

(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2007-510204
(P2007-510204A)

(43) 公表日 平成19年4月19日(2007.4.19)

(51) Int. Cl. F I テーマコード (参考)
G06F 17/21 (2006.01) G06F 17/21 570L 5B009
 G06F 17/21 501T

審査請求 未請求 予備審査請求 未請求 (全 55 頁)

(21) 出願番号 特願2006-536821 (P2006-536821)
 (86) (22) 出願日 平成16年10月22日 (2004.10.22)
 (85) 翻訳文提出日 平成18年6月13日 (2006.6.13)
 (86) 国際出願番号 PCT/US2004/035032
 (87) 国際公開番号 W02005/043412
 (87) 国際公開日 平成17年5月12日 (2005.5.12)
 (31) 優先権主張番号 60/513,306
 (32) 優先日 平成15年10月22日 (2003.10.22)
 (33) 優先権主張国 米国 (US)
 (31) 優先権主張番号 10/889,614
 (32) 優先日 平成16年7月12日 (2004.7.12)
 (33) 優先権主張国 米国 (US)

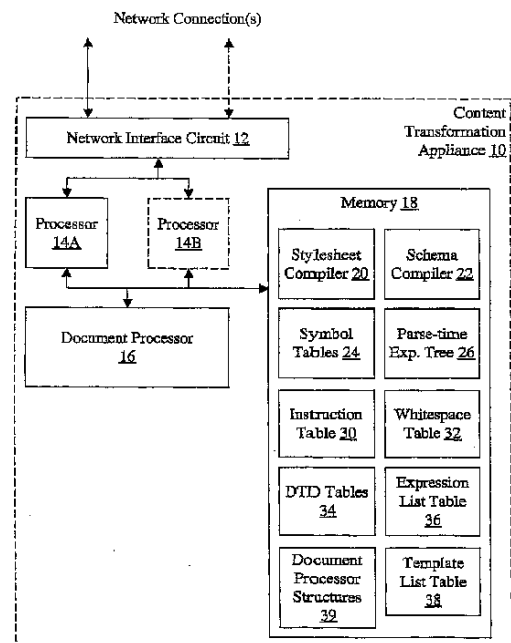
(71) 出願人 593096712
 インテル コーポレーション
 アメリカ合衆国 95052 カリフォル
 ニア州 サンタ クララ ミッション カ
 レッジ プールバード 2200
 (74) 代理人 100078282
 弁理士 山本 秀策
 (74) 代理人 100062409
 弁理士 安村 高明
 (74) 代理人 100113413
 弁理士 森下 夏樹
 (72) 発明者 ラマラオ, カレムブディ ブイ.
 アメリカ合衆国 テキサス 78730,
 オースティン, ブランタース ウッズ
 ドライブ 10105

最終頁に続く

(54) 【発明の名称】 高性能な構造化データ変換用のハードウェア/ソフトウェア・パーティション

(57) 【要約】

本願が提供する装置は、例えば、コンピュータでアクセス可能な媒体であって、その媒体は、実行されるとスタイルシートを1つ以上のデータ構造にコンパイルする複数の命令でコード化される、コンピュータでアクセス可能な媒体と、データ構造および文書を受信するために連結されたハードウェア回路と、を備える装置である。そのハードウェア回路は、そのデータ構造を使用してスタイルシートに指定されたように文書の少なくとも一部分を変換するように構成される。



【特許請求の範囲】

【請求項 1】

実行されるとスタイルシートを 1 つ以上のデータ構造にコンパイルする複数の命令でコード化される、コンピュータでアクセス可能な媒体と、

該データ構造および文書を受信するために連結されたハードウェア回路と

を備える装置であって、該ハードウェア回路は、該データ構造を使用して該スタイルシートにおいて指定されたように該文書の少なくとも一部を変換するように構成される、装置。

【請求項 2】

実行されると、前記複数の命令が、前記スタイルシート内の各ノード識別子にシリアル番号を割り当て、前記データ構造が、該シリアル番号に該ノード識別子をマップする 1 つ以上の記号表を含む、請求項 1 に記載の装置。

10

【請求項 3】

前記ハードウェア回路が、前記文書を解析するように構成された構文解析器を備え、該構文解析器が、該文書内の該ノード識別子を検出するように構成され、前記 1 つ以上の記号表を使用して、該ノード識別子を該シリアル番号に変換するように構成される、請求項 2 に記載の装置。

【請求項 4】

前記記号表に存在しない前記文書内での第 1 のノード識別子の検出に応じて、前記構文解析器が、第 1 のシリアル番号を該第 1 のノード識別子に割り当てるように構成される、請求項 3 に記載の装置。

20

【請求項 5】

前記構文解析器が、前記第 1 のノード識別子と、前記第 1 のシリアル番号とを前記構文解析が生成した第 2 の 1 つ以上の記号表に保存するように構成される、請求項 4 に記載の装置。

【請求項 6】

前記データ構造が、前記スタイルシート内の式を表す 1 つ以上の式ツリーを含み、前記ハードウェア回路が、該式ツリー内に表される該式を前記文書に適用して該式を満たす該文書内のノードを識別するように構成される、請求項 1 に記載の装置。

【請求項 7】

前記スタイルシートに指定されるような所与の式を満たす前記ノードに、1 つ以上のアクションを実行するように構成される変換エンジンをさらに含む、請求項 6 に記載の装置。

30

【請求項 8】

前記アクションを適用すると、少なくともいくつかの式の一部分が評価可能であり、前記ハードウェア回路が、該部分に基づいて前記式の残りを満たすノードをグループ化するように構成される、請求項 7 に記載の装置。

【請求項 9】

2 つ以上の式が共通部分を有する場合、該 2 つ以上の式が、該共通部分に対応する前記式ツリー内のノードを共有する、請求項 6 に記載の装置。

40

【請求項 10】

前記コンピュータでアクセス可能な媒体が、実行されるとスキーマを第 2 の 1 つ以上のデータ構造にコンパイルする第 2 の複数の命令をさらに含む、請求項 1 に記載の装置。

【請求項 11】

前記第 2 のデータ構造が、有効な文書構造を識別するスケルトン表を含む、請求項 10 に記載の装置。

【請求項 12】

前記第 2 のデータ構造が、デフォルト属性値を識別する 1 つ以上の表を含む、請求項 10 に記載の装置。

【請求項 13】

50

前記第2のデータ構造が、要素の所要の属性を識別する1つ以上の表を含む、請求項10に記載の装置。

【請求項14】

スタイルシートを1つ以上のデータ構造へコンパイルするステップと、
ハードウェア回路へと該データ構造および文書を受信するステップと、
該データ構造を使用して、該ハードウェア回路が該スタイルシート内に指定されたように該文書の変換の少なくとも一部を行うステップと
を包含する、方法。

【請求項15】

シリアル番号を前記スタイルシート内の各ノード識別子へ割り当てるステップをさらに包含し、前記データ構造が、該シリアル番号に該ノード識別子をマップする1つ以上の記号表を含む、請求項14に記載の方法。

10

【請求項16】

前記ハードウェア回路が前記文書を解析して該文書内の前記ノード識別子を検出するステップと、

該ハードウェア回路が前記1つ以上の記号表を使用して該ノード識別子を該シリアル番号に変換するステップとをさらに包含する、請求項15に記載の方法。

【請求項17】

前記ハードウェア回路が前記記号表にない前記文書の第1のノード識別子を検出するステップと、

20

該ハードウェア回路が第1のシリアル番号を該第1のノード識別子に割り当てるステップと、

該ハードウェア回路が該第1のノード識別子および前記第1のシリアル番号を、該ハードウェア回路が生成した第2の1つ以上の記号表に保存するステップと

をさらに包含する、請求項16に記載の方法。

【請求項18】

前記データ構造が、前記スタイルシート内の式を表す1つ以上の式ツリーを含み、前記ハードウェア回路が該式ツリー内に表される該式を該文書に適用して該式を満たす該文書内のノードを識別するステップをさらに包含する、請求項11に記載の方法。

【請求項19】

30

前記スタイルシートに指定されるような所与の式を満たす前記ノードへ、1つ以上のアクションを実行するステップをさらに包含する、請求項18に記載の方法。

【請求項20】

前記アクションを適用すると、少なくともいくつかの式の一部分が評価可能であり、前記ハードウェア回路が該部分に基づいて該式の残りを満たすノードをグループ化するステップをさらに包含する、請求項19に記載の方法。

【請求項21】

スキーマを第2の1つ以上のデータ構造へコンパイルするステップをさらに包含する、請求項14に記載の方法。

【請求項22】

40

前記第2のデータ構造が、有効な文書構造を識別するスケルトン表を含む、請求項21に記載の方法。

【請求項23】

前記第2のデータ構造が、デフォルト属性値を識別する1つ以上の表を含む、請求項22に記載の方法。

【請求項24】

前記第2のデータ構造が、要素の所要の属性を識別する1つ以上の表を含む、請求項22に記載の方法。

【発明の詳細な説明】

【技術分野】

50

【0001】

本発明は、拡張マークアップ言語（XML）、標準汎用マークアップ言語（SGML）、ハイパーテキストマークアップ言語（HTML）のような構造化文書、ならびにデータベースおよび/またはファイルシステム内の非構造化データおよび文書を処理および変換する分野に関連する。

【背景技術】

【0002】

コンピュータおよびコンピュータストレージが遍在化するにしたがい、多様な組織によって維持される情報量が急激に増加している。しばしば情報は、ワードプロセッサ文書、表計算ファイル、データベース、ポータブルドキュメントフォーマット（PDF）文書、画像文書（例、さまざまなグラフィック表示フォーマットにスキャンしたもの）、プレーンテキスト等の多くの異なる形式で保存される。さらに、文書はSGML、HTML、XML等のマークアップ言語の形で保存することが可能である。

10

【0003】

あまりにも多くの異なる形式で情報を所有することは、組織の外部のみならず、組織内の情報の共有を複雑にする。最近では、文書の内容の説明や非構造化データおよび/または文書へ構造を提供するために、XMLが標準として浮上してきている。XMLは、文書のマークアップを定義する、柔軟で拡張可能な機構を提供し、説明される情報に対するマークアップのカスタマイズを可能にする。

【0004】

XMLを処理するための手段として実施される1つの機構は、拡張スタイルシート言語（XSL）およびXSLを使用して作成されたスタイルシートである。スタイルシートは、XML文書をXML内で定義される1つのマークアップ定義（または「語彙」）から別の語彙へ、XMLマークアップから、別の構造化文書または非構造化文書形式（プレーンテキスト、ワードプロセッサ、表計算ソフト、データベース、pdf、HTML等）へ、または別の構造化文書あるいは非構造化文書形式からXMLマークアップへ、変換するように作成することが可能である。したがって、スタイルシートは、文書構造を保存した形式から所与のユーザーが希望する形式へ変換することによって、（多くの異なる形式の）組織の情報へのアクセスを簡素化することが可能である。また、他の種類のスタイルシート（例、拡張機能としてHTMLに定義されるカスケーディングスタイルシート、すなわち、CSS）も存在する。

20

30

【0005】

一般的に、文書変換処理は、汎用コンピュータ（例、文書保存、ユーザー機等を管理するサーバー）上で実行するソフトウェアにおいて行われる。そのような書類へのアクセスの著しい滞りが発生している。

【発明の開示】

【課題を解決するための手段】

【0006】

一実施形態において、実行されるとスタイルシートを1つ以上のデータ構造にコンパイルする複数の命令でコード化される、コンピュータアクセス可能媒体と、データ構造および文書を受信するように結合されたハードウェア回路とを備える装置を提供する。ハードウェア回路は、データ構造を使用してスタイルシートに指定されるように、文書の少なくとも一部を変換するように構成される。

40

【発明を実施するための最良の形態】

【0007】

以下の詳細な説明では、図面の簡単な説明において簡単に説明される添付の図面を参照する。

【0008】

本発明に対しては、さまざまな変更および代替形態が可能であるが、その具体的な実施形態を図面中に一例として示し、本願明細書において詳述するものである。しかし、図面

50

およびその詳細な説明は、この発明を開示される特定の形態に限定することを意図するのではなく、逆に、添付の請求請求の範囲により規定される本発明の範囲および精神の範囲内にある全ての変更、均等物および代替案をカバーすることを意図すると理解されたい。

【0009】

図1は、内容変換装置10の一実施形態のブロック図を示す。図1の実施形態において、内容変換装置10は、ネットワークインターフェース回路12と、プロセッサ14Aおよび選択的に14Bのような1つ以上のプロセッサと、文書プロセッサ16と、メモリ18とを含むことが可能である。ネットワークインターフェース回路12は、1つ以上のネットワーク接続を介して、1つまたは複数のネットワークへ結合される。さまざまなコンピュータシステム(図1には示さず)を、前記の1つまたは複数のネットワークに結合することも可能である。ネットワークインターフェース回路12は、プロセッサ14Aおよび14Bにも結合される。これらのプロセッサはメモリ18および文書プロセッサ16に結合され、文書プロセッサ16はメモリ18にも結合される。図示の実施形態において、メモリ18は、スタイルシートコンパイラ20、スキーマコンパイラ22、1つ以上の記号表24、1つ以上の解析時間の式ツリー26、命令表30、余白表32、文書型定義(DTD)表34、式リスト表36、テンプレートリスト表38、および各種の文書プロセッサデータ構造39を保存する。

10

【0010】

内容変換装置10は、ネットワーク接続を介して、文書に適用するスタイルシート、文書に適用するスキーマ、および/または(スタイルシート/スキーマの文書への適用の要求と共に)書類自体を受信することができる。スタイルシートを適用する要求に回答して、内容変換装置10は、スタイルシートを文書に適用して変換済み文書を生成し、それをネットワークを通じて要求元に発信することが可能である。いくつかの実施形態において、内容変換装置10は、要求を受信して、(例、XML(SAX)または文書オブジェクトモデル(DOM)用の単純なアプリケーションプログラミングインタフェース(API)のような定義されたフォーマットに)文書を解析することも可能である。スキーマ(またはDTD)の適用の要求に回答して、内容変換装置10は、スキーマまたはDTDに従って文書を認証し、要求元への成功メッセージまたは(失敗の表示を伴う)失敗メッセージを生成することが可能である。

20

【0011】

いくつかの実施形態において、内容変換装置10は、XPath式を受信し、XMLデータベースへのアクセスに使用することができる。そのような実施形態において、前記式は(下記に詳述する)スタイルシートと同様にコンパイルすることが可能であり、スタイルシートを文書に適用するのと同様な様式でXMLデータベースに適用することが可能である。

30

【0012】

概して、XML文書は階層ツリー構造を持ち、その構造において、そのツリーのルートは文書全体を識別し、文書内の他のノードのそれぞれはルートの子孫の関係になる。さまざまな要素、属性および文書内容がツリーのノードを形成する。前記要素は、その要素が含む内容の構造を定義する。各要素は要素名を持ち、その要素は、それぞれが要素名を含む開始タグおよび終了タグを使用して内容の範囲を定める。要素はサブ要素として他の要素を持つことが可能であり、それが内容の構造をさらに定義することが可能である。加えて、要素は属性を含むことが可能であり(要素名に続いて、開始タグに含まれる)、その属性は、要素または要素の内容の構造についてさらなる情報を提供する名前/値ペアである。XML文書は、XML文書、コメント等を読み込んでアプリケーションへ渡される処理命令を含むことも可能である。ここで使用されている「文書」という用語は、内容を解釈するために使用することのできる、対応する定義済みの構造を有するあらゆる内容を指す。前記内容は、(XML文書、HTML文書、pdf文書、ワード処理文書、データベース等のように)高度に構造化してもよいし、またはプレーンテキスト文書のような単純なものでもよい(その構造は、例えば一連の文字であってもよい)。概して、文書の「ノ

40

50

ード」は、構造定義（例、XMLの要素および/または属性）、および/または文書内容を含むことが可能である。ある特定の形態において、ノードは、要素、属性、処理命令、コメントおよびテキストを含むことが可能である。

【0013】

XSLTスタイルシートは、テンプレートのセットと考えることが可能である。各テンプレートには：(i)ソース文書のツリー構造のノードを選択する式と、(ii)ソース文書の各一致するノードに対してインスタンス化される、出力文書の構造の対応する部分を特定する本体とを含むことが可能である。スタイルシートのソース文書への適用は、ソース文書の各ノードに一致するテンプレートを探す試み、および、出力文書のツリーにおいて、一致するテンプレートの本体をインスタンス化すること、を含む場合がある。テンプレートの本体には、1つ以上の：(i)出力文書においてインスタンス化されるリテラル内容と、(ii)出力文書にコピーされる一致するノードからの内容の選択と、および(iii)出力文書においてインスタンス化されるステートメントの結果と共に、評価されるステートメントと、を含むことが可能である。同時に、インスタンス化される内容および評価されるステートメントは、テンプレートと一致するノード上で行われる「アクション」と称することが可能である。テンプレートの本体には1つ以上の「テンプレート適用」ステートメントを含むことができ、それらは、1つ以上のノードを選択してスタイルシートのテンプレートを選択したノードに適用させる式を含む。よって、テンプレートは効果的にネストされる。テンプレート適用ステートメントへの一致が見つかった場合、その結果のテンプレートは、テンプレート適用ステートメントを含むテンプレートのインスタンス化の範囲内でインスタンス化される。テンプレートの本体の他のステートメントは、ノードに対して一致する式も含むことが可能である（そして、ステートメントは、一致するノード上で評価することが可能である）。XSLTスタイルシートは、本願明細書において一実施例として用いることが可能であるが、一般的に、「スタイルシート」は、ソース文書を変換するためのあらゆる仕様を含むことが可能である。ソース文書および出力文書は同じ言語であってもよいし（例、ソース文書および出力文書は異なるXML語彙であってもよい）、または異なって（例、XMLからpdf等）いてもよい。スタイルシートの別の例は、HTMLおよび/またはXML Queryのために定義されたカスケーディングスタイルシートであってもよい。

10

20

【0014】

スタイルシートで使用される式は、概して、ノード識別子および/またはノード値の間の親/子（または祖先/子孫）関係を指定するノード識別子上の演算子と共に、ノード識別子および/またはノード値を含むことが可能である。ノード識別子は、名前（例、要素名、属性名等）を含んでもよいし、またはノードの種類によって識別する式構造を含んでもよい（例、ノードテスト式が任意のノードと一致することが可能であるか、またはテキストテスト式が任意のテキストノードと一致することが可能である）。場合によっては、名前は特定の名前空間に属することができる。そのような場合、ノード識別子は名前空間に関連する名前でもよい。XMLでは、名前空間は、ユニバーサルリソース識別子（URI）によって識別される名前空間名と関連付けることによって、要素名および属性名を修飾する方法を提供する。したがって、ノード識別子は修飾名であってもよい（任意の名前空間接頭辞、その後にはコロン、その後にはその名前が続く）。本願明細書において使用される場合の名前（例、要素名、属性名等）は、修飾名を含むことが可能である。式はまた述語を含むことが可能であり、それは、ノードを一致させるための1つまたは複数の追加の条件である。述語は、コンテキストノード（以下に定義）として関連ノードによって評価される式であり、その式の結果は真（ノードが式ノードに一致する）または偽（および、ノードが式ノードと一致しない）のいずれかである。したがって、式は文書ツリーに対して一致するノードツリーとして考えることができる。XSLTで使用される式言語であるXPathでは、式は、「コンテキストノード」のコンテキストに対しても評価されることが可能である（すなわち、式はコンテキストノードに相対的であることが可能で、他のノード識別子との関係だけでなく、式の中のノード識別子を、コンテキストノードの祖先

30

40

50

、子孫、親、子として特定する)。所与の文書ノードは、その所与の文書ノードが式の評価を介して選択されれば、式を満たすことが可能である。すなわち、式の中の式ノード識別子が、所与の文書ノード名と、または、その式に指定された文書ノードとの関係と同一の所与の文書ノードとの関係を有する文書ノード名と、一致し、その式の中で使用される任意の値は、その所与の文書名に関連する、対応する値に等しい。文書ノードは、そのノードが所与の式を満たす場合、所与の式のための「一致するノード」とも呼ぶことが可能である。場合によっては、本説明の残部において、式ツリー内のノードと文書内のノードとの区別を明確にし易くすることが可能である。したがって、ノードが式ツリーの一部である場合、そのノードを「式ノード」と称することが可能であり、ノードが処理中の文書の一部である場合、そのノードを「文書ノード」と称することが可能である。

10

【0015】

図示した実施形態において、スタイルシートの文書への適用は、以下の様式において行うことができる。スタイルシートコンパイラ20は、プロセッサ14Aおよび14Bのうちの1つで実行するソフトウェア(すなわち、複数の命令)を含むことが可能であり、文書プロセッサ16で使用するために、スタイルシートを1つ以上のデータ構造およびコードにコンパイルする。文書プロセッサ16はそのデータ構造をソース文書に適用し、出力文書を生成することが可能である。

【0016】

特に、一実施形態では、スタイルシートコンパイラ20は、(文字列比較を伴うことになる)ノード識別子よりもむしろ数字を比較することで文書プロセッサが式評価を行えるように、シリアル番号をノード識別子に割り当てることが可能である。スタイルシートコンパイラ20は、シリアル番号に対するノード識別子のマッピングを記号表24に保存することが可能である。加えて、スタイルシートコンパイラ20は、スタイルシートから式を抽出し、文書プロセッサが式を一致させるために使用する式ツリーのデータ構造を生成することが可能である(例、解析時間の式ツリー26)。さらに、スタイルシートコンパイラ20は、一致する式のそれぞれに実行する命令(また一実施形態において、ランタイム述語の評価を実行する命令)の命令表30を生成することが可能である。その命令表内の命令は、文書プロセッサ16によって実行されると、式が一致した場合に実行するように定義されているアクションを実行させることが可能である。いくつかの実施形態において、命令は実行するアクションを含むことが可能である(つまり、命令とアクションの間は一対一対応とすることが可能である)。他の実施形態において、少なくともいくつかのアクションは、2つ以上の命令を実行することによって実現することができる。スタイルシートコンパイラ20は、ソース文書内でどのように種類の余白を取り扱うのか(例、保存した、取り除いた、等)を定義する余白表32、式リスト表36、およびテンプレートリスト表38を生成することも可能である。

20

30

【0017】

スキーマコンパイラ22も同様に、プロセッサ14Aおよび14Bのうちの1つで実行される命令を含むことが可能である。スキーマコンパイラ22は、DTD表34とともに、1つ以上の記号表24(ノード識別子をシリアル番号に置き換える)を生成するためのスキーマまたはDTDをコンパイルすることができる。概して、DTDまたはスキーマは、許容される文書構造および要求される文書構造の両方の定義を含むことが可能である。したがって、文書の著者は、DTDおよび/またはスキーマを使用して、有効な文書の要求されるおよび許容される構造を記述することが可能である。場合によっては、そのDTDまたはスキーマが属性のデフォルト値も含んでもよい。一実施形態において、DTD/スキーマは、文書内の実体参照を置き換えるために使用される実体宣言、有効な文書に必要な属性である所与の要素の属性、その文書の所与の要素で特定されない場合がある属性の属性デフォルト値、文書の構造の要件(例、特定のサブ要素等の必要最大/最小/特定数)、およびその文書の許容される構造の定義、などのさまざまな情報を含むことが可能である。DTD表34は、実体参照の置換表、必要とされる属性の表、属性デフォルト表、許容される構造を識別するスケルトンツリー(および、該当する場合は、要求される構

40

50

造)を含むことが可能である。

【0018】

文書プロセッサ16は文書を解析し、解析時間の式ツリー内の式ノードに文書ノードを一致させるハードウェア回路構成を含むことが可能である。すなわち、文書を解析し、式ノードに文書ノードを一致させるハードウェア回路構成は、これらの演算をいずれのソフトウェアの命令を実行せずに行うことが可能である。そのハードウェアは、一致する文書ノードの各式に対する解析した内容および表示を保存しているさまざまなデータ構造を生成することが可能である。そのハードウェアは、次いで、命令表30からの命令を、所与の式の各一致する文書ノード上の所与の式に実行して結果を生成し、それらを組み合わせ出力文書を作成することが可能である。一実施形態の更なる詳細を以下に示す。

10

【0019】

上述のように、図示した実施形態において、スタイルシートコンパイラ20およびスキーマコンパイラ22はソフトウェアにおいて実施されており、文書プロセッサ16は、ハードウェアにおいて実施されている。いくつかの実施形態において、内容変換装置10の遂行の極めて重要な要因は、変換要求が起こされ、文書が提供される時の文書の処理であり得る。すなわち、スタイルシートおよび/またはスキーマは、多くの場合、処理される文書の数に比べて、比較的まれにしか変更しない場合がある。所与のスタイルシートは、スタイルシートが(更新したスタイルシートまたは全く異なるスタイルシートへ)変更される前に、複数の文書(例、少なくとも約数十の文書)に適用することが可能である。スキーマおよびそれらが適用される文書も同様な関係を保つことが可能である。したがって、比較的不変である情報をスタイルシート/スキーマから(ソフトウェアを使用して)、専用のカスタムハードウェアが効率よくアクセスできるデータ構造に取り込むことによって、高性能なソリューションを提供することが可能である。加えて、ハードウェアがスタイルシート/スキーマのコンパイルを有することにより、いくつかの実施形態において、異なるスタイルシート/スキーマ言語の実施に対する柔軟性、および/またはカスタムハードウェアを変更する必要もなく、言語仕様に変更を施すための柔軟性を提供することが可能である。例えば、XSLT、XPath、およびXMLスキーマはさらに発展させることが可能であり、これらの言語には将来新しい機能が追加される可能性がある。そのコンパイラは、これらの新しい機能に対処するように構成されることが可能である。使用されるスタイルシート/スキーマを事前に提供することが可能であり、したがって、スタイルシート/スキーマをコンパイルする時間はそれほど重要でなくなる可能性がある。しかし、他の実施形態では、スタイルシートコンパイラ20およびスキーマコンパイラ22の一方またはそれらの両方を、ハードウェアにおいて、またはハードウェアとソフトウェアを組み合わせたものにおいて、実施することが可能である。

20

30

【0020】

ネットワークインターフェース回路12は、ネットワーク接続上の低レベルの電氣的詳細およびプロトコル詳細に対応し、受信したパケットを処理するためにプロセッサ14Aおよび14Bに渡すことが可能である。あらゆる種類のネットワークを使用することが可能である。例えば、いくつかの実施形態において、前記ネットワーク接続は、Gigabit Ethernet(登録商標)接続であってもよい。必要に応じて、複数の接続を提供して、所与のレベルの帯域幅を達成すること、および/またはネットワーク接続に冗長性を提供することができる。

40

【0021】

プロセッサ14Aおよび14Bは、あらゆる種類のプロセッサを含むことが可能である。例えば、一実施形態において、プロセッサ14Aおよび14Bは、PowerPCネットワークプロセッサであってもよい。他の実施形態において、プロセッサ14Aおよび14Bは、ARM、Intel社のIA-32、MIPS等の、他の命令セットアーキテクチャを実施することが可能である。

【0022】

プロセッサ14Aおよび14B、文書プロセッサ16、およびメモリ18への結合には

50

、あらゆる相互接続を使用することが可能である。さらに、プロセッサ 14 A および 14 B は、メモリ 18 へのプロセッサ 14 A および 14 B ならびに文書プロセッサ 16 の接続とは別に、文書プロセッサ 16 に結合することが可能である。例えば、一つの実施形態において、プロセッサ 14 A および 14 B は、1 つ以上の周辺構成要素相互接続 (P C I - X) バスを使用して、文書プロセッサ 16 に結合することが可能である。

【 0 0 2 3 】

場合によっては、D T D、スキーマ、またはスタイルシートを (直接、あるいはスキーマまたはスタイルシートに対するポイントとして) 文書に埋め込むことが可能であることに留意されたい。そのような場合、D T D、スキーマ、またはスタイルシートを文書から取り出して、説明したように、別途提供されたスキーマまたはスタイルシートとして処理する。

10

【 0 0 2 4 】

メモリ 18 は、あらゆる種類の揮発性または不揮発性メモリで構成することが可能である。例えば、メモリ 18 は、1 つ以上の R A M (例、S D R A M、R D R A M、S R A M 等)、バッテリバックアップ R A M またはフラッシュメモリのような不揮発性メモリ、ディスクまたは C D - R O M 等の磁気または光学式記憶装置を含むことが可能である。メモリ 18 は、別々にアクセスできる複数のメモリ (例、プロセッサ 14 A および 14 B へのみアクセスできる 1 つまたは複数のパーティションおよび文書プロセッサ 16 へのみアクセスできる 1 つまたは複数の別のパーティション) を含むことが可能である。

【 0 0 2 5 】

図 1 は、メモリ 18 に保存されているスタイルシートコンパイラ 20 およびスキーマコンパイラ 22 を示す。概して、スタイルシートコンパイラ 20 および / またはスキーマコンパイラ 22 は、あらゆるコンピュータでアクセス可能な媒体にコード化することが可能である。一般的に言えば、コンピュータでアクセス可能な媒体は、コンピュータに命令および / またはデータを提供するために使用する間に、コンピュータがアクセスできる任意の媒体を含むことが可能である。例えば、コンピュータでアクセス可能な媒体には、ネットワークおよび / または無線リンクのような通信媒体を介して伝達される、伝送媒体、または電氣的、電磁的、またはデジタル信号のような信号を介してアクセスできる媒体、に加えて、例えばディスク (固定またはリムーバブル)、C D - R O M または D V D - R O M 等の磁気または光学的媒体のような記憶媒体、R A M (例、S D R A M、R D R A M、S R A M 等)、R O M、フラッシュメモリ等の揮発性または不揮発性メモリ媒体等の記憶媒体、が挙げられる。

20

30

【 0 0 2 6 】

いくつかの実施形態では、コンピュータでアクセス可能な媒体を、コンパイルを行うためにスタイルシートコンパイラ 20 および / またはスキーマコンパイラ 22 を実行することが可能な独立した 1 つまたは複数のコンピュータシステムに含めることが可能である。そのコンパイルから生じるデータ構造 / コードを、(例えば、内容変換装置 10 へのネットワーク接続を介して) 内容変換装置 10 へ通信することが可能である。

【 0 0 2 7 】

本願明細書における説明では、1 つのスタイルシートを 1 つの文書に適用する例を挙げているが、他の例では、複数のスタイルシートの 1 つの文書に適用する例 (必要に応じて、同時に、または連続的に)、および 1 つのスタイルシートを複数の文書に適用する例 (必要に応じて、コンテキスト切り替えと同時に、または連続的に) が挙げられることに留意されたい。

40

【 0 0 2 8 】

次に、図 2 は、文書プロセッサ 16 の一実施形態のブロック構成図を示す。図 2 の実施形態において、文書プロセッサ 16 は、構文解析回路 40 と、式プロセッサ 42 と、変換エンジン 44 と、出力生成器 46 と、バリデータ回路 48 とを含む。構文解析回路 40 は、式プロセッサ 42 および出力生成器 46 に結合される。式プロセッサ 42 は、変換エンジン 44 に結合され、それが出力生成器 46 に結合される。バリデータ 48 は、出力生成

50

器 46 に結合される。図 2 のユニットは、直接互いに結合するか（例、ユニット間の信号線を使用して）、メモリ 18 を介して結合するか（例、ソースユニットは、送信先ユニットと通信する情報をメモリ 18 に書き込み、送信先ユニットはメモリ 18 からその情報を読み込むことができる）、またはその両方によって結合することが可能である。

【 0 0 2 9 】

構文解析回路 40 は、文書を受信してその文書を解析することが可能であり、式プロセッサ 42 およびバリデータ回路 48 に対するイベントを識別し、その解析した内容でデータ構造を生成もする。文書プロセッサ 16 がスタイルシートに従って文書を変換する場合、解析された内容を変換エンジン 44 に対してメモリ 18 のデータ構造に保存することができる。あるいは、その文書が解析されるだけであれば、構文解析回路 40 は、出力生成器 46 に S A X または D O M フォーマットで出力される解析した内容を提供することが可能である。構文解析回路 40 は、メモリ 18 を介して、出力生成器 46 に解析した内容を提供することも可能である。

10

【 0 0 3 0 】

式プロセッサ 42 は、構文解析回路 40 からイベントを受信し（文書から解析した文書ノードを識別して）、構文解析回路 40 が識別する文書ノードを解析時間の式ツリーと比較する。式プロセッサ 42 は、各式の一致する文書ノードのリストを変換エンジン 44 に出力する。変換エンジン 44 は、構文解析回路 40 によって構築された、解析した内容のデータ構造および一致する文書ノードのリストを受信し、命令表 30 の対応する命令を実行して出力文書のための結果を生成する。いくつかの実施形態において、それぞれの命令はその他のものから独立しており、したがって、どの順序でも実行され得る。出力生成器 46 はその結果を順番に再構築し、その出力文書をメモリ 18 に書き込むことが可能である（または、メモリ 18 を介さずに、出力文書をプロセッサ 14 A および 14 B に送信することが可能である）。プロセッサ 14 A および 14 B は、ソフトウェアを実行して出力文書を読み込み、その出力文書を要求元に発信することが可能である。

20

【 0 0 3 1 】

またバリデータ回路 48 は、構文解析回路 40 が送信したイベントを受信することも可能であり、（スケルトンツリーおよび D T D 表 34 で表されるように）スキーマ / D T D を適用し、その文書がスキーマに示されたように有効であるかどうかを判断することが可能である。文書が有効な場合は、バリデータ回路 48 は成功メッセージを生成して出力生成器 46 に発信することが可能である。文書が有効でない場合は、バリデータ回路 48 は（1つまたは複数の失敗の理由を表示する）失敗メッセージを生成し、その失敗メッセージを出力生成器 46 に発信することが可能である。出力生成器 46 は、そのメッセージをメモリ 18 に保存することが可能である（続いてプロセッサ 14 A および 14 B は、そのメッセージを要求元に発信することが可能である）。

30

【 0 0 3 2 】

図 3 は、文書プロセッサ 16 の一部分（具体的には、構文解析回路 40、式プロセッサ 42、および変換エンジン 44）、およびプロセッサ 14 A を示す。図 3 では、内容変換装置 10 の一実施形態に基づいて示した部分の間の通信をより詳細に強調している。また、プロセッサ 14 B は、プロセッサ 14 A に対して説明した様式で演算することが可能である。

40

【 0 0 3 3 】

プロセッサ 14 A は、内容変換装置 10 が連結されている 1つまたは複数のネットワークからパケットを受信することが可能である。データペイロードのパケットは、内容変換装置 10 によって変換される文書を含むことが可能である。加えて、受信した他のパケットは、他の通信（例、スタイルシートまたはスキーマ、あるいは他の内容変換装置 10 との通信）を含むことが可能である。プロセッサ 14 A は、文書を再構築し、その再構築した文書を構文解析回路 40 に渡すことが可能である。

【 0 0 3 4 】

構文解析回路 40 は、プロセッサ 14 A から再構築した文書を受信し、さらにメモリ 1

50

8 から記号表 2 4、D T D 表 3 4、および余白表 3 2 にアクセスする。構文解析回路 4 0 は、文書を解析し、発見した文書ノードに関連するイベントを生成する。より詳しくは、構文解析回路 4 0 は、文書内のノード識別子を記号表 2 4 の対応するシリアル番号に転換し、そのシリアル番号をイベントの一部として式プロセッサ 4 2 へ発信する。加えて、構文解析回路 4 0 は、変換エンジン 4 4 に対する文書の解析した内容を保存する解析した内容の表を生成する。式プロセッサ 4 2 は構文解析器 4 0 からイベントを受信し、(それらのシリアル番号に基づいて) 識別した文書ノードを解析時間の式ツリー 2 6 と比較する。一致する文書ノードが識別され、変換エンジン 4 4 に送信されるテンプレートおよび式一致リストに記録される。

【0035】

変換エンジン 4 4 は、テンプレートおよび式一致リスト、ならびに解析した内容表を受信し、さらに命令表 3 0 受信する。変換エンジン 4 4 は任意のランタイム式を評価し、テンプレートおよび式一致リストからランタイム式を満たさない文書ノードを削除する。加えて、変換エンジン 4 4 は、その式と一致する各文書ノードの各式に対する命令表 3 0 からの命令を実行し、結果を出力生成器 4 6 に出力する。

【0036】

図示の実施形態において、プロセッサ 1 4 A は、再構築した文書をインラインで発信することが可能であり、構文解析回路もイベントを式プロセッサ 4 2 へインラインで発信することが可能である。すなわち、文書の一部がプロセッサ 1 4 A によって受信されて再構築されると、プロセッサ 1 4 A はその文書のその部分を構文解析回路 4 0 へ渡す。構文解析回路 4 0 は、したがって、プロセッサ 1 4 A が文書全体を受信する前に解析を始めることができる。同様に、イベントが識別されると、それらのイベントは式プロセッサ 4 2 に渡される。一方、解析した内容表およびテンプレート/式一致リストは、メモリ 1 8 を介して渡される(変換エンジン 4 4 への通信上に破線の楕円で示す)。本願明細書において使用されるように、データを直接渡す場合、そのデータは「インライン」でソースから受信者に発信されるが、メモリ 1 8 等のメモリでバッファリングされる場合にはそうではない(しかし、ソースまたは受信者は転送のためにデータを一時的にキューに入れることが可能である)。インラインで発信したデータは、メモリを介した伝送よりも待ち時間が少なくなる場合がある。

【0037】

図 4 は、文書を変換するための方法の一実施形態のフローチャートを示す。概して、該方法は、文書変換が複数の段階を含む際に適用することが可能である。スタイルシート内の式は、それらを実行できる最も初期の段階に基づいて分類することが可能である。次いで、各段階の間に、その段階で評価できる式が評価される。したがって、各式を最も初期の可能な段階で評価でき、その後の段階で評価すべき式がほとんど残らない。

【0038】

図示の実施形態において、段階には、コンパイル段階、分解段階、および変換段階が含まれ得る。コンパイル段階において、スタイルシート内の式が(例えば、本実施形態では、コンパイル時間、解析時間、またはランタイムのいずれかとして)特徴づけられる(ブロック 5 0)。加えて、コンパイル段階において、コンパイル時間式が評価される(ブロック 5 2)。解析段階の間に解析時間式が評価される(ブロック 5 4)。変換段階の間にランタイム式が評価される(ブロック 5 6)。

【0039】

いくつかの実施形態において、ランタイム式は、初期に評価可能である部分(例えば解析時間)、およびランタイムの部分に分類することが可能である。初期に評価可能である部分を、評価して、ランタイムの部分に基づいて分類することが可能である。すなわち、式の解析時間部分と一致し、式のランタイムの部分に使用される 1 つまたは複数の同じ値を有する文書ノードを分類する。ランタイム時に、式のランタイムの部分の評価され、グループに対応する値がその式のランタイムの部分を満たしていなければ、その式は削除される。式のランタイムの部分を満たすグループだけが保持され、保持したグループ内の文

10

20

30

40

50

書ノードに命令を実行する。

【0040】

XSLTスタイルシートを実施する一実施形態では、式が祖先ノード参照（//）および述語を含んでいなければ、その式はコンパイル時間となり得る。式が現在のノード、次の兄弟ノードまたは要素値を参照する述語を含んでいなければ、その式は解析時間となり得る。コンパイル時間または解析時間でない式は、ランタイム式である（例、現在のノードを参照する式、または次の兄弟または要素値を参照する述語を含む式）。現行ノードを参照しないランタイム式に関して、上述の述語を含まない部分は、解析時間で評価することが可能である。このコンテキストでは、現在のノードは、コンテキストノードとし、その式がテンプレートに一致した式である場合には、テンプレート本体内のステートメント内で参照されるノード（例、テンプレート本体内のループ構文または他の式）とすることが可能である。

10

【0041】

どの式が解析時間対ランタイムであるかは、一つには、式プロセッサ42のインライン性質に影響を受ける場合がある。すなわち、文書ノードは識別され、式プロセッサ42にインラインで渡される。対照的に、式の処理がインラインでない場合、過去および今後の文書ノードを、特定のノードの処理中に検索する可能性がある。したがって、式の処理がインラインでない場合、現行ノードを参照する式に限り、処理できない場合がある。インライン処理に関して、式ツリーに一致する文書ノードは、概して、式ツリー内の式ノードと以前の文書ノードとの一致に関する情報の保持を含むことが可能である。次いで、以前の文書ノードの子または子孫が構文解析回路40によって識別される場合、当該の子ノード文書ノードは、式ツリー内の以前に一致した式ノードにリンクしたその式ツリー内の次のレベルと比較することが可能である。

20

【0042】

コンパイル時間の式の評価は、「テンプレート適用選択」ステートメント内の式に適用することが可能である。上述のように、「テンプレート適用」ステートメントは、テンプレート本体のコンテキスト内のノードのセットを選択し、テンプレートをそれらのノードに適用する。「テンプレート適用選択」ステートメントは、ノードのセットを選択する式を含む。そのノードのセットは、次いでスタイルシートのテンプレートに適用される。「テンプレート適用選択」ステートメント内の式が上述の与えられたコンパイル時間の定義を満たせば、次いでコンパイラは、そのセット内のノードのテンプレート（もしあれば）のどれが一致するのかを判断する。したがって、テンプレートの一致は、当該の場合に削除することが可能である。一実施形態において、コンパイラは、テンプレート適用選択ステートメント内の式とテンプレート一致条件を含む式との代数的な一致を行うことが可能である。解析時間の式およびランタイムの式の評価は、どのノードが式を満たすかの判断を含むことが可能である。

30

【0043】

一実施形態において、XMLスキーマによって記述されるXML文書のコンテキスト内のXPath式の代数的な一致は、下記のように行うことが可能である。以下の定義は、XPath式およびXMLスキーマの代数的な一致アルゴリズムの説明に有用となり得る。PをXPath式とし、SをXMLスキーマとすると、Pに生じる各要素および属性名がSにおいて宣言され、Sにおいて同じ型（要素または属性）を有する場合に限り、Pを定義することが可能である。PおよびQを2つのXPath式とすると、入力文書Dの各ノードがPと同様にQも満たすSに基づく場合に限り、PをSに対する一致Qと称することが可能である。「単純な」式は、「//」演算子、および、「.」演算子を除外した式とすることが可能である。これらの定義を前提として、一実施形態の代数的な一致を、以下のように式Pおよび一連の1つ以上の単純な式Eに対して実行することが可能である。第一に、式を正規化することが可能である。「/」で始まる式は、正規形である。式Qが、「/」で始まっていなければ、次のように正規化される：(i) Qがいずれかのループ内にあれば、各ループ（各選択式およびQを/演算子で区切る）に対する選択式を有し、

40

50

Qに最も近く、先頭に付加した式の始めとして最外ループへ進む最内ループ選択式を有するQを先頭に付加する。(ii)Qが生じるテンプレートのテンプレート一致条件を有する、(i)から形成される式を先頭に付加する。および(iii)テンプレート一致条件が「/」でない場合、(i)および「//」を有する、(ii)から形成される式を先頭に付加する。(上述のように正規化される)単純な式Eの場合の式ツリーは、述語を無視することを除いては、解析時間の式ツリー26と同様に形成される。(述語に関する例外を以って)同一である式は、したがって、式ツリー内の同じパスにマップさせることが可能であり、その式ツリー内の同じリーフノードと関連する。Pは、その式ツリーと一致する(すなわち、Pの各ノード識別子または演算子は同じ位置でその式ツリーと一致する)。リーフノードが、Pが空になるのと同時に式ツリーに達する場合、そのリーフノードと関連する式は、Pと一致するE内の式である。これらの一致する式は、削除が可能なコンパイル時間の式とすることが可能である。

【0044】

図5は、スタイルシートコンパイラ20のフローチャートを示す一実施形態である。スタイルシートコンパイラ20がソフトウェアに実施される実施形態において、スタイルシートコンパイラ20は、実行すると図5に示す機能を実施する命令を含む。図5のブロックはスタイルシートコンパイラ20の機能を示すが、フローチャートはコンパイラがリスト順にその機能を実行することを示唆するものではなく、また総じて次の機能を開始する前にスタイルシート上の1つの機能を完了させるものではないことに留意されたい。

【0045】

スタイルシートコンパイラ20は、スタイルシート内の式を識別し(ブロック60)、その式をコンパイル時間、解析時間、またはランタイムに分類する。その式は、テンプレート一致ステートメント内、テンプレート適用ステートメント内、およびスタイルシート内のさまざまな他のステートメント内に存在することが可能である。スタイルシートコンパイラ20は、各式の標準形を生成する(ブロック62)。一般的に、異なる方法が論理的に等価であっても、所定の式を表すには多くの異なる方法がある。標準形は、等価式(または、等価である式の部分)を識別する処理を簡素化するために、所定の式を表す特定の方法を指定する。スタイルシートコンパイラ20は、式のノード識別子にシリアル番号を割り当てる(ブロック64)。

【0046】

スタイルシートコンパイラ20は、解析時間の式ツリーを構築するために、その式に共通のプレフィックスコンパクションを行うことが可能である(ブロック66)。上述のように、式は、概して文書内のノード識別子に一致するノード識別子の階層的リストを含むことが可能である。したがって、種々の式が、共通する部分を有することが可能である(特に、階層のより高い(すなわち、ルートにより近い)式ノードは、種々の式に対して同一となり得る)。共通する部分は、したがって、式の第1の部分(その式の「プレフィックス」)に存在することが可能である。当該の式を解析時間の式ツリー内にも併せて圧縮することによって、共通の部分を解析時間の式ツリー内に一度に表すことが可能であり、式が異なり始めれば、残りの式はその共通の部分の子となる。したがって、文書ノードを一致させることが可能な複数の式を並行して評価することが可能であり、解析時間の式ツリーの共通部分において、差異が生じたときにツリー内を分岐することが可能である。解析時間の式ツリーは、より小さくすることが可能であり、そのような場合には、より迅速に処理することが可能である。

【0047】

例えば、2つの式(シリアル番号を割り当てた後)は、/10/15/20/25、および、/10/15/20/30/35となり得る。これらの2つの式は、共通して/10/15/20/を有する。したがって、これらの2つの式は、子としてノード15を有するノード10と、子としてノード20を有するノード15とを含む共通の部分として解析時間の式ツリー内に表すことが可能である。ノード20は、2つの子(25および30)を有することが可能である。ノード30は、子としてノード35を有することが可能で

ある。文書ノードが解析され、式プロセッサ 4 2 に渡されるとき、ノード 2 0 が達するまで、式を文書ノードに対して並行して評価することが可能である。次の子文書ノードは、次いで式ノード 2 5 または 3 0 の 1 つと一致し、またはどちらにも一致し得ない。

【 0 0 4 8 】

上述のように、スタイルシートコンパイラ 2 0 は、解析時間で評価される部分およびランタイムで評価される部分にランタイム式を分類することが可能である。解析時間の部分は、解析時間の式ツリー内に含まれることが可能である。ランタイム述語が生ずる解析時間の式ツリー内の各レベルにおいて、スタイルシートコンパイラ 2 0 は、述語をランタイムで評価することができ、一致する文書ノードがランタイム述語を満たすかどうかによってそれらを保持または破棄することが可能となるように、一致するノードが分類される（および、ランタイム述語が使用する情報を保持することが可能である）ことを示すことが可能である。所定のランタイム式は、式内の種々のレベルにおいて 1 より多いランタイム述語を含むことが可能であるので、複数のレベルの分類が生ずる可能性がある。各分類レベルでは、ランタイム述語に対応する同じ値を有する文書ノードを一緒に分類する。ランタイム述語を評価されるときに、その値がランタイム述語と一致しない場合、文書ノードのそのグループ（および、そのグループの全てのサブグループ）は破棄される。その値が評価したランタイム述語と一致するグループは保持され、変換エンジン 4 4 によって処理される。

10

【 0 0 4 9 】

スタイルシートコンパイラ 2 0 は、構文解析回路 4 0 および / または式プロセッサ 4 2 が使用する複数のデータ構造をメモリ 1 8 に出力することが可能である（ブロック 6 8）。ノード識別子をシリアル番号にマップする 1 つ以上の記号表 2 4 に加えて、解析時間の式ツリーが出力され得る。例えば、一実施形態においては、独立した記号表を要素名および属性名に対して出力されることが可能である。他の実施形態においては、単一の記号表を出力されることが可能である。加えて、スタイルシートコンパイラ 2 0 は、各テンプレートに対する一連の命令を有する命令表 3 0 を出力することが可能であり、それを変換エンジン 4 4 が実行してテンプレート本体がもたらされる。スタイルシート 2 0 は、変換エンジン内で実行された場合、ランタイム述語を評価する各ランタイム述語に対する命令をさらに出力することが可能である。さらに、スタイルシートコンパイラ 2 0 は、後述するようにテンプレートリスト表 3 8 および式リスト表 3 6 を出力することが可能である。

20

30

【 0 0 5 0 】

いくつかの実施形態において、所定のスタイルシートは、1 つ以上の他のスタイルシートを含むこと、および / または 1 つ以上の他のスタイルシートをインポートすることを含み得る。含まれたスタイルシートは、その含まれたスタイルシートが含めるスタイルシートに物理的に移動されるかのように処理される（例、C 言語における `#include` ステートメントの処理と同様）。すなわち、含まれたスタイルシートの本体は、本質的に `include` ステートメントを含めるスタイルシート内で置き換えることが可能である。含まれたスタイルシートと含めるスタイルシート（例、大域的な変数宣言）との間に競合がある場合、含めるスタイルシート内の定義が使用される。XSLT スタイルシートにおいて、含まれたスタイルシートは、含めるスタイルシート内の `xsl:include` 要素に記述することが可能である。一方で、インポートしたスタイルシートは、含めるスタイルシート内のステートメントが明示的に参照されることが可能な独立したスタイルシートとして処理される。明示的な参照は、例えば、テンプレート一致ステートメント内で生じ得る（インポートしたスタイルシートからのテンプレートを使用する場合）。明示的な参照が、インポートするスタイルシートの他の場所である場合、インポートするスタイルシート内の一致するテンプレートは、インポートしたスタイルシート内の一致するテンプレートに優先する。複数のスタイルシートおよび 1 つの一致するテンプレートが、1 つより多いインポートしたスタイルシートに生じる場合、インポートするスタイルシート内にインポートしたスタイルシートをリストする順序は、どの一致するテンプレートが選択されるのかを制御する（例、一致を有する最初にリストされ、インポートしたスタイルシ

40

50

トを使用する)。XSLTスタイルシートにおいて、インポートしたスタイルシートは、`xsl:import`要素内に記述することが可能である。

【0051】

「メイン」(インポートする、または含める)スタイルシートおよび各インポートしたまたは含まれたスタイルシートは、スタイルシートコンパイラ20によって独立してコンパイルすることが可能である。含まれたスタイルシートの場合、メインスタイルシートに対するデータ構造および含まれたスタイルシートは、式プロセッサ42が使用する一連のデータ構造にマージすることができる。インポートしたスタイルシートの場合、データ構造は独立して残すことが可能であり、式プロセッサ42は並行してデータ構造を文書に適用することが可能である。このような実施形態は、一致する式における競合が、上述の競合解消を実施する競合解消ロジックによって処理することが可能であることを除いて、式プロセッサ42の実施形態と同様とすることが可能である。

10

【0052】

いくつかの実施態様において、スタイルシートは、スタイルシートを適用する文書以外の1つ以上の文書を参照するステートメントを含むことが可能である。そのスタイルシートは、参照される文書を処理するために更なるステートメントを含むことが可能である。一実施においては、スタイルシートコンパイラ20は、無条件でどの参照される文書を使用したのかを識別することが可能である(すなわち、スタイルシートをいずれかの文書に適用する各場合において、その参照される文書を使用することが可能である)。内容変換装置10は、入力文書の処理が始まるときに、無条件で参照される文書を取り込むことが可能であり、その無条件で参照される文書は、スタイルシートで参照される順序で解析することが可能である。変換エンジン44が参照される文書を使用する命令を実行するためのものであり、参照される文書の解析がまだ始まっていない場合、変換エンジン44は、別のタスクにコンテキストを切り替えることが可能である。条件つきで参照される文書の解析がまだ始まっていない場合、変換エンジン44は、別のタスクにコンテキストを切り替えることが可能である。条件つきで参照される文書に対して、内容変換装置10は、その文書を使用する命令を実行しようとする変換エンジン44に応じて、その文書を取り込むことが可能であり、その文書は、無条件で参照される文書に対して上述のようなときに解析される。他の実施においては、内容変換装置10は、そのスタイルシートが入力文書に使用されるか、または参照される文書を使用する命令を実行しようとする変換エンジン44に応じて、その文書を取り込むことが可能である場合、スタイルシートに対応する全ての参照される文書を取り込むことが可能である。

20

30

【0053】

図6は、スキーマコンパイラ22の一実施形態を示すフローチャートである。スキーマコンパイラ22をソフトウェアに実施する実施形態において、スキーマコンパイラ22は、実行されると、図6に示す機能を実施する命令を含む。図6のブロックはスキーマコンパイラ22の機能を示すが、フローチャートはコンパイラがリスト順にその機能を実行することを示唆するものではなく、また総じて次の機能を開始する前にスタイルシート上の1つの機能を完了させるものではないことに留意されたい。

【0054】

スタイルシートコンパイラ20と同様に、スキーマコンパイラ22はスキーマ(またはDTD)のノード識別子にシリアル番号を割り当てることが可能である(ブロック70)。スタイルシートコンパイラ20によって所定のノード識別子に割り当てられるシリアル番号は、スキーマコンパイラによって割り当てられるシリアル番号と異なることがある。

40

【0055】

スキーマコンパイラ22は、構文解析回路40が使用するための複数の表を生成することが可能である。例えば、実体参照は文書に含むことが可能であり、スキーマ/DTDは実体値を定義することが可能である。DTD実体参照表は、実体参照を相当値にマップするために作成することが可能である。加えて、スキーマ/DTDは、属性を含むことが可能な所与の要素がその属性を含まない場合、属性のためのデフォルト値を指定することが

50

可能である。DTDのデフォルトの属性リストは、属性およびデフォルトを記録するために作成することが可能である。加えて、許容され、要求される文書構造を識別するスケルトンツリーは、文書が（スキーマ/ DTDにおいて定義したように）有効であるかどうかを決定するバリデータが使用するために作成することが可能である。スキーマコンパイラ 22 は、メモリ 18 に、記号表、 DTD 表、 およびスケルトンツリーを出力する（ブロック 72）。

【0056】

図7は、構文解析回路40の一実施形態のための、構文解析回路40、入力データ構造、および出力データ構造を示す。図示の実施形態において、構文解析回路40が使用する入力データ構造は、DTD実体参照表34AおよびDTD属性リスト34B（DTD表34の一部であってよい）、余白表32、および記号表24を含むことが可能である。構文解析器40は、動的記号表39A（文書プロセッサ構造39の一部）および一連の解析した内容表39B（文書プロセッサ構造39の一部）を作成および使用することが可能である。より詳しくは、図示した実施形態において、解析した内容表39Bは、スケルトン表80、要素インデックス表82、要素名/値表84、属性インデックス表86、属性名/値表88、属性リスト90、処理命令/コメント（PI/C）表92、PIインデックス表94、および要素目次（TOC）96を含むことが可能である。解析した内容表39Bは、変換エンジン44における文書変換に使用することが可能である。いくつかの実施形態において、構文解析回路40は、解析のみの要求に対するSAXまたはDOMフォーマットでの解析した内容を出力するように構成することも可能である。

10

20

【0057】

構文解析回路40は、プロセッサ14A（図7には示さず）から受信したままの状態で文書を解析して、解析した内容表39Bを生成するように構成することが可能である。概して、解析した内容表39Bは、情報をさまざまなノードにリンクさせるポイントとともに、さまざまな種類の文書内容に対する表を含むことができる。図7に示す解析した内容表39Bに関する詳細を下記に示す。加えて、構文解析回路40は、(i)文書内の各コードに対するプレオーダーおよびポストオーダー数を生成すること、(ii)DTD/スキーマからの実体参照を実体値に置き換えること、(iii)所定の実体参照（XML仕様にて説明したように）を対応する文字に置き換えること、(iv)DTD/スキーマからのデフォルト属性および/または属性値を追加すること、(v)CDATAセクションを文字に置き換えること、(vi)スタイルシートが命令するように余白を取り除くか保存し、余白を正規化すること、および(vii)埋め込まれたDTD/スタイルシートまたはDTD/スタイルシートに対する埋め込まれた参照を識別することが可能である。

30

【0058】

実体参照を実体値に置き換えるために、構文解析器40は、DTD実体参照表34Aを使用することができる。文書が実体参照を受けると、構文解析回路40は、DTD実体参照表34A内の実体参照を検索すること、およびDTD実体参照表34Aからの対応する実体値を読み込むことが可能である。実体値は、その文書に対して構文解析回路40が出力する解析した内容に、実体参照を置き換える。一実施態様において、DTD実体参照表34Aは、各エントリがハッシュした実体参照（例、周期的な冗長性コード（CRC）-16 hash）を格納する、複数のエントリを有する最初のセクションと、実体値を含む文字列が格納されるDTD実体参照表34Aの第2のセクションへのポイントとを含むことが可能である。構文解析回路40は、文書内に検出される実体参照をハッシュすること、およびそのハッシュをDTD実体参照表34Aの最初のセクションのハッシュ値と比較して、その一致する実体値を第2のセクションに置くことが可能である。

40

【0059】

文書にデフォルト属性または属性値を追加するために、構文解析回路40は、DTD属性リスト34Bを使用することが可能である。DTD属性リスト34Bは、さまざまな要素名のためのデフォルト属性および/または属性値を含むことが可能であり、構文解析回路40は、その要素に任意のデフォルト属性または属性値が含まれているかどうかを決定

50

するために、文書内の要素開始タグに検出される要素名を検索することが可能である。デフォルトが含まれている場合、構文解析器 40 は、要素クローズが検出されるまで、要素開始タグ内に含まれる属性を追跡することが可能である。DTD 属性リスト 34 B 内の属性および / または属性値が要素に含まれていない場合、構文解析回路 40 は、DTD 属性リスト 34 B からの属性 / 属性値を挿入することが可能である。一実施形態において、DTD 属性リスト 34 B は、ハッシュされた要素名 (例、CRC - 16 hash) を含む最初のセクションと、DTD 属性リスト 34 B の第 2 のセクションへのポインタとを有することが可能である。第 2 のセクションは、ハッシュされた属性名 (例、CRC - 16 hash) と、デフォルトの属性名 / 値が文字列として格納される DTD 属性リスト 34 B の第 3 のセクションへのポインタとを含むことが可能である。構文解析回路 40 は、要素名をハッシュすること、そのハッシュを最初のセクション内のそのハッシュを検索すること、および第 1 の部分に一致が見つかった場合に、ハッシュされた属性名および第 2 のセクションからのポインタを読み取ることが可能である。各属性名が検出されると、構文解析回路 40 は、その属性名をハッシュして、それらを DTD 属性リスト 34 B からのハッシュした属性名と比較することが可能である。要素クローズが検出された場合、文書内で構文解析回路 40 が検出しなかった任意のハッシュされた属性名は、デフォルトが必要な属性とすることが可能であり、そのデフォルトは、DTD 属性リスト 34 B の第 3 のセクションから読み込み、解析した内容表 39 B に挿入することが可能である。

10

【0060】

余白表 32 は、どの要素名がスタイルシートに指定されているように取り除かれる余白を有することになるのかを示すことが可能である。一実施形態においては、余白が取り除かれる各要素名をハッシュすることが可能であり、(例、CRC - 16 hash アルゴリズム)、ハッシュ値は表に格納される。構文解析回路 40 が文書内の要素名を検出する場合、構文解析回路 40 は、その要素名をハッシュし、余白表 32 内でそれを検索することが可能である。一致が見つかった場合、構文解析回路 40 は、その要素から余白を取り除くことが可能である。それ以外ならば、構文解析回路 40 は、要素内に余白を保存することができる。

20

【0061】

上述のように、記号表 24 は、ノード識別子をスタイルシートコンパイラによって割り当てられるシリアル番号にマップすることが可能である。構文解析回路 40 は、記号表 24 を使用して文書内の要素または属性名 (名前空間を使用する場合、名前空間プレフィックスによって修飾される) を式プロセッサ 42 に渡すためにシリアル番号に変換することが可能である。しかし、その文書がスタイルシート内に表されない要素または属性を含む可能性がある。そのような場合、構文解析回路 40 は、シリアル番号を割り当てて、動的記号表 39 A にそのシリアル番号を格納することが可能である。図 8 のフローチャートは、文書内の要素または属性を検出する場合の構文解析回路 40 の動作の一実施形態を示す。

30

【0062】

構文解析回路 40 は、ノード識別子 (例、選択的に名前空間プレフィックスを前置した要素 / 属性名) に対するコンパイラの記号表 24 をスキャンすることが可能である (ブロック 100)。エントリが見つかった場合 (判断ブロック 102、「yes」レグ)、構文解析回路 40 は、シリアル番号をコンパイラの記号表 24 から読み込むことが可能である (ブロック 104)。エントリが見つからなかった場合 (判断ブロック 102、「no」レグ)、構文解析回路 40 は、ノード識別子に対する動的記号表 39 A をスキャンすることが可能である (ブロック 106)。エントリが見つかった場合 (判断ブロック 108、「yes」レグ)、構文解析回路 40 は、シリアル番号を動的記号表から読み込むことが可能である (ブロック 110)。エントリが見つからなかった場合 (判断ブロック 108、「no」レグ)、構文解析回路 40 は、一意のシリアル番号 (コンパイラの記号表 24 にも動的記号表 39 A にもすでに記録されていないシリアル番号) を生成することが可能であり、生成したシリアル番号およびノード識別子によって動的記号表 39 A を更新す

40

50

ることが可能である（ブロック 1 1 2）。任意の場合においても、構文解析回路 4 0 は、イベント内のシリアル番号を式プロセッサ 4 2 に発信することが可能である（ブロック 1 1 4）。

【 0 0 6 3 】

要素は、しばしば同じ名前（例、同じサブ要素または属性の複数のインスタンス）を有する複数の子（要素または属性）を有することに留意されたい。したがって、ノード識別子が入力時に検出された場合、次のノード識別子が同一となる（または、ノード識別子が、検出された次のいくつかの名前の範囲内で繰り返される）可能性がある。いくつかの実施形態において、構文解析回路 4 2 は、1 つ以上の直前に検出した名前および対応するシリアル番号を保持することが可能であり、記号表 2 4 および動的記号表 3 9 A を検索する前に新しく検出されたノード識別子とこれらの名前とを比較することが可能である。

10

【 0 0 6 4 】

いくつかの実施形態においては、コンパイラの記号表 2 4 に一致しないノードを最適化することが可能である。コンパイラがシリアル番号をスタイルシート内の各ノード識別子に割り当てるため、コンパイラの記号表 2 4 に一致しないノードは、解析時間の式ツリー 3 6 内のどのノードとも一致しないことがわかる。構文解析回路 4 0 は、シリアル番号がコンパイラの記号表 2 4 からのもの、または動的記号表 3 9 A からのものであるかどうかのイベント内の表示を含むことが可能である。シリアル番号がコンパイラの記号表 2 4 からのものではない場合、式プロセッサ 4 2 は、イベントを解析時間の式ツリー 3 6 と比較することができない。式プロセッサ 4 2 は、他の目的（例、一部のイベント型に対して、任意の「/」式ノードの子が以降のイベントと一致しない）のための注記を行うことが可能である。

20

【 0 0 6 5 】

一実施形態において、構文解析回路 4 0 が生成し、シリアル番号（および一部の実施形態におけるプレオーダー番号）とともに式プロセッサ 4 2 に発信することが可能である複数のイベントが存在し得る。要素開始タグが検出された場合には、要素開始イベントを生成することが可能であり、シリアル番号をその要素名に対応するシリアル番号とすることが可能である。要素終了タグが検出された場合には、要素終了イベントを生成することが可能であり、シリアル番号をその要素名に対応するシリアル番号とすることが可能である。要素開始タグ内に属性名が検出された場合には、属性名イベントを生成することが可能であり、シリアル番号をその属性名のシリアル番号とすることが可能である。要素開始タグの終了が検出された場合には、要素クローズイベントを生成することが可能であり、シリアル番号をその要素のシリアル番号とすることが可能である。式プロセッサ 4 2 のための所望のスタイルシート / 文書コンテキストを確立するために、構成イベントも生成することが可能である。

30

【 0 0 6 6 】

一実施形態において、記号表 2 4 は、個々の文字のツリーとして配置することが可能である。表内の各エントリは、文字、リーフノード表示、レベル表示の終了、および、そのエントリがノード識別子の最後の文字である場合に、そのエントリの第 1 の子またはそのシリアル番号のいずれかを含むことが可能である。表の最上部から開始し、1 つまたは複数の名前のそれぞれの一意の第 1 の文字がエントリに提供され、そのエントリが非リーフノードとして示され、その名前の次の文字を格納する第 1 のエントリにポインタが設定される。そのポインタでの一連のエントリへのグループ化は、第 1 の子（第 3 の文字）等へのポインタとともに第 1 の文字を有する名前の各一意の第 2 の文字である。所定の名前の終わりの文字に達したとき、リーフノード表示は、エントリがリーフであり、ポインタフィールドがシリアル番号であることを示す。レベル内の最後の一意の文字に達すると、一連の終わりがレベル表示の終わりによって示される。同様に、（名前の第 1 の文字に対する）エントリの第 1 のレベルの終了は、レベル表示の終了を使用して記録することが可能である。したがって、記号表 2 4 のスキャンは、文字単位でツリーを下って、検出した名前と記号表 2 4 との比較を含むことが可能である。

40

50

【 0 0 6 7 】

一実施形態において、動的記号表 3 9 A は、わずかに異なって編成することが可能である。名前は、その名前内の第 1 の文字に基づいて「ピン (b i n)」に格納される。名前の可能な第 1 の文字のそれぞれは、動的記号表 3 9 A へのオフセットとして使用することが可能である。これらのオフセットの各エントリは、ピンポインタ、および、「ピン内の最後のエントリ」ポインタを含むことが可能である。ピンポインタは、名前の残り (すなわち、名前の終わりまでの文字 2)、シリアル ID、および次のピンエントリへのポインタ (すなわち、そのピンエントリをリンクされたリストとすることが可能) を含む文字列である。検出された名前は、ピンエントリ内の文字列と比較することが可能で、一致が検出された場合、シリアル ID を使用することが可能である。それ以外ならば、次のピンエントリを読み込むために、次のピンエントリに対するポインタを使用する。一致を検出せずにピンの終わりに達した場合は、次いで新しいエントリをピン内の名前に追加する (また、シリアル番号を割り当てる)。1 つの特定の実施形態において、各ピンエントリは、複数の文字 (例、2) を格納するように構成された 1 つ以上のサブエントリ、および全ての文字が有効であること、または文字列の終わりを複数の文字内に置くこと、のいずれかを定義する 1 つのコードを含むことが可能である。「ピン内の最後のエントリ」のポインタは、ピン内の最後のエントリを示すことが可能であり、新しいエントリが加えられたときに次のピンポインタを更新するために使用することが可能である。

10

【 0 0 6 8 】

解析した内容表 3 9 B を一実施形態に対して以下に詳細に説明する。構文解析回路 4 0 は、文書の構造 / 内容を識別し、検出した構造 / 内容に基づいてさまざまなデータ構造に対する文書内容を解析した内容表 3 9 B に書き込む。例えば、構文解析回路 4 0 は、要素名 / 値表 8 4 内で検出された要素名 (および、対応する要素値 / テキストノード) を格納することが可能であり、属性名 / 値表 8 8 内で検出された属性名 (および、相当値) を文字列として格納することが可能である。対応するインデックス表 8 2 および 8 6 は、対応する文字列の始めに対するポインタを表 8 4 および 8 8 にそれぞれ格納することが可能である。インデックス表 8 2 および 8 6 は、それぞれ要素 (図 7 の E S / N) または属性 (図 7 の A S / N) のシリアル番号を使用してアドレス指定が行われる。

20

【 0 0 6 9 】

処理命令 / コメント (P I / C) 表 9 2 は、処理命令およびコメントに対応する文字列を格納する。コメントは、要素 T O C 9 6 に格納したポインタによって置かれる文字列として格納することが可能である。処理命令は：処理命令の目標部分 (拡張した名前)、および処理命令の値部分 (文書からの処理命令の残り) の 2 つのストリング値を含むことが可能である。処理命令の目標および処理命令値は、エントリからの一組のポインタによって P I インデックス表 9 4 内に置くことが可能であり、要素 T O C 9 6 からのポインタによってインデックスが付けられる。P I インデックス表 9 4 のエントリは、一組のポインタおよび処理命令に割り当てられるシリアル番号を含むことが可能である。

30

【 0 0 7 0 】

構文解析回路 4 0 は、各要素に対する属性リスト 9 0 を文書内に生成することも可能である。属性リスト 9 0 は、属性名 / 値表 8 8 内の (もしあれば) 属性名および属性値に対するポインタによって、その要素に対応する (シリアル番号による) 属性のリストとすることが可能である。加えて、構文解析回路 4 0 は、各要素のための要素 T O C 9 6 を生成することが可能である。要素 T O C 9 6 は、対応する要素 (例、子要素、テキストノード、コメントノード、および処理命令ノード) の子ノードを識別する。要素 T O C 9 6 内の各エントリは、ノード位置 (子要素の位置をノード内の他の子要素と比較して識別する)、ノード型 (子ノードを要素、テキスト、コメントまたは処理命令として識別する)、および (コメント、処理命令、またはテキストノードのための) ノード内容ポインタまたは (要素ノードのための) 子のプレオーダー番号のいずれか、を含むことが可能である。ノード内容ポインタは、コメントノードのための P I / C 表 9 2 へのポインタ、処理命令ノードのための P I インデックス表 9 4 へのポインタ、またはテキストノードのため

40

50

の要素名/値表 8 4 へのポインタである。一実施形態において、要素 T O C 9 6 をエントリのリンクされたリストとすることが可能であり、したがって、各エントリはリスト内の次のエントリに対するポインタをさらに含むことが可能である。

【 0 0 7 1 】

スケルトン表 8 0 は、文書内の各要素ノードに対するエントリを含むことが可能であり、その要素ノードのプレオーダー番号によってインデックスを付けることが可能である。図示の実施態様では、スケルトン表の任意のエントリは、親ノードプレオーダー番号 (P P R E O) と、要素ノードに対する直前の兄弟のプレオーダー番号 (I P S P R E O) と、要素ノードの子孫であるサブツリー内の最後のプレオーダー番号も含むことが可能な要素ノードのポストオーダー番号 (P S T O) と、要素シリアル番号 (E S / N) と、要素ノードのための属性リスト 9 0 に対する属性リストポインタ (A L P) と、要素ノードのための要素 T O C 9 6 に対する目次ポインタ (T O C P) と、(もしあれば) 要素ノードに対する一致するテンプレートをリストしたテンプレートリスト表 3 8 内のエントリを示すテンプレートリストポインタ (T L P) とを含む。

10

【 0 0 7 2 】

上述の種々のデータ構造は文字列を含むことに留意されたい。一実施形態において、文字列長 (例、文字数) は、その文字列の最初の「文字」として格納することが可能であり、構文解析回路 4 0 は、何文字を読み込むのかを決定するために、その文字列を使用することが可能である。

【 0 0 7 3 】

上述の例の構文解析回路 4 0 およびその出力データ構造は、XML 文書に使用することが可能である。一実施形態において、構文解析回路 4 0 は、XML 文書を解析するように設計されたハードウェア回路構成を含む。いくつかの実施形態においては、構文解析回路 4 0 は、リレーショナルデータベース構造 (例えば S Q L、O r a c l e 等) を解析するように設計されたハードウェア回路構成も含むことが可能である。構文解析回路 4 0 は、図 7 に示す構造と類似の解析したリレーショナルデータベース構造を出力することが可能であり、したがって、式プロセッサ 4 2 および変換エンジン 4 4 は、その入力 XML なのかリレーショナルデータベースなのかを知る必要がない。

20

【 0 0 7 4 】

いくつかの実施態様において、構文解析回路 4 0 は、他の種類の文書を解析するようにプログラム可能である。構文解析回路 4 0 は、1 つ以上の入力型記述子によってプログラム可能になり得る。入力型記述子は、例えば、文書内に構造区切りを記述すること、文書が事実上階層的か、または平板状かを示すこと、階層的な文書が構造の各レベルに対して明示的な終わりを有するかどうかを示すこと、構造の終わりが明示的でない場合、どのようにその終わりを検出すべきかを定義すること、もしあれば、所定の構造的なユニット内に内部構造を定義すること、が可能である。

30

【 0 0 7 5 】

いくつかの実施形態では、構文解析回路 4 0 へのフィルタ処理した文書を生成するために、C P U 1 4 A および 1 4 B が提供する文書をフィルタ処理する、プレ構文解析回路も含むことが可能である。すなわち、構文解析回路 4 0 は、フィルタを通過した文書の一部だけを受信し、構文解析回路 4 0 は、その受信した部分を解析する文書全体として処理することが可能である。プレ構文解析器は、例えば、比較的大きな文書が提供されても、その文書のサブセットだけが対象となる場合に使用することが可能である。プレ構文解析器は、文書を任意の所望の形態にフィルタ処理するようにプログラム可能である (例えば、文書の始めから所定の数の文字をスキップして、複数の文字またはその文書の終わりまで取り込むように ; 文書内容を取り込む前に特定の要素まで、または複数の要素までフィルタ処理するように ; および / または、取り込む文書の一部を識別する、より複雑な式 (例、X P a t h 式) などにプログラム可能である)。プレ構文解析器は、スタイルシートが文書内容を破棄するなどの影響を有する場合に、スタイルシートコンパイラ 2 0 によってユーザーがプログラムすることが可能である。

40

50

【 0 0 7 6 】

図 9 は、式プロセッサ 4 2 の一実施形態のための、式プロセッサ 4 2、入力データ構造、および出力データ構造を示すブロック図である。図示の実施形態では、式プロセッサ 4 2 が使用する入力データ構造は、解析時間の式ツリー 2 6、式リスト表 3 6、およびテンプレートリスト表 3 8 を含む。式プロセッサ 4 2 は、複数の文書プロセッサ構造 3 9 (特に、ノスタック 3 9 C、ノノスタック 3 9 D、ポインタ (Ptr) スタック 3 9 E、および属性 (Attr) スタック 3 9 F) を生成および使用することも可能である。式プロセッサ 4 2 は、変換エンジン 4 4 にテンプレートノ式一致リスト 3 9 G を出力することが可能である。

【 0 0 7 7 】

概して、式プロセッサ 4 2 は、構文解析回路 4 0 からイベントを受信し、その中に識別される文書ノードを解析時間の式ツリー 2 6 内の式ノードに一致させる。文書ノードは、それらがインラインで解析されるように受信される。したがって、任意の所定の時点で、以前に受信した文書ノードは、解析時間の式ツリーの一致した部分を有することが可能であるが、(式全体が構文解析回路 4 0 によって提供される一連の文書ノードに一致した) そのツリーのリーフはまだ達していない。式プロセッサ 4 2 は、以前の文書ノードに一致させて、事実上、次の文書ノードと比較することが可能な解析時間の式ツリー 2 6 内の位置を保持している、解析時間の式ツリー 2 6 の一部を格納するために、スタック 3 9 C 乃至 3 9 F を使用することが可能である。

【 0 0 7 8 】

図示した実施形態は、X Path 式に使用することが可能であり、ノード間の演算子は、親ノ子演算子 (「/」) および子孫ノ祖先演算子 (「//」) を含むことが可能である。したがって、所定の式ノードは、1 つ以上のノ子、および、1 つ以上のノノ子を有することが可能である。所定の式ノードが、ノ子を有し、文書ノードに一致する場合、所定の式ノードをノスタック 3 9 C にプッシュすることが可能である。同様に、所定の式ノードが、ノノ子を有し、文書ノードに一致する場合、所定の式ノードをノノスタック 3 9 D にプッシュすることが可能である。文書ノードが属性である場合、その属性は、Attr スタック 3 9 F に格納することが可能である。いくつかの実施形態において、スタックポインタのトップは、イベントを処理する前にそのスタックの状態を回復できるように、そのイベントの処理の開始時に保存される。Ptr スタック 3 9 E は、ポインタを保存するために使用することが可能である。

【 0 0 7 9 】

ランタイム部分を有する式が式プロセッサ 4 2 において部分的に評価される実施態様において、そのランタイム部分に関する情報は、ランタイム評価を実行し、式のランタイム部分を満たさない一致リスト 3 9 G 内の文書ノードを破棄することが可能であるように、一致リスト 3 9 G に保持することが可能である。したがって、一致リスト 3 9 G は、ランタイム評価を有する式の各部分で分類することが可能である。ランタイム部分が使用する同じ値を有する各文書ノードは、所与のグループ内に含むことが可能である。

【 0 0 8 0 】

図 9 に示すように、一致リスト 3 9 G は、式のためのノードセットを形成する階層的アレイの一群の文書ノードを含むことが可能である。解析時間の式ツリー内の各式にはそのような構造が存在し得る (すなわち、図 9 に示す構造は、1 つの式に対応することが可能である)。主グループ (例、図 9 の P G 0 および P G 1) は、解析時間の式ツリー内のトップレベルのノードに対応することが可能であり、トップレベルのノードを一致させ、それ自身がノードセットのメンバーであるか、またはそのノードセットのメンバーである子孫を有する、各異なる文書ノードに対する異なる主グループが存在し得る。各サブグループレベルは、ランタイム評価 (例えば、一実施態様にけるランタイム述語) に対応することが可能である。ランタイム述語に使用する値は、ポインタとともに次のレベルのグループ (または、ノードリスト自身) に保持することが可能である。式に対する一致が生じた場合、ノードは、サブグループレベルのその値に基づいてグループ内に配置される。すな

10

20

30

40

50

わち、所与のレベルで、ノードは、その値がサブグループと一致するサブグループか、またはそのノードに対して作成される新しいサブグループに含まれる。図の例では、サブグループの第1のレベル(第1のランタイム述語に対応する)は、主グループ0(PG0)からのサブグループ0および1(SG0およびSG1)を含む。同様に、主グループ1(PG1)は、第1のランタイム述語に対応するサブグループSGMおよびSGM+1を含む。第2のランタイム述語は、サブグループSG0のサブグループとしてSGNおよびSGN+1を含む、第2のレベルのサブグループに対応する。この例では、式内に2つのランタイム述語があり、したがって、サブグループSGMおよびSGM+1は、それぞれ潜在的に一致する文書ノードのリスト(図示の例では、ノードN0、N1、およびN2)を示す。一実施態様において、文書ノードは、構文解析器40によって割り当てられるそれらのプレオーダー番号によって一致リスト39G内に表すことが可能である。

10

【0081】

階層構造を使用することで、変換エンジン44は、主グループを選択し、第1のランタイム述語を評価し、その第1のランタイム述語を主グループの各サブグループと比較(第1のランタイム述語を満たさないあらゆるサブグループを破棄)することが可能である。変換エンジン44は、第2のランタイム述語を評価し、第2のランタイム述語を、破棄されていない第1のレベルのサブグループのそれぞれのサブグループと比較し、第2のランタイム述語などを満たさないサブグループを破棄することが可能である。各ランタイム述語を評価した後に構造内に残ったノードは、対応する式を満たすノードセットである。変換エンジン44は、式に対応する命令表30内の命令を検索し、それらの命令をノードセ

20

【0082】

一実施態様において、解析時間の式ツリー26内の第1の式が第2の式の接尾辞である場合(すなわち、第2の式は、第1の式に含まれない接頭辞を含むが、第1の式全体は、第2の式の終わりと同じである)、独立した一致リスト39Gは、第1の式に対して作成され得ない。その代わりに、第2の式に対する一致リスト39Gが作成され、第1の式のトップレベルのノードのグループを含む。第1の式に対応するポイントは、第2の式に対して一致リスト39G内の第1の式のトップレベルのノードのグループを示し得る。

【0083】

一実施態様において、所与の式と一致するノードは、対応するテンプレート本体によってそれら自身を操作することが可能であるか、またはそのノードの値(例、属性値または要素の内容)を操作することが可能である。スタイルシートコンパイラ20は、解析時間の式ツリー26内の各リーフノードに対して、ノードまたはノードの値が必要であるかどうかを示すように構成することが可能である。いくつかの実施態様において、式プロセッサ42は、各ノードに対して、そのノードが一致するテンプレートのリストを出力することが可能である。

30

【0084】

式リスト表36は、スタイルシート内に含まれる式のリストとすることが可能である。スタイルシートコンパイラは、式に式番号を割り当て、その式番号を式リストに保存することが可能である。解析時間の式ツリー内のポイントは、式リスト内のエントリを示すことが可能である。各エントリは、式番号およびグループ化が必要な式ツリーのレベルを示すグループ署名を保存することが可能である。例えば、一実施態様において、グループ署名は、ゼロはそのレベルにグループ化がないことを示し、1はグループ化を示す、式ツリーの各レベルに対するビットを含むことが可能である。いくつかの実施態様では、複数の式が所与のリーフノードに対応することが可能である。例えば、別の式との一致により削除されるコンパイル時間式は、どちらの式もリーフノードと一致することになり得る。加えて、所与のスタイルシートは、複数の場所に等価式を有することが可能である。このような実施態様では、一致する式番号のリストは、式リスト表36の連続したエントリに保存され、そのエントリは、所与のリーフノードに対する最後の一致する式を識別することが可能な最後の式表示を含むことが可能である。一致する式が1つだけである場合、式ポ

40

50

インタによって示される第1のエントリ内の最後の式表示は、最後のエントリを示す状態におけるその最後のエントリ表示を有することが可能である。

【0085】

テンプレートリスト表38は、同様に、所与のリーフノードに対する複数の一致テンプレートを許可するために、テンプレート番号および最後のテンプレート表示を有するエントリを含むことが可能である。解析時間の式ツリー36内のリーフノードは、同様に、1つ以上の一致するテンプレートのためのテンプレートリスト表に対するポインタを含むことが可能である。テンプレートリスト表38は、テンプレート型フィールド（例、インポートしたか否か、テンプレートがモード番号を有するか否か、およびテンプレートが1つ以上のランタイム述語を有するか否か）と、モード番号、インポートされる型に対してインポートしたテンプレートからのスタイルシートを識別するインポート識別子と、テンプレートに対して実行される命令を識別する命令表30へのテンプレート本体の命令ポインタと、1つ以上のランタイム述語を評価するために実行される命令を識別する命令表30への述語ポインタとを、さらに含むことが可能である。

【0086】

図10は、解析時間の式ツリー26のデータ構造の一実施態様のブロック図を示す。図示の実施態様では、解析時間の式ツリーは、エントリ120のような複数のエントリを有し、各エントリが式ツリー内の式ノードに対応する、表を含むことが可能である。各式ノードは、本実施態様において、次の最大3つの異なる種類のゼロ以上の子を有することが可能である：(1)ノ子、文書ツリー内のノードの子である、(2)ノノ子、文書ツリー、文書ツリー内のノードの子孫（直接の子か、または1つ以上のノードのツリーを間接的に介する）である、または(3)属性の子（要素ノードの属性）である。加えて、所与の式ノードは、トップレベルのノードであっても、トップレベルのノード以外であってもよい。一実施態様において、解析時間の式ツリー26は、「フォレスト」という複数のツリーを含むことが可能であり、それぞれがルートを有する。トップレベルのノードは、ツリーのうちの1つのルートであり、そのツリーはトップレベルのノードで始まる1つ以上の式を表すことが可能である。トップレベルのノードは、エントリ120に関して下記に詳細を説明するように、次のレベルでのノードを示すことによって、解析時間の式ツリーのデータ構造の最上位でグループ化することが可能である。

【0087】

エントリ120のフィールドを次に説明する。エントリ120は、トップレベルの式ノードに使用されるトップレベルの型（TLT）フィールドを含む。トップレベルの型は、相対的、絶対的、または祖先としてコード化することが可能である。相対的なトップレベルのノードは、文書ツリー内のコンテキストノードに関連して評価される1つ以上の式を開始する式ノードであるが、絶対的なトップレベルのノードは、文書ツリーのルートノードから評価される1つ以上の式を開始する式ノードである（すなわち、1つまたは複数の式は、ノで始まり、後ろにトップレベルのノード識別子が続く）。祖先のトップレベルのノードは、コンテキストノードの祖先を参照する式の始めである（すなわち、1つまたは複数の式は、ノノで始まり、後ろにトップレベルのノード識別子が続く）。

【0088】

エントリ120は、式ノードのシリアル番号を保存するシリアル番号（S/N）フィールドを含む。S/Nフィールドは、エントリ120に保存した式ノード上の一致（等しいシリアル番号）または不一致を検出するために、構文解析回路40によって発信されるイベント内に識別される文書ノードのシリアル番号と比較する。エントリ120は、エントリ120に保存した式ノードがリーフノードであるかどうか（すなわち、式の終わりに達したかどうか）を識別するリーフノード（LN）フィールドをさらに含む。リーフノード上の一致によって、リーフノードに対応する各式ノード/テンプレートのための一致リスト39G内に記録される文書ノードが生じる。LNフィールドは、例えば、設定時には、式ノードがリーフノードであることを示すビットとし、クリア時には、式ノードがリーフノードではないことを示すビットとすることが可能である。他の実施態様は、ビットの設定およ

10

20

30

40

50

びクリアの意味を逆にするか、または他のコード化を使用することが可能である。

【0089】

パス型のフィールドは、エン트리120内に保存された式ノードからパスリンクの種類を識別することが可能である(例、/、//、またはその両方)。例えば、パス型のフィールドは、各型に対するビットを含むことが可能であり、そのパスの型がこのノードから生じることを示すように設定すること、およびそのパスの型が生じないことを示すようにクリアすることが可能である。他の実施態様は、ビットの設定およびクリアの意味を逆にするか、または他のコード化を使用することが可能である。パス型のフィールドは、「Ptr/」、および、「Ptr//」フィールドを認証することが可能である。Ptr/フィールドは、式ノードの第1の/子に対するポインタを保存することが可能である(および、/子のそれぞれは、Ptr/ポインタによって示されるエン트리から始まる解析時間の式ツリーのデータ構造の連続したエントリに分類することが可能である)。同様に、Ptr//フィールドは、式ノードの第1の//子に対するポインタを保存することが可能である(および、//子のそれぞれは、Ptr//ポインタによって示されるエン트리から始まる、解析時間の式ツリーのデータ構造の連続したエントリに分類することが可能である)。PtrAttrフィールドは、解析時間の式ツリー内の第1の属性ノードへのポインタを保存することが可能である(および、属性のそれぞれは、PtrAttrポインタによって示されるエン트리から始まる、解析時間の式ツリーのデータ構造の連続したエントリに分類することが可能である)。

10

【0090】

EOLフィールドは、エン트리120が現在のツリーレベルの終わりである式ノードを保存するかどうかの表示を保存する。例えば、レベルの終わりを示す解析時間の式ツリーのデータ構造のトップから第1のエン트리からの第1のエントリは最後のトップレベルのノードを示すことが可能である。各ポインタ(例、Ptr/、Ptr//、またはPtrAttr)から始まる、エントリは、レベルの終わりを示すEOLフィールドを有するエントリが達成されるまで、ポインタを含むエントリの子である。EOLフィールドは、例えば、設定時には、式ノードがレベルの終わりであることを示すビットとし、クリア時には、式ノードがレベルの終わりではないことを示すビットとすることが可能である。他の実施態様は、ビットの設定およびクリアの意味を逆にするか、または他のコード化を使用することが可能である。

20

30

【0091】

エン트리120は、式リスト表36内のエントリを示す式リストポインタを保存する式リストポインタ(XLP)フィールドと、上述のように、テンプレートリスト表38内のエントリを示すテンプレートリストポインタを保存するテンプレートリストポインタ(TLP)フィールドとをさらに含む。XLPおよびTLPフィールドは、リーフノードに対して有効とすることが可能である。

【0092】

述語には、本実施態様の解析時間で評価可能となり得るものがあり、述語型(PrTP)フィールドおよび述語データ(PrDT)フィールドを当該の述語を表すために使用することが可能である。例えば、述語タイプフィールドは、評価可能な述語、位置述語、または属性名述語を示さないようにコード化することが可能である。述語データフィールドは、式(例、位置述語のための位置番号、または属性名述語に対する属性名のための属性名またはシリアル番号)から述語データを保存することが可能である。

40

【0093】

図11は、典型的な式ツリー122のブロック図、および式ツリー122に対応する解析時間の式ツリーエン트리120A乃至120Eの対応部分を示す。式ツリー122は、シリアル番号10を有する式ノード124Aを含み、式ノード124Aは、2つの/子ノード124Bおよび124C(シリアル番号15および20)と、1つの//子ノード1240(シリアル番号25)と、属性の子124E(シリアル番号30)とを有する。したがって、式ツリー122は、10/15、10/20、10//25、および、10/

50

attribute : : 30 という式を表す (ノード 1 2 4 A が相対的なトップレベルのノードであると仮定) 。

【 0 0 9 4 】

エントリ 1 2 0 A 乃至 1 2 0 E は、解析時間の式ツリーエントリの S / N、L N、E O L、P t r /、P t r / /、および P t r A t t r フィールドを示す。エントリ 1 2 0 A は、ノード 1 2 4 A に対応するので、シリアル番号 1 0 を含む。ノード 1 2 4 A はリーフノードではないので、L N フィールドはエントリ 1 2 0 A においてゼロである。この例に関して、ノード 1 2 4 A はツリー 1 2 2 内のそのレベルでの唯一のノードであるので、E O L フィールドは 1 となる。エントリ 1 2 0 A の P t r / フィールドは、エントリ 1 2 0 B (第 1 の / 子) を示す。エントリ 1 2 0 A の P t r / / フィールドは、エントリ 1 2 0 D を示し、エントリ 1 2 0 A の P t r A t t r フィールドは、エントリ 1 2 0 E を示す。

10

【 0 0 9 5 】

エントリ 1 2 0 B は、S / N フィールド内に 1 5 を含み、ノード 1 2 4 B が式ツリー 1 2 2 のリーフノードであるので、L N フィールドは 1 となる。しかし、このレベルには別の / 子があるので、E O L フィールドはゼロとなる。エントリ 1 2 0 B の P t r /、P t r / /、および P t r A t t r フィールドは、これがリーフノードであるので、ヌルである。エントリ 1 2 0 C は、S / N フィールド内に 2 0 を含み、ノード 1 2 4 C が式ツリー 1 2 2 のリーフノードであるので、L N フィールドは 1 となる。このレベルではノード 1 2 4 C が最後の / 子であるので、E O L フィールドも 1 となる。また、エントリ 1 2 0 C がリーフノードであるので、エントリ 1 2 0 C の P t r /、P t r / /、および P t r A t t r フィールドはヌルとなる。

20

【 0 0 9 6 】

エントリ 1 2 0 D は、S / N フィールド内に 2 5 を含み、ノード 1 2 4 D が式ツリー 1 2 2 のリーフノードであるので、L N フィールドは 1 となる。このレベルではノード 1 2 4 D が最後の / / 子であるので、E O L フィールドも 1 となる。エントリ 1 2 0 D がリーフノードであるので、エントリ 1 2 0 D の P t r /、P t r / /、および P t r A t t r フィールドはヌルとなる。

【 0 0 9 7 】

エントリ 1 2 0 E は、S / N フィールド内に 2 5 を含み、ノード 1 2 4 E が式ツリー 1 2 2 のリーフノードであるので、L N フィールドは 1 となる。このレベルではノード 1 2 4 E が最後の属性の子であるので、E O L フィールドは 1 となる。エントリ 1 2 0 E がリーフノードであるので、エントリ 1 2 0 E の P t r /、P t r / /、および P t r A t t r フィールドはヌルとなる。

30

【 0 0 9 8 】

図 1 2 A および 1 2 B、1 3、1 4 A および 1 4 B、ならびに 1 5 は、構文解析回路 4 0 が生成し得る各々のイベントのための式プロセッサ 4 2 の一実施態様の演算を示すフローチャートである。各イベントは、検出された文書ノードのシリアル番号 (および、いくつかの実施態様では、文書ノードのプレオーダー番号) を含むことが可能である。式プロセッサ 4 2 は、ハードウェア内に実装することが可能であるので、そのフローチャートは、種々のブロックをハードウェア内で並行して、または必要に応じてハードウェア内でパイプライン化して実行することが可能であっても、ハードウェアの演算を表すことが可能である。フローチャートは、概して一致する文書ノードおよび式ノードを参照することが可能である。上述のように、このような一致は、文書ノードおよび式ノードの一致するシリアル番号を含むことが可能である。さらに、そのフローチャートは、一致リストに出力するノードを参照することが可能である。上述のように、ノードは、いくつかの実施態様では、プレオーダー番号によって一致リスト内に表すことが可能である。

40

【 0 0 9 9 】

図 1 2 A および 1 2 B は、要素開始イベントに応えた、式プロセッサ 4 2 の一実施態様の演算を示す。要素開始イベントは、文書内の要素開始タグの検出に応じて発信すること

50

が可能である。

【0100】

式プロセッサ42は、ノスタック39Cに保存することが可能なあらゆる属性式ノードをポップすることが可能であり、ポインタスタック39Eへのノスタックポインタをプッシュすることが可能である(ブロック130)。新しい要素が始まっているので、あらゆる属性式ノード上のノスタックは一致されず、したがって、必要とされない。要素開始イベントによって識別された要素(図12Aおよび12Bの記述では、より簡潔に要素と称する)が文書のルートノードである場合、更なる処理は行われ(判断ブロック132、「yes」レグ)。ルートノードは、解析時間の式ツリー26内のノードのいずれにも一致することができず、あらゆるトップレベルのノードをルートの子に一致することが可能である。要素がルートノードではないが(判断ブロック132、「no」レグ)、その要素の親がルートノードである場合(判断ブロック134、「yes」レグ)、式プロセッサ42は、ルートノードの子に対する絶対的なトップレベルのノードに対してであっても一致が検出され得るので、解析時間の式ツリー26内の各トップレベルの式ノードを検査することが可能である(ブロック136)。一方で、要素ノードの親要素がルートノードである場合(判断ブロック134、「no」レグ)、式プロセッサ42は、ルートノードの子ではないノードに対する絶対的なトップレベルのノードに対して一致が検出されない(判断ブロック138)ので、解析時間の式ツリー26内の各相対的なトップレベルの式ノードを検査することが可能である(ブロック138)。

10

【0101】

トップレベルのノードのいずれかに対して一致が検出されない場合(判断ブロック140、「no」レグ)、フローチャートは、図12の参照Aを継続する。一致が検出され(判断ブロック140、「yes」レグ)、かつ式ノードがリーフノードである場合(判断ブロック142、「yes」レグ)、解析時間の式ツリー26の式ノードのエントリ内のXLPおよびTLPポインタによって示される1つもしくは複数の式および/または1つもしくは複数のテンプレートに対応する一致リストに、要素ノードが出力される(ブロック144)。一致した式ノードが、リーフノードでない場合(判断ブロック142、「no」レグ)、式プロセッサ42は、一致した式ノードが/または//のいずれかの子を有するかどうかを判断し(それぞれ、判断ブロック146および148)、一致した式ノードが/または//のいずれかの子を有する場合(それぞれ、判断ブロック150および152)、それぞれノスタック39Cおよび/または//ノスタック39D上の一致した式をプッシュする。加えて、ノスタック39Cおよび//ノスタック39Dは、(解析時間の式ツリーエントリ内のPrTPおよびPrDTフィールドを経て示される)解析時間述語の一致の管理に使用する評価フィールドを含むことが可能である。(PrTPフィールドによって示されるような)解析時間述語がある場合、評価フィールドを0に設定することが可能である。それ以外ならば、評価フィールドを2に設定することが可能である。フローチャートは、図12Bの参照Aを継続する。

20

30

【0102】

図12Bの参照Aで、フローチャートは、そのフローチャートを介したパスが、この要素に対するノスタック39Cの第1の検査であるか否かによって別々に演算する(判断ブロック154)。これが、この要素に対するノスタック39Cの第1の検査である場合(判断ブロック154、「yes」レグ)、式プロセッサ42は、その要素の親がノスタック39Cで一致するかどうかを判断する(判断ブロック156)。要素の親がノスタック39Cで一致する場合(判断ブロック156、「yes」レグ)、次いで一致した式ノードのノ子の中の1つをその要素に一致することが可能である。式プロセッサ42は、一致した式ノードの第1のノ子を、一致した式ノードの解析時間の式ツリーエントリ内のPtrノが示すように取り込むことが可能であり(ブロック158)、ノ子上に一致が検出されるかどうかを判断するために(および、一致が検出された場合に図12Aのブロック142乃至152に示すように処理するために)図12Aの参照Bに戻る(判断ブロック156)ことが可能である。要素の親がノスタックで一致しない場合(判断ブロック156、「no」レグ)、式

40

50

プロセッサ 42 は、ノスタック 39D を検査することが可能である (ブロック 160)。同様に、このパスが、この要素に対する第 1 の検査でなく (判断ブロック 154、「no」レグ)、かつ最後のノ子が、ノスタック内の一致した式ノードから取り込まれている場合 (判断ブロック 162、「yes」レグ)、式プロセッサ 42 は、ノスタック 39D を検査することが可能である (ブロック 160)。このパスが、この要素に対する第 1 の検査でなく (判断ブロック 154、「no」レグ)、かつ最後のノ子が、ノスタック内の一致した式ノードから取り込まれていない場合 (判断ブロック 162、「No」レグ)、式プロセッサ 42 は、ノスタック内の一致した式ノードの次のノ子を取り込むことが可能であり、そのフローチャートは、その要素に対する一致が検出されるかどうかを判断するために、図 12A の参照 B を継続することが可能である。したがって、ブロック 154 乃至 164 の演算 (および、図 12A 内のブロック 140 乃至 152 への回帰) を介して、要素の親に一致する式ノードの各ノ子は、その要素によって一致を検索することが可能である。

10

【0103】

いくつかの実施態様において、ある要素の親要素を式プロセッサ 42 で保持することが可能である (例、式プロセッサ 42 は、その要素に対して要素開始イベントは生じているが要素終了イベントは生じていない、多くの要素を保持することが可能である)。この代わりに、他の実施態様では、親要素を構文解析回路 40 で保持し、要素開始イベント内に含むことが可能である。

【0104】

本実施態様において、ノスタック 39D の検索は、ノスタック 39C の検索とは幾分異なる処理を伴うことがある。ノードは、ノスタック 39D 上のあらゆる式ノードのノノ子に一致することが可能である (これは、ノ演算子が式ノードのいずれの子孫を選択し、スタック 39D 上のエントリが、要素開始イベント内で識別される要素の親または祖先である以前の要素と一致させているためである)。したがって、図 12B のフローチャートは、ノスタック 39D 上の各有効なエントリのノノ子の検索を示す。

20

【0105】

ノスタック 39D が、有効なエントリ (または、それ以上有効なエントリ) を持たない場合 (判断ブロック 166「no」レグ)、ノスタックの処理を終了し、要素開始イベントの処理も終了する。ノスタック 39D が有効なエントリを有する場合 (判断ブロック 166「yes」レグ)、式プロセッサ 42 は、エントリ内の Ptrノノが示すように、そのエントリの第 1 のノノ子を取り込む (ブロック 168)。式プロセッサ 42 は、一致が検出されるかどうかを判断するために、ノノ子と比較する (判断ブロック 170)。一致が検出され (判断ブロック 170「yes」レグ)、かつノノ子ノードがリーフノードである場合 (判断ブロック 172「yes」レグ)、ブロック 144 と同様に、その要素が一致リストに出力される (ブロック 174)。ブロック 146 乃至 152 と同様に、ノノ子ノードがリーフノードではなく、かつ一致が検出された場合、式プロセッサ 42 は、ノノ子ノードがノまたはノノ子自身のいずれかを有するかどうかを判断し (それぞれ、判断ブロック 176 および 178)、ノノ子ノードがノまたはノノ子のいずれかを有する場合 (それぞれ、ブロック 180 および 182)、それぞれノスタック 39C およびノまたはノスタック 39D 上のノノ子ノードをプッシュする。加えて、ノスタック 39C またはノスタック 39D 内の評価フィールドは、ブロック 146 乃至 152 に関して上述のように設定することが可能である。

30

40

【0106】

現在のノスタックの最後のノ子が処理されていない場合 (判断ブロック 184「no」レグ)、式プロセッサ 42 は、現在のノスタックエントリの次のノノ子を取り込み (ブロック 186)、フローチャートは次のノノ子に対する判断ブロック 170 を継続する。最後のノ子が処理された場合 (判断ブロック 184、「yes」レグ)、式プロセッサ 42 は、ノスタック 39D 内の次のエントリに進み (ブロック 188)、フローチャートは次のノスタックエントリに対する判断ブロック 166 を継続する。

50

【0107】

図13は、要素終了イベントに応えた、式プロセッサ42の一実施態様の演算を示す。要素終了イベントは、文書内の要素終了タグの検出に応えて、構文解析回路40によって発信することが可能である。

【0108】

要素終了イベントが、文書のルートノードに対するものである場合（判断ブロック190、「yes」レグ）、その文書は完了している（ブロック192）。式プロセッサ42は、スタック39C乃至39Fをクリアすることが可能である。要素終了イベントが、文書のルートノードに対するものでない場合（判断ブロック190、「no」レグ）、式プロセッサ42は、終了要素に対応するノードおよびノスタックエントリをポップすることが可能である（ブロック194）。要素はクローズされているので、要素の全ての子はすでに解析されている。したがって、その要素に対応するノードおよびノスタックエントリ内のいずれのエントリ（すなわち、その要素のシリアル番号を有するエントリ）も、引き続き検出されるノードと一致させることができない。

【0109】

図14Aおよび14Bは、属性名イベントに応えた、式プロセッサ42の一実施態様の演算を示す。属性名イベントは、文書の要素開始タグ内での属性名の検出に応えて、構文解析回路40によって発信することが可能である。属性名は、そのシリアル番号で表すことが可能である。

【0110】

式プロセッサ42は、Attrスタック39Fに、属性名（すなわち、そのシリアル番号）をプッシュすることができる。Attrスタック39Fは、要素クローズコマンド処理（図15）のための属性名を蓄積する。属性の親ノードがルートノードである場合（判断ブロック202、「yes」レグ）、（ルートノードは属性を持たないので）実行する更なる処理はない。一方で、属性の親ノードがルートノードでない場合（判断ブロック202、「no」レグ）、式プロセッサ42は継続する。

【0111】

式プロセッサ42は、属性名に対する一致（また、シリアル番号による一致）のための相対的なトップレベルの式ノードのそれぞれを検査することが可能である（ブロック204）。所与の相対的なトップレベルの式ノードとの一致がない場合（判断ブロック206、「no」レグ）、トップレベルの式ノードが空になる（判断ブロック208、「no」レグ）まで、処理は、次の相対的なトップレベルの式ノードを継続する。トップレベルの式ノードが空になると（判断ブロック208、「yes」レグ）、処理は図14Bの参照Cを継続する。

【0112】

一致が検出され（判断ブロック206、「yes」レグ）、かつノードがリーフノードである場合（判断ブロック210、「yes」レグ）、属性ノードが一致リスト39Gに出力される（ブロック210）。式プロセッサ42は、一致した式ノードがノまたはノのいずれかの子を有するかどうかを判断し（それぞれ、判断ブロック212および214）、一致した式ノードがノまたはノのいずれかの子を有する場合（それぞれ、判断ブロック216および218）、それぞれノスタック39Cおよびノまたはノスタック39D上の一致した式をプッシュする。

【0113】

図14Bの参照Cを継続し、式プロセッサ42は、ノスタック39Cおよびノスタック39D内の属性名の親ノードの一致に対する検査を行う（判断ブロック220）。一致が検出されない場合（判断ブロック220、「no」レグ）、属性名イベントの処理を終了する。一致が検出された場合、式プロセッサ42は、その一致する式ノードが属性名述語（または、一実施態様における予約したコード化）を有するかどうかを判断するために一致する式ノードの式ツリーのPrTPフィールドを検査する。一致する式ノードが属性名述語を有し、かつ評価フィールドの最下位ビットがクリアである場合（すなわち、評価

10

20

30

40

50

フィールドが、0か2のいずれかである場合) (判断ブロック222、「yes」レグ)、式プロセッサ42は、その属性名を一致する式ノードの式ツリーエントリのPrDTフィールドを比較する。属性が一致しない場合(判断ブロック224、「no」レグ)、式プロセッサ42は、もしあれば、/スタック39Cおよび//スタック39D内の次の一致するノードを継続する。一実施態様において、属性が一致する場合(判断ブロック224、「yes」レグ)、式プロセッサ42は、PrTPフィールドが予約したコード化を有するかどうかを検査する(判断ブロック226)。他の実施態様では、判断ブロック226を削除することが可能である。PrTPフィールドが予約したコード化を有する場合(判断ブロック226、「yes」レグ)、式プロセッサ42は、/スタック39Cおよび//スタック39D内の一致するノードに対する評価フィールドのうちの1つのビットを設定することが可能である(ブロック228)。PrTPフィールドが予約したコード化を持たない場合(判断ブロック226、「no」レグ)、式プロセッサ42は、評価フィールドを3に設定することが可能である(ブロック230)。いずれの場合も、属性名イベントの処理を終了することが可能である。一部の実施態様において、式プロセッサ42が属性値の一致を試みる場合、予約したコード化を使用することが可能である。他の実施態様では当該の一致を行わない場合があり、そのような実施態様ではブロック226および228を削除することが可能である。

10

【0114】

判断ブロック222、「no」レグが先行する場合、式プロセッサ42は、PrTPフィールドが解析時間述語がないこと、または1つの位置述語がないことを示すかどうかを判断する(判断ブロック232)。すなわち、式プロセッサ42は、PrTPフィールドが属性名を示すか否かを判断する。PrTPフィールドが、無しまたは位置にコード化されない場合(判断ブロック232、「no」レグ)、式プロセッサ42は、次の一致する/または//スタックに移動するか、またはそれ以上一致するエントリがなければ(それぞれ、判断ブロック234、「yes」および「no」レグ)処理を終了する。PrTPフィールドが、無しまたは位置にコード化される場合、式プロセッサ42は、式ノードが属性の子を有するかどうかを判断する(判断ブロック236)。式ノードが属性の子を持たない場合(判断ブロック236、「no」レグ)、式プロセッサ42は、式プロセッサ42は、次の一致する/または//スタックに移動するか、またはそれ以上一致するエントリがなければ(それぞれ、判断ブロック234、「yes」および「no」レグ)処理を終了する。式ノードが属性の子を有する場合(判断ブロック236、「yes」レグ)、式プロセッサ42は、属性の子を取り込み(ブロック238)、その属性の子を属性名(シリアル番号)と比較する。属性の子が属性名と一致し(判断ブロック240、「yes」レグ)、かつ属性の子ノードがリーフノードである場合(判断ブロック242、「yes」レグ)、式プロセッサ42は、属性名ノードを一致リスト39Gに出力する。属性の一致が検出されるか否かに関わらず、更なる属性の子ノードがある場合(すなわち、属性の子ノードのEOL表示は、レベルの終わりを示さない場合)、次いで式プロセッサ42は、属性の子を取り込み(ブロック238)、ブロック240乃至244を継続する(判断ブロック246、「yes」レグ)。それ以外ならば(判断ブロック246、「no」レグ)、式プロセッサ42は、次の一致する/または//スタックに移動するか、またはそれ以上一致するエントリがなければ(それぞれ、判断ブロック234、「yes」および「no」レグ)処理を終了する。

20

30

40

【0115】

図15は、要素クローズイベントに応えた、式プロセッサ42の一実施態様の演算を示す。要素クローズイベントは、要素開始タグの検出に応じて構文解析回路40によって発信することが可能である(したがって、その要素の全ての属性が、この要素に対する文書内に発見される)。要素クローズイベントに応えて、式プロセッサ42は、/スタック39C内のあらゆる一致するノードの属性の子に対して構文解析回路40が以前に識別した属性名を検査する。

【0116】

50

クローズする要素の親ノードがルートノードである場合（判断ブロック250、「yes」レグ）、更なる処理は行われぬ。クローズする要素の親ノードがルートノードでない場合（判断ブロック250、「no」レグ）、式プロセッサは、属性名を示すPrTPフィールドを有するエントリのためのノスタック39Cを検査する。一致するエントリが見つからない場合（判断ブロック254、「no」レグ）、処理を終了する。一致するエントリが見つかり（判断ブロック254、「yes」レグ）、かつその一致するエントリの評価フィールドが3でない場合（判断ブロック256、「no」レグ）も、処理を終了する。一致するエントリが見つかり（判断ブロック254、「yes」レグ）、かつその一致するエントリの評価フィールドが3である場合（判断ブロック256、「yes」レグ）、ブロック258で処理を継続する。

10

【0117】

式プロセッサ42は、一致する式ノードの属性の子を取り込む（ブロック258）。加えて、式プロセッサ42は、属性スタック39Fのための属性名を取り込む（ブロック260）。式プロセッサ42は、その属性名を比較する。一致が検出された場合（判断ブロック262、「yes」レグ）、属性ノードが一致リスト39Gに出力される（ブロック264）。いずれの場合も、属性スタック39Fの終わりに達してない場合（判断ブロック266、「no」レグ）、属性スタック39F内の次の属性に対してブロック260で処理を継続する。属性スタック39Fの終わりに達して（判断ブロック266、「yes」レグ）、かつ一致する式ノードの最後の属性の子が処理されていない場合（判断ブロック268、「no」レグ）、一致する式ノードの次の属性に対してブロック253で

20

【0118】

図12Aおよび12B、13、14Aおよび14B、および15の上記説明における種々の点において、フローチャートは一致リスト39Gへのノードの出力を参照することに留意されたい。ノードを出力する演算は、解析時間の式ツリー26内の一致するリーフノードに対応する式ノードの1つまたは複数のノードセット構造へのノードの挿入を含むことが可能である。ノードを出力する演算は、解析時間の式ツリー26のリーフノード内のテンプレートリストポインタが示すように、テンプレート番号（またはリスト）

30

【0119】

上述の種々の点において、式ノードは、ノスタック39Cまたはノスタック39D上にプッシュされるとみなされることに留意されたい。スタック39Cおよびスタック39Dへの式ノードのプッシュは、スタック（または、式的一致に使用される式ツリーエントリの一部）へのノードの式ツリーエントリ120のプッシュを含むことが可能である。必要に応じて、更なる情報をエントリ内に含むことが可能である（例、評価フィールドのような一致の進行を示す種々のステータス変数）。

【0120】

図16は、文書に対する解析した内容表39Bおよび一致リスト39Gの受信に応えた変換エンジン44の一実施態様の演算を示すフローチャートを示す。変換エンジン44は、ハードウェア内に実装することが可能であるので、そのフローチャートは、種々のブロックをハードウェア内で並行して、または必要に応じてハードウェア内でパイプライン化して実行することが可能であるが、ハードウェアの演算を表すことが可能である。

40

【0121】

各式に関して、変換エンジン44は、式のあらゆるランタイム部分を評価することが可能である（例、一実施例では、ランタイム述語、ブロック270）。例えば、一実施態様において、テンプレートリスト表38内のポインタは、ランタイム述語を評価するために変換エンジン44が実行する、命令表30内の命令を示すことが可能である。他の実施態

50

様では、ランタイム述語は、他の形態（例、解析時間の式ツリー 26 と同様のランタイム式ツリー）で識別することが可能である。ランタイム述語の評価に応じて、変換エンジン 44 は、ランタイム式を満たすノードセット構造からグループを選択することが可能である（ブロック 272）。変換エンジン 44 は、式（例、テンプレートリスト表 38 内のテンプレート本体のポインタによって命令をおくことが可能である）に対応する命令表 30 からの命令を実行することが可能である。その命令は、選択したグループ内のそれぞれのノード上で実行することが可能であり（ブロック 274）、その結果を出力生成器 46 に出力することが可能である（ブロック 276）。

【0122】

一実施態様において、変換エンジン 44 は、スタイルシートコンパイラ 20 が生成する命令を実行するように設計された、複数のハードウェアプロセッサを含むことが可能である。すなわち、プロセッサの命令セットを定義することが可能で、スタイルシートコンパイラ 20 は命令セット内に命令を生成することが可能である。いくつかの実施態様において、その命令セットは、XSLT 言語への拡張機能に適應するように設計される。変換エンジン 44 は、例えば、特定のノード上で実行する命令をプロセッサのうちの 1 つに送ることが可能であり、結果を生成するためにそのノード上でその命令を実行することが可能である。

10

【0123】

一実施態様において、ランタイム述語を評価するために実行する命令は、類似した述語（例、共通の接頭辞部分を有する述語）を同時に評価し、その述語を評価するためのノードの取り込みを最小限に抑えるように命令を出すことが可能である。例えば、一致するノードの後に続くノードに基づく述語は、互いにグループ化し、同時に評価することが可能である。

20

【0124】

場合によっては、スタイルシート内の変数および/またはパラメータは、式を使用して宣言することが可能であり、その後の命令は、その変数/パラメータを使用することが可能である。変数/パラメータを定義する式は、解析時間の式ツリー 26 内に含むことが可能なので、その式を式プロセッサ 42 によって評価する（または、その式がランタイム述語を含む場合は、部分的に評価する）ことが可能である。ランタイム述語は、他のランタイム述語と同様に評価されることが可能である。いくつかの実施態様において、スタイルシートコンパイラ 20 は、変数/パラメータを評価する前に、変換エンジン 44 によって命令が試みられる可能性を減じるために、その変数/パラメータを使用する命令のかなり前にその変数/パラメータを評価するための命令の指示を試みることが可能である。変換エンジン 44 は、例えば、変数/パラメータを使用し、変数/パラメータを評価する前に試みられる命令をおくことが可能な待ちキューを含むことが可能であり、その命令は、変数/パラメータが評価されるまで、やり直し、および待ちキュー内への置き換えが可能である。他の実施態様では、スタイルシートコンパイラ 20 は、種々の変数/パラメータに依存する命令を明示的に識別することが可能であり、変換エンジン 44 は、各命令の実行を試みる前に、当該の依存関係を検査することが可能である。さらに他の実施態様では、スタイルシートコンパイラ 20 は、その依存関係が満たされる前に所与の命令が実行されないようにするために、その命令を再配置することが可能である。例えば、スタイルシートコンパイラ 20 は、その命令の位相的にソートしたデータの依存関係グラフを構成することが可能であり、所与のレベルで各命令にグループ番号を割り当てることが可能である。変換エンジン 44 は、以前のグループの全ての命令が実行のために選択されるまで、実行のための所与のグループ番号を有する所与の命令を選択することができない場合もある。

30

40

【0125】

上述の種々のデータ構造への（および、種々のデータ構造内の）ポインタは、（示されるデータのアドレスを単独で指定する）全ポインタとするか、または必要に応じて、示されるデータを含む構造の基底アドレスからのオフセットとすることが可能である。

50

【 0 1 2 6 】

(式プロセッサ、更なる実施態様)

次に、式プロセッサ 4 2 の別の実施態様を、図 1 7 乃至 2 4 B に関して説明する。本実施態様は、幾分異なる解析時間の式ツリー構造を使用することが可能であり、更なる式を処理することが可能である。他の実施態様では、他のマークアップ言語を用いることが可能であるが、本実施例では XML のノード構造を使用する。図 1 7 乃至 2 4 B に示される実施態様は、あらゆるノード (例、要素、属性、処理命令、コメント、テキスト等) に適合することが可能であり、文書オーダーに見られるノードを使用して評価することが可能な述語を含む。フローチャートは、概して一致する文書ノードおよび式ノードを参照することが可能である。上述のように、当該の一致は、文書ノードと式ノードとの一致するシリアル番号を含むことが可能である。さらに、フローチャートは、一致リストに出力するノードを参照することが可能である。上述のように、一部の実施態様では、プレオーダー番号によってノードを一致リスト内に表すことが可能である。

10

【 0 1 2 7 】

図 1 7 は、解析時間の式ツリー 2 6 の別の実施態様を示すブロック図であり、エントリ 3 0 0 を含む。エントリ 3 0 0 は、図内のスペースの理由から図 1 7 において 2 行で示す。エントリ 3 0 0 は、解析時間の式ツリー 2 6 内の 1 つの式ノードに対応することが可能であるので、各式ノードに対するエントリ 3 0 0 に類似したエントリが存在し得る。

【 0 1 2 8 】

エントリ 3 0 0 は、トップレベルの型 (T L T) フィールドと、シリアル番号 (S / N) フィールドと、リーフノード (L N) フィールドと、レベルの終わり (E O L) フィールドと、式リストポインタ (X L P) フィールドと、テンプレートリストポインタ (T L P) フィールドと、述語型 (P r T P) フィールドと、図 1 0 に示すエントリ 1 2 0 に類似した述語データ (P r D T) フィールドとを含む。加えて、エントリ 3 0 0 は、ノード型 (N T) フィールドと、子記述子 (C D) フィールドと、P t r / フィールドと、P t r / / フィールドと、P t r / A t t r フィールドと、P t r / / A t t r フィールドと、P t r / P I フィールドと、P t r / / P I フィールドとを含むことが可能である。図 1 7 に示されるフィールドの順序は、メモリ内のフィールドの順序に対応していない場合があることに留意されたい。むしろ、エントリ 3 0 0 のフィールドは、単に解析時間の式ツリーエントリの一実施態様の内容を説明するように示されたものである。

20

30

【 0 1 2 9 】

エントリ 3 0 0 に対応する式ノードは、さまざまな子の式ノードを有することが可能である。エントリ 3 0 0 の C D フィールドは、その式ノードが、どの型の子の式ノードを有するかの表示を保存することが可能である。例えば、図 1 8 は、C D フィールドのコード化の一実施態様を示している表 3 0 2 を含む。図 1 8 に示す実施態様において、C D フィールドは、各子ノードの型およびその型の / または / / 子のためのビットを含むことが可能である。例えば、図示した実施態様には 6 つの子ノード (要素、属性、テキスト、コメント、処理命令 (P I)、および定数を有する処理命令 (P I - l i t e r a l)) がある。各型は、式ノードの / 子または / / 子のいずれかとするのが可能なので、本実施態様では、C D フィールドは 1 2 ビットより成る。対応するビットが設定されると、次いで式ノードは、所与の型 (および、 / または / /) の少なくとも 1 つの子式ノードを有する。例えば、式ノードが少なくとも 1 つの / 子要素ノードを有する場合、C D フィールドのビット 1 1 を設定することが可能である。式ノードが少なくとも 1 つの / / 子要素ノードを含む場合、C D フィールドのビット 1 0 を設定することが可能である。他の実施態様は、ビットの設定およびクリア状態の意味を逆にするか、またはいずれの所望のコード化を使用することが可能である。C D フィールドは、所与の式ノードが、式の一致処理の一部として、所与の型の子を有するかどうかを判断するために使用することが可能である。

40

【 0 1 3 0 】

N T フィールドは、エントリ 3 0 0 に対応する式ノードの型を識別するノード型を保存することが可能である。例えば、図 1 8 は、N T フィールドのための例示的なコード化を

50

示す表 304 を含む。例示した実施態様において、NTフィールドは、3ビットのコード化を含むことが可能であり、その二進値を表 304 の左欄に示す。右欄は、本実施態様のための種々のノード型（例、要素、属性、テキスト、コメント、PI、ノード、およびターゲットを有するPI）をリストしたものである。他の実施態様は、いずれの他のコード化を使用することが可能であり、示される型のあらゆるサブセットまたはスーパーセットをサポートすることが可能である。

【0131】

図示の実施態様では、エン트리 300 は、子ノードエントリに対して解析時間の式ツリー 26 内に 6 のポイントを含むことが可能である。Ptr / Attr ポインタは、式ノードのノードの子である属性ノードを示すことが可能である。ノードの子ノードは、Ptr / Attr ポインタによって示されるエントリから始まる解析時間の式ツリー 26 内にグループ化することが可能である。Ptr / / Attr ポインタは、式ノードのノードの子である属性ノードを示すことが可能である。ノードの子ノードは、Ptr / / Attr ポインタによって示されるエントリから始まる解析時間の式ツリー 26 内にグループ化することが可能である。Ptr / PI ポインタは、式ノードのノードの子であるPIノードを示すことが可能である。ノードの子ノードは、Ptr / PI ポインタによって示されるエントリから始まる解析時間の式ツリー 26 内にグループ化することが可能である。Ptr / / PI ポインタは、式ノードのノードの子である属性ノードを示すことが可能である。ノードの子ノードは、Ptr / / PI ポインタによって示されるエントリから始まる解析時間の式ツリー 26 内にグループ化することが可能である。式ノードの他のノード（すなわち、属性またはPIノードでない）は、（式ノードのノードに対する）Ptr / ポインタおよび（式ノードのノードに対する）Ptr / / ポインタで、解析時間の式ツリー 26 内にグループ化される。

【0132】

図示した実施態様は、属性の子、処理命令の子、および残りの子を置くために独立した一連のポイントを提供するが、他の実施態様では、異なる一連のポイントを実装することが可能である。例えば、一実施態様は、一連のポイント、すなわち、全てのノードおよび全てのノードの子をそれぞれ置くためのノードポイントおよびノードノードポイント、を1つだけ含むことが可能である。他の実施態様は、各ノード型に対してノードおよびノードノードポイントを実装するか、または必要に応じて他の形態のポイントでノードをグループ化することが可能である。

【0133】

PrTPフィールドは、本実施態様では、図 18 の表 306 に示すように、コード化を有することが可能である。本実施態様において、述語型には、述語なし（または、より詳しくは、解析時間の評価可能な述語なし）、位置述語、要素の子の述語、属性名述語、名前述語を有するPIノードテスト、ノードテスト述語、コメントノードテスト述語、PIノードテスト述語、およびテキストノード述語が挙げられる。ノードテスト述語は、式ノードの子または子孫として（いずれの型の）ノードがあることをシンプルにテストすることが可能である。コメントノードテスト、PIノードテスト（名前なし）、およびテキストノードテスト述語は、所与の型のノードの存在をテストすることが可能である。名前によるPIノードテストは、名前を有するPIターゲットを伴うPIノードの存在をテストすることが可能である。他の実施態様は、いずれの他のコード化を使用することが可能であり、示される述語型のあらゆるサブセットまたはスーパーセットをサポートすることが可能である。

【0134】

いくつかの実施態様において、式リスト表 36 およびテンプレートリスト表 38 は、図 9 に関する上述のものと同様の構造を有することが可能である。加えて、一実施態様では、各テンプレートリスト表のエントリは、（該当する場合）どの子ノードがそのテンプレートリスト表のエントリに対応するテンプレート一致式内で参照されるのかを識別するノードIDを含むことが可能である。例えば、テンプレートリスト表 38 は、所与の1つまたは複数の式に一致する最後の要素に基づいて構成することが可能である（すなわち、そ

10

20

30

40

50

の表は、式がその要素の非要素の子を含んでいても、その要素と一致する、最後の要素のための式のリストを含むことが可能である)。属性、テキスト、コメント、またはその要素ノードのための式リスト内の一部の式に含むことが可能な処理命令の子ノードを、ノードIDで識別することが可能である。ノードIDは、子ノードの型を識別することが可能である。加えて、属性および処理命令に関して、ノードIDは、ノード内の他の属性および処理命令のそれぞれに関する位置を識別することが可能である。ゼロのノードIDは、対応する式のための要素ノードの子がないことを示すことが可能である。

【0135】

本実施態様では、構文解析回路40は、式プロセッサ42のための次のようなイベントを生成することができる：トップレベルの要素開始イベント、要素開始イベント、要素終了イベント、属性名イベント、テキストイベント、コメントイベント、処理命令イベント、および構成イベント。構成イベントは、式プロセッサ42のための所望のスタイルシート/文書コンテキストを確立するために生成することが可能である。

10

【0136】

トップレベルの要素開始イベントは、ルートの子である要素の開始を識別することが可能である。ルート以外の他の要素の子である要素の開始は、要素開始イベントによって識別される。例えば、XMLを用いた実施態様において、要素開始イベントは、要素開始タグを検出したことを示すことが可能である。各イベントは、要素のシリアル番号を含むことが可能である。いくつかの実施態様において、イベントは、要素および/または要素のプレオーダー番号の子の位置を含むことも可能である。これらのイベントによって、式プロセッサ42は、解析時間の式ツリー26内の式への要素の一致を試みる事が可能となる。

20

【0137】

要素終了イベントは、(例えば、XMLを用いた実施態様において要素終了タグが検出されている)要素の終わりの検出に応じて生成することが可能である。式プロセッサ42は、その要素終了イベントに応えた要素によって一致したあらゆる式の分岐を消去することが可能である。

【0138】

属性名イベントは、属性名の検出に応じて生成することが可能である。属性名イベントは、属性名のシリアル番号を含むことが可能であり、いくつかの実施態様では、要素が対応する属性のプレオーダー番号を含むことが可能である。式プロセッサ42は、属性名イベントに応じて式ツリー内の式への属性名の一致を試みる事が可能である。

30

【0139】

テキストイベントは、文書内のテキストの検出に応じて生成することが可能である。テキストイベントは、対応する要素のプレオーダー番号を含むことが可能であり、式プロセッサ42に、テキストノードテストまたはテキスト式ノードの一致のための式ツリー内の式を検査させることが可能である。同様に、コメントイベントは、文書内のコメントノードの検出に応じて生成することが可能である。コメントイベントは、対応する要素のプレオーダー番号を含むことが可能であり、式プロセッサ42に、コメントノードテストまたはコメント式ノードの一致のための式ツリー内の式を検査させることが可能である。

40

【0140】

処理命令イベントは、処理命令の検出に応じて生成することが可能である。処理命令イベントは、処理命令のシリアル番号を含むことが可能であり、いくつかの実施態様では、対応する要素のプレオーダー番号を含むことが可能である。式プロセッサ42は、定数の有無にかかわらず、処理命令ノードテストへの、または処理命令の式ノードへの処理命令の一致を試みる事が可能である。

【0141】

図19A - 図19Bは、(要素開始およびトップレベルの要素開始イベントの両方を含む)要素開始イベントに応じて、図17に示す解析時間の式ツリー26を使用した、式プロセッサ42の一実施態様の演算を示すフローチャートを示す。概して、処理には、要素

50

が / スタック 39C および / / スタック 39D 上の式ノードの述語を満たすか、または要素が当該の式ノードの要素の子であるかどうかの検査とともに、要素との一致のための相対的なトップレベルの式ノード（および、イベントがトップレベルの要素開始イベントである場合の非相対的なトップレベルの式ノード）の検査を含むことが可能である。

【0142】

式プロセッサ 42 は、 / および / / スタックポインタをポインタスタック 39E にプッシュすることが可能である（ブロック 310）。より詳しくは、図示した実施態様において、ポインタスタック 39E は、それぞれ / スタックポインタおよび / / スタックポインタのための、 / Ptr スタックおよび / / Ptr スタックを含むことが可能である。この代わりに、他の実施態様では、両ポインタを同じスタック上にプッシュすることが可能である。ポインタは、 / スタック 39C および / / スタック 39D の状態を要素の検出前の状態に回復する（別の要素の、終了する要素の前に一致するような式への一致を可能にする）ために、対応する要素終了イベントが終了したときに、続いてポップさせることが可能である。要素開始イベントが識別する要素は、図 19A - 19B の記述では、より簡潔に要素と称する。トップレベルの要素開始イベントを受信したかどうか（判断ブロック 312）に基づいて、式プロセッサ 42 は、各トップレベルの式ノード（例、絶対的および祖先のトップレベルの式ノードを含む）を検査する（ブロック 316）か、または一致のための解析時間の式ツリーの各相対的なトップレベルの式ノードを検査する（ブロック 314）ことが可能である。すなわち、そのイベントがトップレベルの要素開始イベントである場合（判断ブロック 312、「yes」レグ）、式プロセッサ 42 は、ルートノードの子のための絶対的なトップレベルのノードに対しても一致を検出することが可能であるので、解析時間の式ツリー 26 内の各トップレベルの式ノードを検査することが可能である（ブロック 316）。一方で、そのイベントがトップレベルの要素開始イベントでない場合（判断ブロック 312、「no」レグ）、式プロセッサ 42 は、絶対的なトップレベルのノードに対して一致を検出しない可能性があるため、解析時間の式ツリー 26 内の各相対的なトップレベルの式ノードを検査することが可能である（ブロック 314）。

【0143】

トップレベルのノードのいずれかに対して一致が検出されない場合（判断ブロック 318、「no」レグ）、フローチャートは、図 19B の参照 D を継続する。一致が検出される（判断ブロック 318、「yes」レグ）、かつ式ノードがリーフノードである場合（判断ブロック 320、「yes」レグ）、解析時間の式ツリー 26 の式ノードのエントリ内の XLP および TLP ポインタによって示される 1 つまたは複数の式および / または 1 つまたは複数のテンプレートに対応する一致リストに、要素ノードが出力される（ブロック 322）。式プロセッサ 42 は、一致した式ノードが / または / / 子のいずれかを有するかどうかを判断し（それぞれ、判断ブロック 324 および 326）、一致した式ノードが / または / / 子のいずれかを有する場合（それぞれ、判断ブロック 328 および 330）、それぞれ / スタック 39C および / または / / スタック 39D 上の一致した式をプッシュする。式プロセッサ 42 は、例えば、 / または / / 子があるかどうかを検出するために、解析時間の式ツリー 26 内の式ノードのエントリの CD フィールドを使用することが可能である。加えて、 / スタック 39C および / / スタック 39D は、（解析時間の式ツリーエントリ 300 内の PrTP および PrDT フィールドを経て示される）解析時間述語の一致の管理に使用する評価フィールドを含むことが可能である。（ゼロではない PrTP フィールドによって示されるような）解析時間述語がある場合、評価フィールドを 0 に設定することが可能である。それ以外ならば、評価フィールドを 1 に設定することが可能である。フローチャートは、図 19B の参照 D を継続する。

【0144】

図 19B の参照 D で、 / および / / スタックは、要素が（ / または / / スタックエントリのうちの 1 つに保存される）以前に一致した式ノードの子または述語に一致するかどうかを検索される。 / および / / スタックが空である場合（判断ブロック 332、「yes

10

20

30

40

50

」レグ)、この要素に対する一致は終了する。それ以外ならば(判断ブロック332、「no」レグ)、スタックエントリが選択される。エントリ内の評価フィールドが1に設定される場合、選択したスタックエントリ内の対応する式ノードは、述語を持たないか、またはその述語が以前に解析した文書ノードによって満たされる。したがって(判断ブロック334、「eval=1」レグ)、式プロセッサは、要素が要素の子のいずれかに一致するかどうかを判断するために、選択したスタックエントリ内の式ノードのあらゆる要素の子を検査することが可能である。/要素の子も、//要素の子も、考慮することが可能である。より詳しくは、式ノードが(例えば、解析時間の式ツリーエントリ300のCDフィールド内に示されるように)要素の子を持たない場合(判断ブロック336、「no」レグ)、その要素に対する一致処理は終了する。この代わりに、式プロセッサ42は、
10 処理のために次のスタックのエントリへ進むことが可能である(ブロック362)。式ノードが要素の子を有する場合(判断ブロック336、「yes」レグ)、式プロセッサ42は、式ノードの第1の要素の子を取り込む(ブロック338)。例えば、エントリのPtr/またはPtr//ポインタは、(子の式ツリーエントリ内のNT型のフィールドとともに)要素の子を置くために使用することが可能である。子要素ノードが要素と一致し(判断ブロック340、「yes」レグ)、かつ子要素ノードがリーフノードである場合(判断ブロック342、「yes」レグ)、そのノードが一致リストに出力される(ブロック344)。加えて、一致した子要素ノードがそれぞれ/または//子を有する場合(判断ブロック346および348)、その一致した子要素ノードをそれぞれ/スタック39Cまたは//スタック39Dにプッシュし(ブロック350および352)、評価フィールドは、
20 上述のように、ブロック324乃至330に関して初期化される。子要素ノードが要素と一致するか否かに関わらず、式プロセッサ42は、最後の要素の子が処理されたかどうかを判断する(判断ブロック354)。一致しなければ、次の子要素ノードを取り込み(ブロック338)、同様に処理する。子要素ノードが現在のスタックエントリの最後の要素の子である場合(判断ブロック354、「yes」レグ)、式プロセッサ42は、次のスタックエントリに進み(ブロック362)、要素子および述語の両方を処理することが可能である。

【0145】

選択したスタックエントリのPrTPフィールドが4であるか、または要素の子である場合、要素は、その選択したスタックエントリ内の式ノードの述語を満たすことが可能である。したがって(判断ブロック334、「PrTP=4」レグ)、式プロセッサ42は、要素のシリアル番号を選択したスタックエントリのPrDTフィールドと比較することが可能である(ブロック356)。要素がPrDTフィールドに一致する場合(判断ブロック358、「yes」レグ)、その要素は述語を満たし、式プロセッサ42は、選択したスタックエントリ内の評価フィールドを1に設定する(ブロック360)。いずれの場合も、式プロセッサ42は、次のスタックエントリに進むことが可能である(ブロック362)。
30

【0146】

所与のスタックエントリは、ゼロの評価フィールドおよび4ではないPrTPフィールドを有することが可能であることに留意されたい。そのような場合、式プロセッサ42は、
40 次のスタックエントリに進むことが可能である(ブロック362)。

【0147】

図20は、要素終了イベントに応じて、図17に示す解析時間の式ツリー26を使用した、式プロセッサ42の一実施態様の演算を示すフローチャートを示す。

【0148】

要素終了イベントが文書のルートノードに対するものである場合(判断ブロック370、「yes」レグ)、その文書は完了している(ブロック372)。式プロセッサ42は、スタック39C-39Fをクリアすることが可能である。要素終了イベントが、文書のルートノードに対するものでない場合(判断ブロック370、「no」レグ)、式プロセッサ42は、Ptrスタック39Eからの/および//スタックポインタをポップするこ
50

とが可能である（ブロック 374）。要素は終わっているのに、要素の全ての子はすでに解析されている。したがって、その要素に対応する / および / / スタックエントリ内のいずれのエントリ（すなわち、その要素のシリアル番号を有するエントリ）も、引き続き検出されるノードと一致させることができない。事実上、要素に対する要素開始イベントが検出されたときにプッシュされた / および / / スタックポインタの回復は、終了要素に対応する / スタック 39C および / / スタック 39D 上にエントリをポップさせ、それらの状態をその要素を処理する前の状態に回復させる（その状態は、次の検出した要素を処理するための適切な状態であり得る）。

【0149】

図 21A - 21B は、属性名イベントに応じて、図 17 に示す解析時間の式ツリー 26 を使用した、式プロセッサ 42 の一実施態様の演算を示すフローチャートを示す。属性名イベントによって識別される属性は、図 21A - 21B の記述において「属性」と称する場合がある。概して、処理には、属性が / スタック 39C および / / スタック 39D 上の式ノードの述語を満たすか、または属性が当該の式ノードの属性の子であるかどうかの検査とともに、属性との一致のための相対的なトップレベルの式ノードの検査を含むことが可能である。

【0150】

属性の親ノードがルートノードである場合（判断ブロック 382、「yes」レグ）、（ルートノードは属性を持たないので）実行する更なる処理はない。一方で、属性の親ノードがルートノードでない場合（判断ブロック 382、「no」レグ）、式プロセッサ 42 は継続する。

【0151】

式プロセッサ 42 は、属性に対する一致のための相対的なトップレベルの式ノードのそれぞれを検査することが可能である（ブロック 384）。所与の相対的なトップレベルの式ノードとの一致があり（判断ブロック 386、「yes」レグ）、かつノードがリーフノードである場合（判断ブロック 388、「yes」レグ）、属性ノードが一致リスト 39G に出力される（ブロック 390）。一致の有無に関わらず、トップレベルの式ノードが空になる（判断ブロック 392、「no」レグ）まで、次の相対的なトップレベルの式ノードによって処理を継続することが可能である。トップレベルの式ノードが空になると（判断ブロック 392、「yes」レグ）、処理は図 21B の参照 E を継続する。

【0152】

図 21B の参照 E から継続し、 / および / / スタックは、属性が（ / または / / スタックエントリのうちの 1 つに保存される）以前に一致した式ノードの子または述語に一致するかどうかを検索される。 / および / / スタックが空である場合（判断ブロック 394、「yes」レグ）、この属性に対する一致は終了する。それ以外ならば（判断ブロック 394、「no」レグ）、スタックエントリが選択される。エントリ内の評価フィールドが 1 に設定される場合、選択したスタックエントリ内の対応する式ノードは、述語を持たないか、またはその述語が以前に解析した文書ノードによって満たされる。したがって（判断ブロック 334、「eval = 1」レグ）、式プロセッサ 42 は、属性が属性の子のいずれかに一致するかどうかを判断するために、選択したスタックエントリ内の式ノードのあらゆる属性の子を検査することが可能である。 / 属性の子も、 / / 属性の子も、考慮することが可能である。より詳しくは、選択したスタックエントリ内の式ノードが（例えば、解析時間の式ツリーエントリ 300 の CD フィールドに示されるように）属性の子を持たない場合（判断ブロック 398、「no」レグ）、式プロセッサ 42 は、スタック内の最後の式ノードが処理されたかどうかを判断することが可能である（判断ブロック 400）。その場合（判断ブロック 400、「yes」レグ）、属性に対する処理は終了する。それ以外ならば（判断ブロック 400、「no」レグ）、式プロセッサ 42 は、処理のために次のスタックエントリ（ブロック 410）に進むことが可能である。選択したスタックエントリ内の式ノードが属性の子を有する場合（判断ブロック 398、「yes」レグ）、式プロセッサ 42 は、そのエントリの最初の属性の子を取り込む（ブロック 402）

10

20

30

40

50

。例えば、エントリの P t r / A t t r または P t r / / A t t r ポインタは、属性の子を置くために使用することが可能である。子属性ノードが属性に一致する場合（ブロック 404、「y e s」レグ）、そのノードが一致リストに出力される（ブロック 406）。子属性ノードが属性と一致するか否かに関わらず、式プロセッサ 42 は、最後の属性の子が処理されたかどうかを判断する（判断ブロック 408）。そうでない場合には、次の属性の子ノードが取り込まれ（ブロック 402）、同様に処理される。子属性ノードが選択したスタックエントリ内の式ノードの最後の属性の子である場合（判断ブロック 408、「y e s」レグ）、式プロセッサ 42 は、最後の式ノードが処理されたかどうかを判断することが可能であり（判断ブロック 400）、次のスタックエントリに進むか（ブロック 410）、またはその結果、処理を終了することが可能である。

10

【0153】

選択したスタックエントリの P r T P フィールドが 5 であるか、または属性名である場合、属性は、その選択したスタックエントリ内の式ノードの述語を満たすことが可能である。したがって（判断ブロック 396、「P r T P = 5」レグ）、式プロセッサ 42 は、属性のシリアル番号を選択したスタックエントリの P r D T フィールドと比較することが可能である（ブロック 412）。属性が P r D T フィールドに一致する場合（判断ブロック 414、「y e s」レグ）、その属性は述語を満たし、式プロセッサ 42 は、選択したスタックエントリの評価フィールドを 1 に設定する（ブロック 416）。いずれの場合も、式プロセッサ 42 は、次のスタックエントリに進むことが可能である（ブロック 410）。式プロセッサ 42 は、必要に応じて、進む前に残りの式ノードがあるかどうかを判断

20

【0154】

所与のスタックエントリは、ゼロの評価フィールドおよび 5 ではない P r T P フィールドを有することが可能であることに留意されたい。そのような場合、式プロセッサ 42 は、次のスタックエントリに進むことが可能である（ブロック 410）。

【0155】

図 22A - 22B は、テキストイベントに応じて、図 17 に示す解析時間の式ツリー 26 を使用した、式プロセッサ 42 の一実施態様の演算を示すフローチャートを示す。テキストイベントによって識別されるテキストノードは、図 22A - 22B の記述において、より簡潔に「テキストノード」と称する場合がある。概して、処理には、テキストノードが / スタック 39C および / / スタック 39D 上の式ノードの述語を満たすか、またはテキストノードが当該の式ノードのテキストの子であるかどうかの検査とともに、テキストノードとの一致のための相対的なトップレベルの式ノードの検査を含むことが可能である。

30

【0156】

テキストノードの親ノードがルートノードである場合（判断ブロック 420、「y e s」レグ）、実行する更なる処理はない。一方で、テキストノードの親ノードがルートノードでない場合（判断ブロック 420、「n o」レグ）、式プロセッサ 42 は継続する。

【0157】

式プロセッサ 42 は、テキストノードに対する一致のための相対的なトップレベルの式ノードのそれぞれを検査することが可能である（ブロック 422）。所与の相対的なトップレベルの式ノードとの一致がある場合（判断ブロック 424、「y e s」レグ）、テキストノードが一致リスト 39G に出力される（ブロック 426）。一致の有無に関わらず、トップレベルの式ノードが空になる（判断ブロック 428、「n o」レグ）まで、次の相対的なトップレベルの式ノードによって処理を継続することが可能である。トップレベルの式ノードが空になると（判断ブロック 428、「y e s」レグ）、処理は図 22B の参照 F を継続する。

40

【0158】

図 22B の参照 F から継続し、/ および / / スタックは、テキストノードが（/ または / / スタックエントリのうちの 1 つに保存される）以前に一致した式ノードの子または述

50

語に一致するかどうかを確認するために検索される。ノおよびノノスタックが空である場合（判断ブロック430、「yes」レグ）、このテキストノードに対する一致は終了する。それ以外ならば（判断ブロック430、「no」レグ）、スタックエントリが選択される。エントリ内の評価フィールドが1に設定される場合、選択したスタックエントリ内の対応する式ノードは、述語を持たないか、またはその述語が以前に解析した文書ノードによって満たされる。したがって（判断ブロック432、「eval=1」レグ）、式プロセッサ42は、テキストノードがテキストの子のいずれかに一致するかどうかを判断するために、選択したスタックエントリ内の式ノードのあらゆるテキストの子を検査することが可能である。より詳しくは、式ノードが（例えば、解析時間の式ツリーエントリ300のCDフィールドに示されるように）テキストの子を持たない場合（判断ブロック434、「no」レグ）、式プロセッサ42は、スタック内の最後の式ノードが処理されたかどうかを判断することが可能である（判断ブロック446）。その場合（判断ブロック446、「yes」レグ）、テキストノードに対する処理は終了する。それ以外ならば（判断ブロック446、「no」レグ）、式プロセッサ42は、処理のために次のスタックエントリ（ブロック448）に進むことが可能である。式ノードがテキストの子を有する場合（判断ブロック434、「yes」レグ）、式プロセッサ42は、その式ノードの最初のテキストの子を取り込む（ブロック436）。例えば、エントリのPtrノまたはPtrノノポインタは、（各子ノード内のNTフィールドとともに）テキストの子を置くために使用することが可能である。子テキストノードがテキストノードに一致する場合（ブロック438、「yes」レグ）、そのノードが一致リストに出力される（ブロック440）。子テキストノードがテキストノードと一致するか否かに関わらず、式プロセッサ42は、最後のテキストの子が処理されたかどうかを判断する（判断ブロック442）。そうでない場合には、次のテキストの子ノードが取り込まれ（ブロック436）、同様に処理される。子テキストノードが選択したスタックエントリ内の式ノードの最後のテキストの子である場合（判断ブロック442、「yes」レグ）、式プロセッサ42は、最後の式ノードが処理されたかどうかを判断することが可能であり（判断ブロック446）、次のスタックエントリに進むか（ブロック448）、またはその結果、処理を終了することが可能である。

【0159】

選択したスタックエントリのPrTPフィールドが8（ノードテスト）であるか、またはB（テキストノードテスト）である場合、テキストノードは、その選択したスタックエントリ内の式ノードの述語を満たす（判断ブロック432、「PrTP=8またはB」レグ）。したがって、式プロセッサ42は、選択したスタックエントリ内の評価フィールドを1に設定する（ブロック444）。式プロセッサ42は、次のスタックエントリに進むことが可能である（ブロック448）。式プロセッサ42は、必要に応じて、進む前に残りの式ノードがあるかどうかを判断することが可能である（判断ブロック446）。

【0160】

所与のスタックエントリは、ゼロの評価フィールドおよび8またはBではないPrTPフィールドを有することが可能であることに留意されたい。そのような場合、式プロセッサ42は、次のスタックエントリに進むことが可能である（ブロック448）。

【0161】

図23A-23Bは、コメントイベントに応じて、図17に示す解析時間の式ツリー26を使用した、式プロセッサ42の一実施態様の演算を示すフローチャートを示す。コメントイベントによって識別されるコメントノードは、図23A-23Bの記述においてより簡潔に「コメントノード」と称する場合がある。概して、処理には、コメントノードがノスタック39Cおよびノスタック39D上の式ノードの述語を満たすか、またはコメントノードが当該の式ノードのコメントの子であるかどうかの検査とともに、コメントノードとの一致のための相対的なトップレベルの式ノードの検査を含むことが可能である。

【0162】

コメントノードの親ノードがルートノードである場合（判断ブロック450、「yes」

」レグ)、実行する更なる処理はない。一方で、コメントノードの親ノードがルートノードでない場合(判断ブロック450、「no」レグ)、式プロセッサ42は継続する。

【0163】

式プロセッサ42は、コメントノードに対する一致のための相対的なトップレベルの式ノードのそれぞれを検査することが可能である(ブロック452)。所与の相対的なトップレベルの式ノードとの一致がある場合(判断ブロック454、「yes」レグ)、コメントノードが一致リスト39Gに出力される(ブロック456)。一致の有無に関わらず、トップレベルの式ノードが空になる(判断ブロック458、「no」レグ)まで、次の相対的なトップレベルの式ノードによって処理を継続することが可能である。トップレベルの式ノードが空になると(判断ブロック458、「yes」レグ)、処理は図23Bの参照Gを継続する。

10

【0164】

図23Bの参照Gから継続し、/および//スタックは、コメントノードが(/または//スタックエントリのうちの1つに保存される)以前に一致した式ノードの子または述語に一致するかどうかを確認するために検索される。/および//スタックが空である場合(判断ブロック460、「yes」レグ)、このコメントノードに対する一致は終了する。それ以外ならば(判断ブロック460、「no」レグ)、スタックエントリが選択される。エントリ内の評価フィールドが1に設定される場合、選択したスタックエントリ内の対応する式ノードは、述語を持たないか、またはその述語が以前に解析した文書ノードによって満たされる。したがって(判断ブロック462、「eval=1」レグ)、式プロセッサ42は、コメントノードがコメントの子のいずれかに一致するかどうかを判断するために、選択したスタックエントリ内の式ノードのあらゆるコメントの子を検査することが可能である。より詳しくは、式ノードが(例えば、解析時間の式ツリーエントリ300のCDフィールドに示されるように)コメントの子を持たない場合(判断ブロック464、「no」レグ)、式プロセッサ42は、スタック内の最後の式ノードが処理されたかどうかを判断することが可能である(判断ブロック476)。その場合(判断ブロック476、「yes」レグ)、コメントノードに対する処理は終了する。それ以外ならば(判断ブロック476、「no」レグ)、式プロセッサ42は、処理のために次のスタックエントリ(ブロック478)に進むことが可能である。式ノードがコメントの子を有する場合(判断ブロック464、「yes」レグ)、式プロセッサ42は、その式ノードの最初のコメントの子を取り込む(ブロック456)。例えば、エントリのPtr/またはPtr//ポインタは、(各子ノード内のNTフィールドとともに)コメントの子を置くために使用することが可能である。子コメントノードがコメントノードに一致する場合(ブロック468、「yes」レグ)、そのノードが一致リストに出力される(ブロック470)。子コメントノードがコメントノードと一致するか否かに関わらず、式プロセッサ42は、最後のコメントの子が処理されたかどうかを判断する(判断ブロック472)。そうでない場合には、次のコメントの子ノードが取り込まれ(ブロック466)、同様に処理される。子コメントノードが選択したスタックエントリ内の式ノードの最後のコメントの子である場合(判断ブロック472、「yes」レグ)、式プロセッサ42は、最後の式ノードが処理されたかどうかを判断することが可能であり(判断ブロック476)、次のスタックエントリに進むか(ブロック478)、またはその結果、処理を終了することが可能である。

20

30

40

【0165】

選択したスタックエントリのPrTPフィールドが8(ノードテスト)であるか、または9(コメントノードテスト)である場合、コメントノードは、その選択したスタックエントリ内の式ノードの述語を満たす(判断ブロック462、「PrTP=8または9」レグ)。したがって、式プロセッサ42は、選択したスタックエントリ内の評価フィールドを1に設定する(ブロック474)。式プロセッサ42は、次のスタックエントリに進むことが可能である(ブロック478)。式プロセッサ42は、必要に応じて、進む前に残りの式ノードがあるかどうかを判断することが可能である(判断ブロック476)。

50

【 0 1 6 6 】

所与のスタックエントリは、ゼロの評価フィールドおよび8または9ではないP r T P フィールドを有することが可能であることに留意されたい。そのような場合、式プロセッサ42は、次のスタックエントリに進むことが可能である（ブロック478）。

【 0 1 6 7 】

図24A - 24Bは、処理命令イベントに応じて、図17に示す解析時間の式ツリー26を使用した、式プロセッサ42の一実施態様の演算を示すフローチャートを示す。処理命令イベントにおいて識別される処理命令ノードは、図24A - 24Bの記述においてより簡潔に「処理命令ノード」または「PIノード」と称する場合がある。概して、処理には、PIノードがノスタック39Cおよびノスタック39D上の式ノードの述語を満たすか、またはPIノードが当該の式ノードのPIの子であるかどうかの検査とともに、PIノードとの一致のための相対的なトップレベルの式ノードの検査を含むことが可能である。

10

【 0 1 6 8 】

PIノードの親ノードがルートノードである場合（判断ブロック480、「yes」レグ）、実行する更なる処理はない。一方で、PIノードの親ノードがルートノードでない場合（判断ブロック480、「no」レグ）、式プロセッサ42は継続する。

【 0 1 6 9 】

式プロセッサ42は、PIノードに対する一致のための相対的なトップレベルの式ノードのそれぞれを検査することが可能である（ブロック482）。所与の相対的なトップレベルの式ノードとの一致がある場合（判断ブロック484、「yes」レグ）、PIノードが一致リスト39Gに出力される（ブロック486）。一致の有無に関わらず、トップレベルの式ノードが空になる（判断ブロック488、「no」レグ）まで、次の相対的なトップレベルの式ノードによって処理を継続することが可能である。トップレベルの式ノードが空になると（判断ブロック488、「yes」レグ）、処理は図24Bの参照Hを継続する。

20

【 0 1 7 0 】

図24Bの参照Hから継続し、ノおよびノスタックは、PIノードが（ノまたはノスタックエントリのうちの1つに保存される）以前に一致した式ノードの子または述語に一致するかどうかを確認するために検索される。ノおよびノスタックが空である場合（判断ブロック490、「yes」レグ）、このPIノードに対する一致は終了する。それ以外ならば（判断ブロック490、「no」レグ）、スタックエントリが選択される。エントリ内の評価フィールドが1に設定される場合、選択したスタックエントリ内の対応する式ノードは、述語を持たないか、またはその述語が以前に解析した文書ノードによって満たされる。したがって（判断ブロック492、「eval=1」レグ）、式プロセッサ42は、PIノードがPIの子のいずれかに一致するかどうかを判断するために、選択したスタックエントリ内の式ノードのあらゆるPIの子を検査することが可能である。より詳しくは、選択したエントリ内の式ノードが（例えば、解析時間の式ツリーエントリ300のCDフィールドに示されるように）PIの子を持たない場合（判断ブロック494、「no」レグ）、式プロセッサ42は、スタック内の最後の式ノードが処理されたかどうかを判断することが可能である（判断ブロック512）。その場合（判断ブロック512、「yes」レグ）、PIノードに対する処理は終了する。それ以外ならば（判断ブロック512、「no」レグ）、式プロセッサ42は、処理のために次のスタックエントリ（ブロック514）に進むことが可能である。選択したエントリ内の式ノードがPIの子を有する場合（判断ブロック494、「yes」レグ）、式プロセッサ42は、その式ノードの最初のPIの子を取り込む（ブロック496）。例えば、エントリのPtr/PIまたはPtr//PIポインタは、PIの子を置くために使用することが可能である。子PIノードがPIノードに一致する場合（ブロック498、「yes」レグ）、そのノードが一致リストに出力される（ブロック500）。子PIノードがPIノードと一致するかどうかに関わらず、式プロセッサ42は、最後のPIの子が処理されたかどうかを判断する

30

40

50

(判断ブロック502)。そうでない場合には、次のPIの子ノードが取り込まれ(ブロック496)、同様に処理される。子PIノードが選択したスタックエントリ内の式ノードの最後のPIの子である場合(判断ブロック502、「yes」レグ)、式プロセッサ42は、最後の式ノードが処理されたかどうかを判断することが可能であり(判断ブロック512)、次のスタックエントリに進むか(ブロック514)、またはその結果、処理を終了することが可能である。

【0171】

選択したスタックエントリのPrTPフィールドが8(ノードテスト)であるか、またはA(PIノードテスト)である場合、PIノードは、その選択したスタックエントリ内の式ノードの述語を満たす。したがって、式プロセッサ42は、選択したスタックエントリの評価フィールドを1に設定する(ブロック510)。式プロセッサ42は、次のスタックエントリに進むことが可能である(ブロック514)。式プロセッサ42は、必要に応じて、進む前に残りの式ノードがあるかどうかを判断することが可能である(判断ブロック512)。

【0172】

選択したスタックエントリのPrTPフィールドが6(名前を有するPIノードテスト)である場合、PIノードは、PIノードのPITargetがPrDTフィールドに一致する場合に述語を満たす。式プロセッサ42は、PITargetをPrDTフィールドと比較する(ブロック506)。一致が検出された場合(判断ブロック508、「yes」レグ)、式プロセッサ42は、選択したエントリの評価フィールドを1に設定する(ブロック510)。式プロセッサ42は、次のスタックエントリに進むことが可能である(ブロック514)。式プロセッサ42は、必要に応じて、進む前に残りの式ノードがあるかどうかを判断することが可能である(判断ブロック512)。

【0173】

所与のスタックエントリは、ゼロの評価フィールドおよび6、8またはAではないPrTPフィールドを有することが可能であることに留意されたい。そのような場合、式プロセッサ42は、次のスタックエントリに進むことが可能である(ブロック514)。

【0174】

いくつかの実施態様では、式プロセッサ42をパイプライン化することが可能であることに留意されたい。例えば、その後のパイプライン段階でノードの比較が生じることがあり、次いでそれらのノードを取り込む(および、述語を有するノードに対して評価フィールドを比較する)。このような実施態様では、ノードおよびノードスタックエントリは、評価フィールドに対する潜在的変化がパイプライン内にある場合に設定され得る、インプログレスビット(in-progress bit)を含むことが可能である。設定されると、インプログレスビットは、以降のイベントが比較が行われる前に評価フィールドを読み込まないように、そのエントリがビジーであることを示すことが可能である。

【0175】

上述の種々の点において、式ノードは、ノードスタック39Cまたはノードスタック39D上にプッシュされるとみなされることに留意されたい。スタック39C-スタック39Dへの式ノードのプッシュは、スタック(または、式の一部に一致して使用される式ツリーエントリの一部)へのノードの式ツリーエントリ300のプッシュを含むことが可能である。必要に応じて、更なる情報をエントリ内に含むことが可能である(例、評価フィールドのような一致の進行を示す種々のステータス変数)。

【0176】

上述の説明が完全に理解されると、数多くのバリエーションおよび変更は当業者にとって明らかになるであろう。以下の請求項が全ての当該のバリエーションおよび変更を包含すると解釈されることを意図するものである。

【図面の簡単な説明】

【0177】

【図1】内容変換装置の一実施形態のブロック図である。

10

20

30

40

50

- 【図 2】図 1 に示す文書プロセッサの一実施形態のブロック図である。
- 【図 3】図 2 に示す文書プロセッサおよび図 1 に示すプロセッサの一部を表す、一実施形態のブロック図であり、それらの間の通信およびそれらへの入力を示す。
- 【図 4】スタイルシートのコンパイルおよび式評価方法の一実施形態を示すフローチャートである。
- 【図 5】スタイルシートコンパイラの一実施形態の演算を示すフローチャートである。
- 【図 6】スキーマコンパイラの一実施形態の演算を示すフローチャートである。
- 【図 7】図 2 および図 3 に示す構文解析器の一実施形態のための入力データ構造および出力データ構造の一実施形態を示すブロック図である。
- 【図 8】シリアル番号をノード識別子へ割り当てるための、図 7 に示す構文解析器の一実施形態の演算を示すフローチャートである。 10
- 【図 9】図 2 および 3 に示す式プロセッサの一実施形態のための、入力データ構造および出力データ構造の一実施形態を示すブロック図である。
- 【図 10】図 2 および 9 に示す解析時間の式ツリーの一実施形態における構成ブロック図である。
- 【図 11】式ツリーおよびそれに対応する解析時間の式ツリーエントリの一部分の例である。
- 【図 12 A】要素開始イベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。
- 【図 12 B】要素開始イベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。 20
- 【図 13】要素終了イベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。
- 【図 14 A】属性名イベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。
- 【図 14 B】属性名イベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。
- 【図 15】要素クローズイベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。
- 【図 16】変換エンジンの一実施形態の演算を示すフローチャートである。 30
- 【図 17】図 2 および 9 に示す解析時間の式ツリーの別の実施形態におけるブロック図である。
- 【図 18】図 17 に示す特定のフィールドのための例示的なコード化を示す表のセットである。
- 【図 19 A】要素開始イベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。
- 【図 19 B】要素開始イベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。
- 【図 20】要素終了イベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。 40
- 【図 21 A】属性名イベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。
- 【図 21 B】属性名イベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。
- 【図 22 A】テキストイベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。
- 【図 22 B】テキストイベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。
- 【図 23 A】コメントイベントに応答する式プロセッサの一実施形態の演算を示すフローチャートである。 50

【図23B】コメントイベントに回答する式プロセッサの一実施形態の演算を示すフローチャートである。

【図24A】処理命令イベントに回答する式プロセッサの一実施形態の演算を示すフローチャートである。

【図24B】処理命令イベントに回答する式プロセッサの一実施形態の演算を示すフローチャートである。

【図1】

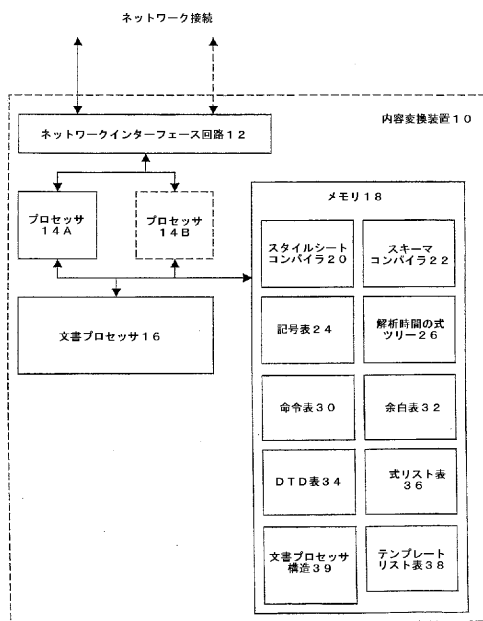


Fig. 1

【図2】

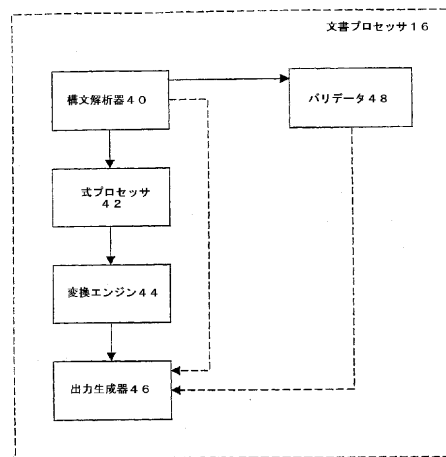
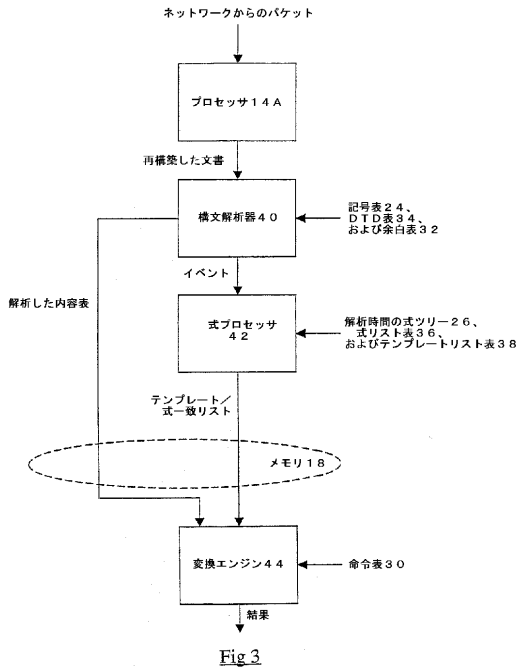
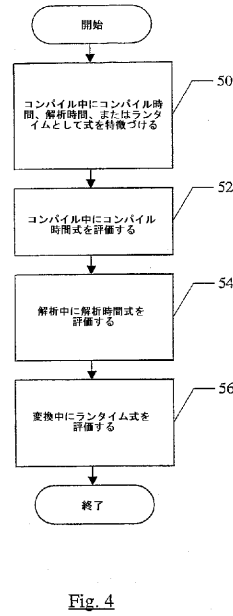


Fig. 2

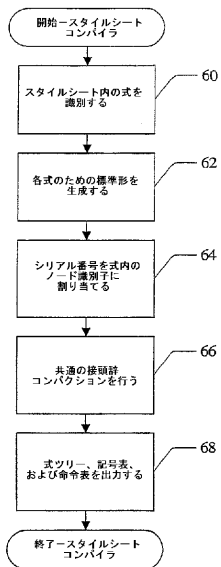
【 図 3 】



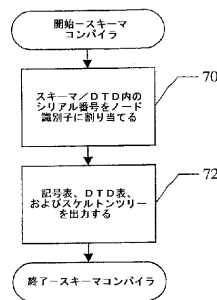
【 図 4 】



【 図 5 】



【 図 6 】



【 図 7 】

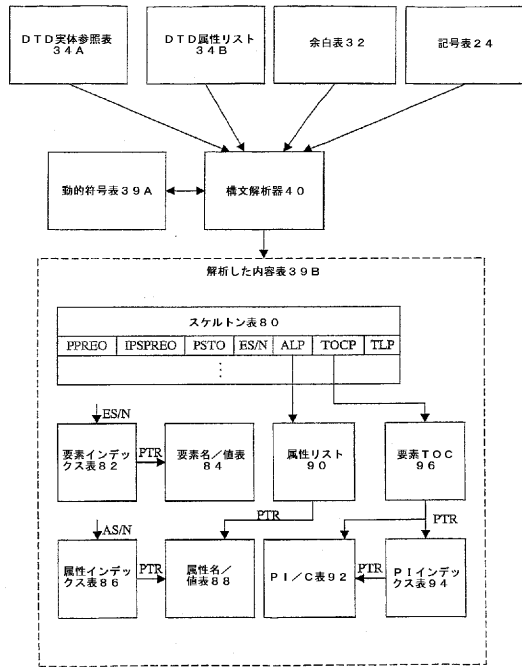


Fig. 7

【 図 8 】

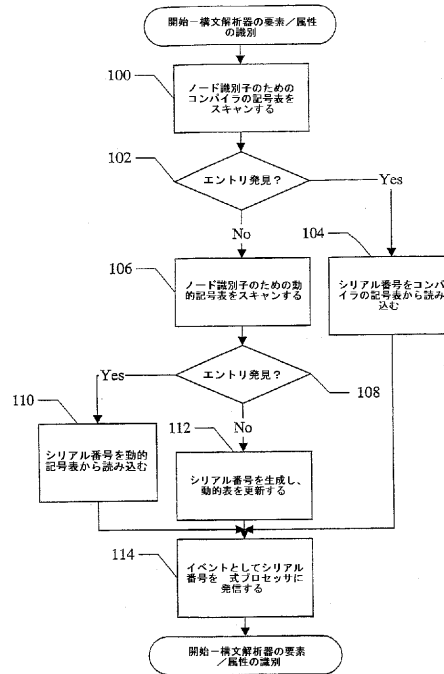


Fig. 8

【 図 9 】

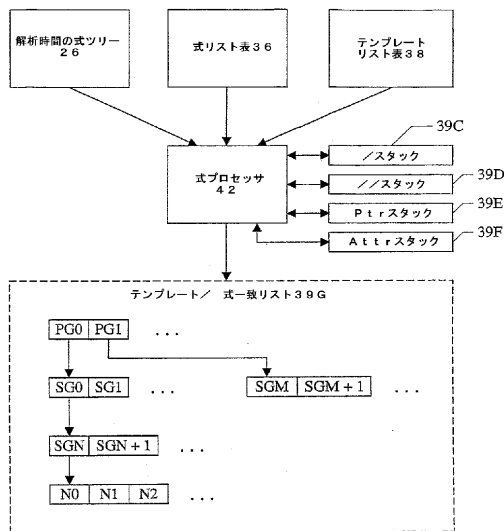


Fig. 9

【 図 10 】

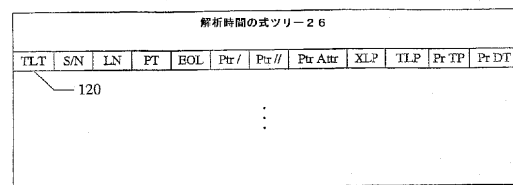


Fig. 10

【 図 11 】

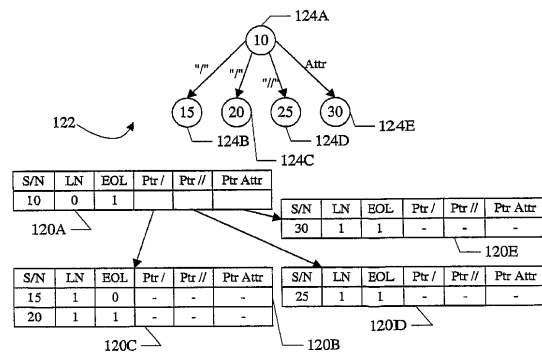


Fig. 11

【 図 1 2 A 】

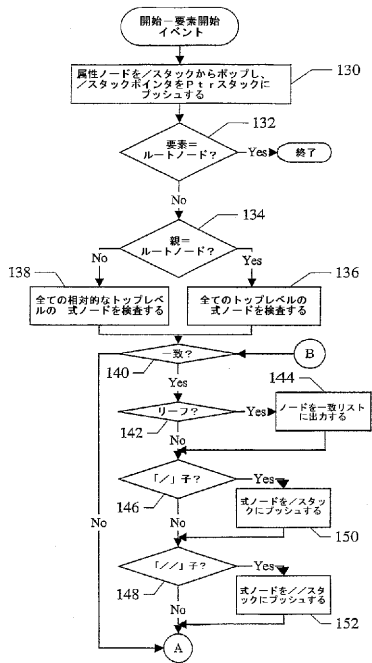


Fig. 12A

【 図 1 2 B 】

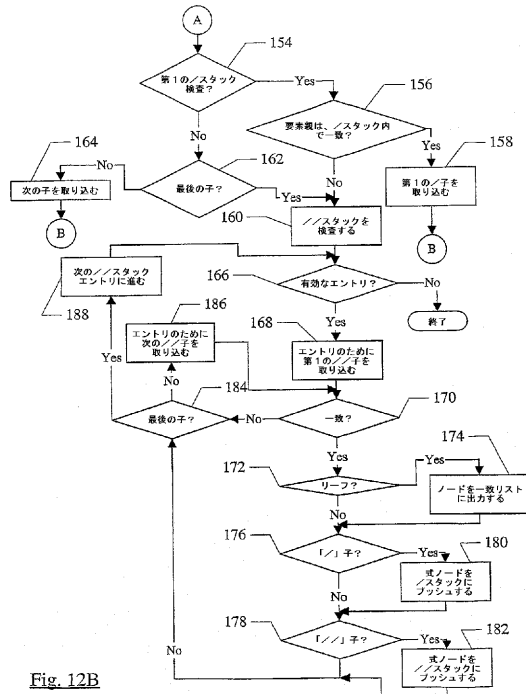


Fig. 12B

【 図 1 3 】

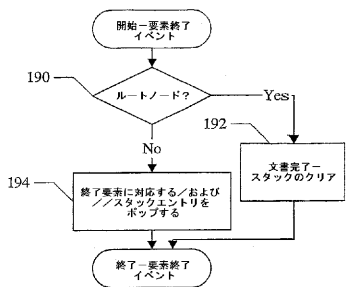


Fig. 13

【 図 1 4 A 】

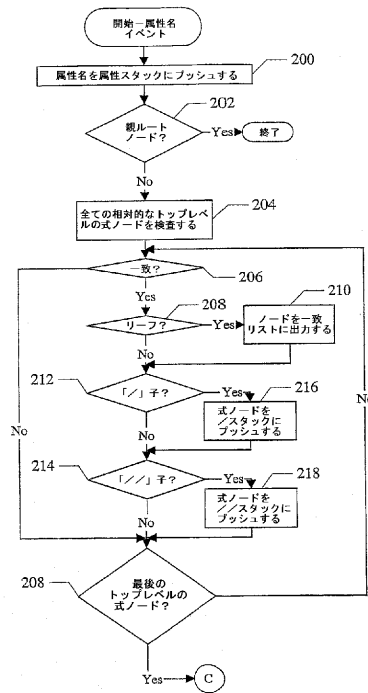


Fig. 14A

【図14B】

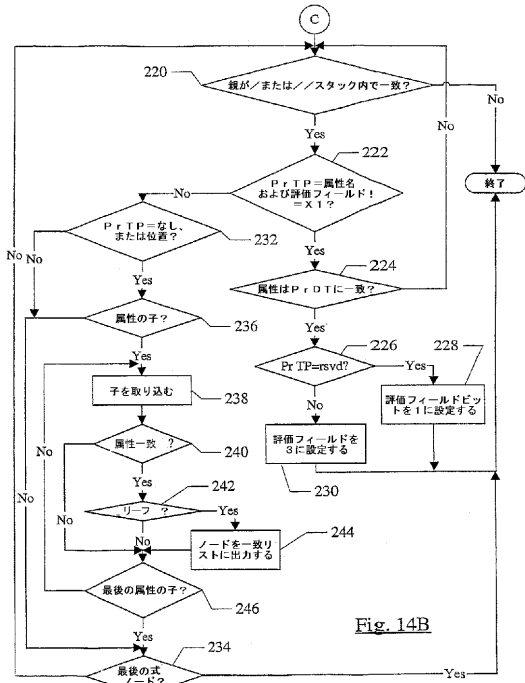


Fig. 14B

【図15】

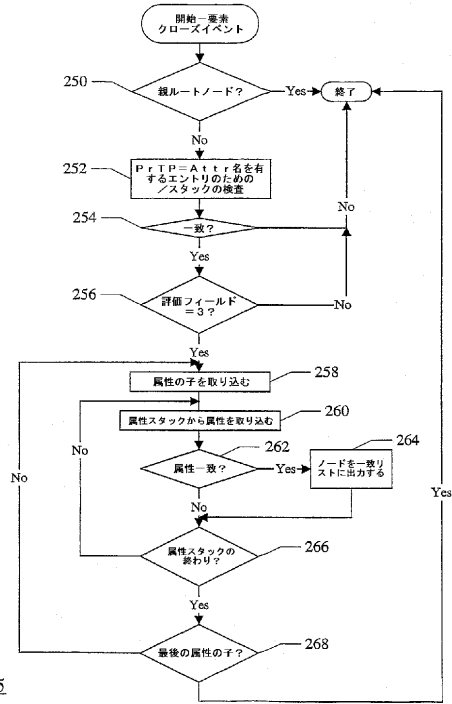


Fig. 15

【図16】

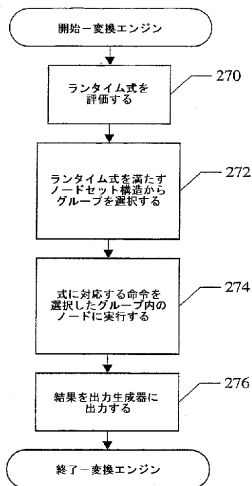


Fig. 16

【図18】

CDフィールド		NTフィールド		PrTPフィールド	
ビット	意味	コード	意味	コード	意味
11	要素、/	000	要素	0000	なし
10	要素、//	001	属性	0001	位置
9	属性、/	010	テキスト	0100	要素の子
8	属性、//	011	コメント	0101	属性名
7	テキスト、/	100	PI	0110	名前付PIノードテスト
6	テキスト、//	101	ノード	1000	ノードテスト
5	コメント、/	110	ターゲット付PI	1001	コメントノードテスト
4	コメント、//	111	予約	1010	PIノードテスト
3	PI、/			1011	テキストノードテスト
2	PI、//				
1	PI一定数、/				
0	PI一定数、//				

Fig. 18

【図17】

解析時間の式ツリー26											
TLT	S/N	LN	CD	EOL	XLP	TLP	PrTP	PrDT	NT	Pr /	Pr //
			Pr / Attr	Pr // Attr	Pr / PI	Pr // PI					

Fig. 17

【図19A】

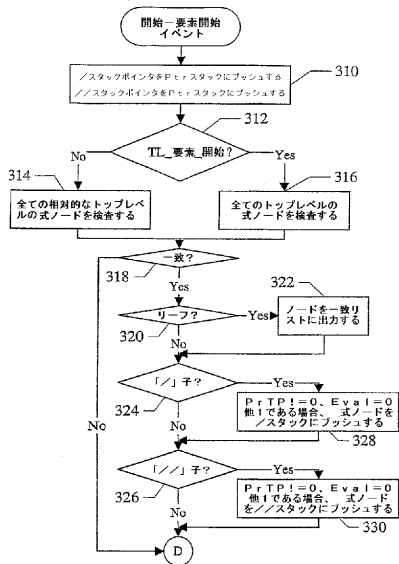


Fig. 19A

【図19B】

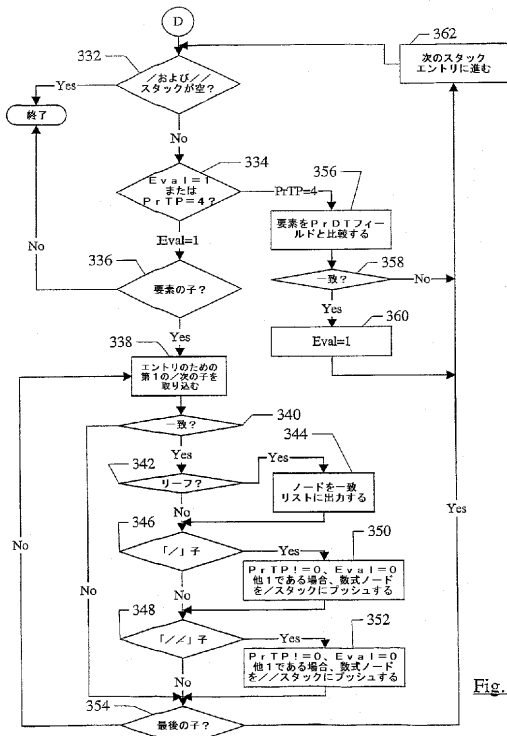


Fig. 19

【図20】

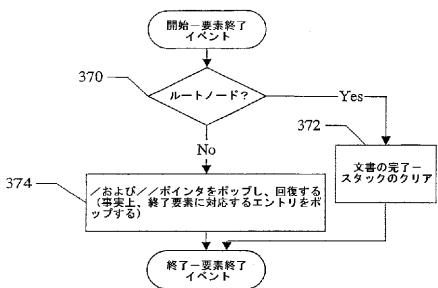


Fig. 20

【図21A】

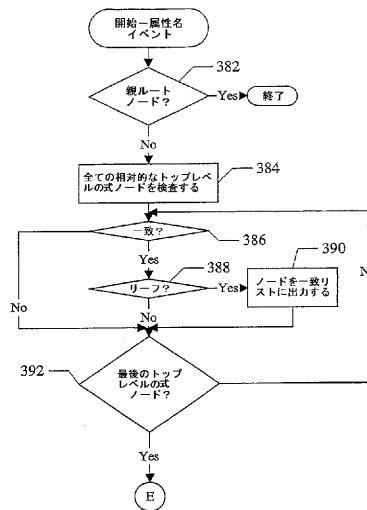


Fig. 21A

【図 2 1 B】

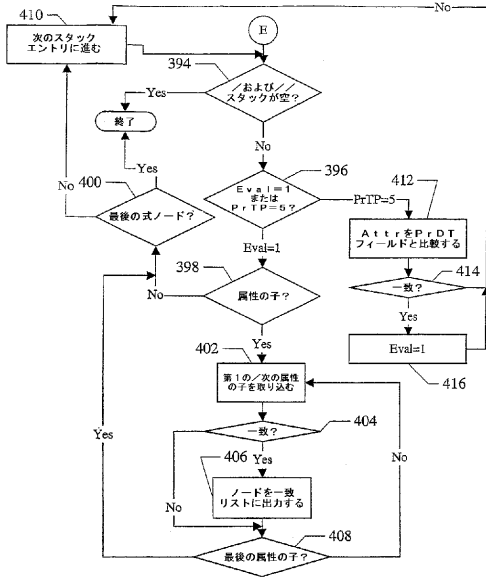


Fig. 21B

【図 2 2 A】

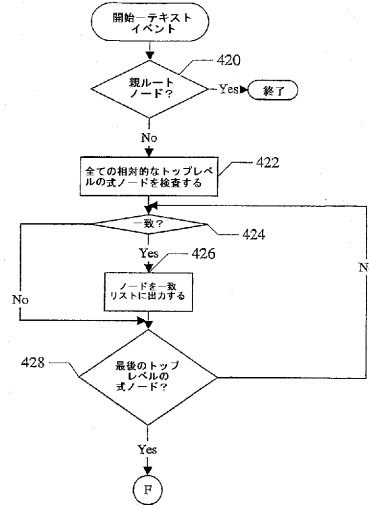


Fig. 22A

【図 2 2 B】

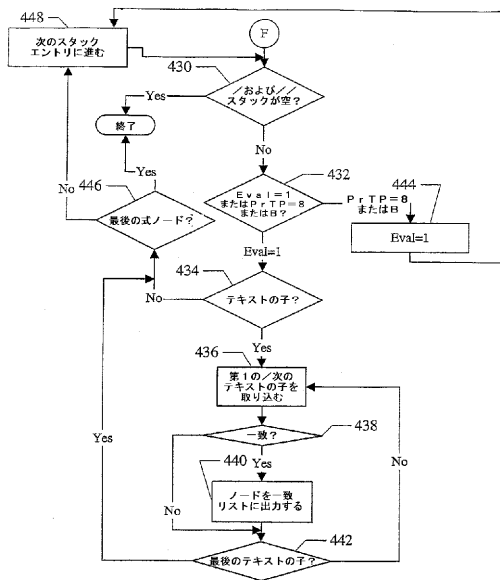


Fig. 22B

【図 2 3 A】

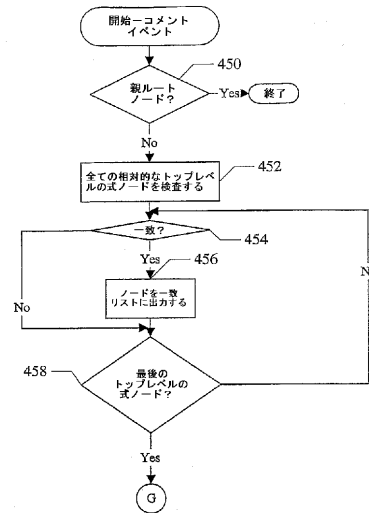


Fig. 23A

【 図 2 3 B 】

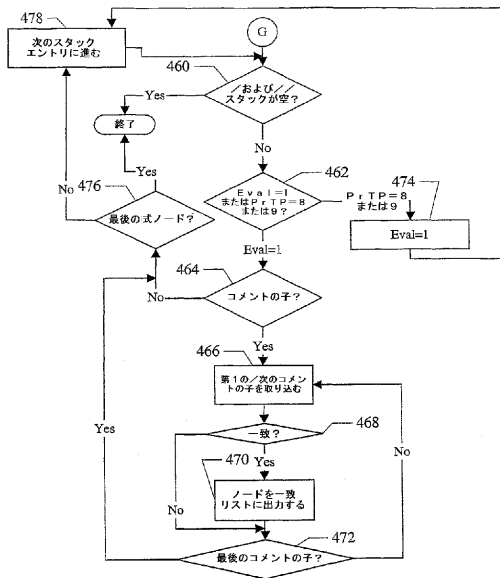


Fig. 23B

【 図 2 4 A 】

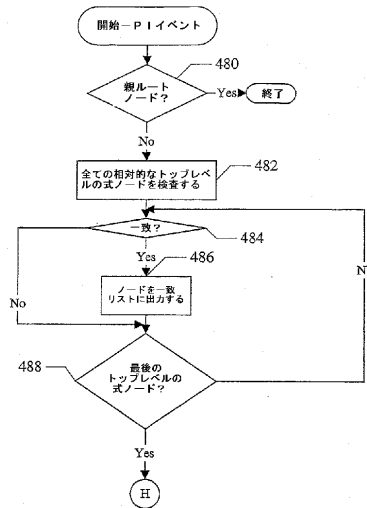


Fig. 24A

【 図 2 4 B 】

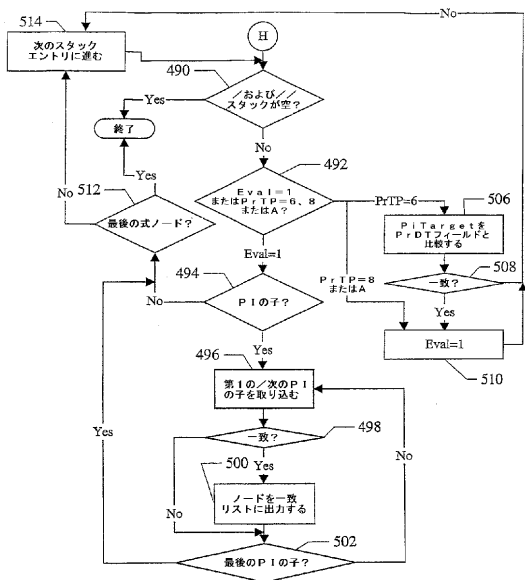


Fig. 24B

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International Application No PCT/US2004/035032
A. CLASSIFICATION OF SUBJECT MATTER IPC 7 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) IPC 7 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		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 01/46837 A2 (DATAPOWER TECHNOLOGY, INC) 28 June 2001 (2001-06-28) page 21, line 24 - page 28, line 14 figure 8B	1-24
X	BARTON C ET AL INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS: "Streaming XPath processing with forward and backward axes" PROCEEDINGS 19TH. INTERNATIONAL CONFERENCE ON DATA ENGINEERING. (ICDE'2003). BANGALORE, INDIA, MARCH 5 - 8, 2003, INTERNATIONAL CONFERENCE ON DATA ENGINEERING. (ICDE), NEW YORK, NY : IEEE, US, vol. CONF. 19, 5 March 2003 (2003-03-05), pages 455-466, XP010678760 ISBN: 0-7803-7665-X the whole document	1-24
-/--		
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C.		<input checked="" type="checkbox"/> Patent family members are listed in annex.
* Special categories of cited documents :		
"A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "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) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed		"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 "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 "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. "&" document member of the same patent family
Date of the actual completion of the international search 31 January 2005		Date of mailing of the international search report 04/02/2005
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016		Authorized officer Dumitrescu, C

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2004/035032

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2003/163285 A1 (NAKAMURA HIROAKI ET AL) 28 August 2003 (2003-08-28) abstract page 1, paragraph 5 - page 2, paragraph 28 page 4, paragraph 62 - page 11, paragraph 280	1-24
P,X	WO 03/094007 A1 (SARVEGA, INC) 13 November 2003 (2003-11-13) abstract page 2 - page 19 figures 1-4	1-24

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2004/035032

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
WO 0146837	A2	28-06-2001	AU 2286601 A	03-07-2001
			CA 2394058 A1	28-06-2001
			EP 1242907 A2	25-09-2002
			JP 2003518291 T	03-06-2003
			US 6772413 B2	03-08-2004
			US 2001056504 A1	27-12-2001
US 2003163285	A1	28-08-2003	JP 2003323332 A	14-11-2003
WO 03094007	A1	13-11-2003	CA 2484355 A1	13-11-2003
			US 2004010754 A1	15-01-2004

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, HU, IE, IT, LU, MC, NL, 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, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW

Fターム(参考) 5B009 QA09 TA11