(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6199314号
(P6199314)
(45)発行日　平成29年9月20日(2017.9.20)　(24)登録日　平成29年9月1日(2017.9.1)

(51)Int.Cl.　F I
G06F 9/44 (2006.01)　G06F 9/06 620A

請求項の数 22 (全 24 頁)

(21)出願番号　特願2014-554754(P2014-554754)
(86)(22)出願日　平成25年1月21日(2013.1.21)
(65)公表番号　特表2015-510181(P2015-510181A)
(43)公表日　平成27年4月2日(2015.4.2)
(86)国際出願番号　PCT/US2013/022351
(87)国際公開番号　WO2013/112388
(87)国際公開日　平成25年8月1日(2013.8.1)
審査請求日　平成28年1月21日(2016.1.21)
(31)優先権主張番号　13/357,623
(32)優先日　平成24年1月25日(2012.1.25)
(33)優先権主張国　米国(US)

(73)特許権者　500046438
マイクロソフト　コーポレーション
ＭＩＣＲＯＳＯＦＴ　ＣＯＲＰＯＲＡＴＩＯＮ
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
One Microsoft Way, Redmond WA 98052-5321 United State of America
(74)代理人　100140109
弁理士　小野　新次郎
(74)代理人　100075270
弁理士　小林　泰
最終頁に続く

(54)【発明の名称】演算子の優先順位のグラフィカル表現

(57)【特許請求の範囲】
【請求項１】
コンピューターが実行する方法であって、
コンピュータープログラムのソースコードをソースコードエディターにおいて見せるステップと、
見せた前記ソースコードから、ソースコードの１つまたは複数の行を含む、前記ソースコードの部分を選択するステップと、
前記ソースコードに関連した言語モデルに基づき、演算子の優先順位を決定するステップであって、演算子の前記優先順位は、ソースコードの選択した前記部分のある演算子が、ソースコードの選択した前記部分の他の演算子に対して、ソースコードの選択した前記部分が実行されたときにどのように実行されるかの優先順位を表す、ステップと、
演算子の前記優先順位を示すグラフィカル表現を提供するステップと、
前記グラフィカル表現に従う演算子の前記優先順位を、ソースコードの選択した前記部分とともにインラインで表示するステップと
を含む、コンピューターが実行する方法。
【請求項２】
請求項１に記載のコンピューターが実行する方法であって、
演算子の前記優先順位を線表現形式で表示するためのグラフィカル表現を指定するステップ
をさらに含む、コンピューターが実行する方法。

【請求項３】
請求項１に記載のコンピューターが実行する方法であって、
演算子の前記優先順位をツリー表現形式で表示するためのグラフィカル表現を指定するステップ
をさらに含む、コンピューターが実行する方法。
【請求項４】
請求項１に記載のコンピューターが実行する方法であって、
演算子の前記優先順位を色付きテキスト表現形式で表示するためのグラフィカル表現を指定するステップ
をさらに含む、コンピューターが実行する方法。
【請求項５】
請求項１に記載のコンピューターが実行する方法であって、
演算子の前記優先順位を数値による順序表現形式で表示するためのグラフィカル表現を指定するステップ
をさらに含む、コンピューターが実行する方法。
【請求項６】
請求項１に記載のコンピューターが実行する方法であって、
演算子の前記優先順位を並列表現形式で表示するためのグラフィカル表現を指定するステップ
をさらに含む、コンピューターが実行する方法。
【請求項７】
請求項１に記載のコンピューターが実行する方法であって、選択する前記ステップは、
ソースコードの選択した前記部分を入力セレクターの使用を介して指定するステップ
をさらに含む、コンピューターが実行する方法。
【請求項８】
請求項１に記載のコンピューターが実行する方法であって、
演算子の前記優先順位を、ソースコードの選択した前記部分が表示される視覚的ディスプレイ・ウィンドウと同じ視覚的ディスプレイ・ウィンドウ内に表示するためのグラフィカル表現を指定するステップ
をさらに含む、コンピューターが実行する方法。
【請求項９】
請求項１に記載のコンピューターが実行する方法であって、
演算子の前記優先順位を、ソースコードの選択した前記部分が表示されるところとは異なる視覚的ディスプレイ・ウィンドウ内に表示するためのグラフィカル表現を指定するステップ
をさらに含む、コンピューターが実行する方法。
【請求項１０】
請求項１から９のうちの何れか一項に記載のコンピューターが実行する方法であって、
演算子の前記優先順位の前記グラフィカル表現の部分が、つぶされ視覚的に表示されないようにするステップ
さらに含む、コンピューターが実行する方法。
【請求項１１】
請求項１から９のうちの何れか一項に記載のコンピューターが実行する方法であって、
ソースコードの選択した前記部分のうちの部分が、つぶされ視覚的に表示されないようにするステップ
さらに含む、コンピューターが実行する方法。
【請求項１２】
少なくとも１つのプロセッサーとメモリとを含むデバイスであって、前記少なくとも１つのプロセッサーは、
ソースコードをソースコードエディターにおいて見せ、

見せた前記ソースコードの部分を選択し、
ソースコードの選択した前記部分についての演算子の優先順位を示すグラフィカル表現を生成し、演算子の前記優先順位は、ソースコードの選択した前記部分のある演算子が、ソースコードの選択した前記部分の他の演算子に対して、ソースコードの選択した前記部分が実行されたときにどのように実行されるかの優先順位を表し、演算子の前記優先順位は前記ソースコードに関連した言語モデルに基づき決定され、
演算子の前記優先順位の前記グラフィカル表現を、ソースコードの選択した前記部分の視覚的ディスプレイとともにインラインで表示する
ように構成された、デバイス。
【請求項１３】
請求項１２に記載のデバイスであって、前記少なくとも１つのプロセッサーは、
演算子の前記優先順位の前記グラフィカル表現を、視覚的に表示される前記ソースコードの選択した前記部分とは別のウィンドウにおいて表示する
ようにさらに構成された、デバイス。
【請求項１４】
請求項１２に記載のデバイスであって、前記少なくとも１つのプロセッサーは、
選択的に、演算子の前記優先順位の前記グラフィカル表現の部分を、ゴースト化されたグラフィック要素と置き換えることを可能にする
ようにさらに構成された、デバイス。
【請求項１５】
請求項１２から１４のうちの何れか一項に記載のデバイスであって、前記少なくとも１つのプロセッサーは、
選択的に、演算子の前記優先順位の前記グラフィカル表現の部分が、つぶされ視覚的に表示されないようにすることを可能にする
ようにさらに構成された、デバイス。
【請求項１６】
請求項１２から１４のうちの何れか一項に記載のデバイスであって、前記少なくとも１つのプロセッサーは、
選択的に、ソースコードの選択した前記部分のうちの部分が、つぶされ視覚的に表示されないようにすることを可能にする
ようにさらに構成された、デバイス。
【請求項１７】
少なくとも１つのコンピューティングデバイスを含むシステムであって、前記少なくとも１つのコンピューティングデバイスは少なくとも１つの処理装置とメモリとを含み、前記少なくとも１つの処理装置は、
ソースコードをソースコードエディターにおいて見せ、
見せた前記ソースコードの部分を選択し、
見せた前記ソースコードの選択した前記部分に関連した構文木を取得し、
見せた前記ソースコードの選択した前記部分についての演算子の優先順位を示すグラフィカル表現を、前記構文木から生成し、演算子の前記優先順位は、見せた前記ソースコードの選択した前記部分のある演算子が、見せた前記ソースコードの選択した前記部分の他の演算子に対して、見せた前記ソースコードの選択した前記部分が実行されたときにどのように実行されるかの優先順位を表し、演算子の前記優先順位は前記ソースコードに関連した言語モデルに基づき決定され、
前記グラフィカル表現に従う演算子の前記優先順位を表示し、
演算子の前記優先順位の前記グラフィカル表現を、見せた前記ソースコードの選択した前記部分とともにインラインで表示する
ように構成された、システム。
【請求項１８】
請求項１７に記載のシステムであって、前記少なくとも１つの処理装置は、前記ソース

コードを、ソースコード・ビューワーにおいて、ウェブブラウザを介して見せるようにさらに構成された、システム。
【請求項１９】
請求項１７に記載のシステムであって、前記少なくとも１つの処理装置は、見せた前記ソースコードの選択した前記部分に関連した前記構文木を、遠隔のソースから取得するようにさらに構成された、システム。
【請求項２０】
請求項１７に記載のシステムであって、演算子の前記優先順位の前記グラフィカル表現は、線表現形式と、ツリー表現形式と、色付きテキスト表現形式と、数値による順序表現形式と、並列表現形式とのうちの少なくとも１つを含む、システム。
【請求項２１】
請求項１７から２０のうちの何れか一項に記載のシステムであって、前記少なくとも１つの処理装置は、
選択的に、演算子の前記優先順位の前記グラフィカル表現の部分が、つぶされ視覚的に表示されないようにすることを可能にする
ようにさらに構成された、システム。
【請求項２２】
請求項１７から２０のうちの何れか一項に記載のシステムであって、前記少なくとも１つの処理装置は、
選択的に、ソースコードの選択した前記部分のうちの部分が、つぶされ視覚的に表示されないようにすることを可能にする
ようにさらに構成された、システム。
【発明の詳細な説明】
【背景技術】
【０００１】
[0001]コンピュータープログラムは、演算子の優先順位に従って式内の演算を実行する。演算子の優先順位は、式内のいずれの演算子が他の演算子に優先するかを指定する規則の集合である。例えば、数式２＋３×４は、加算演算子が乗算演算子に優先すると評価され得る場合、それによって値２０を生じる（すなわち、（２＋３）×４＝２０）。あるいは、同一の数式が、乗算演算子が加算演算子に優先すると評価される場合、それによって値２４を生じる（すなわち、２＋（３×４）＝２４）。演算子の優先順位を指定することがなければ矛盾した結果が生成される可能性がある。
【０００２】
[0002]演算子の優先順位は、多くの場合、コンピュータープログラム（例えば、アプリケーション、スクリプト、など）が書かれているプログラミング言語の文法により指定される。各プログラミング言語は、異なる演算子の優先順位を使用して式を評価することができる。異なる演算子の優先順位は、ユーザーが不慣れなプログラミング言語で書かれたソースコードを見たり編集したりする場合にユーザーに問題を引き起こす恐れがある。さらに、特定のプログラミング言語の文法の微妙な差異は複雑であり、それによって演算子の優先順位付けを確認することを困難とする恐れがある。
【発明の概要】
【発明が解決しようとする課題】
【０００３】
[0003]本概要は、以下の発明を実施するための形態でさらに説明される概念から選択したものを簡単化した形で紹介するために提供される。本概要は、特許請求する主題の主たる特徴または本質的特徴を同定することを意図するものでも、また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したものでもない。
【０００４】
[0004]ソースコードは、演算が実行される態様を指定する言語モデルまたは文法を有するプログラミング言語で書かれる。特に、演算子の優先順位は、プログラミング言語中で

使用される他の演算子に対する演算子の優先順位を指定する。演算子の優先順位のグラフィカル表現は、演算が実行される順序をユーザーが理解することを可能とするためにソースコードと共に表示される。このことは、ソースコードの開発および保守の際に有益である。

【課題を解決するための手段】

【０００５】

[0005]グラフィカル表現は、複数の異なった形式で視覚的に表示され得る。グラフィカル表現の様々な表示形式は、線表現、ツリー表現、数値による順序付け、色で強調されたテキスト、色で強調された演算子、などの形を取ることができる。グラフィカル表現は、結果に影響を与えることなく任意の順序で実行され得る演算を表示することができる。グラフィカル表現の部分は、特定の文字でつぶされかつ／またはゴースト化され得る。

【０００６】

[0006]これらのおよび他の特徴および利点は、以下の詳細な説明を読むことおよび添付の図面を見直すことから明らかになろう。前述の一般的な説明と以下の詳細な説明は共に単に説明のためのものでありかつ特許請求される態様を制限するものではないことを理解すべきである。

【図面の簡単な説明】

【０００７】

【図１】[0007]演算子の優先順位のグラフィカル表現を生成しかつ表示するための第１の例示的システムを示す図である。

【図２】[0008]演算子の優先順位のグラフィカル表現を生成しかつ表示するための第２の例示的システムを示す図である。

【図３】[0009]図３Ａ～３Ｆは、演算の逐次的実行のために書式設定された演算子の優先順位のグラフィカル表現のある実施形態を示す図である。

【図４】[0010]図４Ａ～４Ｊは、演算の並列および逐次的実行のために書式設定された演算子の優先順位のグラフィカル表現のある実施形態を示す図である。

【図５】[0011]式内の演算子のみが演算子の優先順位のグラフィカル表現内に表示されている式のグラフィカル表現を示す図である。

【図６】[0012]演算子の優先順位のグラフィカル表現の部分が選択的にゴースト化されかつつぶされていることを示す図である。

【図７Ａ】[0013]ディスプレイ上の演算子の優先順位のグラフィカル表現の配置のある実施形態を示す図である。

【図７Ｂ】ディスプレイ上の演算子の優先順位のグラフィカル表現の配置のある実施形態を示す図である。

【図７Ｃ】ディスプレイ上の演算子の優先順位のグラフィカル表現の配置のある実施形態を示す図である。

【図８】[0014]動的に生成される演算子の優先順位のグラフィカル表現を示す図である。

【図９】[0015]表示されるソースコードの一部のグラフィカル表現を生成するための例示的方法を示す流れ図である。

【図１０】[0016]動作環境を示すブロック図である。

【図１１】[0017]第１の例示的コンピューティングデバイスを示すブロック図である。

【図１２】[0018]第２の例示的コンピューティングデバイスを示すブロック図である。

【図１３】[0019]例示的サーバーを示すブロック図である。

【発明を実施するための形態】

【０００８】

[0020]様々な実施形態は、ソースコードの選択された部分の演算子の優先順位のグラフィカル表現の生成および表示に関連するものである。演算子の優先順位のグラフィカル表現は、ユーザー（すなわち、開発者、プログラマーなど）がソースコードの選択された部分の実行の流れをより十分に理解することを支援するために視覚的に表示され得る。グラフィカル表現は、ソースコードと同時に同一のウインドウ内にまたは別のウインドウ内に

複数の異なる形式で視覚的に表示され得る。グラフィカル表現の様々な表示形式は、線表現、ツリー表現、数値による順序付け、色で強調されたテキスト、色で強調された演算子、などの形を取ることができる。演算子の優先順位のグラフィカル表現は、演算がソースコードの実行の際に行われる順序を理解するためにユーザーがソースコードを見ることを支援する。この知識は、所期の目的に対してソースコードをデバッグし、テストし、かつ／または編集するためにユーザーにより使用され得る。

【０００９】

[0021]ここで、演算子の優先順位のグラフィカル表現を利用する第１の例示的システムの考察に注意を向ける。図１を参照すると、演算子の優先順位のグラフィカル表現を生成しかつ表示するための例示的システム１００のブロック図が示されている。図１に示されるシステム１００は、特定の接続形態において限定された数の要素を有しているが、システム１００が、所与の実装に対して望まれる別の接続形態においてより多くの要素またはより少ない要素を含み得ることが理解されるであろう。

【００１０】

[0022]システム１００は、限定されないが、モバイル装置、携帯情報端末、モバイルコンピューティングデバイス、スマートフォン、携帯電話、携帯コンピューター、サーバー、サーバーアレイまたはサーバーファーム、ウェブサーバー、ネットワークサーバー、インターネットサーバー、ワークステーション、ミニコンピューター、メインフレームコンピューター、スーパーコンピューター、ネットワーク装置、ウェブ装置、分散型計算機システム、マルチプロセッサーシステムまたはこれらの組み合わせなどの、プログラム可能な命令を実行する能力がある任意の形式の電子装置であり得る、コンピューティングデバイス１０２を含むことができる。

【００１１】

[0023]コンピューティングデバイス１０２は、ソースコード１０８を表示する能力があるソースコード・ビューワー１０４および編集エンジン１０６を含むことができる。ソースコード・ビューワー１０４は、ユーザー（例えば、プログラマー、開発者など）がソースコード１０８を見ることを可能とするソフトウェアアプリケーションであり得る。編集エンジン１０６は、ソースコードを見る、編集する、かつ／または生成するためにプログラマーにより使用されるソフトウェアアプリケーションであり得る。ソースコード１０８は、処理装置により実行される時、定められた仕事に従った方法および／または動作を処理装置に行わせる、一連のコンピュータープログラム命令であり得る。ソースコード１０８は、任意の適切な高級、低級、オブジェクトオリエンテッド、ビジュアル、コンパイルされた、かつ／またはインタープリットされたプログラミング言語を使用して実装され得る、ソフトウェアアプリケーション、プログラムコード、プログラム、プロシージャ、モジュール、コードセグメント、プログラムスタック、ミドルウェア、ファームウェア、メソッド、ルーチン、ウェブページ、実行可能コード、スクリプトファイル、などであり得る。

【００１２】

[0024]ユーザーは、演算子の優先順位をユーザーが見たいと望むソースコード１０８の部分を選択することができると共に、コードはソースコード・ビューワー１０４または編集エンジン１０６内で見ることができる。入力セレクター１１０は、演算子の優先順位に関心があるソースコードの部分をユーザーが特定することに利用され得る。入力セレクター１１０は、カーソル位置１１２および／または蛍光ペン１１４であり得る。カーソル位置１１２は、カーソルが置かれている位置であり、関心があるソースコードの一部を特定する。蛍光ペン１１４は、ソースコードの網掛けされた部分であり、関心のあるソースコードを特定することができる。入力セレクター１１０は、ユーザーが関心のあるソースコードを特定するために使用することができる任意の機構であり得ることに留意すべきである。

【００１３】

[0025]入力アナライザー１１６は、ユーザーまたはプログラムが演算子の優先順位解析

のために選択したソースコードの一部を受信する。入力アナライザー１１６はまた、言語モデルおよびユーザーのオプションに基づき解析に必要な文全体を見つけ出すことができる。入力アナライザー１１６は、ソースコードの静的に選択された部分をソースコード・ビューワー１０４から受信する。ソースコードの静的に選択された部分は、静的なソースコードリストから特定される。ソースコードの静的に選択された部分は、カーソル位置、強調されたテキストなどにより指定され得る。入力アナライザー１１６は、指定されているソースコードの量が、言語モデルまたはユーザーのオプションで指定される、完全な式または文ではないことを認識することができる。この場合には、入力アナライザー１１６は、式または文を完全なものとするために追加のテキストまたは文字をソースコードの選択された部分に組み込むことができる。

【００１４】

[0026]入力アナライザー１１６はまた、ソースコードの動的に選択された部分を動的モジュールセレクター１０７から受信することができる。ソースコードの動的に選択された部分は、ユーザーはソースコードを生成し、ソースコードをテストし、かつ／またはソースコードを編集している間に、実時間で特定される。

【００１５】

[0027]動的モジュールセレクター１０７は、ソースコードのどの部分に対応する演算子の優先順位をユーザーが表示したいかを決定する。いくつかの事例では、ユーザーは、強調されたテキストによりソースコードの一部を特定することができる。別の事例では、動的モジュールセレクター１０７は、カーソル位置１１２などの入力セレクター１１０からソースコードの一部を決定しなければならない。例えば、カーソル位置１１２が、ソースコードの行の中央に置かれている場合、動的モジュールセレクター１０７は、完結した式が得られるようにソースコードの一部に行内のどの文字が含まれるかを決定する。

【００１６】

[0028]さらに、動的モジュールセレクター１０７は、ユーザーが演算子の優先順位解析のために選択しているソースコードの部分を推測しなければならないかもしれない。編集エンジンなどによる、文脈の実時間編集においては、ユーザーが演算子の優先順位解析のためにソースコードの一部を選択していたとしても、ユーザーは、ソースコードのタイピングまたは編集を行うことを終了していない可能性がある。選択され得るソースコードの一部は、言語モデルまたはユーザーのオプションに従った完結した式または文ではない。この状況では、動的モジュールセレクター１０７は、追加の文字を含めるために、ユーザーがタイピングを続け、かつ論理的コード区切りをタイプすることなどを待つべきか否かを決定するために、ユーザーの動作を推測しなければならない。

【００１７】

[0029]例えば、動的モジュールセレクター１０７の役割を示すために、ソースコードの以下の部分を考えることとする。

【００１８】

【数１】

```
void cClass:: DoSomething (int *a, int b, int c, int d) {          (1)
    a += 2;                                                        (2)
    CrashSystem();                                                 (3)
    int e = *a++;                                                  (4)
    int f = *a | ++;                                               (5)
    if ( c + f > b * c * d) {                                      (6)
        OopsDontCrashSystem();                                     (7)
```

【００１９】

[0030]行（５）において、カーソル「｜」が、文字「ａ」の後ろ、「＋＋」の前の行内の中央に位置している。動的モジュールセレクター１０７は、このカーソル位置の後ろに追加のテキストがあることを認識し、かつ完結した式が含まれるように行（５）上のテキストの残りをソースコードの選択された部分内に含めることになろう。従って、動的モジュールセレクター１０７は、ソースコードの選択された部分が「ｉｎｔ　ｆ＝＊ａ＋＋；」であると決定することになる。入力アナライザー１１６が、上記のコード例で指摘された、ソースコードの静的に選択された部分に追加のテキストおよび文字を捕捉するための同一の能力を有することに留意すべきである。

【００２０】

[0031]ソースコードの選択された部分は、選択されたコード部分の構文構造を表すデータ構造を生成するために構文解析部１１８により利用され得る。構文解析部１１８は、選択されたコード部分の構文構造を決定するために、ソースコードが書かれているプログラミング言語の文法などの、言語モデル１２２を利用する。言語モデル１２２または文法は、トークンの集合をプログラミング言語の構文構造に変換するために使用される規則の集合を含む。式を評価するために使用される演算子の優先順位は、文法において固有である。

【００２１】

[0032]構文解析部１１８は、選択されたコード部分の構文構造を表す構文木を生成することができる。構文木は、変数を表す葉ノードおよび式内の変数に適用される演算子を表す分岐点を含むことができる。あるいは、構文解析部１１８は、限定されないが、ハッシュテーブル、ベクトルなどの、選択されたコード部分の構文構造を表すための他のデータ構造を利用することもできる。

【００２２】

[0033]構文解析部１１８は、グラフィカル表現に使用される演算子の優先順位をカスタマイズするために使用される１つまたは複数の構文解析部オプション１２０を受信することができる。例えば、ユーザーは、関数名の評価がソースコードの選択された部分に対する演算子の優先順位で考慮されるべきではないことを構文解析部オプション１２０により指定することができる。同様に、ユーザーは、ソースコードの選択された部分が５０文字未満の文字を含むコードの行を含むべきではないことを構文解析部オプション１２０により指定することができる。その他のユーザー選択もまた、構文解析部オプション１２０により指定され得る。

【００２３】

[0034]グラフィカル表現モジュール１２４は、演算子の優先順位のグラフィカル表現を生成するために構文木を利用する。グラフィカル表現モジュール１２４は、ユーザーが望むグラフィカル表現の形式を指定するグラフィカル表現オプション１２６を受信すること

ができる。例えば、グラフィカル表現オプション１２６は、線表現形式、ツリー表現形式、数値による順序付け形式、色付けテキスト形式、フォント型テキスト形式、などを含むことができる。さらに、グラフィカル表現オプション１２６は、優先順位を意味するために使用される形式の態様を指定することができる。例えば、線表現形式は、各演算上に階層的順序で置かれる線分を有する。演算の優先順位は、最下位線分が最初に演算されることを意味し、かつ最下位線分の上の各線分が最下行の上に示された順序で実行される、下から上への順序で各線分を配置することにより指定され得る。同様に、上から下への順序では、最上部の線分が、最初に演算されることを意味すると指定され得て、かつ最上部の線分の下の各線分が、最上部の線分の下に示される順序で実行される。

【００２４】

[0035]グラフィックエンジン１２８は、ユーザーのためにディスプレイ１３０上にグラフィカル表現を表示するために利用され得る。グラフィックエンジン１２８は、ディスプレイ１３０上のウインドウ内へ、演算子の優先順位のグラフィカル表現などのオブジェクトの描画を管理するソフトウェアアプリケーションであり得る。ディスプレイ１３０は、コンピューティングデバイス１０２のスクリーンまたは視覚的ディスプレイ装置である。

【００２５】

[0036]図２は、演算子の優先順位のグラフィカル表現を生成しかつ表示するための第２の例示的システム２００のブロック図を示す。図２に示されるシステム２００は、特定の接続形態において限定された数の要素を有しているが、システム２００が、所与の実装に対して望まれる別の接続形態においてより多くの要素またはより少ない要素を含み得ることが理解されるであろう。

【００２６】

[0037]システム２００では、サーバー２０２は、サーバー２０２が提供するソースコード・ビューワー１０４によりユーザーが見ることができる大量のソースコードを格納するために利用され得る。ユーザーは、サーバーのソースコード・ビューワー１０４によりソースコードを見るためにユーザーのコンピューティングデバイス２０１からウェブブラウザ２０６を利用することができる。ソースコードがサーバー２０２内に格納されている場合、構文解析部１１８は、サーバーの構文木レポジトリ２１０内に格納されているソースコードの構文木または他の構文表現を生成するために使用され得る。ユーザーが、ソースコードの一部に対する演算子の優先順位を見たいと望む場合、ウェブブラウザ２０６は、サーバー２０２から対応する構文木を要求することができる。サーバー２０２は、グラフィカル表現モジュール１２４がユーザーのグラフィカル表現オプション１２６に従ってディスプレイ１３０上に演算子の優先順位を生成するために使用することができる、構文木をユーザーのコンピューティングデバイス１０２に提供することができる。

【００２７】

[0038]図２を参照すると、システム２００は、ネットワーク２０４を介してサーバー２０２に通信可能に結合されているコンピューティングデバイス２０１を含むことができる。

コンピューティングデバイス２０１およびサーバー２０２は、限定することなく、モバイル装置、携帯情報端末、モバイルコンピューティングデバイス、スマートフォン、携帯電話、携帯コンピューター、サーバー、サーバーアレイまたはサーバーファーム、ウェブサーバー、ネットワークサーバー、インターネットサーバー、ワークステーション、ミニコンピューター、メインフレームコンピューター、スーパーコンピューター、ネットワーク装置、ウェブ装置、分散型計算機システム、マルチプロセッサーシステム、またはこれらの組み合わせなどの、プログラム可能な命令を実行する能力のある任意の形式の電子装置であり得る。ネットワーク２０４は、コンピューティングデバイス２０１とサーバー２０２間の電子的通信を容易にする任意の形式の通信基盤であり得て、かつ図１０に示される通信フレームワークに関連して以下にさらに詳細に説明される。

【００２８】

[0039]コンピューティングデバイス２０１は、ウェブブラウザ２０６、グラフィカル表

現モジュール１２４、グラフィカル表現オプション１２６、グラフィックエンジン１２８およびディスプレイを含むことができる。ウェブブラウザ２０６は、サーバー２０２上でソースコード・ビューワー１０４により提供されるソースコードを見るために使用され得る。ユーザーは、ウェブブラウザ２０６を通じてソースコードの特定の部分に対する演算子の優先順位を要求することができる。ソースコードの構文木または構文表現は、サーバー２０２から得られ得る。グラフィカル表現モジュール１２４は、構文木を受信し、かつグラフィックエンジン１２８によりディスプレイ１３０上に表示される対応した演算子の優先順位を生成することができる。

【００２９】

[0040]サーバー２０２は、ソースコード・ビューワー１０４、入力アナライザー１１６、構文解析部１１８、構文解析部オプション１２０、言語モデル１２２、ソースコード・レポジトリ２０８、および構文木レポジトリ２１０を含むことができる。ソースコードが、ソースコード・レポジトリ２０８に格納するためにサーバー２０２にアップロードされる毎に、構文解析部１１８は、ソースコードに対する構文木を生成するために起動され得る。構文解析部オプション１２０は、ソースの任意の部分が修正されている場合、あるいはソースコードが格納され、見られ、または編集される所定の時点で、構文解析部１１８が構文木を再生成することができることを指定することができる。本実施形態は、この態様に限定されない。

【００３０】

[0041]構文木は、構文木レポジトリ２１０に格納され得る。ソースコード・ビューワー１０４は、構文木レポジトリ２１０から引き出され得るソースコードの特定の部分に関連した構文木に対する要求を受信することができる。入力アナライザー１１６、構文解析部１１８、構文解析部オプション１２０および言語モデル１２２は、図１に関連して上記に説明した同様の態様で動作する。

【００３１】

[0042]図１および図２に示されたシステム１００、２００は、特定の構成内に限定された数の要素を有するが、システム１００、２００が、別の構成においてより多くの要素またはより少ない要素を含むことができることが理解されるべきである。例えば、動的モジュールセレクター１０７、入力アナライザー１１６、構文解析部１１８、および／またはグラフィカル表現モジュール１２４は、ソースコード・ビューワー１０４、編集エンジン１０６、統合開発環境、オペレーティングシステム、クラウドサービス、コンパイラー、クラウドサービス用キャッシュ、電子メールクライアント、オペレーティングシステム、および／またはこれらの組み合わせに組み込まれ得る。さらに、ユーザーは、図２に示されるサーバー２０２上で提供されるソースコードにアクセスするために、コンピューティングデバイス１０２上で、ウェブエディター、ウェブサービスクライアント、などを利用することができる。

【００３２】

[0043]他の実施形態では、図２に示されるシステム２００の要素は、別の構成で配置され得る。例えば、サーバー２０２は、ウェブブラウザ２０６またはクライアント装置１０２上に存在する他のアプリケーションを通じてアクセスされ得る編集エンジン１０６を提供することができる。別の例では、ソースコードは、サーバー２０２に格納され、かつコンピューティングデバイス２０１上に存在するソースコード・ビューワー、編集エンジン、ウェブエディターなどを通じて見ることができる。コンピューティングデバイスからソースコードにアクセスする編集エンジンの場合には、動的モジュールセレクター１０７は、コンピューティングデバイス２０１またはサーバー２０２いずれかに存在し得る。他の実施形態では、ソースコードは、コンピューティングデバイス２０１内に格納され得て、かつソースコードに関連付けられた構文木は、サーバー２０２上に格納され得る。コンピューティングデバイス２０１は、ソースコードの開発および保守の際の様々な時点において、サーバー上の構文解析部が関連した構文木を生成するために、ソースコードをサーバー２０２に送信することができる。本実施形態において、サーバー２０２は、必要とされ

る場合に、コンピューティングデバイス２０１の使用のために構文木を生成し、格納し、かつ提供するように使用される。

【００３３】

[0044]様々な実施形態では、本明細書に記載されたシステム１００、２００は、複数の要素、プログラム、プロシージャ、モジュールを有するコンピューター実装のシステムを含み得る。本明細書で使用されるように、これらの用語は、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、またはソフトウェアいずれかを含む、コンピューター関連のエンティティを指すことを意図している。例えば、要素は、プロセッサー、ハードディスクドライブ、（光および／または磁気記憶媒体の）複数の記憶装置ドライブ上で実行されるプロセス、オブジェクト、実行可能ファイル、実行スレッド、プログラム、および／またはコンピューターとして実装され得る。実例として、コンピューティングデバイス上で実行されるアプリケーションとコンピューティングデバイスは共に、要素であり得る。１つまたは複数の要素は、プロセスおよび／または実行スレッド内に存在し、かつ要素は、所与の実装で望まれる場合、１台のコンピューター上に局在される、かつ／または２つ以上のコンピューター間に分散され得る。本実施形態は、この態様に限定されない。

【００３４】

[0045]システム１００、２００の様々な要素は、様々な線または矢印によって示される様々な形式の通信媒体を介して通信可能に結合され得る。本要素は、互いの間で動作を調整することができる。調整は、一方向または双方向の情報交換を必要とし得る。例えば、本要素は、通信媒体を介して伝達される信号の形で情報を伝達することができる。情報は、様々な信号線に割り当てられた信号として実装され得る。このような割り当てにおいて、各メッセージは、信号である。しかしながら、別の実施形態では、代替としてデータメッセージを用いることができる。このようなデータメッセージは、様々な接続で送信され得る。例示的接続は、パラレルインターフェイス、シリアルインターフェイス、およびバスインターフェイスを含む。

【００３５】

[0046]ここで、グラフィカル表現の様々な実施形態の別の説明に注意を向ける。図３Ａを参照すると、Ｃプログラミング言語で書かれた、Ａ｜｜Ｂ＋Ｃ｜｜Ｄと記載している式３００を含む、ソースコードのある行が示されている。この式には、４つの変数、すなわちＡ、Ｂ、ＣおよびＤがある。この式には、２つの演算子、すなわち算術加算演算子「＋」および論理ＯＲ演算子「｜｜」がある。

【００３６】

[0047]図３Ｂは、式３００に対応した構文木３０２の例示的表示を示している。構文木３０２は、式３００内の４つの変数を含む葉ノードを含んでおり、かつ分岐点は、式３００内の演算子を含んでいる。演算子の優先順位は、終端節点から開始して根ノードへと構文木３０２を最下部から最上部まで通過することにより構文木３０２内に示される。この態様では、構文木３０２は、式３００に対する演算子の優先順位が、Ｂ＋Ｃを最初に計算し、次いでＡ｜｜（Ｂ＋Ｃ）を計算し、最後にＤ｜｜（Ａ｜｜（Ｂ＋Ｃ））を計算することを示している。

【００３７】

[0048]この式に対する演算子の優先順位のグラフィカル表現は、図３Ｃ～図３Ｆに示される任意の形式で表現され得る。例えば、図３Ｃを参照すると、表示ボックス３０４は、式３００に対する演算子の優先順位を示す線表現形式を示す。線表現形式は、階層的順序で表示される一連の線分から構成される。式に最も近い、最下部の線分は、線分下側の演算が最初に実行され、この線分より上位にある線分に関連した各演算が連続して続くことを意味する。特に、線分３０６は、式Ｂ＋Ｃが最初に実行されることを示し、線分３０８は、ＡとＢ＋Ｃの論理ＯＲ演算（すなわち、Ａ｜｜（Ｂ＋Ｃ））が２番目に実行され、かつ線分３１０は、Ｄと式Ａ｜｜（Ｂ＋Ｃ）の論理ＯＲ演算（すなわち、（（Ａ｜｜（Ｂ＋Ｃ））｜｜Ｄ）が最後に実行されることを示す。

【００３８】

[0049]図３Ｄは、演算子の優先順位がまた、表示ボックス３１２内に描かれたツリー表現形式３１４内に表示され得ることを示す。ツリー表現形式３１４は、式内の演算に対する実行順序の階層を描いている。表示ボックス３１２に示されるように、演算Ｂ＋Ｃが最初に実行され、変数Ａとの論理ＯＲ演算（すなわち、Ａ｜｜（Ｂ＋Ｃ））が続き、変数Ｂとの論理ＯＲ演算（すなわち、（（Ａ｜｜（Ｂ＋Ｃ））｜｜Ｄ））が続く。

【００３９】

[0050]図３Ｅは、演算子の優先順位が、表示ボックス３１６内に示される数値順序形式で表示され得ることを示す。数値順序形式は、演算子が実行される順序を表す数字を各演算子上に描く。表示ボックス３１６に示されるように、「＋」演算子の上に置かれた数字「１」で示される演算Ｂ＋Ｃが最初に実行される。変数Ａの隣の論理ＯＲ演算子「｜｜」上に置かれた数字「２」により示される、変数Ａとの論理ＯＲ演算（すなわち、Ａ｜｜（Ｂ＋Ｃ））が２番目に実行される。変数Ｄの隣の論理ＯＲ演算子「｜｜」上に数字「３」を配置することにより示される、変数Ｄとの論理ＯＲ演算（すなわち、（（Ａ｜｜（Ｂ＋Ｃ））｜｜Ｄ））は、３番目に実行される。

【００４０】

[0051]図３Ｆは、異なる演算子の優先順位を描くために演算子の優先順位が異なる色のテキストまたは異なるフォントスタイルのテキストを使用して表示され得ることを示す。例えば、赤色は、第１の演算を表し、青色は、実行されるべき第２の演算を表し、緑色は、第３の演算を表す、などと表し得る。あるいは、異なるフォントスタイルが異なる演算子の優先順位を表すために使用され得る。例えば、Ｔｉｍｅｓ　Ｎｅｗ　Ｒｏｍａｎフォントは、最初の演算を示すために使用され、Ｌｕｃｉｎｄａ　Ｈａｎｄｗｒｉｔｉｎｇフォントは、２番目の演算を示すために使用される、などであり得る。

【００４１】

[0052]例えば、Ａｒｉａｌフォントが、実行されるべき最初の演算を示すために使用され、Ｃｅｎｔｕｒｙ　Ｓｃｈｏｏｌｂｏｏｋフォントが、実行されるべき２番目の演算を示すために使用され、かつＢｌａｃｋａｄｄｅｒ　Ｓｃｒｉｐｔフォントが、実行されるべき３番目の演算を示すために使用されることを仮定する。表示ボックス３２４内で演算子の優先順位を示すためにこれらのフォントスタイルを使用すると、３２６の演算Ｂ＋Ｃは、最初に実行されることが示され、変数Ａとの論理ＯＲ演算（すなわち、Ａ｜｜（Ｂ＋Ｃ））３２８は、２番目に実行されることが示され、かつ変数Ｄとの論理ＯＲ演算（すなわち、（（Ａ｜｜（Ｂ＋Ｃ））｜｜Ｄ））３３０は、変数Ｂと共に３番目に実行されることが示される。

【００４２】

[0053]異なる色が、上記に説明したフォントの代わりに式の優先順位を示すために使用され得ることに留意されるべきである。例えば、式「Ｂ＋Ｃ」に対するテキストは、演算Ｂ＋Ｃが最初に実行されることを示す赤色で表示され得る。式「Ａ｜｜」に対するテキストは、ＡとＢ＋Ｃの論理ＯＲ演算が２番目に実行されることを示す青色で表示され得る。式「｜｜Ｄ」に対するテキストは、この演算が最後に実行されることを示す緑色で表示され得る。さらに、全てのテキスト、線分、数字などの色付けは、ユーザーにより指定された特定のパターン、網掛けの色またはユーザーがグラフィカル表現オプション１２６内で指定した他のパターンに従うことができる。

【００４３】

[0054]図４Ａ～図４Ｊは、演算が並列に実行され得る、Ｃ＋＋プログラミング言語で書かれた式を描く演算子の優先順位のグラフィカル表現の様々な実施形態を示す。図４Ａは、式（ＰＴＲ－＞Ａ）および（Ｙ．Ｖ＜６７）に適用される論理ＯＲ演算が１であるか０であるかを決定するブール式４００を示す。演算子の優先順位は、並列に実行され得て、それによって、図４Ｃ～図４Ｆに示されるように、並列演算として演算子の優先順位のグラフィカル表現を描くことを可能とする。図４Ｇ～図４Ｊは、同一のブール式４００を示し、演算子の優先順位は、同一の演算を逐次的に実行することを表す様々なグラフィカル表現で描かれている。

【００４４】
[0055]図４Ｂは、式４００に対応した構文木４０２を示す。構文木４０２は、図４Ａに示される式４００に対する演算子の優先順位を含む。この例では、式４００に対する演算子の優先順位は、ブール式（Ｙ．Ｖ＜６７）の計算が、ブール式（ＰＴＲ－＞Ａ）が評価されている間に同時に実行されることを示す構文木４０２により示される。次いで、２つの値の論理ＯＲ演算が、計算される。
【００４５】
[0056]図４Ｃは、ＰＴＲ－＞ＡおよびＹ．Ｖ＜６７が、論理的に同時に最初に計算されて、論理ＯＲ演算が続くことを示す、式４００に対する線表現形式を示す。図４Ｄは、式４００に対するツリー表現形式を示し、また図４Ｅは、式４００に対する数値形式を示す。図４Ｆは、異なるフォントスタイルを使用する演算子の優先順位を示す。しかしながら、色付きテキストもまた、演算子の優先順位を示すために図４Ｆに示される異なるフォントスタイルの代わりに使用され得る。例えば、赤色は、「ＰＴＲ－＞Ａ」および「Ｙ．Ｖ＜６７」を表示するために使用されて、これらの演算が最初に実行されることを示し得る。青色は、論理ＯＲ演算「｜｜」を表示するために使用されて、論理ＯＲ演算が最後に実行されることを示し得る。
【００４６】
[0057]図４Ｇは、ＰＴＲ－＞ＡおよびＹ．Ｖ＜６７が表示ボックス４１２内に逐次的態様で計算されることを示す、式４００に対する線表現形式を示す。図４Ｈは、式４００を逐次的に実行するためのツリー表現形式を表示ボックス４１４内に示す。図４Ｉは、式４００を逐次的に実行するための数値形式を表示ボックス４１６内に示す。図４Ｊは、異なるフォントスタイルを使用する演算子の優先順位を表示ボックス４１８内に示す。しかしながら、色付きテキストもまた、演算子の優先順位を示すために図４Ｊに示される異なるフォントスタイルの代わりに使用され得る。例えば、赤色は、「ＰＴＲ－＞Ａ」を表示するために使用され、この演算子の優先順位が最初に実行されることを示すことができ、また緑色は、式「Ｙ．Ｖ＜６７」が２番目に実行されることを示すために使用され得る。青色は、論理ＯＲ演算「｜｜」を表示するために使用されて、論理ＯＲ演算が最後に実行されることを示し得る。
【００４７】
[0058]図５は、式内の演算子のみが、演算子の優先順位を表すために強調されている、ある式に対する演算子の優先順位のグラフィカル表現を示している。図３Ｃ～図３Ｆおよび図４Ｃ～図４Ｊに示されるグラフィカル表現では、式全体のグラフィカル表現を仮定していた。図５では、演算子のみが、表示ボックス５０２内にグラフィカル表現で示されるように、演算子の優先順位を示すためにグラフィカルに表示されている。図５に示されるように、式５００は、ＩＦ（ＲＥＴ＝ＤＯＳＯＭＥＴＨＩＮＧ（））と記載している。この式内の演算子は、ＤＯＳＯＭＥＴＨＩＮＧ（）関数呼び出しおよびＩＦ文評価である。ＤＯＳＯＭＥＴＨＩＮＧ（）関数呼び出しは、関数呼び出し５０４が最初に実行されることを示す色またはパターンで強調され、またＩＦ文は、ＩＦ文５０６が２番目に評価されることを示す色またはパターンで強調されている。
【００４８】
[0059]図６は、演算子の優先順位のグラフィカル表現の一部を選択的に見せることの例示的図を示す。選択的に見せる能力は、特定の式をより綿密に、またはユーザーの好みにより見せる必要がある場合にユーザーにより利用され得る。この例示的図では、式６００は、ブール式ＰＴＲ－＞Ａ｜｜（Ｙ．Ｖ＜６７）である。表示ボックス６０２は、式６００に対応した演算子の優先順位に対する線表現形式を示す。表示ボックス６０４は、式（Ｙ．Ｖ＜６７）をゴースト化し、かつ式をたわいないグラフィック要素で置き換えることをユーザーが選択したことを示す。この例では、言葉「ｓｅｌｅｃｔ」が元々の式（Ｙ．Ｖ＜６７）を置き換え、文字「・」が元々の式に対する線表現上に置かれた。表示ボックス６０４は、式ＰＴＲ－＞Ａが表示の対象外とされたことを示す。ユーザーは、マウスの右ボタン操作でグラフィカル表現内の式のゴースト化を始動することができ、かつマウス

の左ボタン操作で式を対象外とすることを始動することができる。

【００４９】

[0060]図７Ａ～図７Ｃは、演算子の優先順位のグラフィカル表現がソースコードと同一のビュー内に置かれる例示的図を示す。図７Ａは、９行のソースコードを有する例示的ソースコードリスト７０２を示す。式（Ｅ＋Ｆ＞Ｂ＊Ｃ＊Ｄ）は、ユーザーがこの式に対する演算子の優先順位を知りたいと望んでいることを示すために下線が引かれている。式（Ｅ＋Ｆ＞Ｂ＊Ｃ＊Ｄ）がまた、色付きのテキストで強調され、表示され、またはこの式を特定するために任意の他の形式の態様で表示され得ることに留意すべきである。

【００５０】

[0061]図７Ｂは、式に対する演算子の優先順位を表示するために線表現形式が選択されていることを示す。演算子の優先順位は、表示ボックス７０４により示されるように行４および行５内の既存のコードを上書きすることにより関心のある式の直上に表示される。表示ボックス７０４は、不透明、部分的に透明、または完全に透明であり得る。図７Ｃは、関心のある式に対する線表現形式７０８が選択された式の直上の空白行に挿入された（７０６）ことを示す。

【００５１】

[0062]図８は、演算子の優先順位のグラフィカル表現が、ソースコードリストを表示しているウインドウに隣接している別のウインドウ内に表示される例示的図を示す。図８に示されるように、表示８００は、ソースコードのリストを表示する第１のウインドウ８０２、および第１のウインドウ８０２内に示されるソースコードの選択された部分に対する演算子の優先順位のグラフィカル表現を表示する第２のウインドウ８０４から成る。カーソル「｜」は、行５内の文字「Ａ」と文字「＋」間に位置付けられている。行５内のソースコードが選択され、ウインドウ８０４内に線表現形式で演算子の優先順位として表示されている。

【００５２】

[0063]本実施形態についての動作はさらに、様々な例示的方法を参照して説明され得る。特に明記されない限り、代表している方法が、提示された順序で、または任意の特定の順序で実行されなければならないという必要がないことを理解されるべきである。さらに、本方法に関連して説明された様々な取り組みは、逐次または並列の様式で、または逐次および並列の動作の組み合わせで実行され得る。本方法は、所与の設計および処理能力の制約群に対して望まれる、説明された実施形態または別の実施形態の１つまたは複数のハードウェア要素および／またはソフトウェア要素を使用して実装され得る。例えば、本方法は、論理デバイス（例えば、汎用または特定目的コンピューター）による実行のための論理（例えば、コンピュータープログラム命令）として実装され得る。

【００５３】

[0064]図９は、演算子の優先順位を生成するための例示的方法の流れ図を示す。方法９００が、本明細書に記載された１つまたは複数の実施形態により実行される動作の一部または全てを表し得ること、および本方法が、図９に説明されているものより多くの動作または少ない動作を含むことができることに留意すべきである。

【００５４】

[0065]図９を参照すると、ユーザーは、解析のためにソースコードの一部を選択する（ブロック９０２）。ソースコードは、ユーザーによる操作によりソースコード・ビューワー１０４、編集エンジン１０６、またはウェブブラウザ２０６を通じて静的に選択され得る。ユーザーは、ソースコードの一部を強調または特定するために、カーソル、マウスのクリック、またはキー入力などの、入力セレクター１１０を利用する。ソースコードは、編集エンジン１０６または他のソフトウェアアプリケーションを通じて動的に選択され得る。この場合、動的モジュールセレクター１０７は、ユーザーが実時間でソースコードを編集する場合に、ユーザーの入力動作を追跡して、ユーザーが解析することを意図するソースコードの部分を決定するために利用され得る。ユーザーは、ユーザーが解析しようと意図するソースコードの一部を特定するために入力セレクター１１０を利用することがで

きる。しかしながら、動的モジュールセレクター１０７は、動作を完結させるためまたはより有意義な解析結果を得るために、ソースコードの選択された部分に追加の文字またはテキストを含めることができる。

【００５５】

[0066]複数のオプションが、演算子の優先順位のグラフィカル表現に関連して指定され得る（ブロック９０４）。ユーザーは、構文解析部オプション１２０およびグラフィカル表現オプション１２６を指定することができる。構文解析部オプション１２０は、ユーザーが見ることに興味がなく、構文解析部１１８が構文木を用意する場合に無視することができる、特定のソースコード文字、トークンまたは文を表示することができる。グラフィカル表現オプション１２６は、線表現、ツリー表現、色付きテキストなどのグラフィカル表現の形式を指定することができる。

【００５６】

[0067]構文解析部１１８は、ソースコードの選択された部分、構文解析部オプション１２０および言語モデル１２２を受信し、かつ構文木などのソースコードの選択された部分を表す構文構造を生成する（ステップ９０６）。１つまたは複数の実施形態において、構文解析部１１８は、ソースコードがサーバー２０２に格納される時、ソースコードが、全てまたは部分的に、あるいは構文解析部オプション１２０内に設定された設定に従って修正される時、などの指定された時点に構文木を生成することができる（ステップ９０６）。

【００５７】

[0068]グラフィカル表現モジュール１２４は、構文木およびグラフィカル表現オプション１２６を受信し、ソースコードの選択された部分に対する演算子の優先順位を決定し（ブロック９０８）、かつグラフィックエンジン１２８が意図した態様で演算子の優先順位を表示するために使用するグラフィック要素および／または命令を含むデータ構造を生成する（ブロック９１０および９１２）。

【００５８】

[0069]図１０を参照して、ここで例示的動作環境９４０の考察に注目を向ける。動作環境９４０は、例示的なものであり、かつ実施形態の機能に関して何らかの限定を提案することを意図するものではないことに留意すべきである。実施形態は、通信フレームワーク９４４を介して１台または複数台のサーバー（複数可）９４６と通信する１台または複数台のクライアント（複数可）９４２を含む動作環境９４０に適用され得る。動作環境９４０は、ネットワーク環境、分散された環境、マルチプロセッサー環境内に、あるいは遠隔またはローカルの記憶装置にアクセスする単独のコンピューティングデバイスとして構成され得る。

【００５９】

[0070]クライアント９４２は、ハードウェア装置、ソフトウェアモジュールとして、またはこれらの組み合わせとして具現化され得る。このようなハードウェア装置の例は、コンピューター（例えば、サーバー、パソコン、ラップトップ、など）、携帯電話、携帯情報端末、または任意の形式のコンピューティングデバイスなどを含み得るが、限定されない。クライアント９４２はまた、単一の実行パス、複数の並列実行パス（例えば、スレッド、プロセスなど）、または任意の他の態様で実行される命令を有するソフトウェアモジュールとして具現化され得る。

【００６０】

[0071]サーバー９４６は、ハードウェア装置、ソフトウェアモジュールとして、またはこれらの組み合わせとして具現化され得る。このようなハードウェア装置の例は、コンピューター（例えば、サーバー、パソコン、ラップトップ、など）、携帯電話、携帯情報端末、または任意の形式のコンピューティングデバイスなどを含むが、限定されない。サーバー９４６はまた、単一の実行パス、複数の並列実行パス（例えば、スレッド、プロセスなど）、または任意の他の態様で実行される命令を有するソフトウェアモジュールとして具現化され得る。

【００６１】
[0072]通信フレームワーク９４４は、クライアント９４２およびサーバー９４６間の通信を容易にする。通信フレームワーク９４４は、パケット交換ネットワーク（例えば、インターネットなどの公衆ネットワーク、企業イントラネットなどの私設ネットワーク、など）、回線交換ネットワーク（例えば、公衆交換電話網）、または（適切なゲートウェイおよび変換器を伴う）パケット交換ネットワークおよび回線交換ネットワークの組み合わせで使用するために適切な技術などの、任意のよく知られた技術を具現化することができる。クライアント９４２およびサーバー９４６は、１つまたは複数の通信インターフェイス、ネットワークインターフェイス、ネットワークインターフェイスカード、無線装置、無線送受信機、有線および／または無線通信媒体、物理的コネクターなどの、通信フレームワーク９４４と相互運用可能であるように設計されている様々な形式の標準的な通信要素を含むことができる。有線通信媒体の例は、電線、ケーブル、金属リード線、プリント回路基板、バックプレーン、スイッチファブリック、半導体材料、撚り線対電線、同軸ケーブル、光ファイバー、伝搬信号などを含むことができる。無線通信媒体の例は、音響、無線周波数スペクトラム、赤外線、および他の無線媒体を含むことができる。
【００６２】
[0073]各クライアント（複数可）９４２は、クライアント９４２に固有な情報を格納する１つまたは複数のクライアントデータストア（複数可）９４８に結合され得る。各サーバー（複数可）９４６は、サーバー９４６に固有な情報を格納する１つまたは複数のサーバーデータストア（複数可）９５０に結合され得る。
【００６３】
[0074]図１１は、例示的コンピューティングデバイス１０２のブロック図を示す。コンピューティングデバイス１０２は、１つまたは複数のプロセッサー９５２、ディスプレイ１３０、ネットワークインターフェイス９５４、メモリ９５６、およびユーザー入力インターフェイス９５８を有することができる。プロセッサー９５２は、任意の市販のプロセッサーであってもよく、かつデュアルマイクロプロセッサーおよびマルチプロセッサーアーキテクチャを含むことができる。ディスプレイ１３０は、モニター、スクリーン、タッチスクリーンなどの任意の視覚的ディスプレイ装置であり得る。ネットワークインターフェイス９５４は、コンピューティングデバイス１０２と他のネットワーク化された装置間の有線または無線通信を容易にする。ユーザー入力インターフェイス９５８は、コンピューティングデバイス１０２とキーボード、マウスなどの入力装置間の通信を容易にする。
【００６４】
[0075]メモリ９５６は、実行可能なプロシージャ、アプリケーション、およびデータを格納することができる任意のコンピューター読み取り可能な記憶媒体であり得る。コンピューター読み取り可能な媒体は、搬送波により送信される変調されたデータ信号などの、伝搬信号とは関係がない。コンピューター読み取り可能な媒体は、任意の形式の記憶装置（例えば、ランダムアクセスメモリ、リードオンリーメモリなど）、磁気記憶装置、揮発性記憶装置、不揮発性記憶装置、光記憶装置、ＤＶＤ、ＣＤ、フロッピー（登録商標）ディスクドライブなどであり得る。メモリ９５６はまた、１つまたは複数の外部記憶装置、または遠隔に位置する記憶装置を含むことができる。メモリ９５６は、以下の命令およびデータを含むことができる。
【００６５】
・オペレーティングシステム９６０
・ソースコード・ビューワー１０４
・編集エンジン１０６
・動的モジュールセレクター１０７
・入力アナライザー１１６
・構文解析部１１８
・グラフィカル表現モジュール１２４

・構文解析部オプション１２０
・言語モデル１２２
・グラフィカル表現オプション１２６
・グラフィックエンジン１２８
・様々な他のアプリケーションおよびデータ９６２

[0076]図１２は、例示的コンピューティングデバイス２０１のブロック図を示し、また図１３は、例示的サーバー２０２を示す。コンピューティングデバイス２０１は、１つまたは複数のプロセッサーまたは処理装置９６４、ディスプレイ１３０、ネットワークインターフェイス９６６、メモリ９６８、およびユーザー入力インターフェイス９７０を有し得る。プロセッサー９６４は、任意の市販のプロセッサーであってもよく、かつデュアルマイクロプロセッサーおよびマルチプロセッサーアーキテクチャを含むことができる。ディスプレイ１３０は、モニター、スクリーン、タッチスクリーンなどの任意の視覚的ディスプレイ装置であり得る。ネットワークインターフェイス９６６は、コンピューティングデバイス２０１および他のネットワーク化された装置との有線または無線通信を容易にする。ユーザー入力インターフェイス９７０は、コンピューティングデバイス２０１とキーボード、マウスなどの入力装置間の通信を容易にする。

【００６６】

[0077]メモリ９６８は、実行可能なプロシージャ、アプリケーション、およびデータを格納することができる任意のコンピューター読み取り可能な記憶媒体であり得る。コンピューター読み取り可能な媒体は、搬送波により送信される変調されたデータ信号などの、伝搬信号とは関係がない。コンピューター読み取り可能な媒体は、任意の形式の記憶装置（例えば、ランダムアクセスメモリ、リードオンリーメモリなど）、磁気記憶装置、揮発性記憶装置、不揮発性記憶装置、光記憶装置、ＤＶＤ、ＣＤ、フロッピー（登録商標）ディスクドライブなどであり得る。メモリ９６８はまた、１つまたは複数の外部記憶装置、または遠隔に位置する記憶装置を含むことができる。メモリ９６８は、以下の命令およびデータを含むことができる。

【００６７】

・オペレーティングシステム９７２
・ウェブブラウザ２０６
・編集エンジン１０６
・グラフィカル表現モジュール１２４
・グラフィカル表現オプション１２６
・グラフィックエンジン１２８
・様々な他のアプリケーションおよびデータ９７４

[0078]サーバー２０２は、１つまたは複数のプロセッサーまたは処理装置９７６、ディスプレイ１３０、ネットワークインターフェイス９７８、メモリ９８０、およびユーザー入力インターフェイス９８２を有し得る。プロセッサー９７６は、任意の市販のプロセッサーであってもよく、かつデュアルマイクロプロセッサーおよびマルチプロセッサーアーキテクチャを含むことができる。ディスプレイ１３０は、モニター、スクリーン、タッチスクリーンなどの任意の視覚的ディスプレイ装置であり得る。ネットワークインターフェイス９７８は、サーバー２０２と他のネットワーク化された装置間の有線または無線通信を容易にする。ユーザー入力インターフェイス９８２は、サーバー２０２とキーボード、マウスなどの入力装置間の通信を容易にする。

【００６８】

[0079]メモリ９８０は、実行可能なプロシージャ、アプリケーション、およびデータを格納することができる任意のコンピューター読み取り可能な記憶媒体であり得る。コンピューター読み取り可能な媒体は、搬送波により送信される変調されたデータ信号などの、伝搬信号とは関係がない。コンピューター読み取り可能な媒体は、任意の形式の記憶装置（例えば、ランダムアクセスメモリ、リードオンリーメモリなど）、磁気記憶装置、揮発性記憶装置、不揮発性記憶装置、光記憶装置、ＤＶＤ、ＣＤ、フロッピー（登録商標）デ

ィスクドライブなどであり得る。メモリ９８０はまた、１つまたは複数の外部記憶装置、または遠隔に位置する記憶装置を含むことができる。メモリ９８０は、以下の命令およびデータを含むことができる。

【００６９】

・オペレーティングシステム９８４
・ソースコード・ビューワー１０４
・入力アナライザー１１６
・構文解析部１１８
・構文解析部オプション１２０
・言語モデル１２２
・ソースコード・レポジトリ２０８
・構文木レポジトリ２１０
・様々な他のアプリケーションおよびデータ９８６

[0080]主題が、構造的特徴および／または方法論的行為に特有な言語で説明されてきたが、添付の特許請求の範囲内に定義された主題が、上記に説明された特定の特徴または行為に限定される必要がないことが理解されるべきである。むしろ、上記に説明された特定の特徴および行為は、特許請求の範囲を実施する形態の例として開示されている。

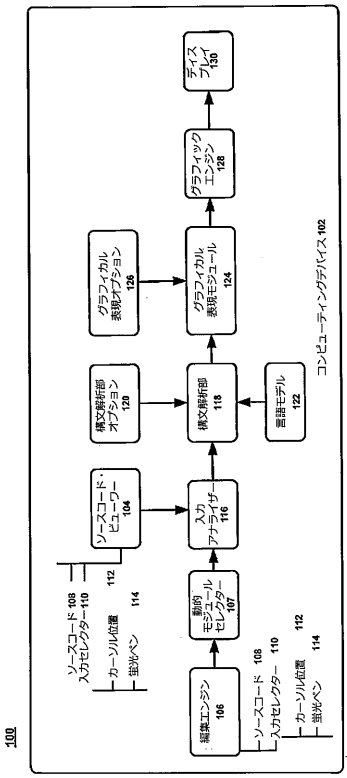
【００７０】

[0081]様々な実施形態は、ハードウェア要素、ソフトウェア要素、または両者の組み合わせを使用して実装され得る。ハードウェア要素の例は、装置、構成要素、プロセッサー、マイクロプロセッサー、回路、回路素子、集積回路、特定用途向け集積回路、プログラム可能論理デバイス、デジタル・シグナル・プロセッサー、フィールド・プログラマブル・ゲートアレイ、メモリ装置、論理ゲートなどを含むことができる。ソフトウェア要素の例は、ソフトウェア構成要素、プログラム、アプリケーション、コンピュータープログラム、アプリケーションプログラム、システムプログラム、機械語プログラム、オペレーティングシステムソフトウェア、ミドルウェア、ファームウェア、ソフトウェアモジュール、ルーチン、サブルーチン、関数、メソッド、プロシージャ、ソフトウェアインターフェイス、アプリケーションプログラムインターフェイス、命令セット、コンピューティングコード、コードセグメント、およびそれらの任意の組み合わせを含むことができる。実施形態が、ハードウェア要素および／またはソフトウェア要素を使用して実装されるか否かを決定することは、所与の実装に対して望まれる、所望の計算速度、電力レベル、帯域幅、演算時間、負荷バランス、メモリ資源、データバス速度および他の設計または処理能力の制約などの任意の数の要因に応じて変化する可能性がある。

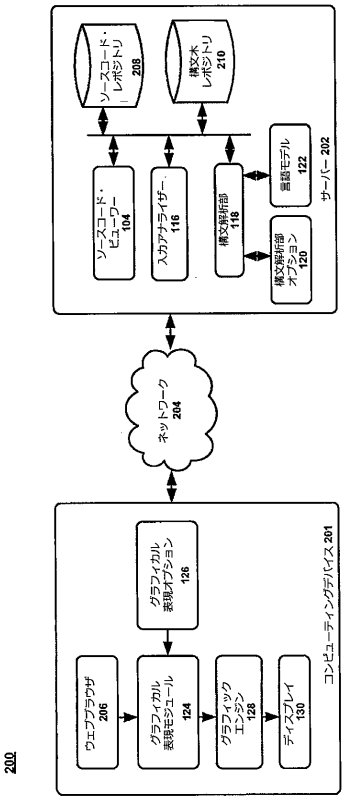
【００７１】

[0082]いくつかの実施形態は、命令またはロジックを格納するための記憶媒体を含むことができる。記憶媒体の例は、揮発性メモリまたは不揮発性メモリ、可換型または非可換型メモリ、消去可能または非消去可能メモリ、書き込み可能または書き換え可能なメモリなどを含む、電子データを格納する能力のある１つまたは複数の形式のコンピューター読み込み可能な記憶媒体を含むことができる。ロジックの例は、プログラム、プロシージャ、モジュール、アプリケーション、コードセグメント、プログラムスタック、ミドルウェア、ファームウェア、メソッド、ルーチンなどの、様々なソフトウェア要素を含むことができる。ある実施形態では、例えば、コンピューター読み取り可能な記憶媒体は、プロセッサーにより実行される場合、記載された実施形態に従ってメソッドおよび／または動作をプロセッサーに実行させる、実行可能なコンピュータープログラム命令を格納することができる。実行可能なコンピュータープログラム命令は、特定の機能をコンピューターに実行させることを指示するための、事前定義のコンピューター言語、態様または構文に従って実装され得る。命令は、任意の適切な高級、低級、オブジェクト指向、ビジュアルな、コンパイルされたおよび／またはインタープリットされたプログラミング言語を使用して実装され得る。

【図１】



【図２】



【図３Ａ】

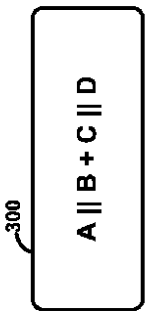


FIG. 3A

【図３Ｂ】

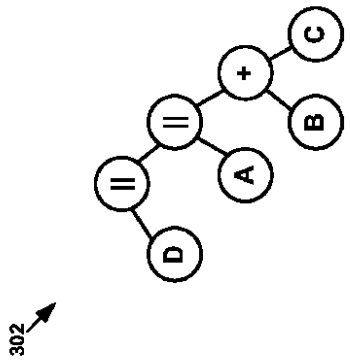


FIG. 3B

【図３Ｃ】

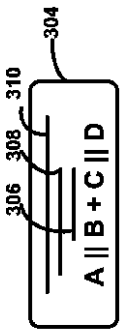


FIG. 3C

【図３Ｄ】

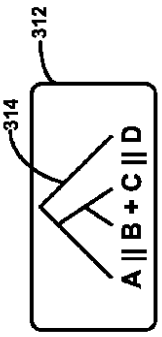


FIG. 3D

【図３Ｅ】

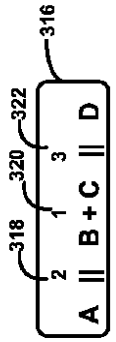


FIG. 3E

【図３Ｆ】

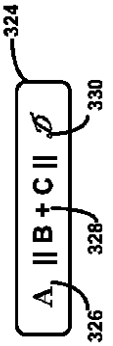


FIG. 3F

【図４Ａ】

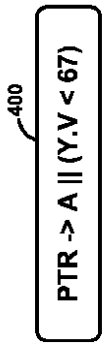


FIG. 4A

【図４Ｂ】

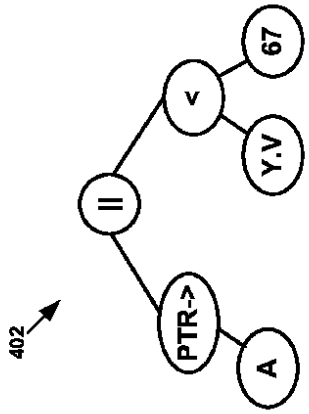


FIG. 4B

【図４Ｃ】

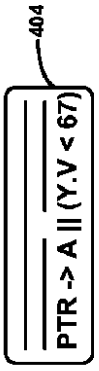


FIG. 4C

【図４Ｄ】

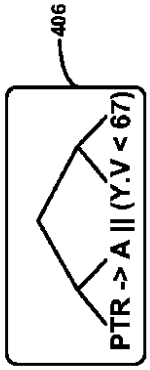


FIG. 4D

【図４Ｅ】

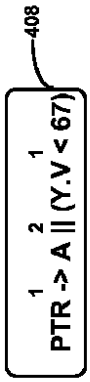


FIG. 4E

【図４Ｆ】

FIG. 4F

【図４Ｇ】

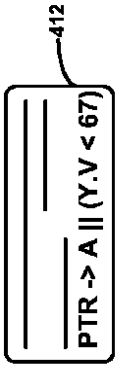


FIG. 4G

【図４Ｊ】

FIG. 4J

【図４Ｈ】

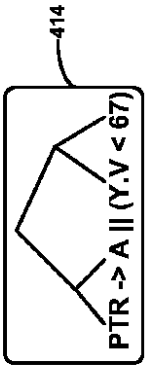


FIG. 4H

【図４Ｉ】

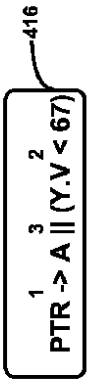


FIG. 4I

【図５】

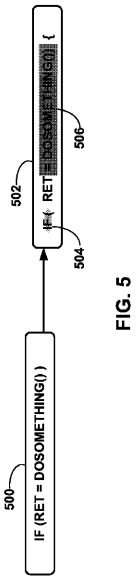


FIG. 5

【図６】

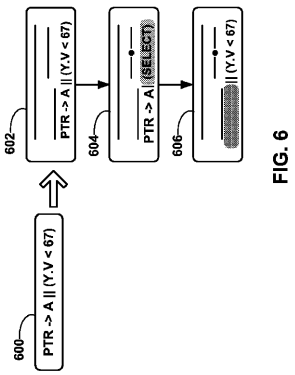


FIG. 6

【図７Ａ】

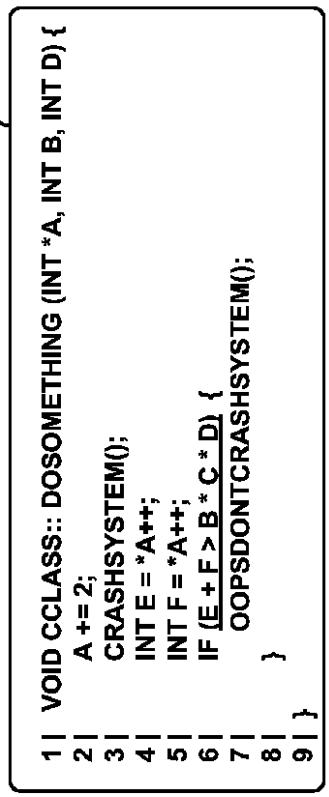


FIG. 7A

【図７Ｂ】

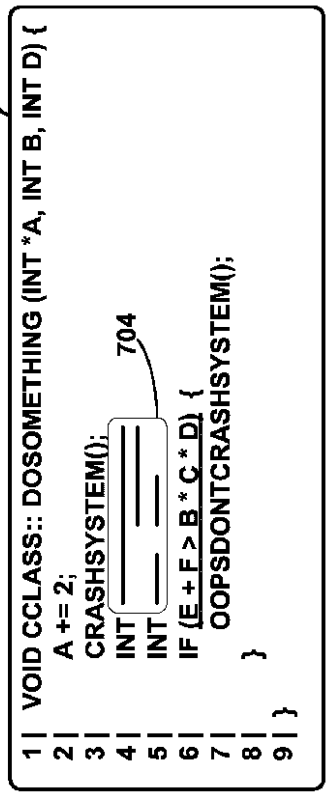


FIG. 7B

【図７Ｃ】

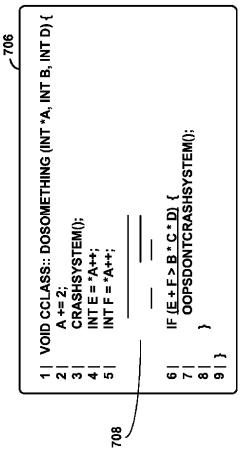
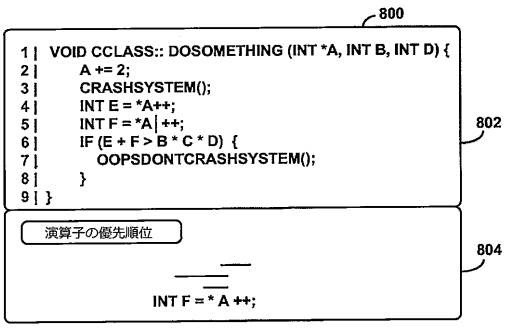
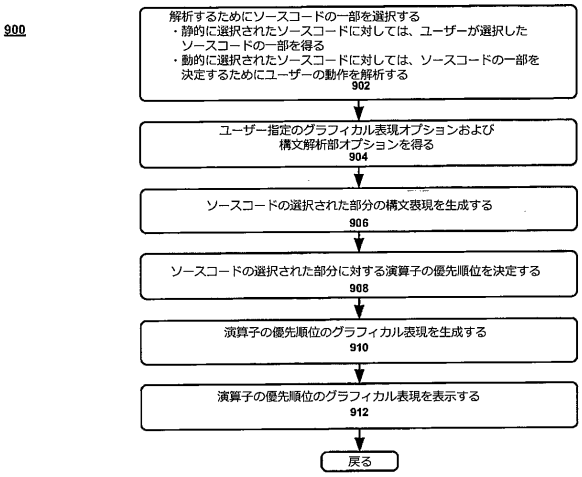


FIG. 7C

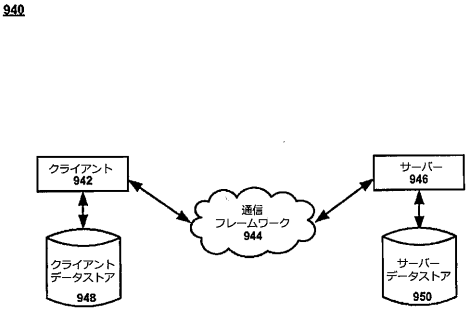
【図８】



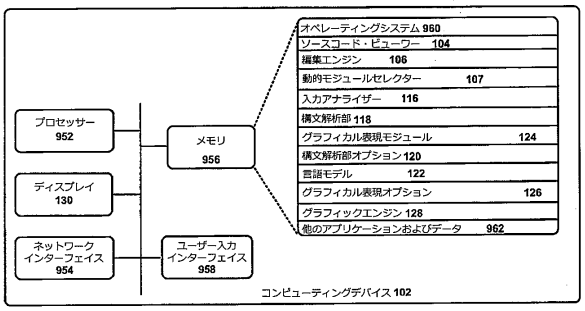
【図９】



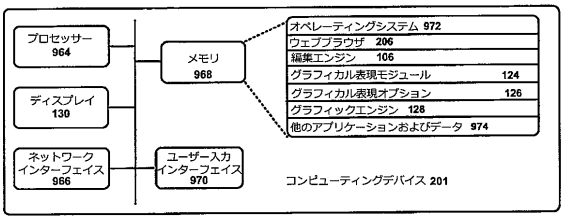
【図１０】



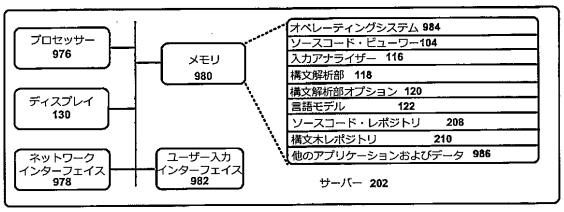
【図１１】



【図１２】



【図１３】



フロントページの続き

(74)代理人 100101373
弁理士 竹内 茂雄
(74)代理人 100118902
弁理士 山本 修
(74)代理人 100153028
弁理士 上田 忠
(72)発明者 ロヴィット，アンドリュー
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，エルシーエイ－インターナショナル・パテンツ

審査官 石川 亮

(56)参考文献 特開平０６－２３１０８１（ＪＰ，Ａ）
特開２００６－１９５８２７（ＪＰ，Ａ）
特開２００９－２６５９９６（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ９／４４
Ｇ０６Ｆ １５／０２
Ｇ０６Ｆ ３／０４８－３／０４８６