

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2011-510372
(P2011-510372A)

(43) 公表日 平成23年3月31日(2011.3.31)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 17/30 (2006.01)	G06F 17/30 140	5B075
G06F 17/21 (2006.01)	G06F 17/30 330B	5B109
	G06F 17/30 350C	
	G06F 17/21 590E	

審査請求 未請求 予備審査請求 未請求 (全 21 頁)

(21) 出願番号 特願2010-541777 (P2010-541777)
 (86) (22) 出願日 平成21年1月7日(2009.1.7)
 (85) 翻訳文提出日 平成22年6月30日(2010.6.30)
 (86) 国際出願番号 PCT/EP2009/050120
 (87) 国際公開番号 W02009/090130
 (87) 国際公開日 平成21年7月23日(2009.7.23)
 (31) 優先権主張番号 08150284.1
 (32) 優先日 平成20年1月15日(2008.1.15)
 (33) 優先権主張国 欧州特許庁(EP)

(71) 出願人 390009531
 インターナショナル・ビジネス・マシーンズ・コーポレーション
 INTERNATIONAL BUSINESS MACHINES CORPORATION
 アメリカ合衆国10504 ニューヨーク州 アーモンク ニュー オーチャードロード
 (74) 代理人 100108501
 弁理士 上野 剛史
 (74) 代理人 100112690
 弁理士 太佐 種一
 (74) 代理人 100091568
 弁理士 市位 嘉宏

最終頁に続く

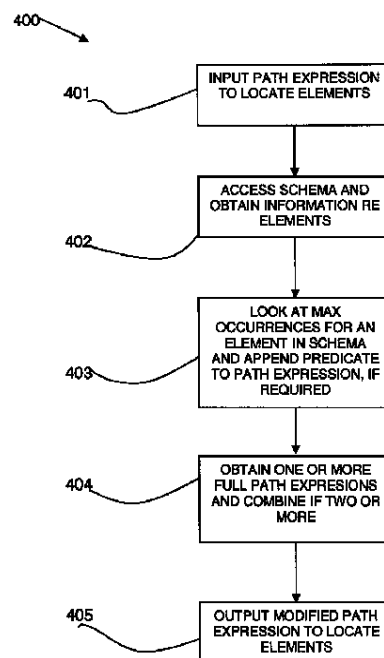
(54) 【発明の名称】 データ構造のナビゲーションのための方法、システム、およびコンピュータ・プログラム (データ構造のナビゲーションのための方法およびシステム)

(57) 【要約】

【課題】 データ構造のナビゲーションのための方法およびシステムを提供する。

【解決手段】 この方法は、ランタイム時にデータ構造(101)内で1つまたは複数の要素(110)を見つけるために、データ構造(101)の照会のための入力パス式(104)を受信することを含む。入力パス式(104)は1つまたは複数のパス・ステップを含む。この方法は、データ構造(101)のスキーマ(103)にアクセスすること、および、1つまたは複数の要素(110)に関する情報を取得することを含む。パス式(104)は、パス・ステップ内で見つかった要素の出現数を制限するために、1つまたは複数のパス・ステップにおけるインデックスを指定するように修正される。一実施形態では、さらにパス式(104)は、任意のワイルドカード文字を拡張して、要素(110)の1つまたは複数の全パスを指定し、2つまたはそれ以上の全パスが存在する場合、それらは合併演算子によって組み合わせられ、修正されたパス式は、ランタイム時にデータ構造(101)を照会(111)するために使用される。

FIG. 4



【特許請求の範囲】**【請求項 1】**

ランタイム時にデータ構造(101)内で1つまたは複数の要素(110)を見つけるために、データ構造(101)の照会のための入力パス式(104)を受信するステップであって、前記入力パス式(104)は1つまたは複数のパス・ステップを含む、受信するステップと、

前記データ構造(101)のスキーマ(103)にアクセスするステップ、および、前記1つまたは複数の要素(110)に関する情報を取得するステップと、

パス・ステップ内で見つかった要素の出現数を制限するために、1つまたは複数のパス・ステップにおけるインデックスを指定するように前記入力パス式(104)を修正するステップと、

ランタイム時に前記データ構造(101)を照会(111)するために前記修正されたパス式を使用するステップと、

を含む、データ構造(101)のナビゲーションのための方法。

【請求項 2】

前記入力パス式(104)が、ワイルドカード文字または下位軸のうちの1つまたは複数を含む、請求項1に記載の方法。

【請求項 3】

前記入力パス式(104)を修正するステップが前記要素(110)の1つまたは複数の全パスを指定し、2つまたはそれ以上の全パスが合併演算子によって組み合わせられる、請求項1または2に記載の方法。

【請求項 4】

前記パス・ステップのうちの1つまたは複数でインデックスを指定するステップが、前記パス要素名に前記インデックスの述語を添付する、請求項1から3のいずれか一項に記載の方法。

【請求項 5】

情報を取得するステップが、前記スキーマ(103)内での属性の最大出現に注目するステップと、これが存在しないかまたは1である場合、1の前記述語を前記パス要素名に添付するステップとを含む、前記請求項のいずれか一項に記載の方法。

【請求項 6】

前記スキーマ(103)が、前記データ構造(101)の要素(110)をそのルートまで追跡するための情報を含む、前記請求項のいずれか一項に記載の方法。

【請求項 7】

前記入力パス式(104)がXPath式であり、前記データ構造(101)が関連付けられたスキーマ(103)を備えたドキュメント・オブジェクト・モデル構造である、前記請求項のいずれか一項に記載の方法。

【請求項 8】

ランタイム時にデータ構造(101)内で1つまたは複数の要素(110)を見つけるために、データ構造(101)の照会のための入力パス式(104)を受信するための手段であって、前記入力パス式(104)は1つまたは複数のパス・ステップを含む、受信するための手段と、

前記データ構造(101)のスキーマ(103)にアクセスし、前記1つまたは複数の要素(110)に関する情報を取得するための手段と、

パス・ステップ内で見つかった要素の出現数を制限するために、1つまたは複数の前記パス・ステップにおけるインデックスを指定するように前記入力パス式(104)を修正するための手段と、

前記データ構造(101)を照会(111)するためにランタイム時に使用するように修正された前記パス式を出力するための手段と、

を備える、データ構造(101)のナビゲーションのためのシステム。

【請求項 9】

前記入力パス式(104)が、ワイルドカード文字または下位軸のうちの1つまたは複数を含む、請求項8に記載のシステム。

【請求項10】

前記入力パス式(104)を修正するための手段が前記要素(110)の1つまたは複数の全パスを指定し、2つまたはそれ以上の全パスが合併演算子によって組み合わせられる、請求項8または9に記載のシステム。

【請求項11】

前記パス・ステップのうちの1つまたは複数でインデックスを指定することが、前記パス要素名に前記インデックスの述語を添付する、請求項8から10のいずれか一項に記載のシステム。

10

【請求項12】

情報を取得するステップが、前記スキーマ(103)内での属性の最大出現に注目するための手段と、これが存在しないかまたは1である場合、1の前記述語を前記パス要素名に添付するための手段とを含む、請求項8から11のいずれか一項に記載のシステム。

【請求項13】

前記スキーマ(103)が、前記データ構造(101)の要素(110)をそのルートまで追跡するための情報を含む、請求項8から12のいずれか一項に記載のシステム。

【請求項14】

前記入力パス式(104)がXPath式であり、前記データ構造(101)が関連付けられたスキーマ(103)を備えたドキュメント・オブジェクト・モデル構造である、請求項8から13のいずれか一項に記載のシステム。

20

【請求項15】

前記システムが、ランタイム時に配置されることになる照会を構築するためのツールキット(105)である、請求項8から14のいずれか一項に記載のシステム。

【請求項16】

前記スキーマ(103)へアクセスしないランタイム・エンジン(102)を含み、前記ランタイム・エンジン(102)が、データ構造(101)内で1つまたは複数の要素(110)を見つけるために前記修正されたパス式を使用する、請求項8から15のいずれか一項に記載のシステム。

【請求項17】

前記ランタイム・エンジン(102)がメッセージ・ブローカであり、前記データ構造(101)がメッセージである、請求項16に記載のシステム。

30

【請求項18】

ランタイム時にデータ構造(101)内で1つまたは複数の要素(110)を見つけるために、データ構造(101)の照会のための入力パス式(104)を受信するステップであって、前記入力パス式(104)は1つまたは複数のパス・ステップを含む、受信するステップと、

前記データ構造(101)のスキーマ(103)にアクセスするステップ、および、前記1つまたは複数の要素(110)に関する情報を取得するステップと、

パス・ステップ内で見つかった要素の出現数を制限するために、1つまたは複数の前記パス・ステップにおけるインデックスを指定するように前記入力パス式(104)を修正するステップと、

40

ランタイム時に前記データ構造(101)を照会(111)するために前記修正されたパス式を使用するステップと、

を実行するための、コンピュータ読み取り可能プログラム・コード手段を備える、コンピュータ読み取り可能ストレージ・メディア上に格納されたコンピュータ・プログラム。

【請求項19】

前記プログラムがコンピュータ上で実行された場合、請求項1から7のいずれか一項に記載の方法を実行するように適合されたプログラム・コード手段を備える、コンピュータ・プログラム。

50

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、データ構造のナビゲーションの分野に関する。具体的に言えば、データ構造を照会するためにパス式 (path expression) を使用するナビゲーションに関する。

【背景技術】**【0002】**

データ構造は、パス式を使用して照会することができる。たとえば XPath (XML Path Language) の何らかの変形などのパス言語を使用した、XML (拡張マークアップ言語) データの格納および照会が可能で、多くのデータベース・システムが存在する。XPath は、ツリー・データ構造を照会する W3C 標準方法である。XPath は、パス式に基づいて XML データを照会可能にするものである。本明細書は XPath 式および照会について言及するが、他のパス言語で作成された他の形のパス式を含むものと解釈されるべきである。パス式とは、XML ドキュメントの階層構造へのパスを指定する任意の式のことである。パス式によって識別される XML ドキュメントの一部は、パス式に一致する任意のパスの終わりにある XML ドキュメントの構造内に常駐する部分である。

10

【0003】

XML ドキュメントは、しばしば、ドキュメント・オブジェクト・モデル (DOM) 構造またはツリーとして表される。パス照会は、これらの構造をトラバースすることによって評価されるため、照会最適化の要件は、トラバースされるノードの数を効率的に減らすことである。

20

【0004】

XML スキーマは、XML ドキュメントの精密な構造形式を定義する。このスキーマはランタイム時に使用できない可能性がある。

【0005】

ドキュメントのスキーマにアクセスできないパス・ランタイム・エンジンを考えてみると、解釈済みの DOM ツリーのみを照会している。ユーザが自由形式パス式を入力する (すなわちパス・ビルダを介さない) オーサリング・ツールは、スキーマにアクセスできない。

30

【先行技術文献】**【非特許文献】****【0006】**

【非特許文献 1】 カリフォルニア大学デービス校、コンピュータ・サイエンス学部、April Kwong および Michael Gertz による「Schema-based Optimization of XPath Expression」

【発明の概要】**【発明が解決しようとする課題】****【0007】**

ユーザが、XML ドキュメント内の任意の要素を、そこまでの全パスを指定する必要なく、名前で見つけられるようにするために、下位 (descendant) (//) パスを使用することが有益であろう。しかしながら、下位軸は、この要素のすべての出現を見つけるために DOM ツリー全体をトラバースする必要があるため、ランタイム・エンジンにとって非常に効率が悪い。

40

【課題を解決するための手段】**【0008】**

カリフォルニア大学デービス校、コンピュータ・サイエンス学部、April Kwong および Michael Gertz による「Schema-based Optimization of XPath Expression」では、スキーマから生成されたパス等価クラスを使用する XPath 式の最適化を開示している。この最適化方法は、XPath 式におけるワイルドカードの拡張を含む。

【0009】

50

本発明の第1の態様によれば、ランタイム時にデータ構造内で1つまたは複数の要素を見つけるために、データ構造の照会のための入力パス式を受信することによって、入力パス式は1つまたは複数のパス・ステップを含む、受信すること、データ構造のスキーマにアクセスし、1つまたは複数の要素に関する情報を取得すること、パス・ステップ内で見つかった要素の出現数を制限するために、1つまたは複数のパス・ステップにおけるインデックスを指定するようにパス式を修正すること、および、ランタイム時にデータ構造を照会するために修正されたパス式を使用することを含む、データ構造のナビゲーションのための方法が提供される。

【0010】

本発明の第2の態様によれば、ランタイム時にデータ構造内で1つまたは複数の要素を見つけるために、データ構造の照会のための入力パス式を受信するための手段によって、入力パス式は1つまたは複数のパス・ステップを含む、受信するための手段と、データ構造のスキーマにアクセスし、1つまたは複数の要素に関する情報を取得するための手段と、パス・ステップ内で見つかった要素の出現数を制限するために、1つまたは複数のパス・ステップにおけるインデックスを指定するようにパス式を修正するための手段と、データ構造を照会するためにランタイム時に使用するように修正されたパス式を出力するための手段とを備える、データ構造のナビゲーションのためのシステムが提供される。

10

【0011】

本発明の第3の態様によれば、ランタイム時にデータ構造内で1つまたは複数の要素を見つけるために、データ構造の照会のための入力パス式を受信するステップによって、入力パス式は1つまたは複数のパス・ステップを含む、受信するステップと、データ構造のスキーマにアクセスし、1つまたは複数の要素に関する情報を取得するステップと、パス・ステップ内で見つかった要素の出現数を制限するために、1つまたは複数のパス・ステップにおけるインデックスを指定するようにパス式を修正するステップと、ランタイム時にデータ構造を照会するために修正されたパス式を使用するステップとを実行するための、コンピュータ読み取り可能プログラム・コード手段を備える、コンピュータ読み取り可能ストレージ・メディア上に格納されたコンピュータ・プログラム製品が提供される。

20

【0012】

好ましくは、費用のかかるメモリ・データ構造の基本テーブルおよび構造体の完全スキャンを招くことなく、XMLドキュメントからデータを取り出すための、パス・ベースの照会に関する方法を提供することによって、パス・ベースの照会の処理時間を改善するソリューションが提供される。好ましくは、照会における要素の反復に関する情報を取得および使用するソリューションが提供される。

30

【0013】

記載された方法およびシステムは、データ構造内の要素が反復可能であり、反復が存在するかどうか、および反復がどれだけあるかを知ることによって、データ構造のナビゲーションを最適化することが可能である、基礎となるモデルを有するデータ構造の照会を実行可能にするものである。

【0014】

本発明は、好ましくはオリジナルのパス式を採用し、スキーマを使用して、これをランタイム・エンジンによってより効率的に処理可能な形に変換する。スキーマは、任意の要素をドキュメント・ルートまで遡って追跡するために必要な情報を含むため、要素への全パスを生成することができる。これによって好ましくは、ランタイム・エンジンが、要素を「検索する」のではなく、直接それへとナビゲートすることが可能になる。

40

【0015】

次に、本発明の諸実施形態について、添付の図面を参照しながら単なる例として説明する。

【図面の簡単な説明】

【0016】

【図1】本発明の一実施形態に従ったシステムを示す概略図である。

50

【図2】本発明の一実施形態に従ったシステムを示す概略図である。

【図3】本発明の一実施形態を内部に実装可能なコンピュータ・システムを示すブロック図である。

【図4】本発明の一実施形態に従ったドキュメント構造を示す概略図である。

【図5】本発明の一実施形態に従った汎用プロセスを示す流れ図である。

【図6】本発明の一実施形態に従ったプロセス例を示す流れ図である。

【発明を実施するための形態】

【0017】

図1を参照すると、データ構造101内の1つまたは複数の要素110を見つけるために、ランタイム・エンジン102によってデータ構造101が照会される、システム100の概略図が示されている。

10

【0018】

データ構造101は、たとえば、ドキュメント・オブジェクト・モデル(DOM)ツリーとして表されるXMLドキュメントとすることができる。他の例では、データ構造101は、同様のツリー・データ構造に解析される非XMLデータである。一実施形態では、データ構造101はメッセージ・ブローカによって照会されるメッセージである。データ構造101は、関連付けられたスキーマ103を有する。

【0019】

パス式104は、1つまたは複数の要素110を見つけるためにパス式104の出現に関してデータ構造101を検索するランタイム・エンジン102へ、照会111として入力される。

20

【0020】

ツールキット105は、ランタイム・エンジン102へと配置される照会111を構築するために使用される。記載されたシステム100では、ツールキット105はパス式104を修正するためにスキーマ103にアクセスする。

【0021】

照会されているデータ構造101のスキーマ103からの情報を使用して、パス式104は、照会されている要素110への全パスおよび他の情報を含むように修正される。これは、要素110に関する検索よりもむしろ要素110の追跡に役立ち、ランタイム・エンジン102の効率を上げる。またスキーマ103が、照会されている要素110の出現

30

数を示す場合、ランタイム・エンジン102は、その特定の要素110の一定の出現数が見つけた後、停止することができる。

【0022】

図2を参照すると、パス式104が入力されたツールキット105の詳細図が示されている。スキーマは、スキーマ情報120を取得するために参照される。修正子121は、照会出力122を提供するために、スキーマ情報120を使用してパス式104を修正する。照会出力122は、データ構造を照会するためにランタイム・エンジンによって使用される。たとえばスキーマを使用して、照会されている要素の出現数を決定することが可能であり、次にこの情報は、照会出力122に述語(predicate)を追加するために、修正子121によって使用される。

40

【0023】

図3を参照すると、ツールキット105を実装するための例示的システムは、バス・システム203を介してメモリ要素に直接または間接的に結合された少なくとも1つのプロセッサ201を含む、プログラム・コードの格納あるいは実行またはその両方に好適な、処理システム200を含む。メモリ要素は、プログラム・コードの実際の実行中に使用されるローカル・メモリ、大容量ストレージ、および、実行中にコードを大容量ストレージから取り出さなければならない回数を減らすために少なくともいくつかのプログラム・コードの一時ストレージを提供するキャッシュ・メモリを、含むことができる。

【0024】

メモリ要素は、読み取り専用メモリ(ROM)204およびランダム・アクセス・メモ

50

リ (RAM) 205 の形の、システム・メモリ 202 を含むことができる。基本入力/出力システム (BIOS) 206 は、ROM 204 に格納可能である。システム・ソフトウェア 207 は、オペレーティング・システム・ソフトウェア 208 を含み、RAM 205 内に格納可能である。ソフトウェア・アプリケーション 210 も RAM 205 内に格納可能である。

【0025】

システム 200 は、磁気ハード・ディスク・ドライブなどの 1 次ストレージ手段 211、ならびに、磁気ディスク・ドライブおよび光ディスク・ドライブなどの 2 次ストレージ手段 212 も、含むことができる。ドライブおよびそれらの関連付けられたコンピュータ読み取り可能メディアは、コンピュータ実行可能命令、データ構造、プログラム・モジュール、およびシステム 200 に関する他のデータの、不揮発性ストレージを提供する。ソフトウェア・アプリケーションは、1 次および 2 次のストレージ手段 211、212、ならびにシステム・メモリ 202 上に格納可能である。

10

【0026】

コンピューティング・システム 200 は、ネットワーク・アダプタ 216 を介した 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、ネットワーク化環境内で動作可能である。

【0027】

入力/出力デバイス 213 は、直接、または介在する I/O コントローラを介して、システムに結合することができる。ユーザは、キーボード、ポインティング・デバイス、または他の入力デバイス (たとえば、マイクロフォン、ジョイスティック、ゲーム・パッド、衛星用アンテナ、スキャナなど) などの入力デバイスを介して、コマンドおよび情報をシステム 200 に入力することができる。出力デバイスは、スピーカ、プリンタなどを含むことができる。ディスプレイ・デバイス 214 も、ビデオ・アダプタ 215 などのインターフェースを介してシステム・バス 203 に接続される。

20

【0028】

パス言語は、データ構造 (XML ドキュメントなど) のツリー表現に基づくものであり、ツリー周辺をナビゲートし、様々な基準によってノードを選択する機能を提供する。パス式は、1 つのノード、現在の「コンテキスト・ノード」から、他のノードまたはノード・セットへと取得するための、一連のステップとして作成される。このステップは、「/」(すなわちパス) 文字によって分離される。各ステップは、軸指定子、ノード・テスト、および述語という、3 つのコンポーネントを有する。

30

【0029】

最も単純なパスは、

【数 1】

/A/B/C

の形を取り、これは、データ構造の最も外側の要素を形成する A 要素の子である B 要素の子である C 要素を選択する。

【0030】

より複雑な式は、デフォルトの「子」軸以外の軸、単純名以外のノード・テスト、または述語を指定することによって、構築可能であり、これは任意のステップ後に角括弧で作成することができる。たとえば、式

40

【数 2】

A//B/*[1]

は、それ自体が子、または他の、現在のコンテキスト・ノードの子である A 要素のより深い下位 (「//」) である、B 要素の子 (「/」) である、第 1 の要素 (「[1]」) を、その名前 (「*」) であれば何でも、選択する (この式は「/」で始まっていない)。ドキュメント内にいくつかの好適な B 要素が存在する場合、これは実際には、それらのす

50

べての第1の子のセットを戻す。軸指定子は、データ構造のツリー表現内のナビゲーション方向を示す。

【0031】

いかなる複雑さの式も、述語として角括弧内に指定可能であり、これは、パス式によって先行ノードが一致することになる前に満たされなければならない。ステップ内の述語数には制限が無く、パス内の最終ステップに限定される必要がない。それらは任意の深さまでネストすることができる。述語内に指定されたパスは、現在のステップの（すなわち、直前のノード・テストの）コンテキストで開始され、そのコンテキストを変更しない。すべての述語は、一致の出現に関して満たされなければならない。述語の順序は有意であり、各述語はロケーション・ステップの選択されたノード・セットを順にフィルタリングする。

10

【0032】

XMLスキーマは、あるタイプのXMLドキュメントの記述であり、通常は、XML自体によって課せられる基本構文制約に加えて、そのタイプのドキュメントの構造およびコンテンツへの制約に関して表される。XMLスキーマは、比較的高レベルの抽象でのドキュメント・タイプの観点を提供する。スキーマは、非XMLデータ構造に関連して使用することもできる。たとえば、XMLスキーマを使用してドキュメント構造をモデル化した後、XSD(XMLスキーマ定義)注釈を使用することによって、各要素の物理フォーマットを記述する。

20

【0033】

XMLスキーマを表すために特別に開発された言語がある。XML仕様固有のドキュメント・タイプ定義(DTD)言語は、比較的機能が制限されているが、スキーマの表現とは別にXMLにおける他の用途も有する、スキーマ言語である。2つの他の非常に普及している、より表現的なXMLスキーマ言語は、XMLスキーマ(W3C)およびRELAX NG(Regular Language for XML Next Generation)である。XMLドキュメントをスキーマと関連付けるためのメカニズムは、スキーマ言語に応じて変化する。関連付けは、XMLドキュメント自体の中のマークアップを介して、または何らかの外部手段を介して、達成可能である。

【0034】

説明されたシステムでは、パス式は以下のような3通りの可能な方法で修正できる。

30

1. ワイルドカード「*」または下位軸「//」の単純な拡張であり、スキーマは1つの可能な解決のみを示唆する。

2. パス合併演算子(unionoperator)「|」と組み合わせられた複数のパスを備える式への、ワイルドカード「*」または下位軸「//」の拡張であり、スキーマは複数の可能な解決を示唆する。

3. インデックス([1])の挿入であり、スキーマは要素が1回のみ現れることを示唆する。同様に、[position()<=n]の挿入であり、スキーマは要素がn回出現可能であることを示唆し、ここでn>1である。

【0035】

例示のために、以下のXMLドキュメントを使用して作業例を示す。

40

【数3】

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <topic>
    <list>
      <item>...</item>
      <item>...</item>
      <item>...</item>
    </list>
  </topic>
  <index>
    <item>...</item>
    <item>...</item>
  </index>
</document>
```

10

【0036】

このドキュメントは、以下のXMLスキーマ(XSD)によってモデル化することができる。

20

【数 4】

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="document">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="topic"/>
        <xsd:element ref="index"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="topic">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="list"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="list">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element minOccurs="0" maxOccurs="unbounded" ref="item"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="item">
    <xsd:complexType/>
  </xsd:element>
  <xsd:element name="index">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element minOccurs="0" maxOccurs="unbounded" ref="item"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

【0037】

スキーマは、各要素の最低および最高の出現を含む、XMLドキュメントの許容構造を記述する。スキーマは、図4に示されたドキュメント構造（スキーマ・ツリー）300を表す。

【0038】

図4に示されたドキュメント構造300は、「document」のルート・ノード311からの2つの分岐310、320を含む。第1の分岐310は、子ノード314「item」を有することが可能な子ノード313「list」を備える、ルート・ノード3

10

20

30

40

50

1 1の子ノード3 1 2「topic」を含む。「item」のノード3 1 4、3 2 3は、ゼロまたはそれ以上の出現を有することができる。他のノード3 1 1、3 1 2、3 1 3、3 2 2は1つの出現のみを含むことができる。

【0039】

次に図5を参照すると、流れ図400は、データ構造をナビゲートするための汎用プロセスを示す。パス式は、データ構造内の1つまたは複数の要素を見つけるために入力される(401)。データ構造に関するスキーマにアクセスし(402)、パス式の要素に関する情報が取得される。要素に関する最大出現がスキーマから参照され(403)、出現を制限する必要がある場合は、述語がパス式に添付される。1つまたは複数の全パス式が取得され(404)、2つまたはそれ以上ある場合は組み合わせられる。

10

【0040】

パス式は、要素に関する取得された情報を参照しながら修正される。修正されたパス式は、データ構造内の1つまたは複数の要素を見つけるためにランタイム・エンジンへと出力される(405)。

【0041】

この作業例では、XPath式「//item」が与えられた場合、下位軸「//」を拡張するため、および「最初の1つのみを選択する」述語「[1]」を追加するために、図6の流れ図に示されるような以下のステップが実施される。

1. 初期の最適化された式「/item」を作成するために、「/」文字のうちの1つを除去する(501)。

20

2. 「item」という名前の要素を検索する「深さ優先(depth first)」様式で、スキーマ・ツリーをトラバースする(502)。

3. 要素が見つかったかどうかを判別し(503)、見つかった場合は続行し、見つからなかった場合、プロセスは終了する(504)。

4. 見つかった要素が第1の要素であるかどうかを判別する(505)。第1の要素である場合は続行し、第1の要素でない場合は、合併演算子を使用して以前の式に添付する(506)。

5. 「item」要素が見つかった場合、スキーマ内の「maxOccurs」属性に注目する(507)。この属性が「1」(この属性が存在しない場合のデフォルトである)であるかどうかを判別し(508)、「1」である場合は、要素名に述語「[1]」を添付する(510)。「1」より大きいかまたは「未制限」の場合、述語を添付しない(509)。

30

6. スキーマ・ツリー内の現在の要素の親に進む(511)。

7. 親がルート・ノードであるかどうかを判別する(512)。ルート・ノードである場合、別の「item」の検索に進む(502)。

8. 親がルート・ノードでない場合、スキーマから名前を抽出し(513)、分離文字「/」によって分離された以前のステップの前にこれを挿入する。

9. ステップ507へと折り返して、必要に応じて要素名に述語を添付し、次の親に進む。「ルート」要素に達するまで続行する。

10. 他の「item」要素が見つかった場合、ツリーのこの分岐について他の最適化されたパス式を作成する。これを、合併演算子「|」によって分離された以前の式に添付する(506)。スキーマ・ツリー内にそれ以上の「item」の出現が見つからなくなるまで、トラバースを続行する。

40

【0042】

以下に、前述のステップに続くような、最適化された式の構築を示す(オリジナルの式「//item」は、各ステップでの変更が下線で示される)。

【数 5】

- /item
- /list/item
- /list[1]/item
- /topic/list[1]/item
- /topic[1]/list[1]/item
- /document/topic[1]/list[1]/item
- /document[1]/topic[1]/list[1]/item
- /document[1]/topic[1]/list[1]/item | /item
- /document[1]/topic[1]/list[1]/item | /index/item
- /document[1]/topic[1]/list[1]/item | /index[1]/item
- /document[1]/topic[1]/list[1]/item | /document/index[1]/item
- /document[1]/topic[1]/list[1]/item | /document[1]/index[1]/item

10

【0043】

さらに汎用ケースでは、一般的な原理は、X Path 式を 1 回に 1 要素ステップずつ進み、ワイルドカード「*」があれば要素名に置き換え、下位軸「//」があれば完全修飾されたパスに置き換えることである。複数の要素がワイルドカードに一致する場合、それぞれが順番に考慮される。後続のステップでは、ある特定の一致が有効なパスへと至ることができず、そのような場合、その一致は廃棄される旨を示すことができる。複数の一致が可能な場合、各式は組み合わせられ、X Path 合併演算子「|」によって分離される。

20

【0044】

「*/topic//item」の式について考えてみる。

1. 「スキーマ・カーソル」をスキーマ・ツリーのルートに設定し、「X Path カーソル」をオリジナル X Path 式の第 1 のステップに設定する。

2. スキーマ・ツリーを、スキーマ・カーソル位置から深さ優先でトラバースする。

3. スキーマ・ツリー内のカーソル要素が X Path ステップに一致する（ワイルドカードがいずれかの要素に一致する）場合、

a. 要素名（ワイルドカードの場合はスキーマ・ツリーから）を最適化された X Path 式に添付する。

b. スキーマ内の「maxOccurs」属性に注目する。この属性が「1」（この属性が存在しない場合のデフォルトである）である場合、要素名に述語「[1]」を添付する。「1」より大きいかまたは「未制限」の場合、述語を添付しない。

30

c. X Path カーソルがオリジナル X Path 式内の最終ステップである場合、反復を停止し、このステージで構築された新しい式を部分リストに追加して、第 2 ステージからトラバースを続行する。

d. X Path カーソルを、オリジナル X Path 式内の次のステップに移動する。次のステップが下位軸「//」によって分離されている場合、

i. 以前に記述されたアルゴリズムを使用して、これを拡張する（拡張アルゴリズム内の「ルート」要素を、ここでは「//」の前の要素であるものと定義する）。

ii. 拡張の結果、合併演算子「|」によって分離された複数のパスが生じた場合、これまでに最適化された式を複製し、各部分の初めに添付する必要がある。各部分は、このアルゴリズムの剰余に対して別々に扱われなければならない。（たとえば、「/document//item」は、拡張前に「/document[1]//item」を生成することになる。このカーソル位置から「//item」を拡張すると、「/topic[1]/list[1]/item|/index[1]/item」を生成することになる。このステップの後、2つの部分は、/document[1]/topic[1]/list[1]/item および /document[1]/index[1]/item となり、各部分は別々に扱われるものとする。）

40

e. スキーマ・カーソルのそれぞれの子について、次に、スキーマ・カーソルを子に設定する。ステージ 3 に再帰する。（このステージで、各子は最適化された式における別

50

々の部分を表すことになり、そのほとんどが、おそらくはデッドエンドであり、破棄されることになる。アルゴリズムの終わりで、合併演算子「|」を使用してすべての剰余部分が組み合せられる。)

一致しない場合、

a. アルゴリズムのステージ2以降に構築されたパスのこの部分を破棄する。スキーマ・カーソルおよびXPathカーソルを、以前ステージ2の際にそれらがいた場所に再設定する。ステージ2からトラバースを続行する。

4. 合併演算子「|」によって分離された部分リストを組み合わせて、最終式を生成する。

【0045】

以下に、前述のステップに続くような、最適化された式の構築を示す(オリジナルの式「*/topic//item」は、各ステップでの変更が下線で示される)。

【数6】

- document
- document[1]
- document[1]/topic
- document[1]/topic[1]
 - /item
 - /list/item
 - /list[1]/item
- document[1]/topic[1]/list[1]/item

10

20

【0046】

「//item」などの式が入力された場合、このいずれかの出現を見つけるために、データ構造全体をトラバースしなければならない。しかしながらこれは、スキーマを参照することによって取得された全パス式「/document/topic/list/item」を指定することによって、ランタイム・エンジンがitemを直接ナビゲートできるような形に変換することができる。

【0047】

スキーマが、itemが反復しないように指定した場合、各パス・ステップ上に明示的indexを配置し、第1の一致、たとえば「/document[1]/topic[1]/list[1]/item」を見つけた場合、検索を停止できる旨を、ランタイム・エンジンに伝えることができる。「item」は反復要素であるが、その上位(ancestor)要素は反復要素でないことに留意されたい。

30

【0048】

データ構造の異なる部分に名前付き要素「item」が現われた場合、それらは合併演算子「|」を使用して組み合わせることができる。たとえば、「/document[1]/topic[1]/list[1]/item|/document[1]/index[1]/item」となる。この最後の式は、所与のスキーマに関するオリジナルの「//item」と等価であるが、そのスキーマにアクセスできないランタイム・エンジンによってより効率的に処理可能な形である。

40

【0049】

スキーマは、要素が所与の回数だけ、たとえば2回出現するように指定することが可能である。XPathステップ「/element[2]」は、第1の2つの要素ではなく第2の要素を戻すことになる。正しい結果を得るために、[position()<=2]を添付すべきである。従来のシステムでは、ユーザは、ツリーをナビゲートするためのランタイム・エンジンを望むことを示し、すべての一致要素を戻すために、「/aaa/bbb」を指定する。しかしながら、時折、戻す要素が1つしかない場合があり、XPathパーザ(parser)は、従来、完全なツリーを解析しなければならないため、多くの不必要な処理を実行しなければならない。ユーザは、たとえば第1番目などの、ある数の要素のみを希望する場合があり、これを示すため、照会内に、たとえば「aaa/bbb[

50

1]」などと断定することができる。以前には、パーザは依然として完全なリストに戻すが、その後、ユーザが必要とする数に戻すように、そのセットを絞り込むものであった。

【 0 0 5 0 】

説明されたシステムでは、ツールキットは、照会されている要素の出現数を決定するために使用される、データ構造スキーマにアクセスする。次に、この情報を使用して、各照会に述語が追加される。したがって、2つの「 a a a / b b b 」要素のみが存在し、ユーザはこうした要素のすべてについて照会していることが判別された場合、最初の2つの要素のみを探した後、解析を停止すべきであることを、パーザに示すように、照会を修正することができる。

【 0 0 5 1 】

説明された方法およびシステムは、 E S B (E n t e r p r i s e S e r v i c e B u s)、および、メッセージを照会するメッセージ・ブローカの形のランタイム・エンジンを含む X M L 処理製品で、適用可能である。

【 0 0 5 2 】

この方法は、照会として使用するよう修正されたパス式を改善するために、ネットワークを介したクライアントへのサービスとして提供することも可能である。

【 0 0 5 3 】

本発明は、完全なハードウェア実施形態、完全なソフトウェア実施形態、または、ハードウェアおよびソフトウェアの両方の要素を含む実施形態の形を取ることができる。好ましい実施形態では、本発明は、ファームウェア、常駐ソフトウェア、マイクロコードなどを含むがこれらに限定されない、ソフトウェア内に実装される。

【 0 0 5 4 】

本発明は、コンピュータまたは任意の命令実行システムによって、あるいはそれらに関して使用するためのプログラム・コードを提供する、コンピュータ使用可能メディアまたはコンピュータ読み取り可能メディアからのアクセスが可能な、コンピュータ・プログラム製品の形を取ることができる。説明の目的で、コンピュータ使用可能メディアまたはコンピュータ読み取り可能メディアは、命令実行システム、装置、またはデバイスによって、あるいはそれらに関して使用するためのプログラムの含有、格納、通信、伝搬、または移送が可能な、任意の装置とすることができる。

【 0 0 5 5 】

メディアは、電子、磁気、光、電磁、赤外線、または半導体のシステム（あるいは装置またはデバイス）、あるいは伝搬メディアとすることができる。コンピュータ読み取り可能メディアの例には、半導体またはソリッド・ステート・メモリ、磁気テープ、取り外し可能コンピュータ・ディスク、ランダム・アクセス・メモリ（ R A M ）、読み取り専用メモリ（ R O M ）、剛性磁気ディスク、および光ディスクが含まれる。光ディスクの現行例には、コンパクト・ディスク読み取り専用メモリ（ C D - R O M ）、コンパクト・ディスク読み取り / 書き込み（ C D - R / W ）、および D V D が含まれる。

【 0 0 5 6 】

本発明の範囲を逸脱することなく、前述の内容に対する改良および修正を実行することが可能である。

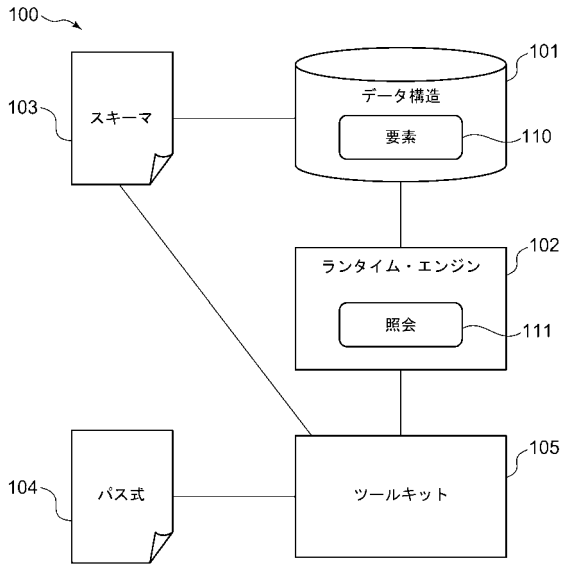
10

20

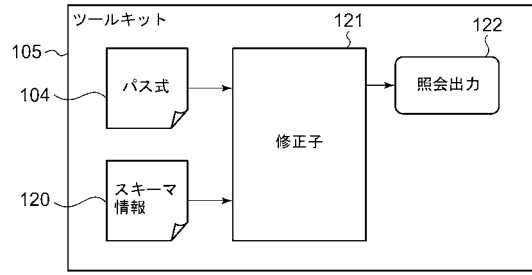
30

40

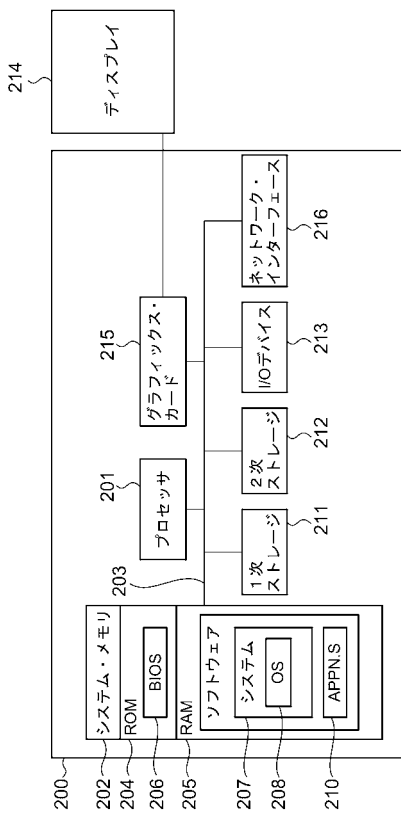
【図1】



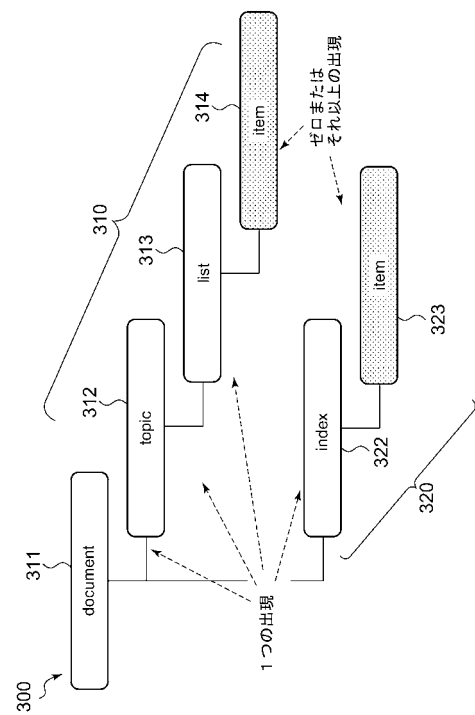
【図2】



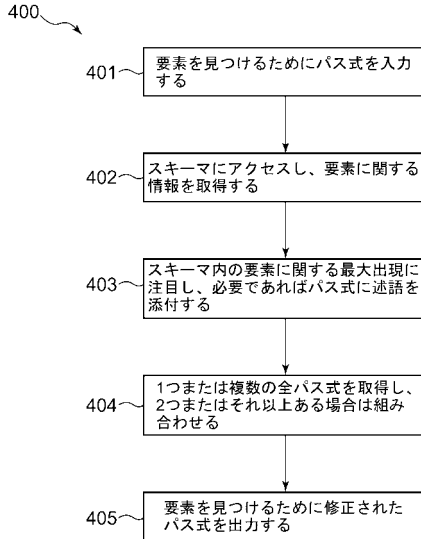
【図3】



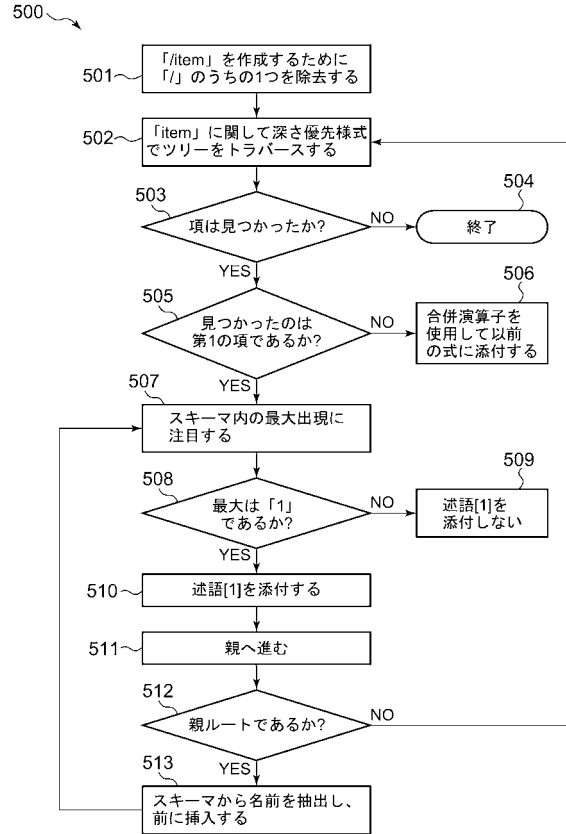
【図4】



【 図 5 】



【 図 6 】



【 手続 補正書 】

【 提出日 】平成22年8月25日 (2010.8.25)

【 手続 補正 1 】

【 補正対象書類名 】特許請求の範囲

【 補正対象項目名 】全文

【 補正方法 】変更

【 補正の内容 】

【 特許請求の範囲 】

【 請求項 1 】

ランタイム時にデータ構造 (1 0 1) 内で1つまたは複数の要素 (1 1 0) を見つけるために、データ構造 (1 0 1) の照会のための入力パス式 (1 0 4) を受信するステップであって、前記入力パス式 (1 0 4) は1つまたは複数のパス・ステップを含む、受信するステップと、

前記データ構造 (1 0 1) のスキーマ (1 0 3) にアクセスするステップ、および、前記1つまたは複数の要素 (1 1 0) に関する情報を取得するステップと、

パス・ステップ内で見つかった要素の出現数を制限するために、1つまたは複数のパス・ステップにおけるインデックスを指定するように前記入力パス式 (1 0 4) を修正するステップと、

ランタイム時に前記データ構造 (1 0 1) を照会 (1 1 1) するために前記修正されたパス式を使用するステップと、

を含む、データ構造 (1 0 1) のナビゲーションのための方法。

【 請求項 2 】

前記入力パス式 (1 0 4) が、ワイルドカード文字または下位軸のうちの1つまたは複数を含む、請求項 1 に記載の方法。

【 請求項 3 】

前記入力パス式(104)を修正するステップが前記要素(110)の1つまたは複数の全パスを指定し、2つまたはそれ以上の全パスが合併演算子によって組み合わせられる、請求項1または2に記載の方法。

【請求項4】

前記パス・ステップのうち1つまたは複数でインデックスを指定するステップが、前記パス要素名に前記インデックスの述語を添付する、請求項1から3のいずれか一項に記載の方法。

【請求項5】

情報を取得するステップが、前記スキーマ(103)内での属性の最大出現に注目するステップと、これが存在しないかまたは1である場合、1の前記述語を前記パス要素名に添付するステップとを含む、前記請求項のいずれか一項に記載の方法。

【請求項6】

前記スキーマ(103)が、前記データ構造(101)の要素(110)をそのルートまで追跡するための情報を含む、前記請求項のいずれか一項に記載の方法。

【請求項7】

前記入力パス式(104)がXPath式であり、前記データ構造(101)が関連付けられたスキーマ(103)を備えたドキュメント・オブジェクト・モデル構造である、前記請求項のいずれか一項に記載の方法。

【請求項8】

ランタイム時にデータ構造(101)内で1つまたは複数の要素(110)を見つけるために、データ構造(101)の照会のための入力パス式(104)を受信するための手段であって、前記入力パス式(104)は1つまたは複数のパス・ステップを含む、受信するための手段と、

前記データ構造(101)のスキーマ(103)にアクセスし、前記1つまたは複数の要素(110)に関する情報を取得するための手段と、

パス・ステップ内で見つかった要素の出現数を制限するために、1つまたは複数の前記パス・ステップにおけるインデックスを指定するように前記入力パス式(104)を修正するための手段と、

前記データ構造(101)を照会(111)するためにランタイム時に使用するよう修正された前記パス式を出力するための手段と、

を備える、データ構造(101)のナビゲーションのためのシステム。

【請求項9】

前記入力パス式(104)が、ワイルドカード文字または下位軸のうち1つまたは複数を含む、請求項8に記載のシステム。

【請求項10】

前記入力パス式(104)を修正するための手段が前記要素(110)の1つまたは複数の全パスを指定し、2つまたはそれ以上の全パスが合併演算子によって組み合わせられる、請求項8または9に記載のシステム。

【請求項11】

前記パス・ステップのうち1つまたは複数でインデックスを指定することが、前記パス要素名に前記インデックスの述語を添付する、請求項8から10のいずれか一項に記載のシステム。

【請求項12】

情報を取得するステップが、前記スキーマ(103)内での属性の最大出現に注目するための手段と、これが存在しないかまたは1である場合、1の前記述語を前記パス要素名に添付するための手段とを含む、請求項8から11のいずれか一項に記載のシステム。

【請求項13】

前記スキーマ(103)が、前記データ構造(101)の要素(110)をそのルートまで追跡するための情報を含む、請求項8から12のいずれか一項に記載のシステム。

【請求項14】

前記入力パス式(104)がXPath式であり、前記データ構造(101)が関連付けられたスキーマ(103)を備えたドキュメント・オブジェクト・モデル構造である、請求項8から13のいずれか一項に記載のシステム。

【請求項15】

前記システムが、ランタイム時に配置されることになる照会を構築するためのツールキット(105)である、請求項8から14のいずれか一項に記載のシステム。

【請求項16】

前記スキーマ(103)へアクセスしないランタイム・エンジン(102)を含み、前記ランタイム・エンジン(102)が、データ構造(101)内で1つまたは複数の要素(110)を見つけるために前記修正されたパス式を使用する、請求項8から15のいずれか一項に記載のシステム。

【請求項17】

前記ランタイム・エンジン(102)がメッセージ・ブローカであり、前記データ構造(101)がメッセージである、請求項16に記載のシステム。

【請求項18】

ランタイム時にデータ構造(101)内で1つまたは複数の要素(110)を見つけるために、データ構造(101)の照会のための入力パス式(104)を受信するステップであって、前記入力パス式(104)は1つまたは複数のパス・ステップを含む、受信するステップと、

前記データ構造(101)のスキーマ(103)にアクセスするステップ、および、前記1つまたは複数の要素(110)に関する情報を取得するステップと、

パス・ステップ内で見つかった要素の出現数を制限するために、1つまたは複数の前記パス・ステップにおけるインデックスを指定するように前記入力パス式(104)を修正するステップと、

ランタイム時に前記データ構造(101)を照会(111)するために前記修正されたパス式を使用するステップと、

を実行するための、コンピュータ読み取り可能プログラム・コード手段を備える、コンピュータ読み取り可能ストレージ・メディア上に格納されたコンピュータ・プログラム。

【請求項19】

プログラムがコンピュータ上で実行された場合、請求項1から7のいずれか一項に記載の方法の各ステップをコンピュータに実行させるようにする、コンピュータ・プログラム

。

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2009/050120

A. CLASSIFICATION OF SUBJECT MATTER INV. G06F17/30		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the International search (name of data base and, where practical, search terms used) EPO-Internal, IBM-TDB		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2005/055355 A1 (MURTHY RAVI [US] ET AL) 10 March 2005 (2005-03-10) abstract paragraphs [0018] - [0085]; figures 1-7	1-19
A	US 2005/065927 A1 (NOURI AHMAD [US] ET AL) 24 March 2005 (2005-03-24) the whole document	1-19
A	US 2003/145047 A1 (UPTON MITCH [US]) 31 July 2003 (2003-07-31) the whole document	1-19
<input type="checkbox"/> Further documents are listed in the continuation of Box C.		<input checked="" type="checkbox"/> See patent family annex.
* Special categories of cited documents :		
A document defining the general state of the art which is not considered to be of particular relevance		*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
E earlier document but published on or after the international filing date		*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)		*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
O document referring to an oral disclosure, use, exhibition or other means		*Z* document member of the same patent family
P document published prior to the international filing date but later than the priority date claimed		
Date of the actual completion of the international search	Date of mailing of the international search report	
6 April 2009	16/04/2009	
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Moon, Timothy	

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2009/050120

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005055355 A1	10-03-2005	WO 2005024670 A1	17-03-2005
US 2005065927 A1	24-03-2005	US 2008140614 A1	12-06-2008
US 2003145047 A1	31-07-2003	NONE	

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW

(72)発明者 コールマン、アンドリュー、ジョン
イギリス国ジー・ユー 3 2 2 ジェイ・ビー ハンプシャー州ピーターズフィールド セルボーン
・クロース 3 7

Fターム(参考) 5B075 ND20 ND35 PP23 QM06 QP10
5B109 NH20 VA02

【要約の続き】

【選択図】 図1