに連れて、これら进行处理し配信し続けることができる。

【0070】

[0075] 以上から、暗示サブスクライバー処理エージェント102が登録申込みしていないイベントについての情報を、暗示サブスクライバー処理エージェント102が受けることを可能にすることにより、システムを通過するイベントに基づいて、ストリーム処理エージェント102を自動的に有効化し、これによって非常に細かい(granular)ストリーム処理ユニットをサポートし、こうしてプログラミング・モデルを大幅に簡素化することができる。更に、ストリームの定義を、そのストリームを介して配信されるイベントのセマンティクス(semantics)に縮減することによって、処理エージェント102がストリーム・トポロジーを知る必要をなくすことができ、代わりに処理することを望む特定のイベントに集中し、これによって更にプログラミング・モデルを簡素化することができる。加えて、本明細書において紹介した技術を利用することによって、トポロジーをコンパイルし直すまたは配備し直す必要なく、複雑なイベント検出および集計ロジックのその場限りの定義をシステムに動的に適用することができる。

【0071】

[0076] 図6は、図1に示したコンポーネントを格納および実行することができる計算デバイス600のコンピューター・アーキテクチャ例を示す。図6に示すコンピューター・アーキテクチャは、従来のサーバー・コンピューター、ワークステーション、デスクトップ・コンピューター、ラップトップ、タブレット、ファブレット(phablet)、ネットワーク・アプライアンス、パーソナル・ディジタル・アシスタント(「PDA」)、電子書籍(e-reader)、ディジタル・セルラ・フォン、または他の計算デバイスを例示し、本明細書において紹介したソフトウェア・コンポーネントの内任意のものを実行するために利用することができる。例えば、図6に示すコンピューター・アーキテクチャは、先に説明し

たソフトウェア・コンポーネントの内任意のものを実行するために利用することもできる。

【 0 0 7 2 】

[0077] 計算デバイス 6 0 0 は、ベースボード 6 0 2 または「マザーボード」を含む。これは、複数のコンポーネントまたはデバイスをシステム・バスまたは他の電気通信経路を介して接続することができる印刷回路ボードである。1つの例示的な形態では、1つ以上の中央処理ユニット(「CPU」) 6 0 4 がチップセット 6 0 6 と共に動作する。CPU 6 0 4 は、計算デバイス 6 0 0 の動作に必要な算術および論理動作を実行する標準的なプログラマブル・プロセッサでよい。

【 0 0 7 3 】

[0078] CPU 6 0 4 は、1つの離散物理状態から次の状態に、これらの状態間で区別し(differentiate)、これらの状態を変化させるスイッチング・エレメントの操作によって遷移させることによって、動作を実行する。スイッチング・エレメントは、一般に、フリップ・フロップのような、2つの二進状態の内1つを維持する電子回路、および論理ゲートのような、1つ以上の他のスイッチング・エレメントの状態の論理組み合わせに基づいて、出力状態を与える電子回路を含むことができる。これらの基本的なスイッチング・エレメントを組み合わせると、レジスター、加算器・減算器、算術論理ユニット、浮動小数点ユニット等を含む、更に複雑な論理回路を作ることができる。

【 0 0 7 4 】

[0079] チップセット 6 0 6 は、CPU 6 0 4 と、ベースボード 6 0 2 上のコンポーネントおよびデバイスの残りのものとの間にインターフェースを設ける。チップセット 6 0 6 は、計算デバイス 6 0 0 において主メモリーとして使用されるRAM 6 0 8 へのインターフェースを設けることもできる。更に、チップセット 6 0 6 は、リード・オンリー・メモリー(「ROM」) 6 1 0 または基本ルーチンを格納するための不揮発性RAM(「NVRAM」)のようなコンピューター読み取り可能記憶媒体へのインターフェースも設けることができる。基本ルーチンは、計算デバイス 6 0 0 を起動し、種々のコンポーネントおよびデバイス間で情報を転送するのを補助する。また、ROM 6 0 1 またはNVRAM は、本明細書において説明した形態による計算デバイス 6 0 0 の動作に必要な他のソフトウェア・コンポーネントも格納することができる。

【 0 0 7 5 】

[0080] 計算デバイス 6 0 0 は、ローカル・エリア・ネットワーク 6 2 0 のようなネットワークを介して、リモート計算デバイスおよびコンピューター・システムへの論理接続を使用して、ネットワーク接続環境において動作することができる。チップセット 6 0 6 は、ギガバイト・イーサネット(登録商標)・アダプターのような、ネットワーク・インターフェース・コントローラー(NIC) 6 1 2 を介してネットワーク接続を設けるための機能を含むことができる。NIC 6 1 2 は、計算デバイス 6 0 0 を他の計算デバイスにネットワーク 6 2 0 を介して接続することができる。尚、複数のNICが計算デバイス 6 0 0 内に存在し、コンピューターを他のタイプのネットワークおよびリモート・コンピューター・システムに接続することも認められてしかるべきである。ローカル・エリア・ネットワーク 6 2 0 は、計算デバイス 6 0 0 が、リモート・サービスおよびリモート・コンピューター 6 5 0 のようなサーバーと通信することを可能にする。

【 0 0 7 6 】

[0081] 計算デバイス 6 0 0 は、計算デバイスに不揮発性ストレージを設ける大容量記憶デバイス 6 1 6 に接続することもできる。大容量記憶デバイス 6 1 6 は、システム・プログラム、アプリケーション・プログラム、他のプログラム・モジュール、およびデータを格納することができ、これについては本明細書において更に詳細に説明した。大容量記憶デバイス 6 1 6 は、チップセット 6 0 6 に接続されたストレージ・コントローラー 6 1 4 を介して、計算デバイス 6 0 0 に接続することができる。大容量記憶デバイス 6 1 6 は、1つ以上の物理記憶ユニットで構成することができる。ストレージ・コントローラー 6 1 4 は、シリアル・アタッチドSCSI(「SAS」)インターフェース、シリアル高

10

20

30

40

50

度技術接続（「SATA」：serial advanced technology attachment）インターフェース、ファイバー・チャネル（「FC」）インターフェース、またはコンピュータと物理記憶ユニットとの間を物理的に接続しデータを転送するための他のタイプのインターフェースを介して、物理記憶ユニットとインターフェースすることができる。また、大容量記憶デバイス616、他の記憶媒体、およびストレージ・コントローラ614は、MultiMediaCard（MMC）コンポーネント、eMMCコンポーネント、セキュア・デジタル（SD）コンポーネント、PCI Expressコンポーネント等も含むことができることも認められてしかるべきである。

【0077】

[0082] 計算デバイス600は、格納される情報を反映するように物理記憶ユニットの物理状態を変換することによって、データを大容量記憶デバイス616に格納することができる。物理状態の具体的な変換は、この説明の異なる実現例では、種々の要因に依存する場合もある。このような要因の例には、物理記憶ユニットを実現するために使用される技術、大容量記憶デバイス616が主ストレージまたは副ストレージのどちらとして特徴付けられるか等を含むことができるが、これらに限定されるのではない。

【0078】

[0083] 例えば、計算デバイス600は、ストレージ・コントローラ614を介して、磁気ディスク・ドライブ・ユニット内の特定の位置の磁気特性、光記憶ユニットにおける特定の位置の反射または屈折特性、あるいはソリッド・ステート記憶ユニットにおける特定のキャパシター、トランジスター、または他のディスクリット・コンポーネントの電気的特性を変化させる命令を発行することによって、情報を大容量記憶デバイス616に格納することができる。物理媒体の他の変換も、本説明の範囲および主旨から逸脱することなく、可能であり、以上の例はこの説明を容易にするために示されたに過ぎない。更に、計算デバイス600は、物理記憶ユニット内の1つ以上の特定の位置の物理状態または特性を検出することによって、大容量記憶デバイス616から情報を読み出すこともできる。

【0079】

[0084] 以上で説明した大容量記憶デバイス616に加えて、計算デバイス600は、プログラム・モジュール、データ構造、または他のデータのような情報を格納し引き出すために、他のコンピュータ読み取り可能記憶媒体にアクセスすることができる。つまり、ランタイム・システム108および他のモジュールは大容量記憶デバイス616に格納されたデータおよびソフトウェアとして表されているが、ランタイム・システム108および/または他のモジュールは、少なくとも部分的に、デバイス600の他のコンピュータ読み取り可能記憶媒体に格納されてもよいことは認められてしかるべきである。本明細書に含まれるコンピュータ読み取り可能媒体の説明は、ソリッド・ステート・ドライブ、ハード・ディスク、またはCD-ROMドライブのような大容量記憶デバイスに言及するが、コンピュータ読み取り可能媒体は、計算デバイス600によってアクセスすることができる任意の入手可能なコンピュータ記憶媒体または通信媒体とすることができることは、当業者によって認められてしかるべきである。

【0080】

[0085] 通信媒体は、コンピュータ読み取り可能命令、データ構造、プログラム・モジュール、または他のデータを、搬送波のような変調データ信号、または他の移送メカニズムに含み、任意の配信媒体を含む。「変調データ信号」という用語は、信号内に情報をエンコードするようにその特性の内1つ以上が変更または設定された信号を意味する。一例として、そして限定ではなく、通信媒体は、有線ネットワークまたは直接有線接続のような有線媒体、ならびに音響、RF、赤外線、および他のワイヤレス媒体のようなワイヤレス媒体を含む。以上の内任意のものの組み合わせも、コンピュータ読み取り可能媒体の範囲に含まれてしかるべきである。

【0081】

[0086] 一例として、そして限定ではなく、コンピュータ記憶媒体は、コンピュータ

10

20

30

40

50

ー読み取り可能命令、データ構造、プログラム・モジュール、または他のデータのような情報の格納のための任意の方法または技術で実現される揮発性および不揮発性、リムーバブルおよび非リムーバブル媒体を含むことができる。例えば、コンピュータ媒体は、RAM、ROM、EPROM、EEPROM、フラッシュ・メモリーまたは他のソリッド・ステート・メモリー技術、CD-ROM、デジタル・バーサタイル・ディスク(「DVD」)、HD-DVD、BLU-RAY、または他の光ストレージ、磁気カセット、磁気テープ、磁気ディスク・ストレージまたは他の磁気記憶デバイス、あるいは所望の情報を格納するために使用することができ計算デバイス600によってアクセスすることができる任意の他の媒体を含むが、これらに限定されるのではない。請求項に限って言えば、「コンピュータ記憶媒体」、「コンピュータ読み取り可能記憶媒体」という句、およびこれらの異形は、波や信号自体、および/または通信媒体を含まない。

10

#### 【0082】

[0087] 大容量記憶デバイス616は、計算デバイス600の動作を制御するために利用されるオペレーティング・システム622を格納することができる。1つの形態によれば、オペレーティング・システムはLINUXオペレーティング・システムを含む。他の形態によれば、オペレーティング・システムは、MICROSOFT CorporationからのWINDOWS(登録商標)オペレーティング・システムを含む。更に他の形態によれば、オペレーティング・システムは、UNIX(登録商標)、Android、Windows Phone、またはiOSオペレーティング・システムを含んでもよい。尚、他のオペレーティング・システムも利用してもよいことは認められてしかるべきである。大容量記憶デバイス616は、ランタイム・システム108および/または以上で説明した他のソフトウェア・コンポーネントおよびデータの内任意のもののような、計算デバイス600によって利用される他のシステムまたはアプリケーション・プログラム、およびデータも格納することができる。また、大容量記憶デバイス616は、本明細書では具体的に識別されなかった他のプログラムやデータも格納することができる。

20

#### 【0083】

[0088] 1つの形態では、大容量記憶デバイス616または他のコンピュータ読み取り可能記憶媒体には、コンピュータ実行可能命令がエンコードされる。コンピュータ実行可能命令は、計算デバイス600内にロードされると、コンピュータを汎用計算システムから、本明細書において説明した形態を実現することができる特殊目的コンピュータに変換する。これらのコンピュータ実行可能命令は、以上で説明したように、CPU604がどのように状態間を遷移するか指定することによって、計算デバイス600を変換する。1つの形態によれば、計算デバイス600は、コンピュータ実行可能命令を格納するコンピュータ読み取り可能記憶媒体にアクセスすることができ、コンピュータ実行可能命令が計算デバイス600によって実行されると、図3および図4に関して先に説明した種々のルーチンを実行する。また、計算デバイス600は、本明細書において説明した他のコンピュータ実装動作の内任意のものを実行するために、コンピュータ読み取り可能記憶媒体を含むこともできる。

30

#### 【0084】

[0089] また、計算デバイス600は、入力デバイス619から入力を受けて処理する1つ以上の入力/出力コントローラ617も含むことができる。入力デバイス619は、キーボード、マウス、マイクロフォン、ヘッドセット、タッチパッド、タッチ・スクリーン、電子スタイラス、または任意の他のタイプの入力デバイスのような、複数の入力デバイスを含むことができる。同様に、入力/出力コントローラ617は、コンピュータ・モニター、フラット・パネル・ディスプレイのようなディスプレイ、デジタル・プロジェクター、プリンター、プロッター、または他のタイプの出力デバイスに出力を供給することができる。尚、計算デバイス600は図6に示すコンポーネントの全てを含まなくてもよく、図6には明示的に示されない他のコンポーネントを含むこともでき、または図6に示すものとは完全に異なるアーキテクチャを利用してもよいことは認められてしかるべきである。

40

50

## 【 0 0 8 5 】

[0090] 本明細書において紹介した開示は、以下の節において検討することもできる。

[0091] 節 1 : イベントを配信するためのコンピューター実装方法であって、この方法は、イベントの発行を受けるステップと、処理エージェントの構成に基づいて、イベントに関連する情報を受けるために処理エージェントを識別するステップであって、処理エージェントがイベントを受けることを明示的に登録申込みしない、ステップと、処理エージェントが有効化されているか否か判定するステップと、処理エージェントが有効化されていないと判定したことに応答して、処理エージェントを有効化するステップと、イベントに関連する情報を処理エージェントに配信するステップと、処理エージェントから 1 つ以上のイベントへの登録申込みを受けるステップと、1 つ以上のイベントを処理エージェントに配信するステップとを含む。

10

## 【 0 0 8 6 】

[0092] 節 2 : 節 1 の方法であって、更に、登録申込みが、1 つ以上の過去のイベントが配信されることを指定するか否か判定するステップと、登録申込みが 1 つ以上の過去のイベントが配信されることを指定すると判定したことに応答して、1 つ以上の過去のイベントを処理エージェントに配信するステップとを含む。

## 【 0 0 8 7 】

[0093] 節 3 : 節 1 および 2 の方法において、待ち行列においてイベントの発行を受けるステップが、イベントを待ち行列において発行する要求を受けるステップと、イベントに対応する仮想ストリームを有効化するステップと、仮想ストリームを介して待ち行列においてイベントを発行するステップとを含む。

20

## 【 0 0 8 8 】

[0094] 節 4 : 節 1 ~ 3 の方法において、1 つ以上のイベントに対する登録申込みが仮想ストリームを定め、登録申込みが、1 つ以上のイベントに対応する仮想ストリームに対する登録申込みである。

## 【 0 0 8 9 】

[0095] 節 5 : 節 1 ~ 4 の方法において、処理エージェントを有効化するステップが、システム・リソースを処理エージェントに割り当てるステップを含み、仮想ストリームに対する登録申込みが仮想ストリームを有効化し、システム・リソースを仮想ストリームに割り当てさせる。

30

## 【 0 0 9 0 】

[0096] 節 6 : 節 1 ~ 5 の方法であって、更に、一定の時間期間にわたって仮想ストリームにおいてイベントが受けられないとき、仮想ストリームに割り当てられたシステム・リソースを取り戻すことによって、仮想ストリームを無効化するステップと、処理エージェントが所与の時間期間にわたってアイドルになったとき、処理エージェントに割り当てられたシステム・リソースを取り戻すことによって、処理エージェントを無効化するステップとを含む。

## 【 0 0 9 1 】

[0097] 節 7 : 節 1 ~ 6 の方法において、処理エージェントの構成が、プログラミングによって得られるか、または処理エージェントの宣言的定義から決定される。

40

[0098] 節 8 : 節 1 ~ 7 の方法において、処理エージェントが有効化されているかまたは有効化されていないかには関係なく、処理エージェントがイベントを配信するためにアドレス可能である。

## 【 0 0 9 2 】

[0099] 節 9 : コンピューター実行可能命令が格納されたコンピューター読み取り可能記憶媒体であって、コンピューター実行可能命令がコンピューターによって実行されると、このコンピューターに、待ち行列において発行されているイベントを受けるために、このイベントに明示的にイベントに登録申込みしていない処理エージェントを決定させ、このイベントに対応する第 1 仮想ストリームを介して、このイベントが待ち行列において発

50



行されており、処理エージェントが有効化されているか否か判定させ、処理エージェントが有効化されていないと判定したことに応答して、処理エージェントを有効化させ、イベントのイベント情報を処理エージェントに配信させ、明示的な登録申込みによって定められる第2仮想ストリームに対する明示登録申込みを処理させ、明示登録申込みが、処理エージェントに配信されたイベント情報に応答して、処理エージェントによって提出され、第2仮想ストリームにおけるイベントを処理エージェントに配信させる。

【0093】

[00100] 節10：節9のコンピューター読み取り可能記憶媒体において、待ち行列において発行されるイベントが永続化される。

[00101] 節11：節9～10のコンピューター読み取り可能記憶媒体であって、更に、コンピューター実行可能命令を含み、コンピューターによって実行されると、このコンピューターに、第2仮想ストリームが、待ち行列において発行された過去のイベントを含むか否か判定させ、第2仮想ストリームが待ち行列において発行された過去のイベントを含むと判定したことに応答して、過去のイベントを第2仮想ストリームにプッシュさせ、処理エージェントに配信させる。

【0094】

[00102] 節12：節9～11のコンピューター読み取り可能記憶媒体であって、更に、コンピューター実行可能命令を含み、コンピューターによって実行されると、このコンピューターに、一定の時間期間にわたって第2仮想ストリームにおいてイベントが受けられないとき、この仮想ストリームに割り当てられたシステム・リソースを取り戻すことによって、第2仮想ストリームを無効化させ、所与の時間期間にわたって処理エージェントがアイドルになったとき、この仮想エージェントに割り当てられたシステム・リソースを取り戻すことによって、この処理エージェントを無効化させる。

【0095】

[00103] 節13：節9～12のコンピューター読み取り可能記憶媒体において、処理エージェントを有効化することが、システム・リソースをこの処理エージェントに割り当てることを含み、第2仮想ストリームに対する登録申込みが、第2仮想ストリームを有効化し、システム・リソースを第2仮想ストリームに割り当てさせる。

【0096】

[00104] 節14：節9～13のコンピューター読み取り可能記憶媒体において、処理エージェントが有効化されているかまたは有効化されていないかには関係なく、処理エージェントが、イベントを配信するためにアドレス可能である。

【0097】

[00105] 節15：節9～14のコンピューター読み取り可能記憶媒体において、処理エージェントがプログラミングによって得られるか、または処理エージェントの宣言的定義に基づいて決定される。

【0098】

[00106] 節16：ランタイムを実行する1つ以上の計算デバイスを含むシステムであって、このランタイムが、イベントの発行を受けて永続化する待ち行列を管理し、第1仮想ストリームにおいてイベントの発行を受け、イベントの発行を待ち行列に転送し、処理エージェントの構成に基づいて、イベントのイベント情報を配信するために処理エージェントを識別し、処理エージェントがイベントを受けを明示的に登録申込みせず、処理エージェントが有効化されているかまたは有効化されていないかには関係なく、イベント情報を配信するためにアドレス可能であり、処理エージェントが有効化されているか否か判定し、処理エージェントが有効化されていないと判定したことに応答して、処理エージェントを有効化し、イベント情報を処理エージェントに配信し、イベント情報が処理エージェントに配信された後に、処理エージェントから第2仮想ストリームにおいてイベントに対する明示的登録申込みを受け、明示的登録申込みが、当該明示的登録申込みが受けられる前に、待ち行列において発行された過去のイベントを指定するか否か判定し、明示的登録申込みが過去のイベントを指定すると判定したことに応答して、待ち行列において

発行された過去のイベントから、過去のイベントを引き出し、過去のイベントを第2仮想ストリームにプッシュし、第2仮想ストリームにおけるイベントを処理エージェントに配信するように構成される。

【0099】

[00107] 節17：節16のシステムにおいて、処理エージェントを有効化する動作が、システム・リソースを処理エージェントに割り当てる動作を含み、第2仮想ストリームに対する登録申込みが、第2仮想ストリームを有効化し、第2仮想ストリームにシステム・リソースを割り当てさせる。

【0100】

[00108] 節18：節16～17のシステムであって、更に、1つ以上の処理エージェントを実行する複数のサーバーを含み、ランタイムが、更に、処理エージェントが有効化された1つのサーバーの障害後に、他のサーバーにおいて処理エージェントを再有効化することによって、処理エージェントを復元するように構成される。

10

【0101】

[00109] 節19：節16～18のシステムにおいて、処理エージェントの構成が、プログラミングによって得られるか、または処理エージェントの宣言的定義から決定される。

【0102】

[00110] 節20：節16～19のシステムにおいて、ランタイムが、更に、一定の時間期間にわたって第2仮想ストリームにおいてイベントが受けられないとき、この仮想ストリームに割り当てられたシステム・リソースを取り戻すことによって、第2仮想ストリームを無効化し、所与の時間期間にわたって処理エージェントがアイドルになったとき、この仮想エージェントに割り当てられたシステム・リソースを取り戻すことによって、この処理エージェントを無効化するように構成される。

20

【0103】

[00111] 以上に基づいて、イベントを登録申込みし、受け、処理するための概念および技術が本明細書において紹介されたことが認められるはずである。本明細書において紹介した主題は、コンピューターの構造的特徴、方法論的アクト、およびコンピューター読み取り可能媒体に特定の文言で記載されたが、添付する特許請求の範囲において定められる発明は、必ずしも本明細書において説明した具体的な特徴、アクト、または媒体に限定されないことは理解されよう。逆に、具体的な特徴、アクト、および媒体は、特許請求の範囲を実現する形態例として開示されたまでである。

30

【0104】

[00112] 以上で説明した主題は、例示として提示したに過ぎず、限定として解釈すべきではない。本明細書において説明した主題には、例示し説明した形態例およびアプリケーション例に従うことなく、そして以下の特許請求の範囲において説明される、本発明の真の主旨および範囲から逸脱することなく、種々の変更(modification)および変化(change)が行われてもよい。

【図 1】

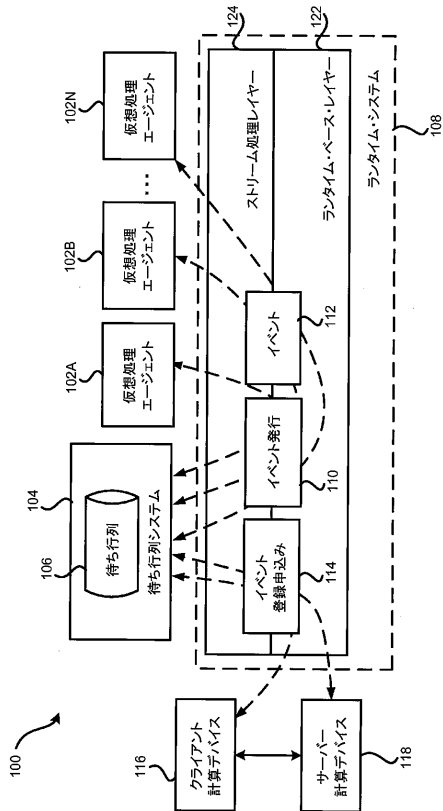


FIG. 1

【図 2】

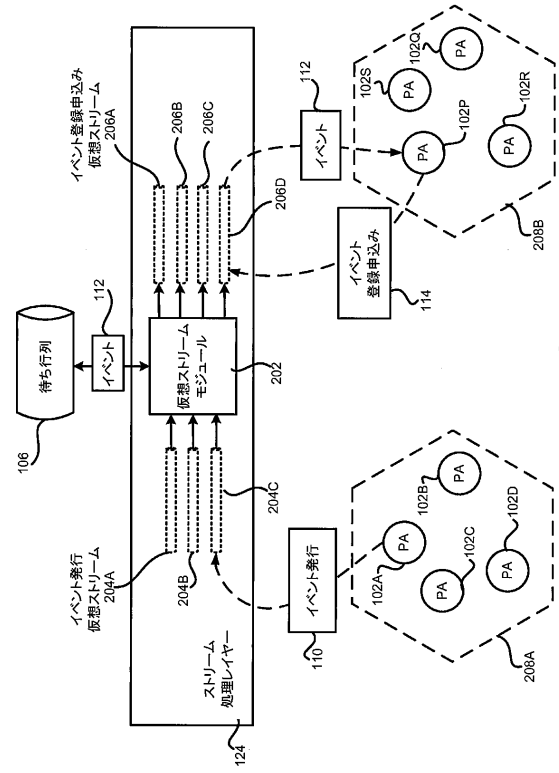


FIG. 2

【図 3】

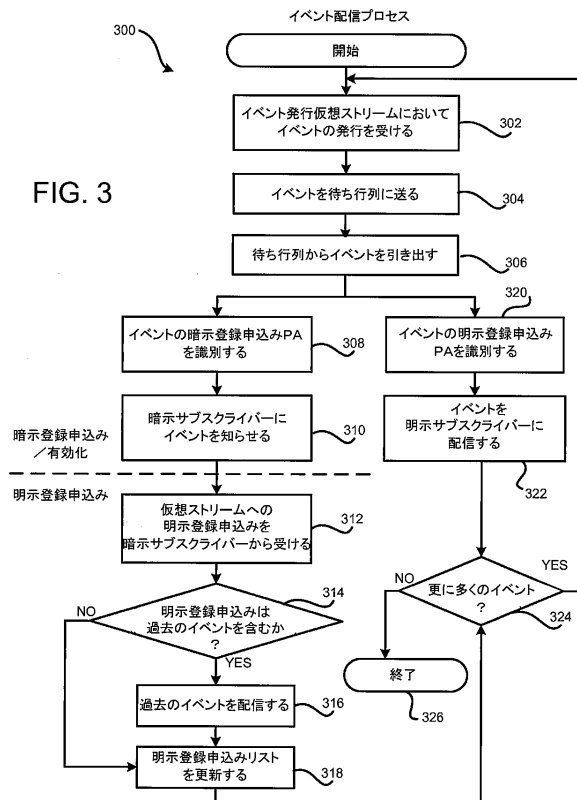


FIG. 3

【図 4】

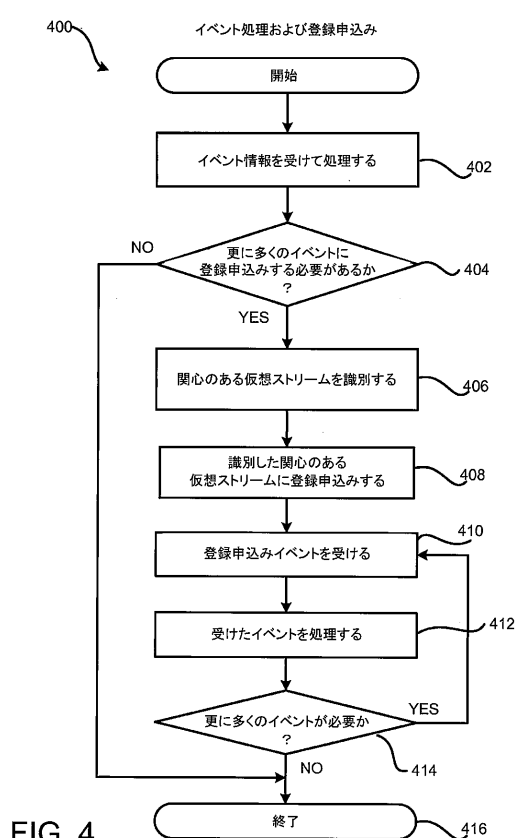
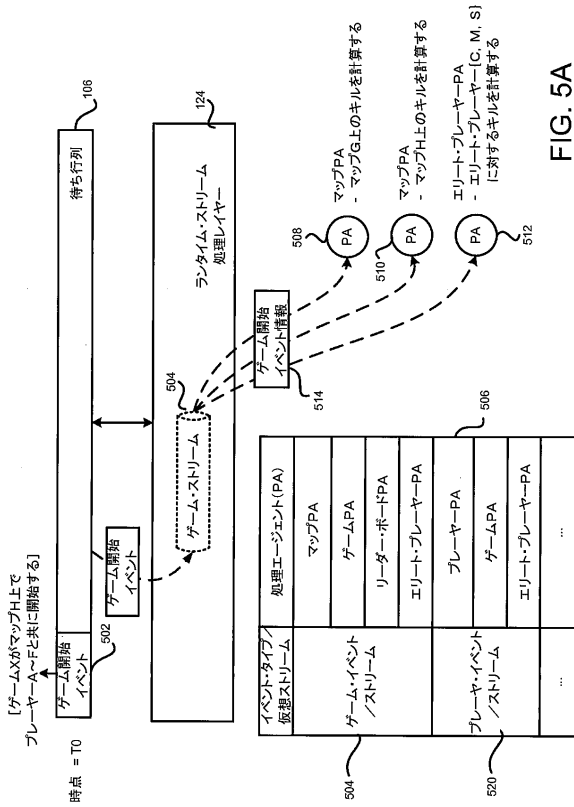
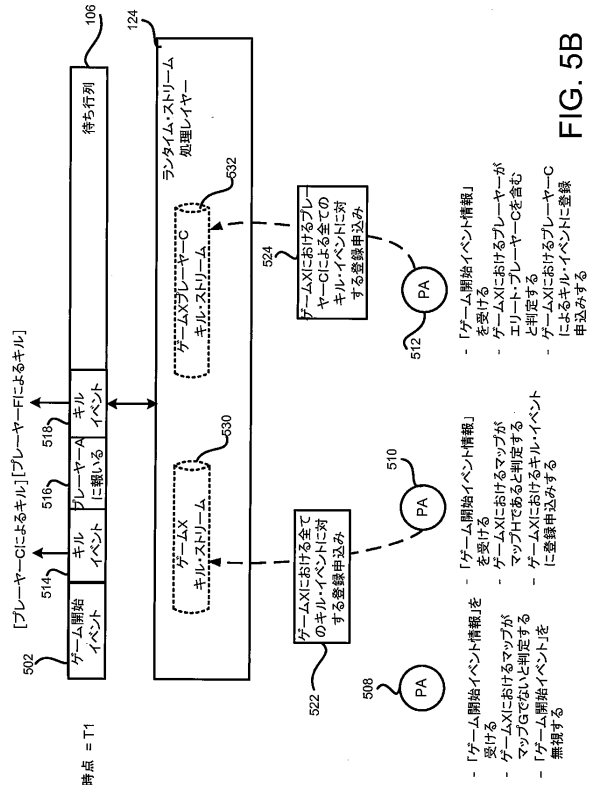


FIG. 4

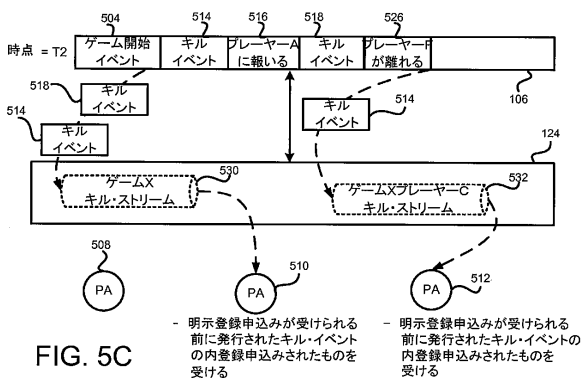
【図 5 A】



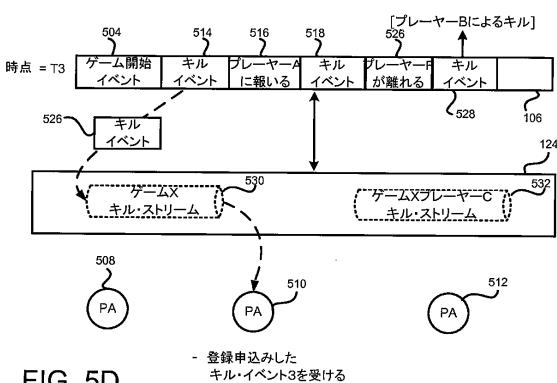
【図 5 B】



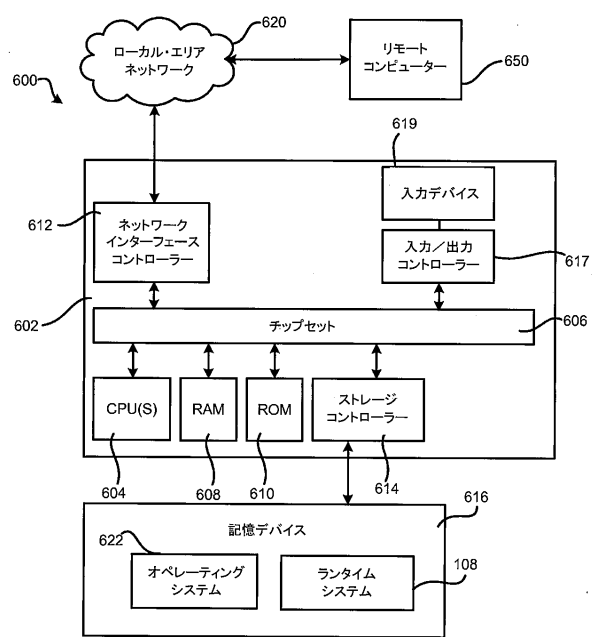
【図 5 C】



【図 5 D】



【図 6】



## フロントページの続き

- (72)発明者 ソムアー, ヘンリー・フーパー  
アメリカ合衆国 ワシントン州 98052 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 テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ(8/1172)
- (72)発明者 バイコフ, セルゲイ  
アメリカ合衆国 ワシントン州 98052 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 テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ(8/1172)
- (72)発明者 メラメド, タミル  
アメリカ合衆国 ワシントン州 98052 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 テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ(8/1172)
- (72)発明者 ロディ, ロバート・ルイス  
アメリカ合衆国 ワシントン州 98052 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 テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ(8/1172)
- (72)発明者 チャン, フェリックス  
アメリカ合衆国 ワシントン州 98052 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 テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ(8/1172)
- (72)発明者 マリウク, マイケル・ウィリアム  
アメリカ合衆国 ワシントン州 98052 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 テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ(8/1172)
- (72)発明者 ヘスキー, アンドリュー・アレクサンダー  
アメリカ合衆国 ワシントン州 98052 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 テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ(8/1172)
- (72)発明者 クリオット, ガブリエル  
アメリカ合衆国 ワシントン州 98052 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 テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ(8/1172)
- (72)発明者 テリン, ヨルゲン  
アメリカ合衆国 ワシントン州 98052 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 テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ(8/1172)
- (72)発明者 ゲラー, アラン・スチュアート  
アメリカ合衆国 ワシントン州 98052 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 テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ(8/1172)

審査官 大桃 由紀雄

- (56)参考文献 米国特許出願公開第2008/0209441(US, A1)  
米国特許出願公開第2007/0117635(US, A1)  
米国特許出願公開第2009/0125550(US, A1)

(58)調査した分野(Int.Cl., DB名)

G 0 6 F      9 / 5 4