(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6529505号
(P6529505)
(45)発行日　令和1年6月12日(2019.6.12)　(24)登録日　令和1年5月24日(2019.5.24)

(51)Int.Cl.　F I
G06F 8/41 (2018.01)　G06F 8/41

請求項の数 8　(全 22 頁)

(21)出願番号	特願2016-549189(P2016-549189)
(86)(22)出願日	平成26年10月14日(2014.10.14)
(65)公表番号	特表2016-537755(P2016-537755A)
(43)公表日	平成28年12月1日(2016.12.1)
(86)国際出願番号	PCT/US2014/060319
(87)国際公開番号	WO2015/057592
(87)国際公開日	平成27年4月23日(2015.4.23)
審査請求日	平成29年9月1日(2017.9.1)
(31)優先権主張番号	14/058,082
(32)優先日	平成25年10月18日(2013.10.18)
(33)優先権主張国	米国(US)

(73)特許権者　314015767
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
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
(74)代理人　100079108
弁理士　稲葉　良幸
(74)代理人　100109346
弁理士　大貫　敏史
(74)代理人　100117189
弁理士　江口　昭彦
(74)代理人　100134120
弁理士　内藤　和彦

最終頁に続く

(54)【発明の名称】インタラクティブ開発環境からソフトウェアアーチファクトをインクリメンタルにコンパイルすること

(57)【特許請求の範囲】
【請求項１】
コンピュータシステムにおいてコードをコンパイルするためのコンパイルシステムであって、
コンパイルされた識別されたコードを格納するコンパイルキャッシュと、
開発環境からコンパイル要求を受け取るインクリメンタルコンパイルコンポーネントであり、前記開発環境において前記コンピュータシステムに対して行われた変更に基づいて、および前記コンパイルキャッシュを検索して、コンパイルされることになるいずれかのコードが、コンパイルされた識別されたコードとして格納されているかどうかを識別することによって、カスタマイズまたは開発されたコンパイルされることになるコードを、それらがコンパイルされていない限り、またはコンパイルされたバージョンが前記コンパイルキャッシュ内に格納されていない限り、識別し、前記識別されたコードをコンパイルし、それらの識別されたコードをランタイム環境にとってアクセス可能にする、インクリメンタルコンパイルコンポーネントと、
前記コンパイルシステムの機能部分であるコンピュータプロセッサであり、前記インクリメンタルコンパイルコンポーネントによってアクティブ化されて、コンパイルされることになる前記コードを識別し、前記コンパイルキャッシュを検索し、および前記識別されたコードをコンパイルする、コンピュータプロセッサとを含み、
前記コンパイルシステムのメモリ使用およびオペレーションを管理する管理コンポーネントをさらに含み

前記インクリメンタルコンパイルコンポーネントが、前記識別されたコードをコンパイルして、前記コンパイルされた識別されたコードを入手し、前記コンパイルされた識別されたコードを前記コンパイルキャッシュ内に格納し、
前記管理コンポーネントが、前記コンパイルシステムからのメモリ使用がしきい値レベルを超えるかどうかを断続的に決定し、そうである場合には、前記コンパイルキャッシュの少なくとも一部を無効にし、
前記管理コンポーネントが、前記コンパイルシステムが所定のしきい値時間にわたってアイドルであるかどうかを断続的に決定し、そうである場合には、前記コンパイルシステムをシャットダウンする、コンパイルシステム。
【請求項２】
前記管理コンポーネントが、いずれかの開発環境インスタンスが稼働しているかどうかを断続的に決定し、そうでない場合には、前記コンパイルシステムをシャットダウンする、請求項１に記載のコンパイルシステム。
【請求項３】
前記コンパイルキャッシュ、前記インクリメンタルコンパイルコンポーネント、および前記コンピュータプロセッサが、統合開発環境（IDE）の一部である、請求項１に記載のコンパイルシステム。
【請求項４】
前記ＩＤＥが、前記コードを作成およびカスタマイズする開発入力を受け取るように構成されている、請求項３に記載のコンパイルシステム。
【請求項５】
コンピュータ実施方法であって、
コードをコンパイルしたいという要求を受け取るステップと、
開発環境において行われたコンピュータシステムに対する変更に基づいて、およびコンパイルされてコンパイルキャッシュ内に格納されている識別されたコードに基づいて、コンパイルする前記コードを、それらがコンパイルされていない限り、またはコンパイルされたバージョンが前記コンパイルキャッシュ内に格納されていない限り、前記コンピュータシステム内のカスタマイズまたは開発されたコードのサブセットとして識別するステップと、
前記識別されたコードをコンパイルして、コンパイルされた識別されたコードを入手するステップとを含み、
前記コンパイルされた識別されたコードを前記コンパイルキャッシュ内に格納するステップと、
前記コンパイルされた識別されたコードを、稼働されるランタイム環境へ送信するステップと、
要求を受け取る前記ステップ、前記コードを識別する前記ステップ、前記識別されたコードをコンパイルする前記ステップ、および前記コンパイルされた識別されたコードを格納する前記ステップを繰り返すステップと、
前記コンパイルキャッシュのメモリ使用がしきい値を超えるかどうかを断続的に決定し、そうである場合には、前記コンパイルキャッシュの少なくとも一部を無効にするステップと、
コンパイルエージェントがしきい値時間にわたってアイドルであるかどうかを断続的に決定し、そうである場合には、前記コンパイルエージェントをシャットダウンするステップと
をさらに含む、コンピュータ実施方法。
【請求項６】
コンパイル要求を受け取る前記ステップ、前記コードを識別する前記ステップ、およびコンパイルする前記ステップが、コンパイルエージェントによって実行され、前記コンピュータ実施方法が、
前記開発環境のいずれかのインスタンスが稼働しているかどうかを断続的に決定し、そ

うでない場合には、前記コンパイルエージェントをシャットダウンするステップをさらに含む、請求項５に記載のコンピュータ実施方法。
【請求項７】
前記コードを開発する前記開発環境において開発入力を受け取るステップと、
コンパイル可能なコードを前記開発入力に基づいて作成またはカスタマイズするステップと
をさらに含む、請求項５に記載のコンピュータ実施方法。
【請求項８】
コンピュータ実行可能命令を格納しているコンピュータ可読ストレージメディアであって、前記コンピュータ実行可能命令が、コンピュータによって実行されたときに、
コンピュータシステムにおいてコードをコンパイルしたいという要求を受け取るステップと、
開発環境において行われたコンピュータシステムに対する変更に基づいて、およびコンパイルされてコンパイルキャッシュ内に格納されている識別されたコードに基づいて、コンパイルする前記コードを、それらがコンパイルされていない限り、またはコンパイルされたバージョンが前記コンパイルキャッシュ内に格納されていない限り、前記コンピュータシステム内のカスタマイズまたは開発されたコードのサブセットとして識別するステップと、
前記識別されたコードをコンパイルして、コンパイルされた識別されたコードを入手するステップと、
前記コンパイルされた識別されたコードを前記コンパイルキャッシュ内に格納するステップと、
前記コンパイルされた識別されたコードを、稼働されるランタイム環境へ送信するステップと、
要求を受け取る前記ステップ、前記コードを識別する前記ステップ、前記識別されたコードをコンパイルする前記ステップ、および前記コンパイルされた識別されたコードを格納する前記ステップを繰り返すステップと、
前記コンパイルキャッシュのメモリ使用がしきい値を超えるかどうかを断続的に決定し、そうである場合には、前記コンパイルキャッシュの少なくとも一部を無効にするステップと、
コンパイルエージェントがしきい値時間にわたってアイドルであるかどうかを断続的に決定し、そうである場合には、前記コンパイルエージェントをシャットダウンするステップとを含む方法を前記コンピュータに実行させる、コンピュータ可読ストレージメディア。
【発明の詳細な説明】
【背景技術】
【０００１】
[0001] 多くのソフトウェア開発者は、ソフトウェアを開発するためにインタラクティブ開発環境（IDE）を使用する。それらの開発者は、コンピュータシステム内のタイプのモデルを開発するために、およびそれらのモデルをカスタマイズするためにＩＤＥを使用する。
【０００２】
[0002] 例示的な統合開発環境は、例示的には、開発する必要があるコードを開発者が開発およびテストすることができるように、およびコンピュータシステムを要望に応じてカスタマイズするために複数の異なるツールを含む開発ツールまたは開発環境である。例として、ＩＤＥは、コンピュータプログラマがソフトウェアを開発することを可能にするソースコードエディタ、１つまたは複数のビルドオートメーションツール、およびデバッガを含むことができる。いくつかのＩＤＥは、例示的には、コンパイラ、インタープリタ、またはそれらの両方を含む。それらは、グラフィカルユーザインターフェースの構築を簡略化するためにバージョンコントロールシステムおよびさまざまなツールを含むことがで

きる。それらは、オブジェクト指向ソフトウェア開発に伴って使用するためにクラスブラウザ、オブジェクトブラウザ、およびクラス階層図を含むこともできる。したがって開発者は、ＩＤＥを使用して、コードおよびメタデータを、コードおよびメタデータに対するカスタマイゼーションとともに生成することができ、それらは、所与の組織において使用するためのシステムを開発する際に利用することができる。

【０００３】

[0003] ＩＤＥを使用してソフトウェアを生成またはカスタマイズする際に、アプリケーション開発者は、アプリケーション内で特定のコンセプト（タイプとも呼ばれる）をモデル化し、必要な場合には、コードを書く。開発者がしばしばＩＤＥを使用する大きなアプリケーションは、数千の異なるタイプを含むことがある。したがって、これらのタイプを開発およびカスタマイズすることは、比較的大きなタスクである。

【０００４】

[0004] 例として、いくつかのコンピュータシステムは、数ある中でも、エンタープライズリソースプランニング（ERP）システム、顧客関係管理（CRM）システム、基幹業務（LOB）システムなどのビジネスシステムを含む。これらのタイプのコンピュータシステムはしばしば、モデル化およびカスタマイズされる何千もの異なるタイプを有する。例として、いくつかのそのようなビジネスシステムはしばしば、多くのその他のタイプは言うまでもなく、単独で、数千の異なる形態を有する。

【０００５】

[0005] ビジネスシステムは、多数のタイプを有するコンピュータシステムの唯一のタイプではない。たとえば、ゲーミングシステム、またはさまざまなその他のタイプのシステムもしばしば、ソフトウェアシステムにおいてモデル化される何千もの異なるタイプを有する。

【発明の概要】

【発明が解決しようとする課題】

【０００６】

[0006] そのようなコンピュータシステムは、インタープリットされたコードまたはコンパイルされたコードによって表すことができる。開発者は、コンパイルされたコードを開発またはカスタマイズしている際にしばしば、その開発またはカスタマイゼーション上で作動を行いたいと望み、次いで、そのコードを稼働させて、それが自分の要望どおりに作動しているということを確かめる。コンパイルされたコード環境においては、これは、新たに開発またはカスタマイズされたコードを用いてシステムを稼働させるためにコンパイラがシステム全体を再コンパイルすることを必要としてきた。その理由は、現在、コンパイラは主として、コンパイルが必要とされる場合に稼働する実行ファイルにすぎないということである。コンパイルが完了したときに、コンパイラは、自分のメモリと、自分が構築したあらゆるキャッシュとを失う。開発環境においては、これは、非常に時間を浪費することがあり、プロジェクトの開発フェーズに時間およびコストを加えることがあり、それは、開発者経験におけるフラストレーションにつながることもある。

【０００７】

[0007] 上述の論考は、一般的な背景情報のために提供されているにすぎず、特許請求される主題の範囲を決定する際の補助として使用されることを意図されているものではない。

【課題を解決するための手段】

【０００８】

[0008] インタラクティブ開発環境が、モデル化されたタイプを開発またはカスタマイズするための開発者入力を受け取る。コンパイルエージェントが、開発者が開発またはカスタマイズしているモデル化されたタイプをコンパイルしたいという要求をＩＤＥから受け取る。コンパイルエージェントは、コンパイラを長期的なサービスとしてホストし、コンパイラは、以前にコンパイルされたタイプのキャッシュを保持し、開発者によって行われた変更に基づいて、個別にロード可能なタイプのうちのどれが再コンパイルされることに

なるかを決定し、それらの識別されたタイプのみをコンパイルする。再コンパイルされたタイプも、キャッシュ内に格納される。

【０００９】

[0009] 上述の論考は、一般的な背景情報のために提供されているにすぎず、特許請求される主題の範囲を決定する際の補助として使用されることを意図されているものではない。

【００１０】

[0010] この「課題を解決するための手段」は、コンセプトのうちの選択されたものを、簡略化された形式で紹介するために提供されており、それらのコンセプトは、以降の「発明を実施するための形態」においてさらに説明されている。この「発明の概要」は、特許請求される主題の鍵となる特徴または必要不可欠な特徴を識別することを意図されているものではなく、特許請求される主題の範囲を決定する際の補助として使用されることを意図されているものでもない。特許請求される主題は、背景技術において記載されているあらゆるまたはすべての不利な点を解決する実施態様に限定されるものではない。

【図面の簡単な説明】

【００１１】

【図１】[0011] １つの例示的な開発アーキテクチャーのブロック図である。

【図２】[0012]開発中にコンパイルオペレーションを実行する際の図１において示されているアーキテクチャーのオペレーションの一実施形態を示す流れ図である。

【図３】[0013]図１において示されているコンパイルエージェントのための管理コンポーネントの、そのコンパイルエージェントのオペレーションを管理する際の一実施形態を示す流れ図である。

【図４Ａ】[0014]図１において示されているアーキテクチャーの、クラウドコンピューティングアーキテクチャーにおいて展開されるさまざまな実施形態を示す図である。

【図４Ｂ】図１において示されているアーキテクチャーの、クラウドコンピューティングアーキテクチャーにおいて展開されるさまざまな実施形態を示す図である。

【図５】[0015]モバイルデバイスのさまざまな実施形態を示す図である。

【図６】モバイルデバイスのさまざまな実施形態を示す図である。

【図７】モバイルデバイスのさまざまな実施形態を示す図である。

【図８】モバイルデバイスのさまざまな実施形態を示す図である。

【図９】モバイルデバイスのさまざまな実施形態を示す図である。

【図１０】[0016] １つの例示的なコンピューティング環境のブロック図である。

【発明を実施するための形態】

【００１２】

[0017] 図１は、開発アーキテクチャー１００の１つの例示的なブロック図を示している。図１は、開発アーキテクチャー１００が、インタラクティブ開発環境（IDE）１０２をコンパイルエージェント１０４およびメタデータ／コードストア１０６とともに含むということを示している。図１はまた、ＩＤＥ１０２およびコンパイルエージェント１０４が、ＩＤＥ１０２を使用して開発またはカスタマイズされたコンピュータシステムをホストするランタイム環境１０８と対話することができるということを示している。

【００１３】

[0018] 加えて図１は、開発者１１０が、例示的には、ランタイム環境１０８によってサービス提供されるコンピュータシステムにおいて稼働されるアプリケーション要素１１２（たとえば、タイプ）の開発またはカスタマイゼーションを実行するためにＩＤＥ１０２と対話するということを示している。アプリケーション要素のうちのそれぞれは、例示的にはメタデータ１１４を含み、コード１１６も含むことができる。図１はまた、ＩＤＥ１０２が、例示的には、プロセッサ１１８およびデータストア１２０を含むということを示している。

【００１４】

[0019] コンパイルエージェント１０４は、例示的には、ＩＤＥ１０２によって送信され

るコンパイル要求１２２において要求されるコンパイルオペレーションを実行する。コンパイルエージェント１０４は、例示的には、キャッシュ１２４、管理コンポーネント１２６、およびプロセッサ１２８を含む。コンパイルオペレーションが実行された後に、エージェント１０４は、例示的には、応答１３０をＩＤＥ１０２に提供することができる。これらのオペレーションのすべてについては、以降で図２および図３に関連してさらに詳細に論じる。

【００１５】

[0020] コンパイルエージェント１０４はまた、例示的には、コンパイルされたタイプ１３２をランタイム環境１０８に提供し、ランタイム環境１０８では、それらのコンパイルされたタイプ１３２をランタイム中の実行用としてデータストア１３４内に格納することができる。ランタイム環境１０８は、例示的には、アプリケーションサーバ１３６およびランタイムトランスレータ／ロケータ１３８を含む。ランタイムトランスレータ／ロケータ１３８は、例示的には、コンピュータシステムを稼働させるのに必要とされるアプリケーション要素に関するタイプを見つけて、それらのタイプを要望に応じてロードする。それらのタイプは、例示的には、データストア１３４からアプリケーションサーバ１３６内にロードされ、アプリケーションサーバ１３６では、それらのタイプをランタイム中に実行することができる。

【００１６】

[0021] メタデータ／コードストア１０６は、例示的には、さまざまな異なるタイプのアプリケーション要素（たとえば、タイプ）に対応するメタデータおよびコードを格納する。メタデータ／コードストア１０６は、たとえば、ＩＤＥ１０２およびコンパイルエージェント１０４によってアクセス可能である。

【００１７】

[0022] 図２は、ソフトウェアシステムを開発する際の図１において示されている開発アーキテクチャー１００のオペレーションの一実施形態を示す流れ図である。開発者１１０は、例示的には、ランタイム環境１０８によって使用されるアプリケーションサーバにおけるデバッギングまたは開発のためにアプリケーション要素を選択するために、ＩＤＥ１０２によって生成されたユーザインターフェース表示と対話するということがわかるであろう。開発者１１０は、別個の開発者デバイス（たとえば、パーソナルコンピュータ、タブレット、別のモバイルデバイスなど）を通じて、または直接ＩＤＥ１０２と対話することができる。開発者１１０は、ネットワークを介してＩＤＥ１０２と対話することもできる。開発者１１０は、あくまでも例示のために、図１においてはＩＤＥ１０２と直接対話しているところを示されている。

【００１８】

[0023] ＩＤＥ１０２は最初に、例示的には、開発者がモデル化またはカスタマイズしたいと望むアプリケーション要素（たとえば、コンパイル可能なタイプ）を識別する開発者入力を受け取る。これは、図２におけるブロック１５０によって示されている。それに応じて、ＩＤＥ１０２は、例示的には、識別されたアプリケーション要素のソースコード表示をメタデータ／コードストア１０６から、またはランタイム環境１０８から入手する。これは、図２におけるブロック１５２によって示されている。そのソースコード表示は、メタデータ１１４、コード１１６、またはその他の情報１１８も含むことができる。

【００１９】

[0024] 任意選択で、コンパイルエージェント１０４は、例示的には、開発オペレーションまたはカスタマイゼーションオペレーションが実行された後に、開発またはカスタマイズされたコードを開発者１１０が稼働させるためにロードされることになるすべての必要とされるアプリケーション要素（たとえば、コンパイル可能なタイプ）をプリロードするためにストア１０６からのメタデータおよびコードにアクセスすることもできる。エージェント１０４は、例示的には、それらのメタデータおよびコードを、個別にロード可能なタイプへとコンパイルし、それらのタイプをキャッシュ１２４内にキャッシュする。これは、図２におけるブロック１５４によって示されている。次いでＩＤＥ１０２は、識別さ

れたアプリケーション要素に関する開発者のカスタマイゼーション入力を受け取り、それによって開発者は、ＩＤＥ１０２を使用して、コンパイル可能なタイプを実際にカスタマイズまたは開発する。カスタマイゼーション入力を受け取ること、およびアプリケーション要素をカスタマイズすることは、図４におけるブロック１５６によって示されている。

【００２０】

[0025] どこかの時点で、開発者１１０は、既存のモデルに対して所望の数のカスタマイゼーションを行った後に、または複数のモデルをゼロから開発した後に、コードをコンパイルエージェント１０４によってコンパイルしたいと望む場合がある。これによって、開発者１１０は、新たに開発またはカスタマイズされたコードを稼働させて、そのコードが要望どおりに稼働するかどうかを見ることができるようになるであろう。コンパイルエージェント１０４は既に、データストア１０６からの複数のモデル化されたタイプにアクセスしたこと、およびそれらのタイプをキャッシュ１２４内にプリロードしたことがある可能性があるということを想起されたい。また、コンパイルエージェント１０４は、開発者１１０によって、以前に行われた変更、または以前に実行された開発に基づいてＩＤＥ１０２から受け取られていた複数のコンパイル可能なタイプを既にコンパイルしていた可能性がある。それらのコンパイルされたタイプは、既にキャッシュ１２４内にある可能性もある。したがって、コンパイルエージェント１０４は、ＩＤＥ１０２からコンパイル要求１２２を受け取る。一実施形態においては、コンパイルエージェント１０４は、サービス契約メソッド呼び出しを通じてコンパイル要求１２２においてコンパイルパラメータを受け取る。例として、明らかにされるサービス契約は、下記のテーブル１において示されているものと同様のものであることが可能である。

【００２１】

【表１】

表 1

```
[ServiceContract(ProtectionLevel=ProtectionLevel.None)]
internal interface ICompilationService
{
    /// <summary>
    /// コンパイルサービスが起動および稼働していて、要求を受け入れる用意ができている
    /// ということを確実にする
    /// </summary>
    [OperationContract]
    void EnsureServiceRunning();

    /// <summary>
    /// コンパイル要求を表す
    /// </summary>
    /// <param name="parameters">コンパイルパラメータ、コンパイルするのはどのクラスか、
どのﾓｼﾞｭｰﾙかなど</param>
    ///   <param   name="stdOutput">このコンパイルから生成される標準的な出力
          </param>
    ///    <param    name="stdError">このコンパイルによって生成される標準的なエラー
          </param>
    /// <returns></returns>
    [OperationContract]
    int Compile(Parameters parameters, out string stdOutput, out string stdError);
  }
```

【００２２】

[0026] コンパイルエージェント１０４はその他の方法でもコンパイル要求１２２を受け取ることができるということがわかるであろう。開発者１１０によってＩＤＥ１０２において行われた変更または開発を識別するコンパイル要求をコンパイルエージェント１０４に受け取らせることは、図２におけるブロック１５８によって示されている。

【００２３】

[0027] 次いでコンパイルエージェント１０４は、開発者１１０によって行われた開発およびカスタマイゼーションを開発者１１０によって稼働させることおよび点検することが可能になるようにコンパイルされることを必要とする要素（たとえば、モデル化されたタイプ）を識別するためにキャッシュ１２４にアクセスする。一実施形態においては、コンパイルエージェント１０４は、どの特定のモデル化されたタイプが開発者１１０によって変更されたか、またはどれが加えられたかを識別することによってこれを行い、それらのモデル化されたタイプのみを、および開発またはカスタマイズされたコードが稼働されるためにはコンパイルされなければならないその他の任意のモデル化されたタイプをコンパイルする。

【００２４】

[0028] コンパイルエージェント１０４は、キャッシュ１２４を検査して、コンパイルされることになるモデル化されたタイプのうちのいずれかが既にコンパイルされてキャッシュ１２４内に格納されているかどうかを決定する。それらのタイプが既にコンパイルされてキャッシュ１２４内に格納されている場合には、それらのタイプを再コンパイルする必要はない。これは、それらのモデル化されたタイプが、独立してロード可能なアセンブリとしてモデル化およびコンパイルされているためである。したがって、現在カスタマイズまたは開発されているモデル化されたタイプのみが、コンパイルされることを必要とし、その他のいかなるモデル化されたタイプも、既にコンパイルされている限り、およびコンパイルされたバージョンがキャッシュ１２４内に格納されている限り、再コンパイルされることを必要としない。コンパイルされることになる要素を識別することは、図２におけるブロック１６０によって示されている。

【００２５】

[0029] 次いでコンパイルエージェント１０４は、コンパイルされることを必要とする識別された要素のみに関するメタデータおよびコードをデータストア１０６からロードする。これは、図２におけるブロック１６２によって示されている。例として、ＩＤＥ１０２は、カスタマイズされたタイプおよび新たに開発されたタイプをコンパイルエージェント１０４によるアクセス用としてデータストア１０６内に格納することができる。エージェント１０４は、それらのタイプがコンパイルされなければならないということを識別した場合には、コンパイルを実行することができるようにデータストア１０６内のメタデータおよびコード（もしあれば）にアクセスする。

【００２６】

[0030] コンパイルエージェント１０４は、コンパイルを実行することを必要とされるメタデータおよびコードのすべてを有すると、識別された要素（たとえば、モデル化されたタイプ）を、個別にロード可能なタイプへとコンパイルし、それらのコンパイルされたタイプ１３２を、ランタイム環境１０８にとって利用可能にする。これは、図２におけるブロック１６４によって示されている。

【００２７】

[0031] 次いでコンパイルエージェント１０４は、新たにコンパイルされた要素をキャッシュ１２４内に格納し、コンパイルが完了しているということを示す応答１３０をＩＤＥ１０２へ送信する。これは、図２におけるブロック１６６によって示されている。その時点で、ランタイム環境１０８は、コンパイルされたばかりのコードを稼働させることができ、それによって開発者１１０は、コードが要望どおりに機能するかどうかを決定することができる。それを行う際に、ランタイムトランスレータ／ロケータ１３８は、コンパイルされたタイプ１３２を識別することができ、アプリケーションサーバ１３６は、それらのタイプ１３２をロードして稼働させることができる。

【００２８】

[0032] プロセス全体を通じて、管理コンポーネント１２６は、例示的には、エージェント１０４の自己管理を実行する。これについては、以降で図３に関連してさらに詳細に説明する。しかしながら、簡潔に言えば、管理コンポーネント１２６は、例示的には、コンパイル１０４によって使用されているメモリフットプリントをモニタする。メモリフットプリントがしきい値レベルに達した場合には、管理コンポーネント１２６は、キャッシュ１２４を無効にして、たとえば、メモリの大部分を解放することができる。同様に、コンパイルエージェント１０４が所定の量の時間にわたってアイドルである場合、またはＩＤＥインスタンス１０２が現在インスタンス化されていない場合には、エージェント１０４は、自分自身をシャットダウンすることができる。自己管理を実行することは、図２においてブロック１６８によって示されており、エージェント１０４自身をシャットダウンするかどうかを決定することは、ブロック１７０によって示されている。

【００２９】

[0033] ブロック１７０において、管理コンポーネント１２６がエージェント１０４をシャットダウンすべきであるということが決定された場合には、エージェント１０４はシャ

ットダウンされる。これは、図２におけるブロック１７２によって示されている。

【００３０】

[0034] 図３は、管理コンポーネント１２６のオペレーションの一実施形態をより詳細に示す流れ図である。ＩＤＥ１０２がコンパイル要求１２２をエージェント１０４に最初に提供したときに、エージェント１０４は、その時点で稼働している可能性があり、または稼働していない可能性もあるということに留意されたい。エージェント１０４が稼働していない場合には、管理コンポーネント１２６は、コンパイル要求１２２が受け取られているということを自動的に検知し、コンパイルエージェント１０４を起動する。ＩＤＥ１０２からの要求を受け取ること、およびエージェント１０４がまだ始動されていない場合にエージェント１０４をスタートアップすることは、図３におけるブロック１８０によって示されている。

【００３１】

[0035] 図２に関して上述されているように、次いでコンパイルエージェント１０４は、要求されたコンパイルオペレーションをキャッシングオペレーションとともに実行する。すなわち、コンパイルエージェント１０４は、（キャッシュ１２４を検索して、どのタイプが、コンパイルまたは再コンパイルされることを必要としているかを決定した後に）必要とされるコンパイル可能なタイプのうちのすべてをコンパイルし、それらの新たにコンパイルされたタイプをキャッシュ１２４内に格納する。これは、図３におけるブロック１８２によって示されている。

【００３２】

[0036] 管理コンポーネント１２６は、その内部で自分が稼働している（またはそれとともに自分が稼働している）オペレーティングシステムに対して断続的にクエリーを行って、コンパイルエージェント１０４のメモリ消費が所与のしきい値を超えているかどうかを決定する。これは、図３におけるブロック１８４によって示されている。コンパイルエージェント１０４のメモリ消費が所与のしきい値を超えている場合には、管理コンポーネント１２６は、キャッシュ１２４を無効にする、または解放する。すなわち、管理コンポーネント１２６は、キャッシングオペレーション（コンパイルされたタイプをキャッシュすること）を再び新たに開始する。これは、ブロック１８６によって示されている。もちろん、管理コンポーネント１２６は、コンパイルエージェント１０４のメモリフットプリントを減少させるためにその他のオペレーションを実行することもでき、キャッシュを無効にすることは、一例にすぎない。

【００３３】

[0037] ブロック１８４において、メモリ消費がしきい値を超えていないということが決定された場合に、またはブロック１８６においてキャッシュが無効にされた後に、管理コンポーネント１２６は、依然として稼働しているＩＤＥ１０２のインスタンスがいくらかでもあるかどうかを決定する。これは、図３におけるブロック１８８によって示されている。コンパイルエージェント１０４の単一のインスタンスは、たとえば、ＩＤＥ１０２の複数のインスタンスにサービス提供することができる。ＩＤＥ１０２のいずれのインスタンスも稼働していない場合には、コンパイルエージェント１０４が稼働している必要はなく、コンパイルエージェント１０４は、ブロック１９０によって示されているように自分自信をシャットダウンする。

【００３４】

[0038] しかしながら、ブロック１８８において、依然として稼働しているＩＤＥ１０２のインスタンスがあるということが決定された場合には、管理コンポーネント１２６は、コンパイルエージェント１０４がしきい値量の時間にわたってアイドルであるかどうかを決定する。これは、図３におけるブロック１９２によって示されている。例として、開発者１１０は、依然としてコードを書いているが、そのコードをまだデバッグしていない可能性がある。開発者１１０は、ひと休みしている、またはコードをコンパイルしたいと望む開発フェーズにいないだけである可能性もある。いずれのケースにおいても、ブロック１９２においてコンパイルエージェント１０４がしきい値量の時間にわたってアイドルで

ある場合には、管理コンポーネント１２６は、ブロック１９０によって示されているようにコンパイルエージェント１０４をシャットダウンする。しかしながら、ブロック１９２において、コンパイルエージェント１０４がしきい値量の時間にわたってアイドルではないということが決定された場合には、処理は、単にブロック１８０へ戻り、ブロック１８０では、エージェント１０４は、ＩＤＥ１０２からのさらなるコンパイル要求１２２を待つ。

【００３５】

[0039] したがって、コンパイルエージェント１０４は、コンパイル要求１２２を受け取った場合は常に、開発者１１０によって行われた変更に基づいて、およびどのタイプが既にコンパイルされてキャッシュ１２４内に格納されているかに基づいて、コンパイルまたは再コンパイルされることを必要としているコンパイル可能なタイプのみをコンパイルまたは再コンパイルするということがわかる。これによって、コードが適切に稼働するかどうかを見るために開発者１１０が待たなければならないコンパイル時間が著しく減少する。これによって、開発者経験が向上し、それによって、コードを開発、カスタマイズ、またはデバッグする際に必要とされる時間および労力を減少させることができる。

【００３６】

[0040] 上述の論考は、データストア１２０、データストア１０６、およびデータストア１３４を含む複数のデータストアを示しているということにも留意されたい。これらは、３つの独立したデータストアとして示されているが、単一のデータストア内に形成することもできる。加えて、それらのデータストア内のデータは、複数のさらなるデータストア内に格納することもできる。また、データストアは、それらのデータストアにアクセスする環境もしくはエージェントもしくはコンポーネントにとってローカルにあることが可能であり、またはそれらのデータストアは、そこからリモートにあること、およびそれらの環境、コンポーネント、もしくはエージェントによってアクセスできることが可能である。同様に、いくつかのデータストアは、ローカルにあることが可能であり、その一方でその他のデータストアは、リモートにある。

【００３７】

[0041] プロセッサ１１８および１２８、ならびにサーバ１３６は、例示的には、関連付けられているメモリおよびタイミング回路（別途、図示されてはいない）を伴うコンピュータプロセッサを含む。それらは、それらが属するエージェントまたは環境の機能部分であり、例示的には、その環境またはエージェント内のその他のアイテムによってアクティブ化され、その他のアイテムの機能を容易にする。

【００３８】

[0042] 図１は、コンパイルエージェント１０４を、ＩＤＥ１０２とは別個のものとして示しているということもわかるであろう。しかしながら、コンパイルエージェント１０４はＩＤＥ１０２の一部であることが可能であるということもわかるであろう。また、図１は、さまざまな異なるブロックを、それぞれのブロックに関連付けられている機能とともに示している。より多くの機能がそれぞれのブロックによって実行されるようにブロックどうしを統合することができ、または機能がさらに分散されるようにそれらのブロックを分割することができるということがわかるであろう。

【００３９】

[0043] また、開発者１１０がＩＤＥ１０２を操作およびコントロールするために対話するユーザインターフェース表示は、例示的には、ユーザによって作動できる入力メカニズムを有し、それらの入力メカニズムは、アイコン、テキストボックス、チェックボックス、タイル、ドロップダウンメニューなど、さまざまな異なる形態を取ることができる。それらは、例示的には、ポイントアンドクリックデバイス（トラックボールまたはマウスなど）、ボタン、ジョイスティック、サムパッド、サムスイッチ、仮想またはハードウェアキーボードまたはキーパッドを使用して作動することができる。また、ユーザインターフェース表示がタッチセンシティブスクリーン上に表示される場合には、タッチジェスチャーによって、ユーザの指、スタイラスを用いて、といった具合に、ユーザ入力メカニズム

をアクティブ化することができる。表示を行うデバイスが話声認識コンポーネントを有している場合には、話声コマンドを使用してユーザ入力メカニズムを作動させることができる。

【００４０】

[0044] 図４Ａは、図１において示されているアーキテクチャー１００の、ただし、その要素がクラウドコンピューティングアーキテクチャー５００内に配置されているブロック図である。図４Ａは、複数の代替構成を示している。たとえば、コンパイルエージェント１０４は、クラウド５０２の内部および外部の両方において示されている。これは、エージェント５０２をアーキテクチャー１００内のその他のアイテムと同様にさまざまな異なるロケーションに配置することができるということを示すことを意図されている。クラウドコンピューティングは、計算サービス、ソフトウェアサービス、データアクセスサービス、およびストレージサービスを提供し、それらのサービスは、それらのサービスを配信するシステムの物理的なロケーションまたは構成をエンドユーザが知っているということを必要としない。さまざまな実施形態においては、クラウドコンピューティングは、適切なプロトコルを使用してインターネットなどのワイドエリアネットワークを介してサービスを配信する。たとえば、クラウドコンピューティングプロバイダは、ワイドエリアネットワークを介してアプリケーションを配信し、それらのアプリケーションには、ウェブブラウザまたはその他の任意のコンピューティングコンポーネントを通じてアクセスすることができる。アーキテクチャー１００のソフトウェアまたはコンポーネント、ならびに対応するデータは、リモートロケーションにおけるサーバ上に格納することができる。クラウドコンピューティング環境内のコンピューティングリソースどうしをリモートデータセンターロケーションにおいて統合することができ、またはそれらを分散させることもできる。クラウドコンピューティングインフラストラクチャーは、共有データセンターを通じてサービスを配信することができるが、それらの共有データセンターは、ユーザにとっては単一のアクセスポイントのように見える。したがって、本明細書において説明されているコンポーネントおよび機能は、クラウドコンピューティングアーキテクチャーを使用してリモートロケーションにおいてサービスプロバイダから提供することができる。あるいは、それらのコンポーネントおよび機能を従来のサーバから提供することもでき、またはそれらのコンポーネントおよび機能を直接、またはその他の方法でクライアントデバイス上にインストールすることもできる。

【００４１】

[0045] この説明は、パブリッククラウドコンピューティングおよびプライベートクラウドコンピューティングの両方を含むことを意図されている。クラウドコンピューティング（パブリックおよびプライベートの両方）は、リソースの実質的にシームレスなプーリングを提供し、ならびに基礎をなすハードウェアインフラストラクチャーを管理および構成する必要性を減少させる。

【００４２】

[0046] パブリッククラウドは、ベンダーによって管理され、典型的には、同じインフラストラクチャーを使用して複数の消費者をサポートする。また、パブリッククラウドは、プライベートクラウドとは対照的に、ハードウェアを管理することからエンドユーザを解放することができる。プライベートクラウドは、組織自体によって管理することができ、インフラストラクチャーは、典型的にはその他の組織との間で共有されない。組織は、それでもなお、インストレーションおよび修理など、ある程度ハードウェアを保持する。

【００４３】

[0047] 図４Ａにおいて示されている実施形態においては、いくつかのアイテムは、図１において示されているアイテムと同様であり、それらのアイテムには、同様の番号が付いている。図４Ａは、ＩＤＥ１０２、コンパイルエージェント１０４、およびランタイム環境１０８をすべてクラウド５０２（パブリック、プライベート、または、複数の部分がパブリックであってその他の部分がプライベートである組合せであることが可能である）内に配置することができるということを具体的に示している。したがって開発者１１０は、

クラウド５０２を通じてそれらのシステムにアクセスするためにユーザインターフェース表示５０５を用いてユーザデバイス５０４を使用する。

【００４４】

[0048] 図４Ａはまた、クラウドアーキテクチャーの別の代替実施形態を示している。図４Ａは、アーキテクチャー１００のいくつかの要素はクラウド５０２内に配置され、その一方でその他の要素はクラウド５０２内に配置されないことも考えられるということを示している。例として、データストア１０６、１２０、１３４をクラウド５０２の外部に配置して、クラウド５０２を通じてデータストア１０６、１２０、１３４にアクセスすることができる。別の実施形態においては、コンパイルエージェント１０４は、クラウド５０２の外部にあることが可能である。それらがどこに配置されるかにかかわらず、それらには、デバイス５０４によって直接、ネットワーク（ワイドエリアネットワークまたはローカルエリアネットワークのいずれか）を通じてアクセスすることができ、それらをサービスによってリモートサイトにおいてホストすることができ、またはクラウドを通じたサービスとしてそれらを提供することができ、もしくはクラウド内に存在する接続サービスによってそれらにアクセスすることができる。本明細書においては、これらのアーキテクチャーのすべてが考えられる。

【００４５】

[0049] 図４Ｂは、別のクラウドベースのアーキテクチャーのブロック図を示している。図４Ｂは、開発者１１０がリモートアクセスデバイス５０７を使用して開発者デバイス５０４およびＩＤＥ１０２にアクセスするという点を除いて図４Ａと同様である。本明細書においては、図４Ａおよび図４Ｂにおいて表されているすべてのさまざまな構成が考えられる。

【００４６】

[0050] アーキテクチャー１００またはその複数の部分をさまざまな異なるデバイス上に配置することができるということもわかるであろう。それらのデバイスのいくつかは、サーバ、デスクトップコンピュータ、ラップトップコンピュータ、タブレットコンピュータ、またはその他のモバイルデバイス、たとえば、パームトップコンピュータ、携帯電話、スマートフォン、マルチメディアプレーヤ、携帯情報端末などを含む。

【００４７】

[0051] 図５は、アーキテクチャー１００（またはその複数の部分）を展開することができる、ユーザのまたはクライアントのハンドヘルドデバイス１６として使用することができるハンドヘルドまたはモバイルコンピューティングデバイスの例示的な一実施形態の簡略化されたブロック図である。図６～図９は、ハンドヘルドまたはモバイルデバイスの例である。

【００４８】

[0052] 図５は、アーキテクチャー１００のコンポーネントを稼働させることができる、もしくはアーキテクチャー１００と対話する、またはそれらの両方を行うクライアントデバイス１６のコンポーネントの一般的なブロック図を提供している。デバイス１６においては、通信リンク１３が提供されており、通信リンク１３は、ハンドヘルドデバイスがその他のコンピューティングデバイスと通信することを可能にし、いくつかの実施形態のもとでは、スキャニングなどによって自動的に情報を受け取るためのチャネルを提供する。通信リンク１３の例は、赤外線ポート、シリアル／ＵＳＢポート、ケーブルネットワークポート、たとえばイーサネットポート、および、１つまたは複数の通信プロトコルを通じた通信を可能にするワイヤレスネットワークポートを含み、そうした通信プロトコルとしては、ジェネラルパケットラジオサービス（GPRS）、ＬＴＥ、ＨＳＰＡ、ＨＳＰＡ＋、ならびにその他の３Ｇおよび４Ｇ無線プロトコル、１Ｘｒｔｔ、および、ネットワークへのセルラーアクセスを提供するために使用されるワイヤレスサービスであるショートメッセージサービス、ならびに、ネットワークへのローカルワイヤレス接続を提供する８０２．１１および８０２．１１ｂ（Wi-Fi）プロトコルおよびブルートゥースプロトコルが含まれる。

【００４９】
[0053] その他の実施形態のもとでは、アプリケーションまたはシステムは、ＳＤカードインターフェース１５に接続されている取り外し可能なセキュアデジタル（SD）カード上で受け取られる。ＳＤカードインターフェース１５および通信リンク１３は、バス１９に沿ってプロセッサ１７（これは、図１からのプロセッサ１１８もしくは１２８またはサーバ１３６を具体化することもできる）と通信し、バス１９はまた、メモリ２１および入力／出力（I/O）コンポーネント２３、ならびにクロック２５およびロケーションシステム２７に接続されている。
【００５０】
[0054] Ｉ／Ｏコンポーネント２３は、一実施形態においては、入力および出力オペレーションを容易にするために提供されている。デバイス１６のさまざまな実施形態に関するＩ／Ｏコンポーネント２３は、入力コンポーネント、たとえば、ボタン、タッチセンサ、マルチタッチセンサ、光センサまたはビデオセンサ、音声センサ、タッチスクリーン、近接センサ、マイクロフォン、傾斜センサ、および重力スイッチ、ならびに出力コンポーネント、たとえば、ディスプレイデバイス、スピーカー、およびまたはプリンタポートを含むことができる。その他のＩ／Ｏコンポーネント２３を使用することもできる。
【００５１】
[0055] クロック２５は、例示的には、時間および日付を出力するリアルタイムクロックコンポーネントを含む。それは、例示的には、プロセッサ１７のためのタイミング機能を提供することもできる。
【００５２】
[0056] ロケーションシステム２７は、例示的には、デバイス１６の現在の地理的ロケーションを出力するコンポーネントを含む。これは、たとえば、グローバルポジショニングシステム（GPS）受信機、ＬＯＲＡＮシステム、推測航法システム、セルラー三角測量システム、またはその他の測位システムを含むことができる。それは、たとえば、所望の地図、ナビゲーションルート、およびその他の地理的機能を生み出すマッピングソフトウェアまたはナビゲーションソフトウェアを含むこともできる。
【００５３】
[0057] メモリ２１は、オペレーティングシステム２９、ネットワーク設定３１、アプリケーション３３、アプリケーション構成設定３５、データストア３７、通信ドライバ３９、および通信構成設定４１を格納する。メモリ２１は、すべてのタイプの有形の揮発性および不揮発性のコンピュータ可読メモリデバイスを含むことができる。メモリ２１は、（後述されている）コンピュータストレージメディアを含むこともできる。メモリ２１は、コンピュータ可読命令を格納し、それらのコンピュータ可読命令は、プロセッサ１７によって実行されたときに、そのプロセッサに、コンピュータによって実施されるステップまたは機能をそれらの命令に従って実行させる。同様に、デバイス１６は、さまざまなビジネスアプリケーションを稼働させることができるクライアントビジネスシステム２４を有することができる。プロセッサ１７は、その他のコンポーネントによって、それらのコンポーネントの機能を容易にするためにアクティブ化することもできる。
【００５４】
[0058] ネットワーク設定３１の例は、プロキシ情報、インターネット接続情報、およびマッピングなどのものを含む。アプリケーション構成設定３５は、アプリケーションを特定の企業またはユーザ向けに調整する設定を含む。通信構成設定４１は、その他のコンピュータと通信するためのパラメータを提供し、ＧＰＲＳパラメータ、ＳＭＳパラメータ、接続ユーザ名、およびパスワードなどのアイテムを含む。
【００５５】
[0059] アプリケーション３３は、デバイス１６上に以前に格納されたアプリケーション、または使用中にインストールされるアプリケーションであることが可能であるが、これらは、オペレーティングシステム２９の一部であること、またはデバイス１６の外部でホストすることも可能である。

【００５６】
[0060] 図６は、デバイス１６がタブレットコンピュータ６００である一実施形態を示している。図６においては、コンピュータ６００は、ディスプレイスクリーン６０２とともに示されている。スクリーン６０２は、タッチスクリーンであることが可能であり（したがって、ユーザの指からのタッチジェスチャーを使用して、アプリケーションと対話することができ）、またはペンもしくはスタイラスからの入力を受け取るペン対応インターフェースであることが可能である。スクリーン６０２は、オンスクリーン仮想キーボードを使用することもできる。もちろん、スクリーン６０２は、たとえばワイヤレスリンクまたはＵＳＢポートなどの適切な接続メカニズムを通じてキーボードまたはその他のユーザ入力デバイスに接続することもできる。コンピュータ６００は、例示的には、音声入力を受け取ることもできる。

【００５７】
[0061] 図７および図８は、使用することができるデバイス１６のさらなる例を提供しているが、その他のデバイスを使用することもできる。図７においては、フィーチャーフォン、スマートフォン、またはモバイル電話４５が、デバイス１６として提供されている。電話４５は、電話番号をダイヤルするための一式のキーパッド４７と、アプリケーションイメージ、アイコン、ウェブページ、写真、およびビデオを含むイメージを表示することができるディスプレイ４９と、ディスプレイ上に表示されるアイテムを選択するためのコントロールボタン５１とを含む。この電話は、ジェネラルパケットラジオサービス（GPRS）および１Ｘｒｔｔなどのセルラー電話信号、ならびにショートメッセージサービス（SMS）信号を受け取るためのアンテナ５３を含む。いくつかの実施形態においては、電話４５はまた、ＳＤカード５７を受け入れるセキュアデジタル（SD）カードスロット５５を含む。

【００５８】
[0062] 図８のモバイルデバイスは、携帯情報端末（PDA）５９、またはマルチメディアプレーヤ、またはタブレットコンピューティングデバイスなど（以降では、ＰＤＡ５９と呼ぶ）である。ＰＤＡ５９は、インダクティブスクリーン６１を含み、インダクティブスクリーン６１は、スタイラス６３（またはその他のポインタ、たとえばユーザの指）の位置を、そのスタイラスがスクリーンの上に位置したときに感知する。これは、ユーザがスクリーン上のアイテムを選択すること、強調表示すること、および移動させること、ならびに描画を行うこと、および文字を書くことを可能にする。ＰＤＡ５９はまた、複数のユーザ入力キーまたはボタン（ボタン６５など）を含み、それらのキーまたはボタンは、ユーザが、ディスプレイ６１上に表示されるメニューオプションまたはその他の表示オプションをスクロールすることを可能にし、また、ユーザがディスプレイ６１に触れることなくアプリケーションを変更すること、またはユーザ入力機能を選択することを可能にする。図示されてはいないが、ＰＤＡ５９は、その他のコンピュータとのワイヤレス通信を可能にする内蔵アンテナおよび赤外線送信機／受信機、ならびにその他のコンピューティングデバイスへのハードウェア接続を可能にする接続ポートを含むことができる。そのようなハードウェア接続は、典型的には、シリアルまたはＵＳＢポートを通じてその他のコンピュータに接続するクレードルを通じて行われる。したがって、これらの接続は、非ネットワーク接続である。一実施形態においては、モバイルデバイス５９はまた、ＳＤカード６９を受け入れるＳＤカードスロット６７を含む。

【００５９】
[0063] 図９は、電話がスマートフォン７１であるという点を除いて図７と同様である。スマートフォン７１は、アイコンまたはタイルまたはその他のユーザ入力メカニズム７５を表示するタッチセンシティブディスプレイ７３を有する。メカニズム７５は、ユーザによって、アプリケーションを稼働させること、通話すること、データ転送オペレーションを実行することなどを行うために使用することができる。一般には、スマートフォン７１は、モバイルオペレーティングシステムに基づいて構築され、フィーチャーフォンよりも進んだコンピューティング機能および接続を提供する。

【００６０】
[0064] その他の形態のデバイス１６も可能であるということに留意されたい。
【００６１】
[0065] 図１０は、アーキテクチャー１００またはその複数の部分を（たとえば）展開することができるコンピューティング環境の一実施形態である。図１０を参照すると、いくつかの実施形態を実施するための例示的なシステムが、汎用コンピューティングデバイスをコンピュータ８１０の形態で含む。コンピュータ８１０のコンポーネントは、処理ユニット８２０（プロセッサ１１８もしくは１２８またはサーバ１３６を含むことができる）と、システムメモリ８３０と、そのシステムメモリを含むさまざまなシステムコンポーネントを処理ユニット８２０に結合するシステムバス８２１とを含むことができるが、それらには限定されない。システムバス８２１は、メモリバスまたはメモリコントローラと、ペリフェラルバスと、さまざまなバスアーキテクチャーのうちのいずれかを使用するローカルバスとを含むいくつかのタイプのバス構造のうちのいずれかであることが可能である。例として、そのようなアーキテクチャーは、インダストリースタンダードアーキテクチャー（ISA）バス、マイクロチャネルアーキテクチャー（MCA）バス、エンハンストＩＳＡ（EISA）バス、ビデオエレクトロニクススタンダーズアソシエーション（VESA）ローカルバス、および、メザニンバスとしても知られているペリフェラルコンポーネントインターコネクト（PCI）バスを含むが、それらには限定されない。図１に関連して説明されているメモリおよびプログラムを図１０の対応する部分において展開することができる。
【００６２】
[0066] コンピュータ８１０は、典型的には、さまざまなコンピュータ可読メディアを含む。コンピュータ可読メディアは、コンピュータ８１０によってアクセスすることができる任意の利用可能なメディアであることが可能であり、揮発性のメディアおよび不揮発性のメディア、取り外し可能なメディアおよび取り外し不能なメディアの両方を含む。例として、コンピュータ可読メディアは、コンピュータストレージメディアおよび通信メディアを含むことができるが、それらには限定されない。コンピュータストレージメディアは、変調されたデータ信号または搬送波とは異なり、変調されたデータ信号または搬送波を含まない。コンピュータストレージメディアは、コンピュータ可読命令、データ構造、プログラムモジュール、またはその他のデータなどの情報の格納のための任意の方法またはテクノロジーにおいて実装される揮発性のメディアおよび不揮発性のメディア、取り外し可能なメディアおよび取り外し不能なメディアの両方を含むハードウェアストレージメディアを含む。コンピュータストレージメディアとしては、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュメモリ、もしくはその他のメモリテクノロジー、ＣＤ－ＲＯＭ、デジタル多用途ディスク（DVD）、もしくはその他の光ディスクストレージ、磁気カセット、磁気テープ、磁気ディスクストレージ、もしくはその他の磁気ストレージデバイス、または、所望の情報を格納するために使用可能で、コンピュータ８１０によってアクセス可能なその他の任意のメディアが含まれるが、それらには限定されない。通信メディアは、典型的には、伝送メカニズムにおいてコンピュータ可読命令、データ構造、プログラムモジュール、またはその他のデータを具体化し、任意の情報伝達メディアを含む。「変調されたデータ信号」という用語は、その信号の特性のうちの１つまたは複数が、その信号内で情報をエンコードするような様式で設定または変更されている信号を意味する。限定ではなく、例として、通信メディアは、有線ネットワークまたは直接有線接続などの有線メディア、ならびに、音響メディア、ＲＦメディア、赤外線メディア、およびその他のワイヤレスメディアなどのワイヤレスメディアを含む。上記のいずれの組合せも、コンピュータ可読メディアの範囲内に含まれるものである。
【００６３】
[0067] システムメモリ８３０は、コンピュータストレージメディアを読み取り専用メモリ（ROM）８３１およびランダムアクセスメモリ（RAM）８３２などの揮発性のメモリおよび／または不揮発性のメモリの形態で含む。基本入出力システム８３３（BIOS）は、スタートアップ中などにコンピュータ８１０内の要素どうしの間において情報を伝達するのを

補助する基本ルーチンを含み、典型的にはＲＯＭ８３１内に格納されている。ＲＡＭ８３２は、典型的には、処理ユニット８２０にとってすぐにアクセス可能な、および／または処理ユニット８２０によって現在操作されているデータモジュールおよび／またはプログラムモジュールを含む。限定ではなく、例として、図１０は、オペレーティングシステム８３４、アプリケーションプログラム８３５、その他のプログラムモジュール８３６、およびプログラムデータ８３７を示している。

【００６４】

[0068] コンピュータ８１０は、その他の取り外し可能な／取り外し不能な、揮発性の／不揮発性のコンピュータストレージメディアを含むこともできる。単なる例として、図１０は、取り外し不能な不揮発性の磁気メディアとの間で読み取りまたは書き込みを行うハードディスクドライブ８４１と、取り外し可能な不揮発性の磁気ディスク８５２との間で読み取りまたは書き込みを行う磁気ディスクドライブ８５１と、ＣＤ　ＲＯＭまたはその他の光メディアなどの取り外し可能な不揮発性の光ディスク８５６との間で読み取りまたは書き込みを行う光ディスクドライブ８５５とを示している。例示的な動作環境において使用することができるその他の取り外し可能な／取り外し不能な、揮発性の／不揮発性のコンピュータストレージメディアとしては、磁気テープカセット、フラッシュメモリカード、デジタル多用途ディスク、デジタルビデオテープ、ソリッドステートＲＡＭ、ソリッドステートＲＯＭなどが含まれるが、それらには限定されない。ハードディスクドライブ８４１は、典型的には、インターフェース８４０などの取り外し不能なメモリインターフェースを通じてシステムバス８２１に接続されており、磁気ディスクドライブ８５１および光ディスクドライブ８５５は、典型的には、インターフェース８５０などの取り外し可能なメモリインターフェースによってシステムバス８２１に接続されている。

【００６５】

[0069] 代替として、または追加として、本明細書において説明されている機能は、少なくとも部分的に、１つまたは複数のハードウェアロジックコンポーネントによって実行することができる。限定ではなく、例として、使用することができるハードウェアロジックコンポーネントの例示的なタイプとしては、フィールドプログラマブルゲートアレイ（FPGA）、プログラム固有集積回路（ASIC）、プログラム固有標準製品（ASSP）、システムオンアチップシステム（SOC）、結合プログラム可能論理回路（CPLD）などが含まれる。

【００６６】

[0070] 図１０において示されている上述のドライブおよびそれらの関連付けられているコンピュータストレージメディアは、コンピュータ８１０のためのコンピュータ可読命令、データ構造、プログラムモジュール、およびその他のデータの格納を提供する。たとえば、図１０においては、ハードディスクドライブ８４１は、オペレーティングシステム８４４、アプリケーションプログラム８４５、その他のプログラムモジュール８４６、およびプログラムデータ８４７を格納するものとして示されている。これらのコンポーネントは、オペレーティングシステム８３４、アプリケーションプログラム８３５、その他のプログラムモジュール８３６、およびプログラムデータ８３７と同じであること、または異なっていることが可能であるという点に留意されたい。ここでは少なくとも、オペレーティングシステム８４４、アプリケーションプログラム８４５、その他のプログラムモジュール８４６、およびプログラムデータ８４７が、少なくとも異なるコピーであるということを示すために、それらに異なる番号を割り当てている。

【００６７】

[0071] ユーザは、キーボード８６２、マイクロフォン８６３、およびポインティングデバイス８６１、たとえば、マウス、トラックボール、またはタッチパッドなどの入力デバイスを通じてコンピュータ８１０にコマンドおよび情報を入力することができる。その他の入力デバイス（図示せず）は、ジョイスティック、ゲームパッド、衛星放送受信用アンテナ、スキャナなどを含むことができる。これらおよびその他の入力デバイスは、システムバスに結合されているユーザ入力インターフェース８６０を通じて処理ユニット８２０に接続されている場合が多いが、パラレルポート、ゲームポート、またはユニバーサルシ

リアルバス（USB）などのその他のインターフェース構造およびバス構造によって接続することもできる。ビジュアルディスプレイ８９１またはその他のタイプのディスプレイデバイスも、ビデオインターフェース８９０などのインターフェースを介してシステムバス８２１に接続されている。コンピュータは、モニタに加えて、スピーカー８９７およびプリンタ８９６などのその他の周辺出力デバイスを含むこともでき、これらには、周辺出力インターフェース８９５を通じて接続することができる。

【００６８】

[0072] コンピュータ８１０は、リモートコンピュータ８８０などの１つまたは複数のリモートコンピュータへの論理接続を使用して、ネットワーク化された環境内で操作される。リモートコンピュータ８８０は、パーソナルコンピュータ、ハンドヘルドデバイス、サーバ、ルータ、ネットワークＰＣ、ピアデバイス、またはその他の一般的なネットワークノードであることが可能であり、典型的には、コンピュータ８１０に関連する上述の要素のうちの多くまたはすべてを含む。図１０において示されている論理接続は、ローカルエリアネットワーク（LAN）８７１およびワイドエリアネットワーク（WAN）８７３を含むが、その他のネットワークを含むこともできる。そのようなネットワーキング環境は、オフィス、企業規模のコンピュータネットワーク、イントラネット、およびインターネットにおいてよく見受けられる。

【００６９】

[0073] ＬＡＮネットワーキング環境において使用される場合には、コンピュータ８１０は、ネットワークインターフェースまたはアダプタ８７０を通じてＬＡＮ８７１に接続される。ＷＡＮネットワーキング環境において使用される場合には、コンピュータ８１０は、典型的には、インターネットなどのＷＡＮ８７３上で通信を確立するためのモデム８７２またはその他の手段を含む。モデム８７２は、内蔵型または外付け型であることが可能であり、ユーザ入力インターフェース８６０またはその他の適切なメカニズムを介してシステムバス８２１に接続することができる。ネットワーク化された環境においては、コンピュータ８１０に関連して示されているプログラムモジュール、またはそれらの一部をリモートメモリストレージデバイス内に格納することができる。限定ではなく、例として、図１０は、リモートアプリケーションプログラム８８５を、リモートコンピュータ８８０上に存在するものとして示している。示されているネットワーク接続は、例示的なものであり、コンピュータどうしの間において通信リンクを確立するその他の手段を使用することもできるということが理解できるであろう。

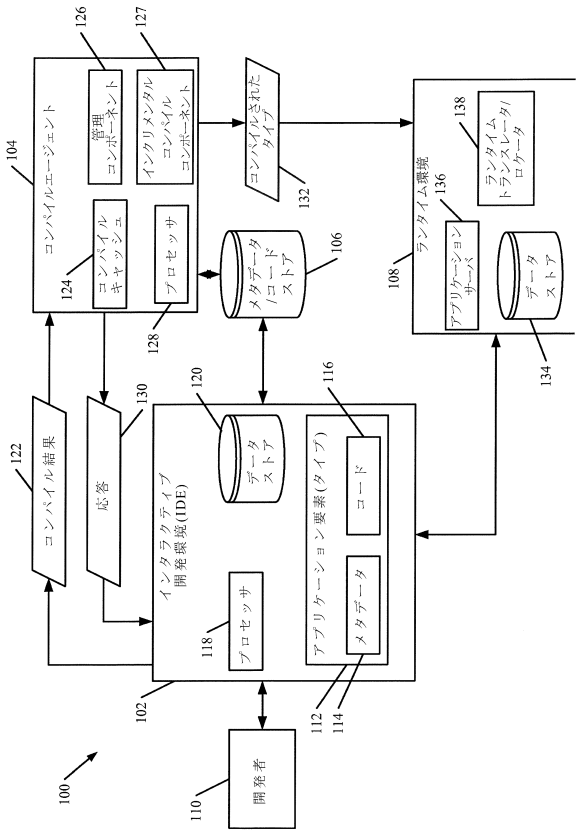
【００７０】

[0074] 本明細書において説明されているさまざまな実施形態をさまざまな方法で組み合わせることができるということにも留意されたい。すなわち、１つまたは複数の実施形態の複数の部分を１つまたは複数のその他の実施形態の複数の部分と組み合わせることができる。本明細書においては、このすべてが考えられる。

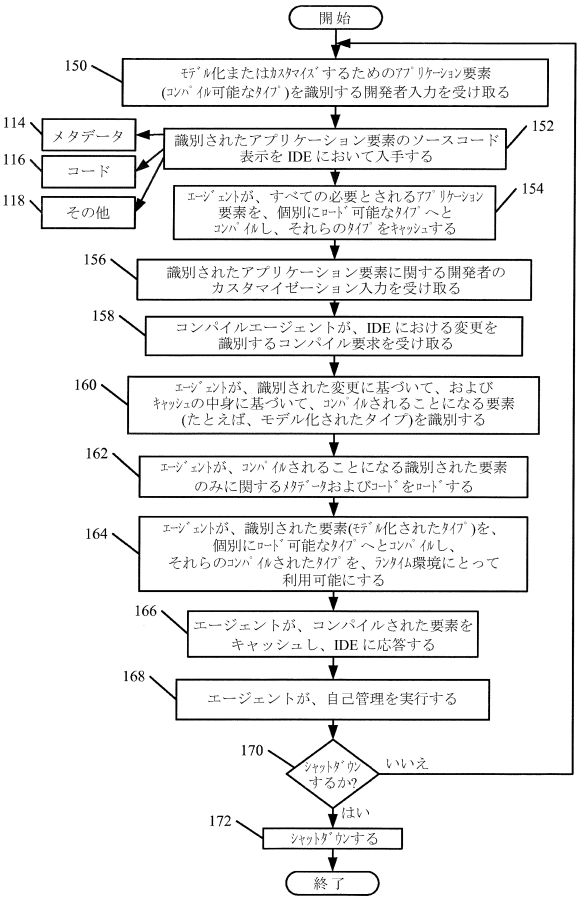
【００７１】

[0075] 本主題は、構造的な特徴および／または方法論的な行為に特有の言葉で説明されているが、添付の特許請求の範囲において定義されている本主題は、上述の特定の特徴または行為に必ずしも限定されるものではないということを理解されたい。むしろ、上述の特定の特徴および行為は、特許請求の範囲を実施する例示的な形態として開示されている。

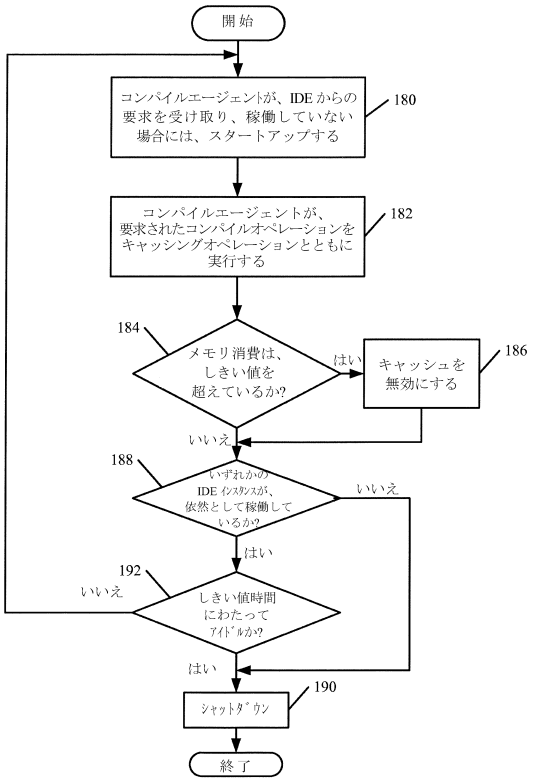
【図１】


126
127
管理コンポーネント
インクリメンタルコンパイルコンポーネント
104
コンパイルエージェント
コンパイルキャッシュ
124
プロセッサ
128
コンパイルされたタイプ
132
メタデータ/コードストア
106
ランタイム環境
136
ランタイムトランスレータ/ローダー
138
アプリケーションサーバ
108
データストア
134
コンパイル結果
122
応答
130
インタラクティブ開発環境(IDE)
120
データストア
116
アプリケーション要素(タイプ)
コード
メタデータ
プロセッサ
118
102
112
114
開発者
110
100

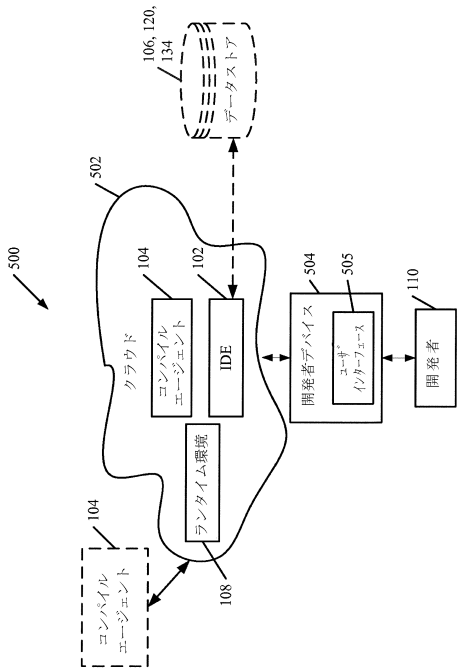
【図２】


開始
150
ﾓﾃﾞﾙ化またはｶｽﾀﾏｲｽﾞするためのｱﾌﾟﾘｹｰｼｮﾝ要素(ｺﾝﾊﾟｲﾙ可能なﾀｲﾌﾟ)を識別する開発者入力を受け取る
114
メタデータ
116
コード
118
その他
識別されたアプリケーション要素のソースコード表示をIDEにおいて入手する
152
ｴｰｼﾞｪﾝﾄが、すべての必要とされるｱﾌﾟﾘｹｰｼｮﾝ要素を、個別にﾛｰﾄﾞ可能なﾀｲﾌﾟへとｺﾝﾊﾟｲﾙし、それらのﾀｲﾌﾟをｷｬｯｼｭする
154
156
識別されたアプリケーション要素に関する開発者のカスタマイゼーション入力を受け取る
158
コンパイルエージェントが、IDEにおける変更を識別するコンパイル要求を受け取る
160
ｴｰｼﾞｪﾝﾄが、識別された変更に基づいて、およびｷｬｯｼｭの中身に基づいて、ｺﾝﾊﾟｲﾙされることになる要素(たとえば、モデル化されたタイプ)を識別する
162
ｴｰｼﾞｪﾝﾄが、ｺﾝﾊﾟｲﾙされることになる識別された要素のみに関するﾒﾀﾃﾞｰﾀおよびｺｰﾄﾞをﾛｰﾄﾞする
164
ｴｰｼﾞｪﾝﾄが、識別された要素(ﾓﾃﾞﾙ化されたﾀｲﾌﾟ)を、個別にﾛｰﾄﾞ可能なﾀｲﾌﾟへとｺﾝﾊﾟｲﾙし、それらのｺﾝﾊﾟｲﾙされたﾀｲﾌﾟを、ﾗﾝﾀｲﾑ環境にとって利用可能にする
166
エージェントが、コンパイルされた要素をキャッシュし、IDEに応答する
168
エージェントが、自己管理を実行する
170
ｼｬｯﾄﾀﾞｳﾝするか?
いいえ
はい
172
ｼｬｯﾄﾀﾞｳﾝする
終了

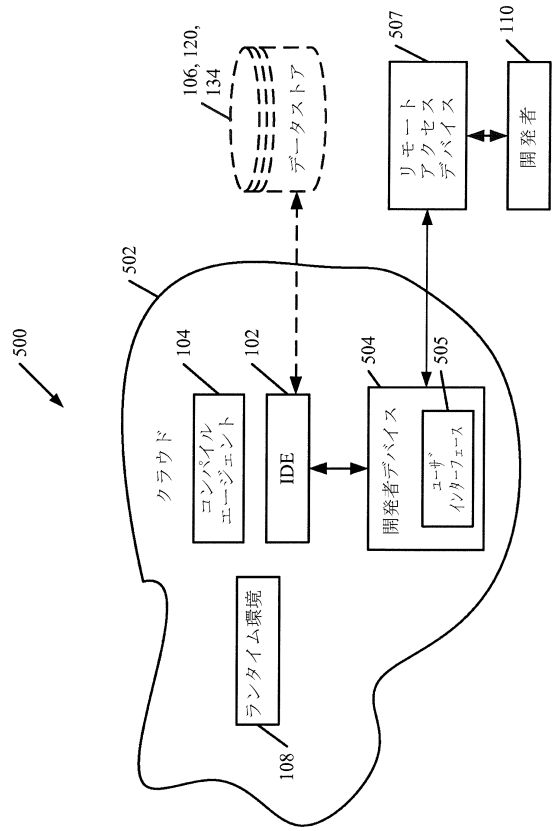
【図３】


開始
コンパイルエージェントが、IDEからの要求を受け取り、稼働していない場合には、スタートアップする
180
コンパイルエージェントが、要求されたコンパイルオペレーションをキャッシングオペレーションとともに実行する
182
184
メモリ消費は、しきい値を超えているか?
はい
キャッシュを無効にする
186
いいえ
188
いずれかのIDEインスタンスが、依然として稼働しているか?
いいえ
はい
192
しきい値時間にわたってアイドルか?
いいえ
はい
ｼｬｯﾄﾀﾞｳﾝ
190
終了

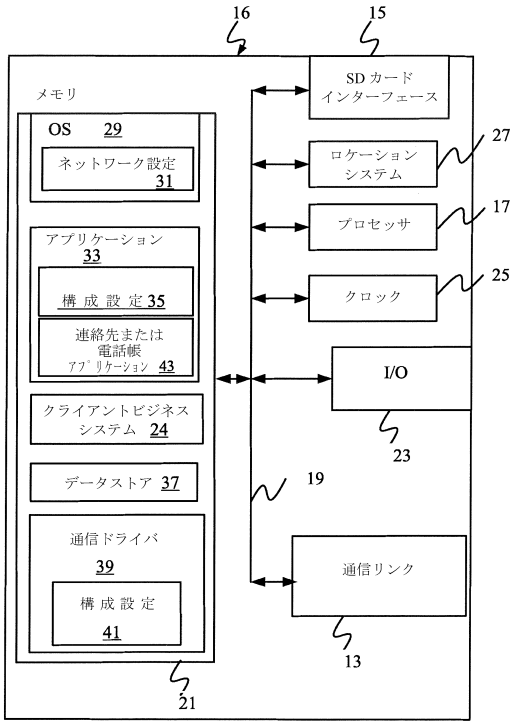
【図４Ａ】


106, 120, 134
データストア
502
500
104
コンパイルエージェント
102
IDE
クラウド
ランタイム環境
108
504
505
ユーザーインターフェース
開発者デバイス
110
開発者
104
コンパイルエージェント

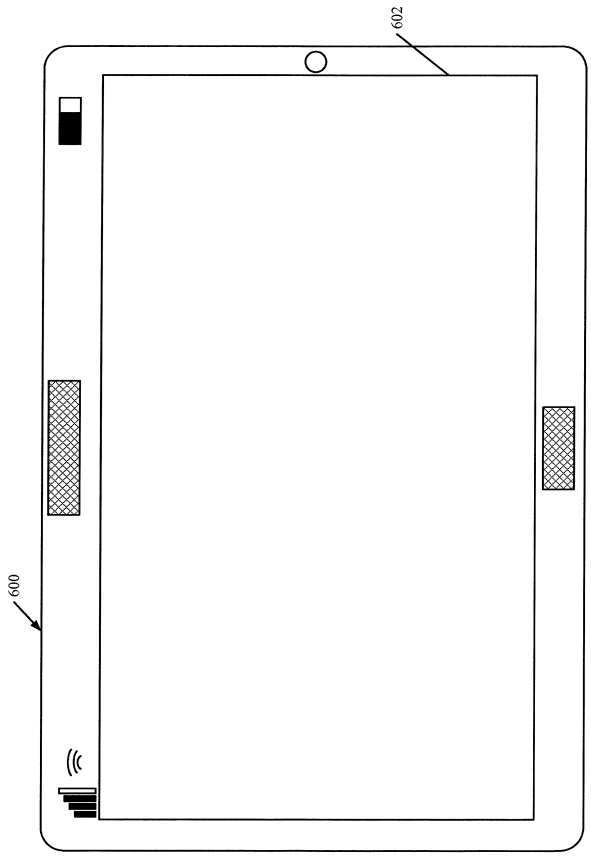
【図４Ｂ】


500
502
クラウド
104
コンパイル
エージェント
102
IDE
504
開発者デバイス
505
ユーザ
インターフェース
ランタイム環境
108
106, 120,
134
データストア
507
リモート
アクセス
デバイス
110
開発者

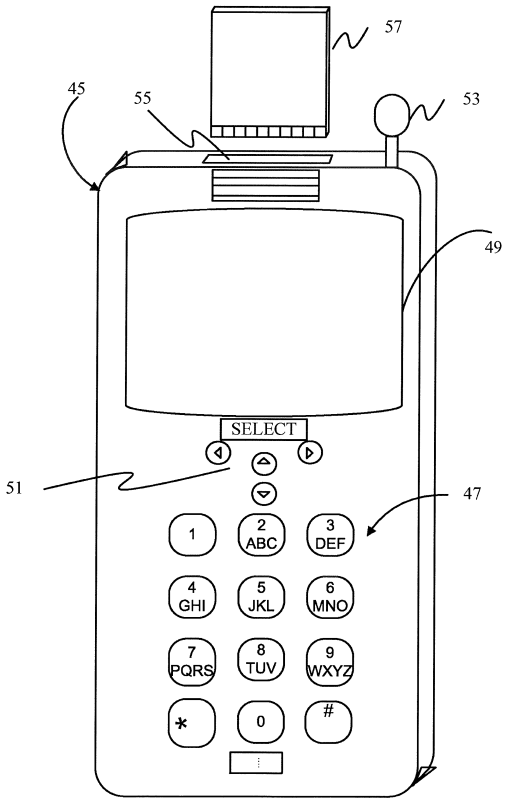
【図５】


16
15
メモリ
OS 29
ネットワーク設定
31
アプリケーション
33
構成設定 35
連絡先または
電話帳
アプリケーション 43
クライアントビジネス
システム 24
データストア 37
通信ドライバ
39
構成設定
41
21
SD カード
インターフェース
ロケーション
システム
27
プロセッサ
17
クロック
25
I/O
23
19
通信リンク
13

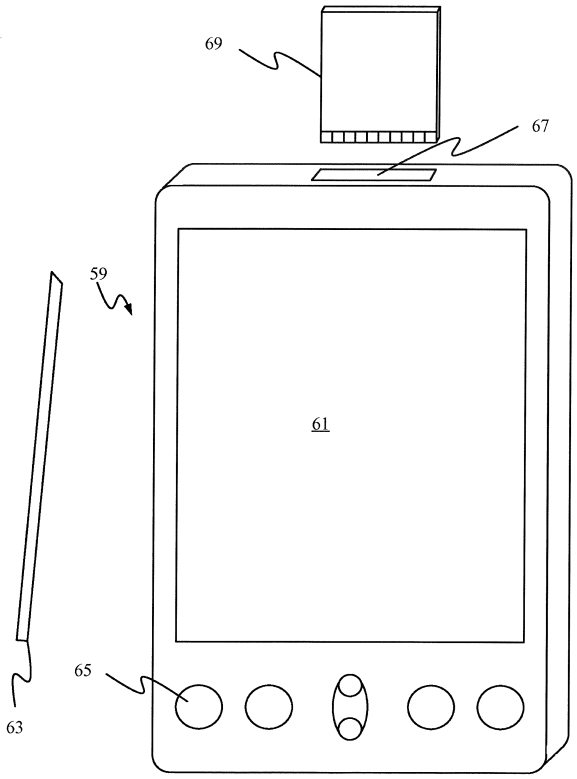
【図６】


602
600

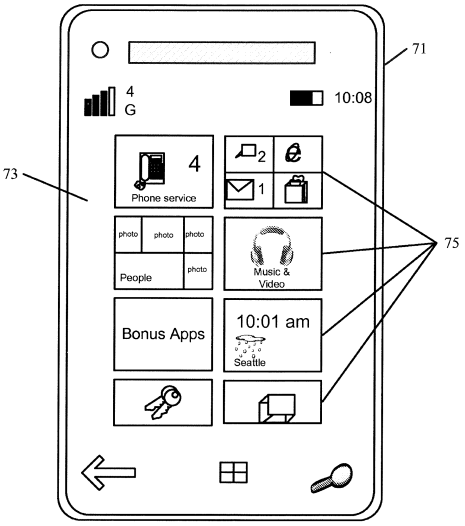
【図７】


57
45
55
53
49
SELECT
51
47
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
*
0
#

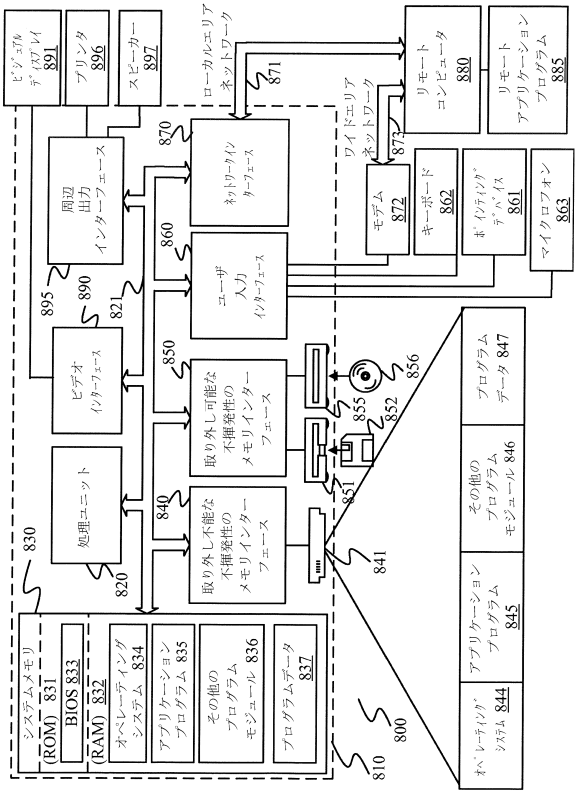
【図８】


69
67
59
61
65
63

【図９】


71
4
G
10:08
73
4
Phone service
2
1
photo
photo
photo
photo
People
75
Music &
Video
Bonus Apps
10:01 am
Seattle

【図１０】


ビデオディスプレイ 891
プリンタ 896
スピーカー 897
ローカルエリアネットワーク 871
リモートコンピュータ 880
リモートアプリケーションプログラム 885
ワイドエリアネットワーク 873
870
ネットワークインターフェース
周辺出力インターフェース
モデム 872
キーボード 862
ポインティングデバイス 861
マイクロフォン 863
860
ユーザ入力インターフェース
895
890
821
ビデオインターフェース
850
取り外し可能な不揮発性のメモリインターフェース
856
855
852
857
840
取り外し不能な不揮発性のメモリインターフェース
841
830
処理ユニット
820
システムメモリ
(ROM) 831
BIOS 833
(RAM) 832
オペレーティングシステム 834
アプリケーションプログラム 835
その他のプログラムモジュール 836
プログラムデータ 837
プログラムデータ 847
その他のプログラムモジュール 846
アプリケーションプログラム 845
オペレーティングシステム 844
800
810

フロントページの続き

(74)代理人　100108213
弁理士　阿部　豊隆
(74)代理人　100188189
弁理士　阪　和之
(72)発明者　アガグ，カリド
アメリカ合衆国，ワシントン州　９８０５２－６３９９，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　コーポレーション内，エルシーエー　－　インターナショナル　パテンツ（８／１１７２）
(72)発明者　ナラヤナン，スリヤ
アメリカ合衆国，ワシントン州　９８０５２－６３９９，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　コーポレーション内，エルシーエー　－　インターナショナル　パテンツ（８／１１７２）

審査官　今城　朋彬

(56)参考文献　特開２０１０－２６７２６４（ＪＰ，Ａ）
特開２００２－０８２８１１（ＪＰ，Ａ）
米国特許第０５８５４９３２（ＵＳ，Ａ）
特開２００７－２３３４７２（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　８／４１
Ｇ０６Ｆ　１１／３６
Ｇ０６Ｆ　１１／０７