(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5123291号
(P5123291)

(45)発行日　平成25年1月23日(2013.1.23)　(24)登録日　平成24年11月2日(2012.11.2)

(51)Int.Cl.　　F I
G06F 12/00 (2006.01)　　G06F 12/00 547Q
G06F 9/44 (2006.01)　　G06F 9/06 620A

請求項の数 12　(全 24 頁)

(21)出願番号　特願2009-509563 (P2009-509563)
(86)(22)出願日　平成19年3月15日 (2007.3.15)
(65)公表番号　特表2009-535730 (P2009-535730A)
(43)公表日　平成21年10月1日 (2009.10.1)
(86)国際出願番号　PCT/US2007/006576
(87)国際公開番号　WO2007/130227
(87)国際公開日　平成19年11月15日 (2007.11.15)
審査請求日　平成22年2月26日 (2010.2.26)
(31)優先権主張番号　60/746,439
(32)優先日　平成18年5月4日 (2006.5.4)
(33)優先権主張国　米国 (US)
(31)優先権主張番号　11/464,874
(32)優先日　平成18年8月16日 (2006.8.16)
(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
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
(74)代理人　100077481
弁理士　谷　義一
(74)代理人　100088915
弁理士　阿部　和夫
(72)発明者　アンダース　ヘルスバーグ
アメリカ合衆国　９８０５２　ワシントン州　レッドモンド　ワン　マイクロソフト　ウェイ　マイクロソフト　コーポレーション　インターナショナル　パテンツ内

最終頁に続く

(54)【発明の名称】プログラミング言語における式ツリーの深い埋め込みのためのジェネリックインターフェイス

(57)【特許請求の範囲】
【請求項１】
式ツリーを介してクエリを埋め込み、データ操作を実行するためのアプリケーションプログラミングインターフェイスを使用してデータ操作を実行させるコンピュータ実行可能命令を記憶したコンピュータ可読記憶媒体であって、前記アプリケーションプログラミングインターフェイスはジェネリックインターフェイスを含み、前記コンピュータ実行可能命令が実行させる方法は、
コンピュータ上の処理装置が、前記ジェネリックインターフェイスを介して実行される標準クエリ演算子の拡張メソッドを使用したデータ操作の実行要求を受信するステップであって、前記標準クエリ演算子の拡張メソッドは、異なる種類の複数のデータストアのうちのいずれか１つに対して実行可能であり、かつ特定の種類のデータストアに固有のものではない、受信するステップと、
前記処理装置が、前記受信した実行要求に係るデータのコレクションおよびクエリに基づいて、前記式ツリーを作成するステップと、
前記処理装置が、前記ジェネリックインターフェイスを介して、前記標準クエリ演算子の拡張メソッドを使用して、データストアに対するデータ操作を実行するステップであって、前記ジェネリックインターフェイスの引数は、デリゲードではなく前記作成した式ツリーである、実行するステップと
を備えたことを特徴とするコンピュータ可読記憶媒体。
【請求項２】

前記異なる種類の複数のデータストアは、リレーショナル型データストア、ＸＭＬ（拡張可能なマーク付け言語）型データストア、およびオブジェクト指向型データストアを含むことを特徴とする請求項１に記載のコンピュータ可読記憶媒体。

【請求項３】

前記標準クエリ演算子の拡張メソッドは、Ｗｈｅｒｅ、Ｓｅｌｅｃｔ、ＳｅｌｅｃｔＭａｎｙ、フィルタ関数、グループ関数、および変換関数のうちの少なくとも１つを含むことを特徴とする請求項１に記載のコンピュータ可読記憶媒体。

【請求項４】

前記処理装置は、クエリプロセッサであることを特徴とする請求項１に記載のコンピュータ可読記憶媒体。

【請求項５】

前記標準クエリ演算子の拡張メソッドは、ローカルまたはリモート経由で実行されることを特徴とする請求項１に記載のコンピュータ可読記憶媒体。

【請求項６】

前記ジェネリックインターフェイスは、ＩＥｎｕｍｅｒａｂｌｅインターフェイスまたはＩＥｎｕｒｎｅｒａｂｌｅ＜Ｔ＞インターフェイスを継承し、、かつＩＱｕｅｒｙａｂｌｅインターフェイスおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞インターフェイスを含み、

前記コンピュータ実行可能命令が実行させる方法は、

前記処理装置が、前記ＩＱｕｅｒｙａｂｌｅインターフェイスおよび前記ＩＱｕｅｒｙａｂｌｅ＜Ｔ＞インターフェイスにより前記ＩＥｎｕｍｅｒａｂｌｅインターフェイスおよび前記ＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞インターフェイスのそれぞれをミラーリングするために、前記ＩＥｎｕｍｅｒａｂｌｅインターフェイス、前記ＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞インターフェイス、および前記標準クエリ演算子の拡張メソッドを、前記ＩＱｕｅｒｙａｂｌｅインターフェイスおよび前記ＩＱｕｅｒｙａｂｌｅ＜Ｔ＞インターフェイスを用いて複製するステップ

をさらに備えたことを特徴とする請求項１に記載のコンピュータ可読記憶媒体。

【請求項７】

前記式ツリーは、前記標準クエリ演算子の拡張メソッドを表す、未コンパイルかつ未検査のデータ構造を含むことを特徴とする請求項１に記載のコンピュータ可読記憶媒体。

【請求項８】

前記コンピュータ実行可能命令が実行させる方法は、前記処理装置が、前記作成した式ツリーを、デリゲートにコンパイルするステップをさらに備えたことを特徴とする請求項７に記載のコンピュータ可読記憶媒体。

【請求項９】

前記コンピュータ実行可能命令が実行させる方法は、前記処理装置が、前記作成した式ツリーを、デリゲートの文字列表現を用いて解析するステップをさらに備えたことを特徴とする請求項１に記載のコンピュータ可読記憶媒体。

【請求項１０】

前記コンピュータ実行可能命令が実行させる方法は、

前記処理装置が、前記作成した式ツリーを文字列解析し、フリー変数を識別するステップと、

前記処理装置が、前記識別したフリー変数に引数をバインドするステップと

をさらに備えたことを特徴とする請求項１に記載のコンピュータ可読記憶媒体。

【請求項１１】

式ツリーを介してクエリを埋め込みデータ操作を実行するためのアプリケーションプログラミングインターフェイスを使用してデータ操作を実行する、１つまたは複数の処理装置およびメモリを備えたコンピュータシステムであって、前記アプリケーションプログラミングインターフェイスはジェネリックインターフェイスを含み、前記コンピュータシステムは、

前記ジェネリックインターフェイスを介して実行される標準クエリ演算子の拡張メソッ

ドを使用したデータ操作の実行要求を受信する受信手段であって、前記標準クエリ演算子の拡張メソッドは、異なる種類の複数のデータストアのうちのいずれか１つに対して実行可能であり、かつ特定の種類のデータストアに固有のものではない、受信手段と、
前記受信した実行要求に係るデータのコレクションおよびクエリに基づいて、前記式ツリーを作成する作成手段と、
前記ジェネリックインターフェイスを介して、前記標準クエリ演算子の拡張メソッドを使用して、データストアに対するデータ操作を実行する実行手段であって、前記ジェネリックインターフェイスの引数は、デリゲードではなく前記作成した式ツリーである、実行手段と
をさらに備えたことを特徴とするコンピュータシステム。

【請求項１２】
式ツリーを介してクエリを埋め込み、データ操作を実行するためのアプリケーションプログラミングインターフェイスを使用してデータ操作を実行させるコンピュータ実行方法であって、前記アプリケーションプログラミングインターフェイスはジェネリックインターフェイスを含み、前記コンピュータ実行方法は、
コンピュータ上の処理装置が、前記ジェネリックインターフェイスを介して実行される標準クエリ演算子の拡張メソッドを使用したデータ操作の実行要求を受信するステップであって、前記標準クエリ演算子の拡張メソッドは、異なる種類の複数のデータストアのうちのいずれか１つに対して実行可能であり、かつ特定の種類のデータストアに固有のものではない、受信するステップと、
前記処理装置が、前記受信した実行要求に係るデータのコレクションおよびクエリに基づいて、式ツリーを作成するステップと、
前記処理装置が、前記ジェネリックインターフェイスを介して、前記標準クエリ演算子の拡張メソッドを使用して、データストアに対するデータ操作を実行するステップであって、前記ジェネリックインターフェイスの引数は、デリゲードではなく前記作成した式ツリーである、実行するステップと
を備えたことを特徴とするコンピュータ実行方法。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、プログラミング言語における式ツリーの深い埋め込みのためのジェネリックインターフェイスに関する。

【背景技術】
【０００２】
グローバル通信ネットワーク（たとえば、インターネット）の出現は、膨大量のデータへのアクセスを可能にしている。人々は、毎日のように、非構造化データや構造化データにアクセスして、クエリを行う。非構造化データは、報告書、電子メール、スプレッドシート、およびその他の種類の文書を作成して、格納し、取り出すために使用され、アトミックレベルで非構造化形式に格納された任意のデータで構成される。つまり、非構造化コンテンツにおいては、概念的な定義はなく、データ型の定義も存在せず、テキスト文書において単語はただ単語でしかない。非構造化データのコンテンツ検索に使用される現在の技術では、名前などのエンティティにタグ付けするか、またはキーワードおよびメタタグを適用することが求められる。したがって、非構造化データを機械可読なものにするには、人間の介入が必要となる。構造化データは、アトミックなデータ型に対して強制的な構成を有する任意のデータである。構造化データは、あらかじめ定められたデータ型および理解されている関係に対するクエリおよびレポートを可能にする技術によって管理される。

【０００３】
プログラミング言語は、プログラマによる指定および効率的な実行を実現するために、進化し続けている。コンピュータ言語の初期の時代には、低水準マシンコードが広く行き

渡っていた。マシンコードにより、コンピュータプログラムまたはコンピュータプログラムを構成する命令は、機械語またはアセンブリ言語で作成されており、ハードウェア（たとえば、マイクロプロセッサ）によって実行されるものであった。これらの言語は、コンピューティングハードウェアを制御する効率的な手段を提供したが、プログラマにとっては理解して高度な論理を展開することが非常に困難であった。

【０００４】

その後、さまざまな抽象化の層をもたらす言語が導入された。それに応じて、プログラマは、高水準のソース言語を使用してさらに高い水準でプログラムを作成することができるようになり、プログラムは、コンパイラまたはインタープリター（解釈プログラム）を介して、ハードウェアによって理解される低水準機械語に変換することができるようになった。プログラミングの更なる進歩は、追加の抽象化の層をもたらして、より高度なプログラミング論理がかつてないほど迅速に指定できるようになった。しかし、こうした進歩は、処理コストを伴うことなくして達成されるものではない。

【発明の開示】

【発明が解決しようとする課題】

【０００５】

主流のプログラミング言語におけるデータベース統合の状態には、かなり大幅な改善の余地がある。ｘＢａｓｅ、Ｔ／ＳＱＬ、およびＰＬ／ＳＱＬのような専用データベースプログラミング言語が多数存在するが、これらの言語は、弱く不十分な拡張可能タイプの体系を有し、オブジェクト指向プログラミングをほとんどまたは全くサポートせず、専用のランタイム環境を必要とする。同様に、Ｃ＃、ＶＢ.ＮＥＴ、C++、およびＪａｖａ（登録商標）などの汎用プログラミング言語があふれているが、これらの言語でのデータアクセスは通常、強い型定義とコンパイル時検査に欠ける煩雑なＡＰＩを通じて行われる。加えて、そのようなＡＰＩには、データ、データコレクションなどにクエリを行うための汎用インターフェイスを提供する機能がない。

【課題を解決するための手段】

【０００６】

本明細書において説明される一部の態様について基本的な理解をもたらすために、本技術革新を簡略にまとめた要約を以下に示す。この要約は、請求項に係る主題の広範囲に及ぶ概要ではない。この要約は、請求項に係る主題の主要または重要な要素を特定すること、あるいは主題の技術革新の範囲を明確に示すことのいずれも意図されていない。この要約の唯一の目的は、これ以降に提示されるさらに詳細な説明の前置きとして簡略化した形で請求項に係る主題の概念を一部提示することである。

【０００７】

主題の技術革新は、データ操作の実施を可能にするために式ツリー表現の作成を容易にするシステムおよび／または方法に関する。ミラーコンポーネントは、ジェネリックインターフェイスＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞を使用して、さまざまなデータ操作および／またはクエリが実施されるようにする式ツリー表現を作成することができ、ＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞はＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞をミラーリングする。ＩＥｎｕｍｅｒａｂｌｅコンポーネントは、特にデータのコレクションで機能し、固有の堅固な特性を備えるジェネリックコレクション型を提供することができる。加えて、ＩＥｎｕｍｅｒａｂｌｅコンポーネントは、そのような演算子が各々（たとえば、任意の適切なネームスペースで定義されるように）その引数としてデリゲートを取ることができるように、クエリ標準クエリ演算子拡張関数のセットを採用することができる。ミラーコンポーネントは、ＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｒｎｅｒａｂｌｅ＜Ｔ＞を複製して、ＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞を使用することにより作成された式ツリーのクエリを行えるようにするジェネリックインターフェイスを提供することができ、標準クエリ演算子拡張メソッドのＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞バージョンは、それらの引数としてデリゲートではなく作成された式ツリーを取る。

【０００８】
本請求項に係る主題の１つの態様に従って、ミラーコンポーネントは、クエリプロセッサを実施することができる。クエリプロセッサは、Ｗｈｅｒｅ、Ｓｅｌｅｃｔ、ＳｅｌｅｃｔＭａｎｙ、フィルタ関数、グループ関数、変換関数、ならびにＩＥｎｕｍｅｒａｂｌｅインターフェイスおよび／またはＩＥｎｕｒｎｅｒａｂｌｅ＜Ｔ＞インターフェイスと互換性のある適切な任意の関数など、ＩＥｎｕｍｅｒａｂｌｅに関連付けられているデータ操作を使用することができるが、これらに限定されることはない。ミラーコンポーネントは、データのコレクションの式ツリー表現を作成することができ（たとえば、クエリはデータのコレクションを生成することができ、データのコレクションは式ツリーによって表される）、表現は任意の適切なクエリプロセッサにとってよりジェネリックでクエリが扱いやすいデータの型であってもよい。
【０００９】
さらに、ミラーコンポーネントは、プログラミング言語および／またはフォーマットにかかわりなく、ローカル、リモート、有線経由などで、データにクエリが行われるようにするクエリプロセッサにプラグ可能なアーキテクチャを提供することができる。たとえば、データのコレクションを表す式ツリーを作成することができ（たとえば、クエリはデータのコレクションを生成することができ、データのコレクションは式ツリーによって表される）、次いで、さまざまなデータ操作をそこで（たとえば、リモート、ローカルなどで）実施することができる。さらに、そのようなデータには、ローカルコンピュータのメモリ内のデータにクエリを行う場合と同様に、リモート位置でクエリを行うことができる。本請求項に係る主題のその他の態様において、データ操作の実施を可能にするために式ツリー表現の作成を容易にする方法が提供される。
【００１０】
以下の説明および添付の図面は、本請求項に係る主題の特定の例示的な態様を詳細に示す。しかし、これらの態様は、本技術革新の原理が採用されうるさまざまな方法のほんの一部を示すに過ぎず、本請求項に係る主題は、そのような態様およびそれらの等価物をすべて含むことが意図されている。本請求項に係る主題のその他の利点および新規の特徴は、本技術革新の以下の詳細な説明を図と併せて検討すれば明らかになろう。
【発明を実施するための最良の形態】
【００１１】
本請求項に係る主題は、全体を通じて類似する要素を参照するために類似する番号が使用されている図面を参照して説明される。以下の説明において、説明の目的で、本主題に係る技術革新の十分な理解をもたらすため、数多くの具体的な詳細が示される。しかし、本請求項に係る主題が、それらの具体的な詳細なくして実施できることは、明らかであろう。その他の例において、本主題の技術革新の説明を容易にするため、よく知られている構造体および装置がブロック図に示されている。
【００１２】
本明細書において使用されている、「コンポーネント（ｃｏｍｐｏｎｅｎｔ）」、「システム（ｓｙｓｔｅｍ）」、「インターフェイス（ｉｎｔｅｒｆａｃｅ）」、「プロセッサ（ｐｒｏｃｅｓｓｏｒ）」、「クエリ（ｑｕｅｒｙ）」、「オペレーション（ｏｐｅｒａｔｉｏｎ）」などの用語は、ハードウェア、ソフトウェア（たとえば、実行中の）、および／またはファームウェアのいずれかのコンピュータ関連エンティティを示すことを意図している。たとえば、コンポーネントは、プロセッサ上で実行するプロセス、プロセッサ、オブジェクト、実行可能モジュール、プログラム、および／またはコンピュータであってもよい。例として、サーバ上で実行しているアプリケーションおよびサーバはいずれもコンポーネントにすることができる。１つまたは複数のコンポーネントはプロセス内に常駐することができ、コンポーネントは、１つのコンピュータ上でローカライズすることおよび／または２つ以上のコンピュータ間に分散することができる。
【００１３】
さらに、本請求項に係る主題は、開示される主題を実施するためにコンピュータを制御

するソフトウェア、ファームウェア、ハードウェア、またはこれらの任意の組み合わせを生成するための標準プログラミングおよび／またはエンジニアリング技法を使用する方法、機器、または製造品として実施することができる。本明細書において使用される「製造品（ａｒｔｉｃｌｅ　ｏｆ　ｍａｎｕｆａｃｔｕｒｅ）」という用語には、任意のコンピュータ可読装置、キャリア、または媒体からアクセス可能なコンピュータプログラムを含むことが意図されている。たとえば、コンピュータ可読媒体には、磁気記憶装置（たとえば、ハードディスク、フロッピー（登録商標）ディスク、磁気ストリップなど）、光ディスク（たとえば、コンパクトディスク（ＣＤ）、デジタル多用途ディスク（ＤＶＤ）など）、スマートカード、およびフラッシュメモリ素子（たとえばカード、スティック、キードライブなど）を含めることができるが、これらに限定されることはない。さらに、搬送波を採用して、電子メールを送受信する際、あるいはインターネットまたはローカルエリアネットワーク（ＬＡＮ）などのネットワークにアクセスする際に使用されるようなコンピュータ可読電子データを搬送することができることを理解されたい。もちろん、当業者であれば、本請求項に係る主題の範囲または精神を逸脱することなく、この構成に多くの変更を加えることができることも理解されよう。さらに、「例示的（ｅｘｅｍｐｌａｒｙ）」という単語は、本明細書において、例、事例、または例証としての機能を果たすことを意味するために使用される。本明細書において「例示的（ｅｘｅｍｐｌａｒｙ）」と説明されている任意の態様または設計は、必ずしも、その他の態様または設計よりも好ましいかまたは有利であると解釈される必要はない。

【００１４】

ここで、図１を参照すると、図１は、データ操作の実施を可能にするために式ツリー表現の作成を容易にするシステム１００を示す。システム１００は、ジェネリックインターフェイスを提供して、データに実施するデータ操作を受信することができるＩＥｎｕｍｅｒａｂｌｅコンポーネント１０４を実施することによりそのようなデータにクエリを行うことができるミラーコンポーネント１０２を含む。ＩＥｎｕｍｅｒａｂｌｅコンポーネント１０４は、特にデータのコレクションで機能し、固有の固定した特性を備えるジェネリックコレクション型を提供することができる。たとえば、ＩＥｎｕｍｅｒａｂｌｅコンポーネント１０４は、ジェネリックコレクション型と互換性のある固有のプログラミング言語を使用して、Ｗｈｅｒｅ、Ｓｅｌｅｃｔ、ＳｅｌｅｃｔＭａｎｙ、フィルタ関数、グループ関数、変換関数など、さまざまな標準クエリエクステンダを、データのコレクションに採用することができるが、これらに限定されることはない。加えて、ＩＥｎｕｍｅｒａｂｌｅコンポーネント１０４は、そのような演算子が各々（たとえば、任意の適切なネームスペースで定義されるように）その引数としてデリゲートを取ることができるように、クエリ標準クエリ演算子拡張関数のセットを採用することができる。

【００１５】

ミラーコンポーネント１０２は、ＩＥｎｕｍｅｒａｂｌｅコンポーネント１０４と、特にＩＥｎｕｍｅｒａｂｌｅインターフェイスおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞に関連付けられている関数を複製することができる。そのようなＩＥｎｕｍｅｒａｂｌｅコンポーネント１０４の複製により、ミラーコンポーネント１０２は、ＩＱｅｒｙａｂｌｅ＜Ｔ＞と、型が実施できる非ジェネリックの対の一方のＩＱｕｅｒｙａｂｌｅを提供して、式ツリーを介するクエリの深い埋め込みがサポートされることを知らせることができる。ミラーコンポーネント１０２は、データのコレクションおよびクエリの式ツリー表現を採用することができる。クエリとデータのコレクションを式ツリーとして表すことにより、データ型は式ツリー表現に基づくクエリプロセッサにさらに役立つので、ＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞インターフェイスは、データ操作を実施するために使用することができるジェネリックインターフェイスとなりうる。したがって、システム１００は、リレーショナルデータ、Ｅｘｔｅｎｓｉｂｌｅ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ（ＸＭＬ）オブジェクト、オブジェクト指向言語、プログラミング言語などへのクエリを容易にする。

【００１６】

ミラーコンポーネント１０２は、データのコレクション（図示せず）および行われるべきクエリに基づいて式ツリー表現を作成することができる。たとえば、データのコレクションは、述部関数に基づいてフィルタリングすることができ、通常は、ＩＥｎｕｍｅｒａｂｌｅインターフェイスおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞インターフェイスは特定の型、演算、特性、および／またはフォーマットに制限されていた。しかし、ミラーコンポーネント１０２が、ＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｒｎｅｒａｂｌｅ＜Ｔ＞関数を複製／ミラーリングすることができるジェネリックインターフェイスを提供できるようにすることにより、式ツリー表現はデータのコレクションとクエリから作成されて、クエリ、データ操作などをそこで実施することができる。一般に、式ツリー表現により、任意の適切なクエリプロセッサは、それと共に（Ｗｈｅｒｅ、Ｓｅｌｅｃｔ、ＳｅｌｅｃｔＭａｎｙ、フィルタ関数、グループ関数、変換関数などの）データ操作を実施することができることを理解されたい。つまり、システム１００は、プログラミング言語および／またはフォーマットにかかわりなく、ローカル、リモート、有線経由などで、データにクエリが行われるようにするクエリプロセッサにプラグ可能なアーキテクチャを提供する。たとえば、システム１００は、クエリ式の表現が作成され、次いでデータに送信されて、リモートに実施されるようにすることができる。さらに、そのようなデータには、ローカルコンピュータのメモリ内のデータにクエリを行う場合と同様に、リモート位置でクエリを行うことができる。

【００１７】

図２は、クエリプロセッサによるデータ操作を採用するためにデータのコレクションおよびクエリの式ツリー表現の使用を容易にするシステム２００を示す。システム２００は、データのコレクション２０２に対して実施されるデータ操作を受信することができるＩＥｎｕｍｅｒａｂｌｅコンポーネント１０４を含むことができ、そのようなデータ操作は、たとえば、Ｗｈｅｒｅ、Ｓｅｌｅｃｔ、ＳｅｌｅｃｔＭａｎｙ、フィルタ関数、グループ関数、変換関数、およびＩＥｎｕｍｅｒａｂｌｅインターフェイスおよび／またはＩＥｎｕｒｎｅｒａｂｌｅ＜Ｔ＞インターフェイスと互換性のある適切な任意の関数などであってもよい。ミラーコンポーネント１０２は、任意の適切なクエリプロセッサにとってよりジェネリックでクエリが扱いやすいデータの型であってもよいデータのコレクションおよびクエリの式ツリー表現を作成することができる。一般に、ミラーコンポーネント１０２および作成される式ツリー表現は、ＩＥｎｕｍｅｒａｂｌｅコンポーネント１０４に関連するデータ演算子（たとえばＩＥｎｕｍｅｒａｂｌｅインターフェイス、ＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞標準クエリ演算子など）が、ＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞の特性、フォーマット、データ構造などによって制約されるのではなく、任意の特定のプログラミング言語によって実施されるようにすることができる。

【００１８】

ミラーコンポーネント１０２は、ＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞インターフェイスと標準クエリ演算子拡張メソッドでＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞インターフェイスと標準クエリ演算子拡張メソッドを複製する。たとえば、ＩＥｎｕｍｅｒａｂｌｅベースの操作がデリゲート引数を持つ場合（たとえば、要素の変換などの操作、グループ要素への操作など）、ＩＱｕｅｒｙａｂｌｅベースの演算子は、それらの引数として式ツリー表現を持つ。このミラーリングにより、システム２００は、標準インターフェイスＩＱｕｅｒｙａｂｌｅ＜Ｔ＞と非ジェネリックの対の一方のＩＱｕｅｒｙａｂｌｅを定義して、これらが式ツリーを介するクエリの深い埋め込みがサポートすることを知らせることができる。

【００１９】

さらに、式ツリー表現を作成すると、クエリプロセッサ２０４が実施されてクエリ結果２０６を提供することができる。ミラーコンポーネント１０２は、任意の適切なクエリプロセッサ２０４が、ＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞によって複製される任意のＩＥｎｕｍｅｒａｂｌｅインターフェイスおよび／またはＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞インターフェイスに基づいて、データのコレクション２０２にクエリを行

うことができるように、式ツリー表現を作成することを理解されたい。つまり、システム２００は、任意の適切なクエリプロセッサとプラグ可能にして、クエリ結果を提供することができる。

【００２０】

図３は、式ツリーによって表すことができるデータのコレクションおよびクエリにクエリを行うためにプラグ可能クエリプロセッサの実施を容易にするシステム３００を示す。システム３００は、コードの式ツリー表現を作成して、プログラミング言語、型、フォーマットなどにかかわりなく任意のクエリプロセッサを使用してデータのコレクション２０２にクエリが行われるようにすることができるミラーコンポーネント１０２を含むことができる。たとえば、クエリプロセッサは、ＩＥｎｕｍｅｒａｂｌｅおよび／またはＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞に関連付けられている特定のデータ型、特性、フォーマットなどに従うことなく、ＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞を介してＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞に関連するデータ操作を実施することができる。

【００２１】

ＩＥｎｕｍｅｒａｂｌｅコンポーネント１０４は、ミラーコンポーネント１０２内に示されているが、ＩＥｎｕｍｅｒａｂｌｅコンポーネント１０４はスタンドアロンのコンポーネント、ミラーコンポーネント１０２に組み入れられたコンポーネント、ＩＱｕｅｒｙａｂｌｅコンポーネント３０２に統合されたコンポーネント、および／またはその任意の組み合わせであってもよいことを理解されたい。ＩＱｕｅｒｙａｂｌｅコンポーネント３０２はデータ操作を受信することができ、そのようなデータ操作は、ＩＥｎｕｍｅｒａｂｌｅコンポーネント１０４およびそれぞれの関数からミラー解除することができる。したがって、ＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞に関連付けられているデータ操作は、ＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞に関連付けられているデータフォーマットではなく、式ツリー表現の実施に基づいてよりジェネリックな方法で、ＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞を介して実施することができる。このようにして、ミラーコンポーネント１０２は、ＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞をミラーリングすることによってクエリを行うことができる式ツリー表現を作成することができる。特定のデータのコレクションの式ツリーが作成される場合、式ツリーは、別個のデータ操作および／またはクエリについて再作成される必要はないことを認識および理解されたい。つまり、式ツリー表現は、再使用することができる。

【００２２】

式ツリー表現は、データのクエリを行えるようにするため、任意の適切なクエリプロセッサ３０４によって使用されうることを理解されたい。クエリプロセッサ３０４は、任意の適切なクエリ操作および／またはデータ操作の使用を可能にするために「プラグイン」することができる。さらに、クエリプロセッサ$_{I}$からクエリプロセッサ$_{N}$まで（ここでＮは正の整数である）のような、任意の数のクエリプロセッサ３０４があってもよい。クエリプロセッサ３０４は、式ツリー表現を使用してさまざまなデータのコレクションのクエリを行えるようにするために「プラグイン」することができる。

【００２３】

図４は、データにクエリを行うジェネリックインターフェイスを提供するためにＩＥｎｕｍｅｒａｂｌｅインターフェイスおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞インターフェイスのミラーリングを容易にするシステム４００を示す。システム４００は、式ツリー表現を使用してクエリの深い埋め込みをサポートすることを型が知らせることができるようにジェネリックインターフェイスを提供することができるミラーコンポーネント１０２を含むことができる。ミラーコンポーネント１０２はさらに、ＩＥｎｕｍｅｒａｂｌｅと全く同様に任意の標準シーケンス演算子を実施することができるが、式ツリーをデリゲートではなく入力として取ることができる。システムはさらに、ＩＱｕｅｒｙａｂｌｅインターフェイスを実施することができるＩＱｕｅｒｙａｂｌｅコンポーネント４０２を含むこと

ができる。たとえば、以下の擬似コードが実施されてもよい。

【００２４】

【表１】

```
public interface IQueryable : IEnumerable {
    Expression Expression { get; }
    Type ElementType { get; }
    IQueryable CreateQuery(Expression expression);
    object Execute(Expression expression);
}

public interface IQueryable<T> : IEnumerable<T>, IQueryable {
    IQueryable<S> CreateQuery<S>(Expression expression);
    S Execute<S>(Expression expression);
}
```

【００２５】

読み取り専用プロパティＥｘｐｒｅｓｓｉｏｎは、クエリが現在表す式ツリーを返すが、ＥｌｅｍｅｎｔＴｙｐｅプロパティは、ＩＱｕｅｒｙａｂｌｅによって表されるコレクションの要素型を返す。ＣｒｅａｔｅＱｕｅｒｙメソッドは、式ツリーを与えられてｑｕｅｒｙａｂｌｅを作成する「仮想」コンストラクタである。同様に、Ｅｘｅｃｕｔｅは、各自がもはやｑｕｅｒｙａｂｌｅではない値を返すクエリのファクトリメソッドである。

【００２６】

ミラーコンポーネント１０２はさらに、シーケンス演算子の使用を容易にすることができるＩＯｒｄｅｒｅｄＱｕｅｒｙａｂｌｅコンポーネント４０４を含むことができる。たとえば、以下の擬似コードは、ＩＯｒｄｅｒｅｄＱｕｅｒｙａｂｌｅコンポーネント４０４によって実施されてもよい。

【００２７】

【表２】

```
public interface IOrderedQueryable : IQueryable { }
  public interface IOrderedQueryable<T> : IQueryable<T>, IOrderedQueryable {
}
```

【００２８】

ＩＯｒｄｅｒｅｄＱｕｅｒｙａｂｌｅおよびＩＯｒｄｅｒｅｄＱｕｅｒｙａｂｌｅ＜Ｔ＞は、さまざまなＯｒｄｅｒＢｙシーケンス演算子によって使用されるＯｒｄｅｒｅｄＳｅｑｕｅｎｃｅ型をミラーリングする。

【００２９】

既存のＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞は、（たとえば、任意の適切なネームスペースで定義されるように）すべてがその引数としてデリゲートを取る標準クエリ演算子（Ｗｈｅｒｅ、Ｓｅｌｅｃｔ、ＳｅｌｅｃｔＭａｎｙなど）で拡張される。たとえば、拡張メソッドＷｈｅｒｅは、ソースコレクションおよびデリゲートを取り、述部が保持するすべての値をもたらす。

【００３０】

【表３】

```
    public static IEnumerable<T> Where<T>(this IEnumerable<T> source,
Func<T, bool> predicate) {
        foreach (T element in source) {
          if (predicate(element)) yield return element;
        }
    }
```

【００３１】

ＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞の場合、ミラーコンポーネント１０２は、デリゲートではなく式ツリーを使用して標準クエリ演算子を実施する拡張を導入することができる。たとえば、Ｗｈｅｒｅの可能な実施を以下に示す。これは述部引数にデリゲートではなく式ツリーを取るが、その他の点に関しては、ＩＥｎｕｍｅｒａｂｌｅで定義されるＷｈｅｒｅ拡張メソッドのシグニチャとシグニチャ同型であることに留意されたい。

【００３２】

【表４】

```
public static IQueryable<T> Where<T>(this IQueryable<T> source,
Expression<Func<T, bool>> predicate) {
 return source.CreateQuery<T>(
          Expression.Call(

((MethodInfo)MethodBase.GetCurrentMethod()).MakeGenericMethod(typeof(T)),
              null,
              new Expression[] { source.Expression, predicate }
              ));
 }
```

【００３３】

その他の標準シーケンス演算子の実施は非常に類似しており、「ソースで自らを呼び出すこと（ｃａｌｌｉｎｇ　ｔｈｅｍｓｅｌｖｅｓ　ｏｎ　ｔｈｅ　ｓｏｕｒｃｅ）」に対応する式ツリーを各々効果的に作成する。その結果、標準クエリ演算子の実施は、完全に汎用であり、式ツリーを消費しようとする（たとえば、深い埋め込み）任意のアプリケーションプログラマブルインターフェイス（ＡＰＩ）によって使用されてもよい。

【００３４】

標準クエリ演算子の実施をミラーコンポーネント１０２が提供することに加えて、ミラーコンポーネント１０２は、ＩＥｎｕｍｅｒａｂｌｅをＱｕｅｒｙａｂｌｅに返すことができるＴｏＱｕｅｒｙａｂｌｅ（）演算子を提供することができる。たとえば、以下の擬似コードが使用されてもよい。

【００３５】

【表５】

```
public static IQueryable<T> ToQueryable<T>(this IEnumerable<T>
source) {}
public static IQueryable ToQueryable(this IEnumerable source) {}
```

【００３６】

ソースの動的型がすでにＩＱｕｅｒｙａｂｌｅである場合、この演算子は減退する。それ以外の場合、定数式としてソースを含む新しいＩＱｕｅｒｙａｂｌｅインスタンスが作成される。

【００３７】

ミラーコンポーネント１０２はまた、式ツリーのデリゲートへのコンパイルを容易にするヘルパーコンポーネント４０６を含むこともできる。つまり、ヘルパーコンポーネント４０６は、以下のような所定の式ツリーから統合言語（ＬＩ）を動的に生成するヘルパーメソッドＴ　Ｃｏｍｐｉｌｅ＜Ｔ＞（このＥｘｐｒｅｓｓｉｏｎ＜Ｔ＞ｆ）を介して式ツリーをデリゲートにコンパイルするためのメソッドを提供することができる。

【００３８】

【表６】

```
Expression<Func<Customer, bool>> e = c => c.City == "London";
Func<Customer, bool> f = e.Compile();
```

【００３９】
さらに、ミラーコンポーネント１０２は、式ツリーに文字列を解析するヘルパーメソッドを提供して、文字列内のフリー変数をバインドするためのリゾルバを提供することができる解析文字列ヘルパーコンポーネント４０８を含むことができる。解析文字列ヘルパーコンポーネント４０８は、デリゲートの文字列表現を解析するためのヘルパーメソッドのセットを導入することができる。一般的な場合、パーサーは、この式が表すデリゲートのパラメータのリストと、式内のフリー変数の名前を与えられてその名前がバインドされているメンバ情報（ｍｅｍｂｅｒ　ｉｎｆｏ）を返すネームリゾルバを取る。
【００４０】
【表７】

```
public static Expression<T> Parse<T>(string expr)
  public static Expression<T> Parse<T> (Parameters params, string expr,
NameResolver resolver)
```

【００４１】
図５は、本主題の技術革新によるさまざまな関数の例示的な関係５００を示す。例示的な関係５００は、ＩＱｕｅｒｙａｂｌｅ＜Ｔ＞５０８がＩＥｎｕｍｅｒａｂｌｅ５０２をミラーリングし、ＩＱｕｅｒｙａｂｌｅ５０４がＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞５０６をミラーリングすることを示している。つまり、ＩＱｕｅｒｙａｂｌｅ：ＩＥｎｕｍｅｒａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞：ＩＱｕｅｒｙａｂｌｅ、ＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞である。
【００４２】
図６は、クエリプロセッサでのデータ操作を採用するためにデータのコレクションの式ツリー表現の使用を容易にするインテリジェンス（処理機能）を採用するシステム６００を示す。システム６００は、以前の図で説明されているように、すべて実質的に各々のコンポーネントと類似していてもよいミラーコンポーネント６０２、ＩＥｎｕｍｅｒａｂｌｅコンポーネント６０４、およびＩＱｕｅｒｙａｂｌｅコンポーネント６０６を含むことができる。システム６００はさらに、インテリジェントコンポーネント６０８を含む。インテリジェントコンポーネント６０８は、式ツリー表現の作成を容易にしてデータ操作を実施できるようにするためにミラーコンポーネント６０２によって使用されてもよい。たとえば、インテリジェントコンポーネント６０８は、式ツリー表現、データ操作、クエリ、クエリ演算子、データのコレクション、ＩＥｎｕｍｅｒａｂｌｅおよびＩＱｕｅｒｙａｂｌｅに関連するミラーリング機能などを推論することができる。
【００４３】
インテリジェントコンポーネント６０８は、イベントおよび／またはデータを介して取り込まれた観察のセットから、システム、環境、および／またはユーザの状態について推察をもたらすかまたは推論することができることを理解されたい。推論は、たとえば、特定のコンテキストまたはアクションを識別するために採用することができるか、または、状態に対する確率分布を生成することができる。推論は、確率的なもの、つまり、データおよびイベントの検討に基づく関心対象の状態にわたる確率分布の計算となりうる。推論はまた、イベントおよび／またはデータのセットからさらに高レベルのイベントを構成するために採用される技法を示すこともできる。そのような推論は、イベントが密接な一時的隣接性で相関しているかどうかにかかわらず、およびイベントとデータが１つまたは複数のイベントおよびデータソースから生じているかどうかにかかわらず、結果として、観察されたイベントおよび／または格納されたイベントデータのセットからの新しいイベントまたはアクションの構築をもたらす。さまざまな分類（明示的および／または暗黙的にトレーニングされた）方式および／またはシステム（たとえば、サポートベクトルマシン、ニューラルネットワーク、エキスパートシステム、ベイズ信頼ネットワーク、ファジー論理、データ融合エンジンなど）は、本請求項に係る主題に関連して自動および／または推論されたアクションの実行と共に採用されてもよい。

【００４４】
分類子は、入力属性ベクトルｘ＝（ｘ１、ｘ２、ｘ３、ｘ４、ｘｎ）を、入力がクラスに属する信頼度にマップする関数、つまりｆ（ｘ）＝ｃｏｎｆｉｄｅｎｃｅ（ｃｌａｓｓ）である。そのような分類は、自動的に実行されるようユーザが望むアクションを予知または推論するために、確率および／または統計ベースの分析を採用することができる（たとえば、分析ユーティリティとコストを考慮に入れる）。サポートベクトルマシン（ＳＶＭ）は、採用することができる分類子の一例である。ＳＶＭは、可能な入力のスペースで超曲面を見出すことによって動作するが、超曲面は非トリガーイベントからトリガー基準を分割しようと試みる。直観的に、このことが、近似ではあるがトレーニングデータと同一ではないデータの検査について分類を正しいものにする。その他の有向および無向モデル分類の手法は、たとえば、単純ベイズ、ベイズネットワーク、決定木、ニューラルネットワーク、ファジー論理モデルを含み、独立のさまざまなパターンを提供する確率分類モデルが採用されてもよい。本明細書において使用される分類はまた、優先度のモデルを開発するために使用される統計回帰も含む。

【００４５】
ミラーコンポーネント６０２はさらに、ユーザとミラーコンポーネント６０２に結合された任意のコンポーネントとの間の対話を容易にするためにさまざまなタイプのユーザインターフェイスを提供するプレゼンテーションコンポーネント６１０を使用することができる。示されているように、プレゼンテーションコンポーネント６１０は、ミラーコンポーネント６０２と共に使用することができる別個のエンティティである。しかし、プレゼンテーションコンポーネント６１０および／または同様の表示コンポーネントが、ミラーコンポーネント６０２および／またはスタンドアロンのユニットに組み入れられてもよいことを理解されたい。プレゼンテーションコンポーネント６１０は、１つまたは複数のグラフィカルユーザインターフェイス（ＧＵＩ）、コマンドラインインターフェイスなどを提供することができる。たとえば、データをロード、インポート、読み取りなどする領域または手段をユーザに提供するＧＵＩが表示されてもよく、ＧＵＩはそのような結果を提示する領域を含むこともできる。これらの領域は、ダイアログボックス、静的コントロール、ドロップダウンメニュー、リストボックス、ポップアップメニュー、編集コントロール、コンボボックス、ラジオボタン、チェックボックス、プッシュボタン、およびグラフィックボックスを備える既知のテキストおよび／またはグラフィック領域を備えることができる。加えて、移動のための垂直および／または水平スクロールバー、領域を表示可能にするかどうかを決定するツールバーボタンなど、提示を容易にするためのユーティリティが採用されてもよい。たとえば、ユーザは、ミラーコンポーネント６０２に結合されたコンポーネントおよび／または組み込まれたコンポーネントのうちの１つまたは複数と対話することもできる。

【００４６】
ユーザはまた、たとえば、マウス、ローラーボール、キーパッド、キーボード、ペン、および／または音声起動など、さまざまな装置を介して情報を選択して提供するために領域と対話することもできる。通常、キーボード上のプッシュボタンまたはエンターキーなどの機構は、検索を開始するために後の情報の入力に採用されてもよい。しかし、本請求項に係る主題がそのように限定されないことを理解されたい。たとえば、単にチェックボックスを強調表示することは、情報の伝達を開始することができる。もう１つの例において、コマンドラインインターフェイスが採用されてもよい。たとえば、コマンドラインインターフェイスは、（たとえば、表示装置上のテキストメッセージおよび音声トーンを介して）ユーザにテキストメッセージの提供により情報を入力するよう促すことができる。次いでユーザは、インターフェイスのプロンプトに表示されたオプションに対応する英数字入力、またはプロンプトに提示された質問に対する回答など、適切な情報を提供することができる。コマンドラインインターフェイスは、ＧＵＩおよび／またはＡＰＩと共に採用することができることを理解されたい。加えて、コマンドラインインターフェイスは、限定的なグラフィックサポートおよび／または低帯域幅通信チャネルにより、ハードウェ

ア（たとえば、ビデオカード）および／または表示装置（たとえば、モノクロおよびＥＧＡ）と共に採用することができる。

【００４７】

図７～図９は、本請求項に係る主題による方法を示す。説明を簡単にするため、方法は、一連の動作として示され、説明される。本主題の技術革新は、たとえば動作がさまざまな順序および／または同時に生じる可能性もあるように、示されている動作および／または動作の順序によって限定されないこと、また本明細書において提示および説明されない他の動作があることを、理解および認識されたい。さらに、示されているすべての動作が、本請求項に係る主題による方法を実施するために必要ではない場合もある。加えて、当業者であれば、方法は代替として、状態図またはイベントを介して一連の相互関係のある状態として表すことができることを理解するであろう。さらに、これ以降および本明細書全体を通じて開示される方法は、そのような方法をコンピュータに容易にトランスポートおよび転送するために製造品に格納することができることを理解されたい。本明細書において使用される製造品という用語は、任意のコンピュータ可読装置、キャリア、または媒体からアクセス可能なコンピュータプログラムを含むことが意図されている。

【００４８】

図７は、データ操作の実施を可能にするために式ツリー表現の作成を容易にする方法７００を示す。参照番号７０２において、データ操作を受信することができる。１つの例において、データ操作は、データのコレクションに対して実施することができ、そのようなデータ操作は、たとえば、Ｗｈｅｒｅ、Ｓｅｌｅｃｔ、ＳｅｌｅｃｔＭａｎｙ、フィルタ関数、グループ関数、変換関数、およびＩＥｎｕｍｅｒａｂｌｅインターフェイスおよび／またはＩＥｎｕｒｎｅｒａｂｌｅ＜Ｔ＞インターフェイスなどと互換性のある適切な任意の関数などであってもよい。もう１つの例において、データ操作は、ユーザ、エンティティ、マシン、ローカルマシン、リモートマシンなどから受信することができる。

【００４９】

参照番号７０４において、ＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞は、ミラーリングされて、データにクエリを行うジェネリックインターフェイスを提供することができる。固有の堅固な特性を持つＩＥｎｕｍｅｒａｂｌｅに関連付けられているデータに基づいて、ジェネリックインターフェイスは、ＩＥｎｕｍｅｒａｂｌｅ、および特に、ＩＥｎｕｍｅｒａｂｌｅインターフェイスとＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞インターフェイスおよびそれらの標準クエリ演算子拡張メソッドに関連付けられている関数を複製することができる。そのようなＩＥｎｕｍｅｒａｂｌｅの複製により、標準インターフェイスＩＱｅｒｙａｂｌｅ＜Ｔ＞と非ジェネリックの対の一方のＩＱｕｅｒｙａｂｌｅは、式ツリーを介するクエリの深い埋め込みがサポートされることを任意の適切な型に知らせることができる。データのコレクションを式ツリーとして表すことにより、データ型は式ツリー表現に基づくクエリプロセッサにさらに役立ち、ＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞がミラーされるようになるので、ＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞インターフェイスは、データ操作を実施するために使用することができるジェネリックインターフェイスとなりうる。

【００５０】

図８は、式ツリーによって表すことができるデータのコレクションにクエリを行うためにプラグ可能クエリプロセッサの実施を容易にする方法８００を示す。参照番号８０２において、データのコレクションに実施されるべきデータ操作を受信することができる。参照番号８０４において、データのコレクションの式ツリー表現を作成することができる。式ツリー表現は、データのコレクションに関連付けることができる。データのコレクションとクエリを式ツリーとして表すことにより、データ型は式ツリー表現に基づくクエリプロセッサにさらに役立つので、ＩＱｕｅｒｙａｂｌｅインターフェイスおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞インターフェイスは、少なくとも１つのデータ操作を実施するジェネリックインターフェイスとなりうる。

【００５１】

参照番号８０６において、データ操作は、たとえばクエリプロセッサを使用して式ツリーで実施することができる。次いで、データ操作は、ＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞機能をミラーリングするＩＱｕｅｒａｂｌｅおよびＩＱｕｅｒａｂｌｅ＜Ｔ＞インターフェイスを採用する技法を使用してジェネリックに適用することができる。ＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞の機能をミラーリングすることにより、データ操作は、通常ＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞に関連付けられている厳密で堅固なデータコレクションではなく、式ツリーに採用することができる。したがって、方法８００は、リレーショナルデータ、Ｅｘｔｅｎｓｉｂｌｅ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ（ＸＭＬ:拡張マックアップ言語）オブジェクト、オブジェクト指向言語、プログラミング言語などにクエリを行うことを容易にする。一般に、式ツリー表現により、任意の適切なクエリプロセッサは、それと共に（Ｗｈｅｒｅ、Ｓｅｌｅｃｔ、ＳｅｌｅｃｔＭａｎｙ、フィルタ関数、グループ関数、変換関数などの）データ操作を実施することができることを理解されたい。

【００５２】

図９は、データのコレクションの式ツリー表現を使用してクエリプロセッサによるデータ操作を採用する方法９００を示す。参照番号９０２において、データのコレクションに実施されうるデータ操作を受信することができる。データ操作は、Ｗｈｅｒｅ、Ｓｅｌｅｃｔ、ＳｅｌｅｃｔＭａｎｙ、フィルタ関数、グループ関数、変換関数、およびＩＥｎｕｍｅｒａｂｌｅインターフェイスおよび／またはＩＥｎｕｒｎｅｒａｂｌｅ＜Ｔ＞インターフェイスと互換性のある適切な任意の関数などであってもよいが、これらに限定されることはない。参照番号９０４において、データのコレクションを表す式ツリーは、インターフェイスＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｒｎｅｒａｂｌｅ＜Ｔ＞をミラーリングする２つのインターフェイスＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞を使用して作成することができる。

【００５３】

参照番号９０６において、データ操作は、式ツリーに実施することができる。ＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞の機能をミラーリングすることにより、データ操作は、通常ＩＥｎｕｍｅｒａｂｌｅおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞に関連付けられている厳密で固定したデータコレクションではなく、（たとえばＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞を使用して）式ツリーに採用することができる。データのコレクションを式ツリーとして表すことにより、データ型は式ツリー表現に基づくクエリプロセッサにさらに役立つので、ＩＱｕｅｒｙａｂｌｅおよびＩＱｕｅｒｙａｂｌｅ＜Ｔ＞インターフェイスは、データ操作を実施するために使用することができるジェネリックインターフェイスとなりうる。

【００５４】

参照番号９０８において、任意の適切なクエリプロセスは、プラグインされて、データのコレクションを表す式ツリーにクエリを行うことができる。一般に、式ツリー表現により、任意の適切なクエリプロセッサは、それと共に（Ｗｈｅｒｅ、Ｓｅｌｅｃｔ、ＳｅｌｅｃｔＭａｎｙ、フィルタ関数、グループ関数、変換関数などの）データ操作を実施することができることを理解されたい。つまり、方法９００は、プログラミング言語および／またはフォーマットにかかわりなく、ローカル、リモート、有線経由などで、データにクエリが行われるようにするクエリプロセッサにプラグ可能なアーキテクチャを提供する。たとえば、方法９００は、クエリ式の表現が作成され、次いでデータに送信されて、リモートに実施されるようにすることができる。さらに、そのようなデータには、ローカルコンピュータのメモリ内のデータにクエリを行う場合と同様に、リモート位置でクエリを行うことができる。

【００５５】

図１０は、本主題の技術革新に従って実施コード（たとえば、実行可能モジュール、中間言語など）を生成するために使用することができるコンパイラ環境１０００を示すブロック図である。コンパイラ環境１０００は、フロントエンドコンポーネント１０２０、コ

ンバータコンポーネント１０３０、バックエンドコンポーネント１０４０、エラーチェッカーコンポーネント１０５０、シンボルテーブル１０６０、解析ツリー１０７０、および状態１０８０を含むコンパイラ１０１０を含む。コンパイラ１０１０は、ソースコードを入力として受け入れ、実装コードを出力として生成する。入力は、本明細書において説明されるように、区切り文字付きプログラマチック式または修飾された識別子を含むことができるが、これらに限定されることはない。コンパイラ環境のコンポーネントとモジュールの間の関係は、データの主要な流れを示す。その他のコンポーネントおよび関係は、明確かつ簡略にするために示されていない。実施に応じて、コンポーネントは、追加、省略、複数モジュールに分割、他のモジュールおよび／または他のモジュールの構成に結合することができる。

【００５６】

コンパイラ１０１０は、要素のシーケンスの処理に関連付けられているソースコードを有するファイルを入力として受け入れることができる。ソースコードは、さまざまな式および関連する関数、メソッドおよび／またはその他のプログラマチック構成体を含むことができる。コンパイラ１０１０は、構成体を分析してコードを生成または注入するために１つまたは複数のコンポーネントと共にソースコードを処理することができる。

【００５７】

フロントエンドコンポーネント１０２０は、ソースコードを読み取り、字句解析を実行する。基本的に、フロントエンドコンポーネント１０２０は、ソースコードの文字のシーケンス（たとえば、英数字）を読み取って、特に、定数、識別子、演算子記号、キーワード、句読点を示す構文要素またはトークンに変換する。

【００５８】

コンバータコンポーネント１０３０は、トークンを中間表現に解析する。たとえば、コンバータコンポーネント１０３０は、構文を検査して、トークンを式またはその他の統語構造にグループ化することができるが、これがステートメントツリーに合体する。概念的には、これらのツリーは解析ツリー１０７０を形成する。さらに、必要に応じて、コンバータモジュール１０３０は、ソースコードに使用されるシンボル名および型情報を関連する特性と共にリストするシンボルテーブル１０３０にエントリを入れることができる。

【００５９】

状態１０８０は、受信または取り出したソースコードを処理して解析ツリー１０７０を形成する際に、コンパイラ１０１０の進行状況を追跡するために採用することができる。たとえば、異なる状態値は、コンパイラ１０１０がクラス定義または関数の開始時点にあるか、クラスメンバを宣言したばかりであるか、または式を完了したことを示す。コンパイラは、進行に応じて、継続的に状態１０８０を更新する。コンパイラ１０１０は、状態１０８０を部分的または完全に外部エンティティに公開するが、これはコンパイラ１０１０に入力を提供することができる。

【００６０】

ソースコードの構造体またはその他の信号に基づいて（あるいは、それ以外に機会が認識される場合）、コンバータコンポーネント１０３０または別のコンポーネントは、効率的で適切な実行を容易にするために対応するコードを注入することができる。コンバータコンポーネント１０３０または他のコンポーネントにコーディングされたルールは、望ましい機能を実施するため、およびコードが注入されるべき位置または他の操作が実行されるべき位置を識別するために行うべきことを示す。注入されたコードは通常、１つまたは複数の位置で追加されたステートメント、メタデータ、またはその他の要素を含むが、この用語はまた既存のソースコードの変更、削除、または修正を含むこともできる。注入されたコードは、１つまたは複数のテンプレートとして、または何らかの他の形態で格納することができる。加えて、シンボルテーブル操作および解析ツリー変換が行われうることを理解されたい。

【００６１】

シンボルテーブル１０６０および解析ツリー１０７０に基づいて、バックエンドコンポ

ーネント１０４０は、中間表現を出力コードに変換することができる。バックエンドコンポーネント１０４０は、中間表現を、ターゲットプロセッサで、またはターゲットプロセッサにより実行可能な命令に、変数のメモリ割り振りに、というように変換する。出力コードは実際のプロセッサにより実行可能であってもよいが、仮想プロセッサにより実行可能な出力コードもまた提供されてもよい。

【００６２】

さらに、フロントエンドコンポーネント１０２０およびバックエンドコンポーネント１０４０は、コード最適化などの追加の機能を実行することができ、単一フェーズまたは複数フェーズとして記述されている操作を実行することができる。コンパイラ１０１０のコンポーネントのその他のさまざまな態様は、事実上従来のものであり、同等の機能を実行するコンポーネントと置き換えることができる。加えて、ソースコードの処理中のさまざまな段階において、エラーチェッカーコンポーネント１０５０は、語彙構造のエラー、構文エラー、さらに意味エラーなどのエラーを確認することができる。エラーを検出すると、チェッカーコンポーネント１０５０は、コンパイルを一時停止して、エラーを示すメッセージを生成することができる。

【００６３】

本請求項に係る主題のさまざまな態様を実施する追加のコンテキストを提供するため、図１１～図１２および以下の解説は、主題の技術革新のさまざまな態様が実施されうる適切なコンピューティング環境の簡潔な概要を示すことを意図している。たとえば、以前の図で説明されているように、式ツリー表現を使用してデータのコレクションにデータ操作を提供するためにジェネリックインターフェイスを提供するミラーコンポーネントは、そのような適切なコンピューティング環境において実施することができる。本請求項に係る主題は、上記で、ローカルコンピュータおよび／またはリモートコンピュータ上で稼働するコンピュータプログラムのコンピュータ実行可能命令の一般的なコンテキストにおいて説明してきたが、当業者であれば、本主題の技術革新がまた、その他のプログラムモジュールとの組み合わせでも実施できることを理解するであろう。一般に、プログラムモジュールは、特定のタスクを実行しおよび／または特定の抽象データ型を実施するルーチン、プログラム、コンポーネント、データ構造などを含む。

【００６４】

さらに、本発明の方法が、シングルプロセッサまたはマルチプロセッサコンピュータシステム、ミニコンピュータ、メインフレームコンピュータ、ならびにパーソナルコンピュータ、ハンドヘルドコンピューティング装置、マイクロプロセッサベースおよび／またはプログラマブル消費者電化製品（家電）など、各々が１つまたは複数の関連する装置と動作可能に通信することができる、他のコンピュータシステム構成により実施できることを当業者は理解するであろう。説明されている本請求項に係る主題の態様はさらに、特定のタスクが通信ネットワークを通じてリンクされたリモート処理装置によって実行される分散コンピューティング環境においても実施することができる。しかし、本主題の技術革新の態様のすべてではないとしても、その一部は、スタンドアロンコンピュータにおいて実施することができる。分散コンピューティング環境において、プログラムモジュールは、ローカルおよび／またはリモートの記憶装置に配置することができる。

【００６５】

図１１は、本請求項に係る主題が対話することのできるサンプルのコンピューティング環境１１００を示す概略ブロック図である。システム１１００は、１つまたは複数のクライアント１１１０を含んでいる。クライアント１１１０は、ハードウェアおよび／またはソフトウェア（たとえば、スレッド、プロセス、コンピューティング装置）であってもよい。システム１１００はさらに、１つまたは複数のサーバ１１２０も含む。サーバ１１２０は、ハードウェアおよび／またはソフトウェア（たとえば、スレッド、プロセス、コンピューティング装置）であってもよい。サーバ１１２０は、たとえば、本主題の技術革新を採用することによって変換を実行するスレッドを収容することができる。

【００６６】

クライアント１１１０およびサーバ１１２０間の１つの可能な通信は、２つ以上のコンピュータプロセス間で伝送されるように適用されたデータパケットの形態であってもよい。システム１１００は、クライアント１１１０およびサーバ１１２０間の通信を容易にするために採用することができる通信フレームワーク１１４０を含む。クライアント１１１０は、クライアント１１１０にローカルな情報を格納するために採用することができる１つまたは複数のクライアントデータストア１１５０に操作可能に接続されている。同様に、サーバ１１２０は、サーバ１１２０にローカルな情報を格納するために採用することができる１つまたは複数のサーバデータストア１１３０に操作可能に接続されている。

【００６７】

図１２を参照すると、本請求項に係る主題のさまざまな態様を実装するための例示的な環境１２００はコンピュータ１２１２を含む。コンピュータ１２１２は、処理装置１２１４、システムメモリ１２１６、およびシステムバス１２１８を含む。システムバス１２１８は、限定はしないが、システムメモリ１２１６を含むシステムコンポーネントを処理装置１２１４に接続している。処理装置１２１４は、さまざまな使用可能なプロセッサのいずれであってもよい。デュアルマイクロプロセッサおよび他のマルチプロセッサアーキテクチャも、処理装置１２１４として採用することができる。

【００６８】

システムバス１２１８は、ＩＳＡ（Ｉｎｄｕｓｔｒｉａｌ Ｓｔａｎｄａｒｄ Ａｒｃｈｉｔｅｃｔｕｒｅ）、ＭＳＡ（Ｍｉｃｒｏ－Ｃｈａｎｎｅｌ Ａｒｃｈｉｔｅｃｔｕｒｅ）、ＥＩＳＡ（Ｅｘｔｅｎｄｅｄ ＩＳＡ）、ＩＤＥ（Ｉｎｔｅｌｌｉｇｅｎｔ Ｄｒｉｖｅ Ｅｌｅｃｔｒｏｎｉｃｓ）、ＶＬＢ（ＶＥＳＡ Ｌｏｃａｌ Ｂｕｓ）、ＰＣＩ（Ｐｅｒｉｐｈｅｒａｌ Ｃｏｍｐｏｎｅｎｔ Ｉｎｔｅｒｃｏｎｎｅｃｔ）、Ｃａｒｄ Ｂｕｓ、ＵＳＢ（Ｕｎｉｖｅｒｓａｌ Ｓｅｒｉａｌ Ｂｕｓ）、ＡＧＰ（Ａｄｖａｎｃｅｄ Ｇｒａｐｈｉｃｓ Ｐｏｒｔ）、ＰＣＭＣＩＡ（Ｐｅｒｓｏｎａｌ Ｃｏｍｐｕｔｅｒ Ｍｅｍｏｒｙ Ｃａｒｄ Ｉｎｔｅｒｎａｔｉｏｎａｌ Ａｓｓｏｃｉａｔｉｏｎ）バス、Ｆｉｒｅｗｉｒｅ（ＩＥＥＥ１２９４）、およびＳＣＳＩ（Ｓｍａｌｌ Ｃｏｍｐｕｔｅｒ Ｓｙｓｔｅｍｓ Ｉｎｔｅｒｆａｃｅ）を含む任意のさまざまな使用可能なバスアーキテクチャを使用するメモリバスもしくはメモリコントローラ、周辺バスもしくは外部バス、および／またはローカルバスを含む複数のタイプのバス構造のいずれであってもよいが、これらに限定されることはない。

【００６９】

システムメモリ１２１６は、揮発性メモリ１２２０および不揮発性メモリ１２２２を含む。起動時などにコンピュータ１２１２内の要素間で情報を転送する基本ルーチンを含むＢＩＯＳ（ｂａｓｉｃ ｉｎｐｕｔ／ｏｕｔｐｕｔ ｓｙｓｔｅｍ）は、不揮発性メモリ１２２２に格納される。限定的ではなく例示的に、不揮発性メモリ１２２２は、読み取り専用メモリ（ＲＯＭ）、プログラマブルＲＯＭ（ＰＲＯＭ）、電気的プログラマブルＲＯＭ（ＥＰＲＯＭ）、電気的消去可能プログラマブルＲＯＭ（ＥＥＰＲＯＭ）、またはフラッシュメモリを含むことができるが、これらに限定されることはない。揮発性メモリ１２２０は、ランダムアクセスメモリ（ＲＡＭ）を含むが、これは外部キャッシュメモリとしての役割を果たす。限定的ではなく例示的に、ＲＡＭは、スタティックＲＡＭ（ＳＲＡＭ）、ダイナミックＲＡＭ（ＤＲＡＭ）、シンクロナスＤＲＡＭ（ＳＤＲＡＭ）、ダブルデータレートＳＤＲＡＭ（ＤＤＲ ＳＤＲＡＭ）、拡張ＳＤＲＡＭ（ＥＳＤＲＡＭ）、シンクリンクＤＲＡＭ（ＳＬＤＲＡＭ）、ラムバスダイレクトＲＡＭ（ＲＤＲＡＭ）、ダイレクトラムバスダイナミックＲＡＭ（ＤＲＤＲＡＭ）、およびラムバスダイナミックＲＡＭ（ＲＤＲＡＭ）などの多くの形態で使用可能である。

【００７０】

コンピュータ１２１２はまた、取り外し可能／固定式、揮発性／不揮発性のコンピュータ記憶媒体も含む。図１２は、たとえばディスクストレージ１２２４を示している。ディスクストレージ１２２４は、磁気ディスクドライブ、フロッピー（登録商標）ディスクドライブ、テープドライブ、Ｊａｚドライブ、Ｚｉｐドライブ、ＬＳ－１００ドライブ、フ

ラッシュメモリカード、またはメモリスティックなどの装置を含むが、これらに限定されることはない。さらに、ディスクストレージ１２２４は、ストレージ媒体を単独で含めることも、コンパクトディスクＲＯＭ装置（ＣＤ－ＲＯＭ）、追記型ＣＤドライブ（ＣＤ－Ｒドライブ）、書き換え可能ＣＤドライブ（ＣＤ－ＲＷドライブ）、またはデジタル多用途ディスクＲＯＭドライブ（ＤＶＤ－ＲＯＭ）などの光ディスクドライブを含む他のストレージ媒体との組み合わせで含めることもできるが、これらに限定されることはない。システムバス１２１８へのディスクストレージ装置１２１４の接続を容易にするため、通常インターフェイス１２２６などの取り外し可能または固定式インターフェイスが使用される。

【００７１】

図１２は、適切なオペレーティング環境１２００で説明されているユーザと基本コンピュータリソースとの間で仲介としての機能を果たすソフトウェアについて説明していることを理解されたい。そのようなソフトウェアは、オペレーティングシステム１２２８を含む。オペレーティングシステム１２２８は、ディスクストレージ１２２４に格納することができ、コンピュータシステム１２１２のリソースを制御して割り当てるように動作する。システムアプリケーション１２３０は、システムメモリ１２１６またはディスクストレージ１２２４に格納されているプログラムモジュール１２３２およびプログラムデータ１２３４を通じてオペレーティングシステム１２２８によるリソースの管理を利用する。本請求項に係る主題は、さまざまなオペレーティングシステムまたはオペレーティングシステムの組み合わせにより実装できることを理解されたい。

【００７２】

ユーザは、入力装置１２３６を通じてコマンドまたは情報をコンピュータ１２１２に入力する。入力装置１２３６は、マウスなどのポインティングデバイス、トラックボール、スタイラス、タッチパッド、キーボード、マイクロフォン、ジョイスティック、ゲームパッド、衛星放送受信アンテナ、スキャナ、テレビチューナカード、デジタルカメラ、デジタルビデオカメラ、Ｗｅｂカメラなどを含んでいるが、これらに限定されることはない。これらのおよびその他の入力装置は、インターフェイスポート１２３８を介してシステムバス１２１８経由で処理装置１２１４に接続している。インターフェイスポート１２３８は、たとえば、シリアルポート、パラレルポート、ゲームポート、およびユニバーサルシリアルバス（ＵＳＢ）を含む。出力装置１２４０は、入力装置１２３６と同じタイプのポートの一部を使用する。したがって、たとえば、ＵＳＢポートは、コンピュータ１２１２への入力を提供し、コンピュータ１２１２から出力装置１２４０に情報を出力するために使用されてもよい。出力アダプタ１２４２は、他の出力装置１２４０の中でも、特殊なアダプタを必要とするモニタ、スピーカ、およびプリンタのような出力装置１２４０があることを示すために提供されている。出力アダプタ１２４２は、限定的ではなく例示的に、出力装置１２４０とシステムバス１２１８との間の接続手段を提供するビデオおよびサウンドカードを含んでいる。他の装置および／または装置のシステムがリモートコンピュータ１２４４のような入力および出力機能を提供することに留意されたい。

【００７３】

コンピュータ１２１２は、リモートコンコンピュータ１２４４など、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するネットワーク化された環境において動作することができる。リモートコンピュータ１２４４は、パーソナルコンピュータ、サーバ、ルータ、ネットワークＰＣ、ワークステーション、マイクロプロセッサベースの機器、ピアデバイスまたはその他の共通ネットワークノードなどであってもよく、通常はコンピュータ１２１２に関連して説明されている要素の多くまたはすべてを含む。簡潔にするため、メモリ記憶装置１２４６のみがリモートコンピュータ１２４４と共に示されている。リモートコンピュータ１２４４は、ネットワークインターフェイス１２４８を通じてコンピュータ１２１２に論理的に接続され、次いで通信接続１２５０を介して物理的に接続されている。ネットワークインターフェイス１２４８は、ローカルエリアネットワーク（ＬＡＮ）およびワイドエリアネットワーク（ＷＡＮ）などの有線および／または無線通信ネッ

トワークを含む。ＬＡＮ技術は、Ｆｉｂｅｒ Ｄｉｓｔｒｉｂｕｔｅｄ Ｄａｔａ Ｉｎｔｅｒｆａｃｅ（ＦＤＤＩ）、Ｃｏｐｐｅｒ Ｄｉｓｔｒｉｂｕｔｅｄ Ｄａｔａ Ｉｎｔｅｒｆａｃｅ（ＣＤＤＩ）、イーサネット（登録商標）、トークンリングなどを含む。ＷＡＮ技術は、二地点間リンク、統合デジタル通信網（ＩＳＤＮ）およびその変種などの回線交換ネットワーク、パケット交換ネットワーク、およびデジタル加入者線（ＤＳＬ）を含むが、これらに限定されることはない。

【００７４】

通信接続１２５０は、ネットワークインターフェイス１２４８をバス１２１８に接続するために採用されるハードウェア／ソフトウェアを示す。通信接続１２５０は、説明を明確にするためにコンピュータ１２１２の内部に示されているが、コンピュータ１２１２の外部にすることもできる。ネットワークインターフェイス１２４８への接続に必要なハードウェア／ソフトウェアは、通常の電話品質のモデム、ケーブルモデム、およびＤＳＬモデムを含むモデム、ＩＳＤＮアダプタ、ならびにイーサネット（登録商標）カードなどの内部および外部技術を含むが、これらは例示のために示したに過ぎない。

【００７５】

以上説明してきた内容は、本主題の技術革新の実施例を含む。本請求項に係る主題を説明するために、すべての考えられるコンポーネントまたは方法の組み合わせについて説明することはもちろん不可能であるが、本主題の技術革新の多くの更なる組み合わせおよび置換が可能であることを当業者であれば理解するであろう。したがって、本請求項に係る主題は、添付の特許請求の範囲の精神および範囲内に入るそのようなすべての変更、修正、および変形を含むことが意図されている。

【００７６】

特に、および上記で説明されているコンポーネント、装置、回路、システムなどによって実行されるさまざまな機能に関して、そのようなコンポーネントを説明するために使用される用語（「手段（ｍｅａｎｓ）」の参照を含む）は、特に明記のない限り、たとえ開示されている構造と構造的に等価ではないとしても、説明されているコンポーネントの指定された機能（たとえば、機能等価物）を実行する任意のコンポーネント、および本請求に係る主題の本明細書に説明される例示的な態様において機能を実行する任意のコンポーネントに対応することが意図されている。この点において、さらに、本技術革新は、システムと、本請求項に係る主題のさまざまな方法の動作および／またはイベントを実行するためのコンピュータ実行可能命令を有するコンピュータ可読媒体とを含むことが理解されよう。

【００７７】

加えて、本主題の技術革新の特定の特徴が複数の実施態様のうちの１つのみに関して開示されている場合もあるが、そのような特徴は、任意の所定のまたは特定の用途にとって望ましく有利なものとなりうるので、その他の実施態様の１つまたは複数のその他の特徴と組み合わせることもできる。さらに、「含む（ｉｎｃｌｕｄｅ）」、および「含む（ｉｎｄｌｕｄｉｎｇ）」という用語ならびにその変形が詳細な説明または特許請求の範囲で使用される限りにおいて、それらの用語は、「備える（ｃｏｍｐｒｉｓｉｎｇ）」と同様の方法で包括的であることが意図されている。

【図面の簡単な説明】

【００７８】

【図１】データ操作の実施を可能にするために式ツリー表現の作成を容易にする例示的なシステムを示すブロック図である。

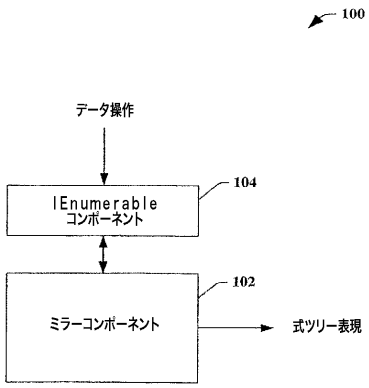
【図２】クエリプロセッサでのデータ操作を採用するためにデータのコレクションの式ツリー表現の使用を容易にする例示的なシステムを示すブロック図である。

【図３】データのコレクションにクエリを行うためにプラグ可能クエリプロセッサの実施を容易にし、クエリは式ツリーによって表すことができる例示的なシステムを示すブロック図である。

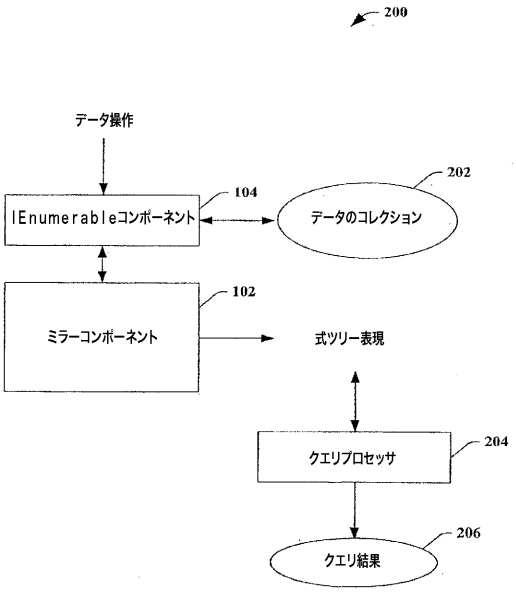
【図４】データにクエリを行うジェネリックインターフェイスを提供するためにＩＥｎｕ

ｍｅｒａｂｌｅインターフェイスおよびＩＥｎｕｍｅｒａｂｌｅ＜Ｔ＞インターフェイスのミラーリングを容易にする例示的なシステムを示すブロック図である。
【図５】本主題の技術革新によるさまざまな関数の例示的な関係を示すブロック図である。
【図６】クエリプロセッサでのデータ操作を採用するためにデータのコレクションの式ツリー表現の使用を容易にする例示的なシステムを示すブロック図である。
【図７】データ操作の実施を可能にするために式ツリー表現を作成する例示的な方法を示す図である。
【図８】式ツリーによって表すことができるデータのコレクションにクエリを行うためにプラグ可能クエリプロセッサの実施を容易にする例示的な方法を示す図である。
【図９】クエリプロセッサによるデータ操作を採用するためにデータのコレクションの式ツリー表現を使用する例示的な方法を示す図である。
【図１０】コンパイラオペレーティング環境の例を示す概略ブロック図である。
【図１１】本請求項に係る主題の新規な態様が採用されうる例示的なネットワーキング環境を示す図である。
【図１２】本請求項に係る主題に従って採用されうる例示的なオペレーティング環境を示す図である。

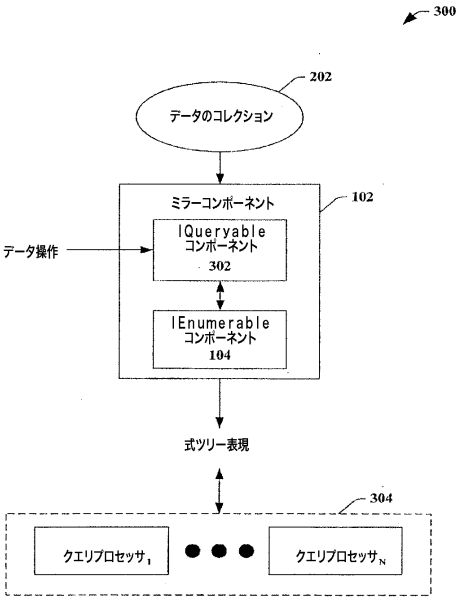
【図１】



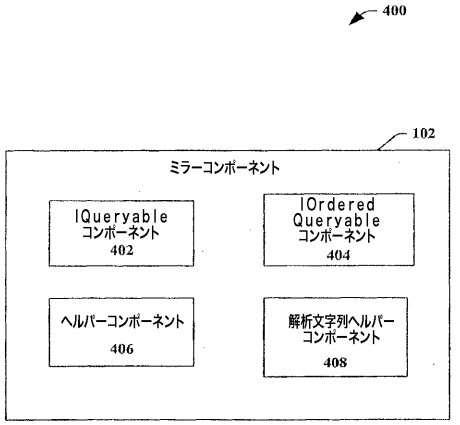
【図２】



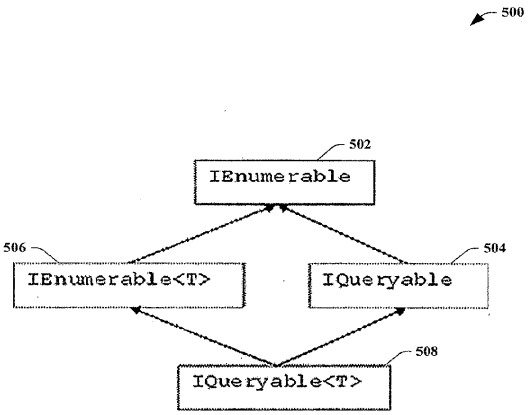
【図３】


300
202
データのコレクション
ミラーコンポーネント
102
データ操作
IQueryable
コンポーネント
302
IEnumerable
コンポーネント
104
式ツリー表現
304
クエリプロセッサ1
クエリプロセッサN

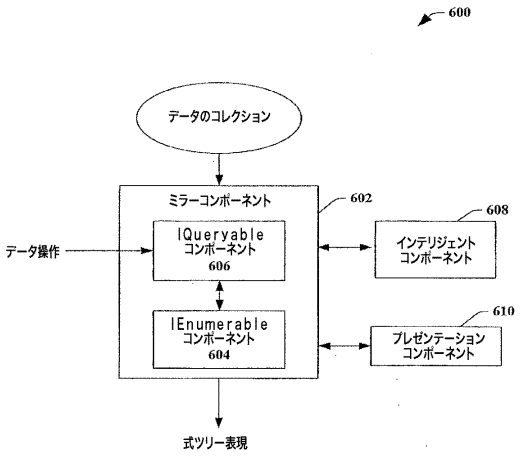
【図４】


400
102
ミラーコンポーネント
IQueryable
コンポーネント
402
IOrdered
Queryable
コンポーネント
404
ヘルパーコンポーネント
406
解析文字列ヘルパー
コンポーネント
408

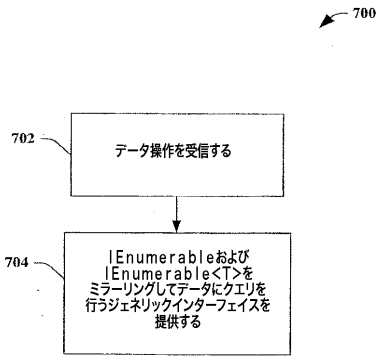
【図５】


500
502
IEnumerable
506
IEnumerable<T>
504
IQueryable
508
IQueryable<T>

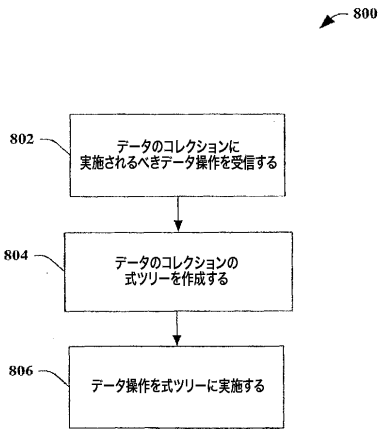
【図６】


600
データのコレクション
ミラーコンポーネント
602
データ操作
IQueryable
コンポーネント
606
608
インテリジェント
コンポーネント
IEnumerable
コンポーネント
604
610
プレゼンテーション
コンポーネント
式ツリー表現

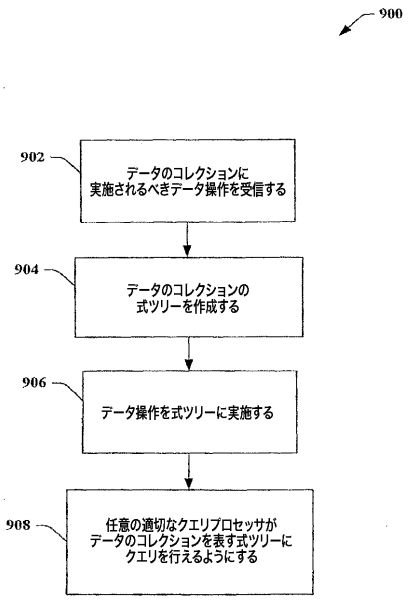
【図７】


700
702
データ操作を受信する
704
IEnumerableおよび
IEnumerable<T>を
ミラーリングしてデータにクエリを
行うジェネリックインターフェイスを
提供する

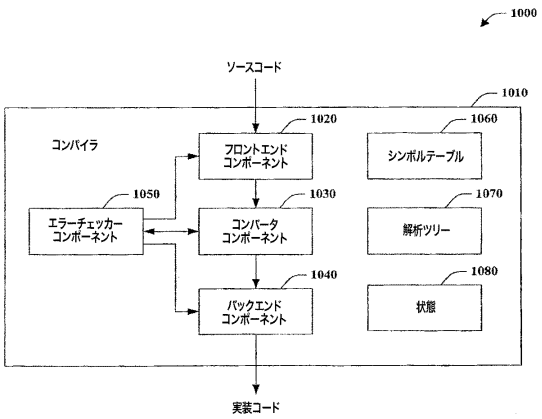
【図８】


800
802
データのコレクションに
実施されるべきデータ操作を受信する
804
データのコレクションの
式ツリーを作成する
806
データ操作を式ツリーに実施する

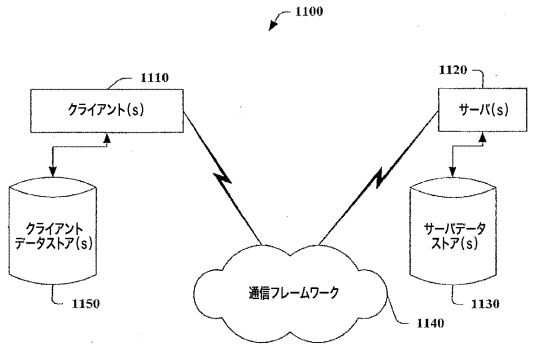
【図９】


900
902
データのコレクションに
実施されるべきデータ操作を受信する
904
データのコレクションの
式ツリーを作成する
906
データ操作を式ツリーに実施する
908
任意の適切なクエリプロセッサが
データのコレクションを表す式ツリーに
クエリを行えるようにする

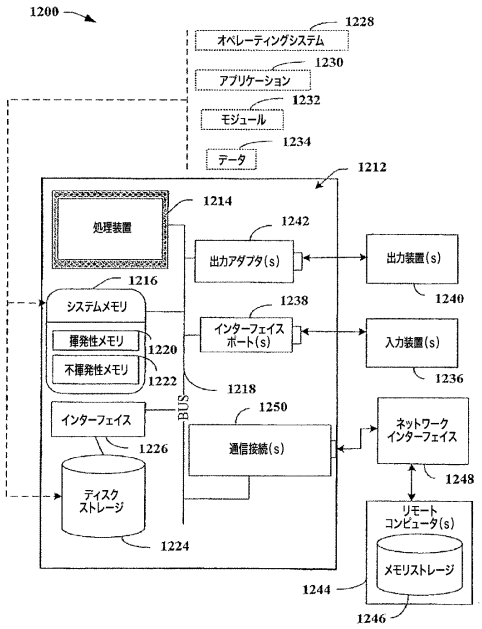
【図１０】


1000
ソースコード
1010
コンパイラ
1020
フロントエンド
コンポーネント
1060
シンボルテーブル
1050
エラーチェッカー
コンポーネント
1030
コンパータ
コンポーネント
1070
解析ツリー
1040
バックエンド
コンポーネント
1080
状態
実装コード

【図１１】


1100
1110
クライアント(s)
1120
サーバ(s)
クライアント
データストア(s)
サーバデータ
ストア(s)
通信フレームワーク
1150
1130
1140

【図１２】


1200
1228
オペレーティングシステム
1230
アプリケーション
1232
モジュール
1234
データ
1212
1214
処理装置
1242
出力アダプタ(s)
出力装置(s)
1240
1216
システムメモリ
1238
インターフェイス
ポート(s)
揮発性メモリ
1220
入力装置(s)
1236
不揮発性メモリ
1222
1218
BUS
1250
インターフェイス
1226
通信接続(s)
ネットワーク
インターフェイス
1248
ディスク
ストレージ
1224
リモート
コンピュータ(s)
メモリストレージ
1244
1246

フロントページの続き

(72)発明者 マシュー ジェイ. ウォーレン
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション インターナショナル パテンツ内
(72)発明者 ヘンリクス ヨハネス マリア マイヤー
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション インターナショナル パテンツ内
(72)発明者 ディネシュ シー. クルカルニ
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション インターナショナル パテンツ内
(72)発明者 マッズ トージャーセン
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション インターナショナル パテンツ内

審査官 田川 泰宏

(56)参考文献 室住 正晴, 次世代標準「ＳＱＬ９９」 最終回, ＩＮＴＥＲＯＰ ＭＡＧＡＺＩＮＥ, 日本, ソフトバンクパブリッシング株式会社, ２０００年 2月 1日, 第10巻,第2号, p.95-102
佐々木 整, ＪＤＢＣによるデータベース連係のポイント, ＪＡＶＡ ＰＲＥＳＳ, 日本, （株）技術評論社, ２００５年 6月１８日, 第42巻, p.43-53

(58)調査した分野(Int.Cl., ＤＢ名)
G06F 12/00
G06F 9/44