

(19) 日本国特許庁(JP)

## (12) 公開特許公報(A)

(11) 特許出願公開番号

特開2005-174340

(P2005-174340A)

(43) 公開日 平成17年6月30日(2005.6.30)

(51) Int.CI.<sup>7</sup>

G06F 9/44

G06F 17/21

F 1

G06F 9/06 620K

G06F 17/21 580J

テーマコード(参考)

5B009

5B076

審査請求 未請求 請求項の数 18 O L (全 19 頁)

(21) 出願番号 特願2004-355669 (P2004-355669)  
 (22) 出願日 平成16年12月8日 (2004.12.8)  
 (31) 優先権主張番号 10/731,597  
 (32) 優先日 平成15年12月9日 (2003.12.9)  
 (33) 優先権主張国 米国(US)

(71) 出願人 500046438  
 マイクロソフト コーポレーション  
 アメリカ合衆国 ワシントン州 98052  
 2-6399 レッドモンド ワン マイクロソフト ウェイ  
 (74) 代理人 100077481  
 弁理士 谷 義一  
 (74) 代理人 100088915  
 弁理士 阿部 和夫  
 (72) 発明者 ブライアン エム. ジョーンズ  
 アメリカ合衆国 98052 ワシントン  
 州 レッドモンド ワン マイクロソフト  
 ウェイ マイクロソフト コーポレーション内

最終頁に続く

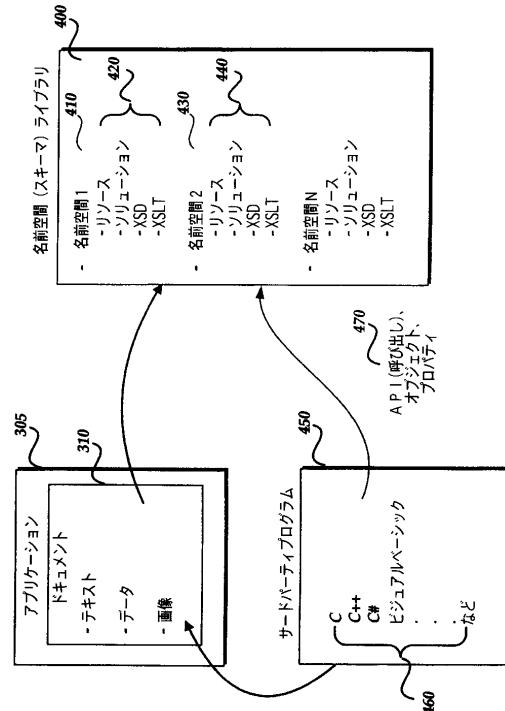
(54) 【発明の名称】ソフトウェアアプリケーションにおけるネーム空間あるいはスキーマライブラリ支援のためのプログラム可能なオブジェクトモデル

## (57) 【要約】

【課題】ユーザー/プログラマがリソースをカスタマイズし、その他操作のため、XMLリソースのネーム空間あるいはスキーマライブラリにアクセスし、ソフトウェアアプリケーションドキュメントとのユーザー/プログラマのXML機能の効用を高める。

【解決手段】プログラム可能なオブジェクトモデルによって、ユーザーは、XMLスキーマファイルおよび関連するXMLベースのリソースを含むネーム空間あるいはスキーマライブラリを、そのスキーマファイルおよびXMLベースのリソースを1つあるいは複数のドキュメントに関連付けるために、および、そのスキーマファイルおよびXMLベースのリソースに関連付けられた機能をカスタマイズするために、プログラムでアクセスし利用する。

【選択図】 図4



**【特許請求の範囲】****【請求項 1】**

拡張可能なマークアップ言語（XML）のスキーマライブラリのリソースにアクセスするプログラム可能なオブジェクトモデルにおいて、

XMLスキーマライブラリにおいて識別されたリソースにユーザーがプログラムでアクセスすることを可能にするアプリケーション・プログラミング・インターフェースと、

前記アプリケーション・プログラミング・インターフェースが、ドキュメントに適用されたXMLマークアップへのXMLスキーマファイルの関連付けを要求するメッセージ呼び出しを備えることと、および

前記アプリケーション・プログラミング・インターフェースが、前記ドキュメントに適用された前記XMLマークアップへの前記XMLスキーマファイルの関連付けに対応して、前記XMLスキーマライブラリから返却値を受け取るよう機能することと、

を備えることを特徴とするプログラム可能なオブジェクトモデル。

**【請求項 2】**

前記アプリケーション・プログラミング・インターフェースがさらに、前記ドキュメントに適用される前記XMLマークアップへの前記XMLスキーマファイルの関連付けの削除を要求するメッセージ呼び出しを備えること、および

前記アプリケーション・プログラミング・インターフェースがさらに、前記ドキュメントに適用される前記XMLマークアップへの前記XMLスキーマファイルの前記関連付けの削除に対応して、前記XMLスキーマライブラリから返却値を受け取るよう機能すること

、  
を特徴とする請求項1に記載のプログラム可能なオブジェクトモデル。

**【請求項 3】**

拡張可能なマークアップ言語（XML）のスキーマライブラリのリソースにアクセスするプログラム可能なオブジェクトモデルにおいて、

XMLスキーマライブラリにおいて識別されたリソースにユーザーがプログラムでアクセスすることを可能にするアプリケーション・プログラミング・インターフェースと、

前記アプリケーション・プログラミング・インターフェースが、ドキュメントに適用されるXMLマークアップへのXSLT変換の関連付けを要求するメッセージ呼び出しを備えることと、および

前記アプリケーション・プログラミング・インターフェースが、前記ドキュメントに適用される前記XMLマークアップへの前記XSLT変換の関連付けに対応して、前記XMLスキーマライブラリから返却値を受け取るよう機能することと、

を備えることを特徴とするプログラム可能なオブジェクトモデル。

**【請求項 4】**

前記アプリケーション・プログラミング・インターフェースがさらに、前記ドキュメントに適用される前記XMLマークアップへの前記XSLT変換の関連付けの削除を要求するメッセージ呼び出しを備えること、および

前記アプリケーション・プログラミング・インターフェースがさらに、前記ドキュメントに適用される前記XMLマークアップへの前記XSLT変換の前記関連付けの削除に対応して、前記XMLスキーマライブラリから返却値を受け取るよう機能すること、

を特徴とする請求項3に記載のプログラム可能なオブジェクトモデル。

**【請求項 5】**

拡張可能なマークアップ言語（XML）のスキーマライブラリのリソースにアクセスするプログラム可能なオブジェクトモデルにおいて、

XMLスキーマライブラリにおいて識別されたリソースにユーザーがプログラムでアクセスすることを可能にするアプリケーション・プログラミング・インターフェースと、

前記アプリケーション・プログラミング・インターフェースが、ドキュメントに適用されるXMLマークアップへの1つあるいは複数のXMLベースのリソースの関連付けを要求するメッセージ呼び出しを備えることと、および

10

20

30

40

50

前記アプリケーション・プログラミング・インターフェースが、前記ドキュメントに適用される前記XMLマークアップへの前記1つあるいは複数のXMLベースのリソースの関連付けに対応して、前記XMLスキーマライブラリから返却値を受け取るよう機能することと、

を備えることを特徴とするプログラム可能なオブジェクトモデル。

#### 【請求項6】

前記アプリケーション・プログラミング・インターフェースがさらに、前記ドキュメントに適用された前記XMLマークアップへの前記1つあるいは複数のXMLベースのリソースの関連付けの削除を要求するメッセージ呼び出しを備えること、および

前記アプリケーション・プログラミング・インターフェースがさらに、前記ドキュメントに適用される前記XMLマークアップへの前記1つあるいは複数のXMLベースのリソースの前記関連付けの削除に対応して、前記XMLスキーマライブラリから返却値を受け取るよう機能すること、10

を特徴とする請求項5に記載のプログラム可能なオブジェクトモデル。

#### 【請求項7】

拡張可能なマークアップ言語(XML)のリソースにアクセスする方法において、

オブジェクト指向のメッセージ呼び出しによりXMLスキーマライブラリを呼び出すことと、

前記XMLスキーマライブラリにおいて識別された機能に関連付けられたソフトウェアオブジェクトに関連付けられたオブジェクトプロパティを、前記XMLスキーマライブラリに渡すことと、および20

前記メッセージ呼び出しあり前記XMLスキーマライブラリに渡された前記オブジェクトプロパティに対応して、前記XMLスキーマライブラリに渡された前記オブジェクトプロパティに関連付けられた前記XMLスキーマライブラリにおいて識別された前記機能へのアクセスを受け付けること、

を備えることを特徴とする方法。

#### 【請求項8】

前記XMLスキーマライブラリにオブジェクトプロパティを渡すことが、新しいXMLネーム空間を作成し、XMLネーム空間の集合に前記の新しいXMLネーム空間を加えるメソッドプロパティを渡すことを含み、前記の新しいXMLネーム空間に関連付けられたスキーマファイルへのパス、および前記の新しいXMLネーム空間のための統一的リソース識別子が、前記メソッドのオブジェクトのパラメーターとして、前記XMLスキーマライブラリに渡されることを特徴とする請求項7に記載の方法。30

#### 【請求項9】

前記XMLスキーマライブラリにオブジェクトプロパティを渡すことが、前記XMLスキーマライブラリにXMLネーム空間を登録するソリューションマニフェストをインストールするメソッドプロパティを渡すことを含むこと、を特徴とする請求項7に記載の方法。。

#### 【請求項10】

前記XMLスキーマライブラリにオブジェクトプロパティを渡すことが、数字インデックスあるいは検索キーワードを用いて、XMLリソースの集合から個別のXMLリソースにアクセスするメソッドプロパティを渡すことを含み、個別のXMLリソースに関連付けられた数字インデックスが、前記メソッドプロパティを持つパラメーターとして渡されることを特徴とする請求項7に記載の方法。40

#### 【請求項11】

前記XMLスキーマライブラリに前記オブジェクトプロパティを渡すことが、前記XMLスキーマライブラリにおいて識別された、指定されたネーム空間に関連付けられたエイリアス名を制御するオブジェクトプロパティを渡すことを含むことを特徴とする請求項7に記載の方法。

#### 【請求項12】

10

20

30

40

50

前記 XMLスキーマライブラリに前記オブジェクトプロパティを渡すことが、指定された XMLネーム空間の指定された XMLスキーマファイルを指定されたドキュメントに添付するメソッドプロパティを渡すことを含み、前記指定されたドキュメントに対するポインタが前記 XMLスキーマライブラリに前記メソッドプロパティのパラメーターとして渡されることを特徴とする請求項 7 に記載の方法。

**【請求項 13】**

前記 XMLスキーマライブラリに前記オブジェクトプロパティを渡すことが、指定されたネーム空間に関連付けられた、デフォルトの XSLT 変換を指し示すオブジェクトプロパティを渡すことを含むことを特徴とする請求項 7 に記載の方法。

**【請求項 14】**

前記 XMLスキーマライブラリにオブジェクトプロパティを渡すことが、ネーム空間オブジェクトの集合から XMLネーム空間オブジェクトを削除するメソッドプロパティを渡すことを含むことを特徴とする請求項 7 に記載の方法。

**【請求項 15】**

前記 XMLスキーマライブラリにオブジェクトプロパティを渡すことが、新しい XSLT 変換を作成し、XSLT 変換の集合に前記の新しい XSLT 変換を加えるメソッドプロパティを渡すことを含み、前記の新しい XSLT 変換に対するポインタが、前記メソッドプロパティへのパラメーターとして、前記 XMLスキーマライブラリに渡されることを特徴とする請求項 7 に記載の方法。

**【請求項 16】**

数字のインデックスあるいは検索キーワードを用いて XSLT 変換の集合に含まれる個別の XSLT 変換にアクセスするメソッドプロパティをさらに備え、前記 XMLスキーマライブラリにおける要求された XSLT 変換の位置を表している数字のインデックスが前記メソッドプロパティを持つ前記 XMLスキーマライブラリへのパラメーターとして渡されることを特徴とする請求項 7 に記載の方法。

**【請求項 17】**

前記 XMLスキーマライブラリにオブジェクトプロパティを渡すことが、前記 XMLスキーマライブラリにおいて識別された XSLT 変換に関連付けられたエイリアス名を制御するオブジェクトプロパティを渡すことを含むことを特徴とする請求項 7 に記載の方法。

**【請求項 18】**

前記 XMLスキーマライブラリに前記オブジェクトプロパティを渡すことが、XSLT 変換の集合から XSLT 変換を削除するメソッドプロパティを渡すこと、を特徴とする請求項 7 に記載の方法。

**【発明の詳細な説明】**

**【技術分野】**

**【0001】**

本発明はプログラム可能なオブジェクトモデルに関する。より詳細には、本発明はソフトウェアアプリケーションでのネーム空間(Namespace)あるいはスキーマライブラリ(schema library)支援のためのプログラム可能なオブジェクトモデルに関する。

**【背景技術】**

**【0002】**

コンピュータ・ソフトウェア・アプリケーションは、ユーザーが仕事、教育および自由時間に自分のための各種ドキュメントを作成することを可能にする。たとえば、一般的な文書処理アプリケーションでは、ユーザーが手紙、論文、本、覚書、などを作成することを可能にする。表計算アプリケーションは、ユーザーが各種の英数字データを格納し、操作し、印刷し、そして表示することを可能にする。このようなアプリケーションは、豊富な編集、フォーマッティング、印刷、計算、およびオンライン、オフラインの編集をはじめとする多くのよく知られた長所を有している。

**【0003】**

**【特許文献 1】**米国特許出願第 10 / 187,060 号、名称「Word Processing Document

10

20

30

40

50

t Stored in a Single XML File that may be Manipulated by Applications that Understanding XML」(パブリケーション番号: 20040210818)

【特許文献2】米国特許出願第10/184,190号、名称「System and Method for Providing Namespace Related Information」(パブリケーション番号: 20040002937)

【特許文献3】米国特許出願第10/185,940号、名称「System and Method for Obtaining and Using Namespace Related Information for Opening XML Documents」(パブリケーション番号: 20040205470)

【特許文献4】米国特許出願第10/164,260号、名称「Mechanism for Downloading Software Components from a Remote Source for Use by a Local Software Application」(パブリケーション番号: 20040003389)

#### 【発明の開示】

#### 【発明が解決しようとする課題】

#### 【0004】

大部分のコンピュータ・ソフトウェア・アプリケーションは、あらゆる潜在的ユーザーにより必要とされ、あるいは期待される機能を提供するために必要なプログラミングを全て含んでいるわけではない。多くのプログラマは、しばしば彼等自身のプログラムの中の既存のアプリケーションの能力を利用することや、あるいはアプリケーションの機能をカスタマイズすることを望み、そしてそれをユーザーあるいは行動の特定のセットのために一層適合させる。たとえば金融業界で働くプログラマは、会計報告書を編集する金融アナリストから成るユーザーの人々のために、ワードプロセッサーをカスタマイズすることを望む可能性がある。近年、拡張可能なマークアップ言語(Extensible Markup Language)が、多くのユーザーにとって交換可能なデータフォーマットとして広く用いられている。XML機能のユーザーはしばしば、そのユーザーにより編集され、あるいは作成されるドキュメントに、1つあるいは複数のXMLスキーマファイル、あるいはXMLベースのソリューションを添付し(attach)、あるいは関連付ける(associate)。しかしながら、ユーザー／プログラマは、XMLスキーマファイル、あるいはXMLベースのソリューションの機能を所与のドキュメントに適用する能力において限界がある。なぜなら、ユーザー／プログラマはXMLスキーマファイルあるいは他のXMLベースのソリューションを含むネーム空間あるいはスキーマライブラリに直接かつ容易にアクセスすることができないからである。

#### 【0005】

したがって、プログラム可能なオブジェクトモデルに関する技術には、ユーザー／プログラマが、リソースをカスタマイズし、あるいはその他操作をするために、XMLリソースのネーム空間あるいはスキーマライブラリにアクセスできるようにし、ソフトウェア・アプリケーション・ドキュメントを有する、ユーザー／プログラマが使用するXML機能を向上させる必要性がある。本発明が為されたのはこれらおよび他の考慮があつてのものである。

#### 【課題を解決するための手段】

#### 【0006】

本発明は、ユーザーがXMLスキーマファイルおよび関連するXMLベースのリソースを含むネーム空間あるいはスキーマライブラリにプログラムで(programmatically)アクセスし利用することを可能とし、これらのXMLベースのリソースを1つあるいは複数のドキュメントと関連付け、これらのXMLベースのリソースと関連付けられた機能をカスタマイズする方法およびシステムを提供する。ユーザーあるいはプログラマはネーム空間あるいはスキーマライブラリへのアクセスを一旦獲得すると、そのユーザーは、関連付けられたドキュメントの中のXMLデータをXMLスキーマファイルにプログラムで関連付けることができる。逆に、ユーザーは、XMLスキーマファイルと、ドキュメントに含まれるXMLデータとの関連付け(associations)を検出し、削除することができる。さらに、ユーザーは、プログラムで、変換ファイルをドキュメントに含まれるXMLデータに関連

10

20

30

40

50

付け、ドキュメントに含まれる XML データに関連付けられた既存の変換ファイルを検出し、削除することができる。さらに、ユーザーは、XML ベースのドキュメントソリューションおよび他のドキュメントソリューションに関連付けられた他のファイルおよび実行可能なソフトウェアを、ドキュメントに含まれる XML データに関連付けることができる。さらに、ユーザーは、ドキュメントに含まれる XML データとの関連付けから XML ベースのソリューションおよび他の種類の実行可能なソフトウェアの関連付けを検出し、削除することができる。

#### 【 0 0 0 7 】

本発明のこれらおよび他の特徴、利点および形態は、以下に開示される実施形態の詳細な記述を吟味頂くことから、および添付された図画および請求範囲を参照されることにより、より明確に理解され、認識されるであろう。

10

#### 【 発明を実施するための最良の形態 】

#### 【 0 0 0 8 】

本発明の実施形態は、ユーザーが、XML スキーマファイルおよび XML ベースのソリューションおよびリソースのネーム空間 /スキーマライブラリをプログラムで呼び出すことを可能とし、それらのファイル、ソリューションおよびリソースの、1つあるいは複数のドキュメントとの関連付けを制御する方法およびシステムを対象とする。これらの実施形態は、本発明の精神あるいは範囲から逸脱することなく結合される場合があり、他の実施形態が利用される場合があり、そして、構造的変更が為される場合がある。したがって、次の詳細な記述は限定する意味で解釈されるべきではなく、本発明の範囲はその添付された請求範囲、およびそれと同等な物により定義される。

20

#### 【 0 0 0 9 】

さて、いくつかの図を通して、同様の数字が同様な要素を表す図面を参照して、本発明の形態およびその例示的な動作環境を説明する。図 1 および以下の説明は本発明を実施することができる適切なコンピュータ処理環境を簡潔に一般的に説明するためのものである。本発明は、パーソナルコンピュータのオペレーティングシステム上で動作するアプリケーションプログラムと連動して実行されるプログラムモジュールとの一般的な文脈において記述されるが、当業者は、本発明を他のプログラムモジュールと結合して実施することが可能であることを認識するであろう。

30

#### 【 0 0 1 0 】

一般的にプログラムモジュールは、ルーチン、プログラム、構成要素、データ構造、および特定のタスクを遂行し、あるいは特定の抽象データ型を実施する他の種類の構造を備える。さらに、当業者は、携帯型の装置、マルチプロセッサーシステム、マイクロプロセッサベースのあるいはプログラム可能な家庭用電化製品、ミニコンピュータ、メインフレームコンピュータなどを含む他のコンピュータシステム構成を用いて本発明を実施できることを理解するであろう。また、本発明は、通信ネットワークを介して接続された遠隔処理装置によってタスクが遂行される分散コンピュータ処理環境で実施することができる。分散コンピュータ処理環境では、プログラムモジュールは、ローカルメモリ記憶装置およびリモートメモリ記憶装置の両方に置くことができる。

40

#### 【 0 0 1 1 】

図 1 には、本発明の種々の実施形態を実施するパーソナルコンピュータの例示的コンピュータキテクチャー 2 が記述される。図 1 で示されるコンピュータキテクチャーは、中央処理装置 (CPU) 4、ランダムアクセスメモリ (RAM) 8、読み取り専用メモリ (ROM) 10 を含むシステムメモリ 6、およびメモリを CPU 4 に結合するシステムバス 12 を備える従来式のパーソナルコンピュータを示す。起動の間等の際に、コンピュータの中の要素の間で情報を転送する役割を担う基本的ルーチンを含む基本入出力システムが ROM 10 に格納される。パーソナルコンピュータ 2 は、さらに、オペレーティングシステム 16、アプリケーションプログラム 305 等のアプリケーションプログラム、およびデータを格納する大容量記憶装置 14 を備える。

50

#### 【 0 0 1 2 】

大容量記憶装置 14 は、バス 12 に接続された大容量記憶装置コントローラ（図示せず）を通して C P U 4 に接続される。大容量記憶装置 14 およびその関連付けられたコンピュータで読み取り可能な媒体は、パーソナルコンピュータ 2 のために不揮発性のストレージを提供する。本明細書に含まれるコンピュータで読み取り可能な媒体の記述は、ハードディスクあるいは C D - R O M ドライブ等の大容量記憶装置を指すが、コンピュータ読み取り可能な媒体は、パーソナルコンピュータ 2 がアクセスできる任意の利用可能な媒体であればよいことを当業者は理解すべきである。

#### 【 0 0 1 3 】

たとえば、限定するものではないが、コンピュータで読み取り可能な媒体は、コンピュータの記憶媒体および通信媒体を備えることができる。コンピュータの記憶媒体は、コンピュータ読み取り可能命令、データ構造、プログラムモジュール、他のデータ等の情報を記憶するための任意の方法あるいは技術により実現された揮発性および不揮発性の、着脱可能および固定の媒体を含む。コンピュータの記憶媒体は、限定するものではないが、R A M 、 R O M 、 E P R O M 、 E E P R O M 、フラッシュメモリ、他の固体のメモリ技術、C D - R O M 、 D V D 、他の光学的記憶域、磁気カセット、磁気テープ、磁気のディスク記憶域、他の磁気記憶装置、所望の情報を格納するために用いることができコンピュータによりアクセス可能な他の任意の媒体を含む。

#### 【 0 0 1 4 】

本発明の種々の実施形態によれば、パーソナルコンピュータ 2 は、インターネット等の T C P / I P ネットワーク 18 を介してリモートコンピュータへの論理的接続を用いて、ネットワーク化された環境で動作できる。パーソナルコンピュータ 2 は、バス 12 に接続されたネットワークインターフェースユニット 20 を通して T C P / I P ネットワーク 18 と接続できる。ネットワークインターフェースユニット 20 は、他の種類のネットワークおよびリモートコンピュータシステムに接続するために利用できることを理解すべきである。パーソナルコンピュータ 2 は、キーボードあるいはマウス（図示せず）を含む多くの装置からの入力を受け取り、処理する入力 / 出力コントローラ 22 を含むことができる。同様に、入力 / 出力コントローラ 22 は、表示スクリーン、プリンター、あるいは他のタイプの出力装置に出力を提供することができる。

#### 【 0 0 1 5 】

以上、簡潔に言及したように、Redmond, Washington の M I C R O S O F T C O R P O R A T I O N からの W i n d o w s X P (登録商標) オペレーティングシステム等の、ネットワーク化されたパーソナルコンピュータの動作を制御するにふさわしいオペレーティングシステム 16 を含む、多くのプログラムモジュールおよびデータファイルを、パーソナルコンピュータ 2 の大容量記憶装置 14 および R A M 8 に格納することができる。大容量記憶装置 14 および R A M 8 は、1つあるいは複数のアプリケーションプログラムを格納することができる。大容量記憶装置 14 および R A M 8 は特に、電子的ドキュメント 310 を作成し編集するためのアプリケーションプログラム 305 を格納することができる。たとえば、アプリケーションプログラム 305 は文書処理アプリケーションプログラム、表計算アプリケーション、問い合わせアプリケーション、などを備えることができる。他の種類の電子的ドキュメントを作成し編集するためのアプリケーションプログラムを本発明の種々の実施形態で用いることができる。スキーマファイル 330 およびネーム空間 / スキーマライブラリ 400 を以下に説明する。

#### 【 0 0 1 6 】

本発明の例示的な実施形態が、オブジェクト指向プログラミング環境における異なるソフトウェアオブジェクト間の通信により実施される。本発明の実施形態に関する以下の説明のために、オブジェクト指向プログラミング環境の構成要素を簡潔に記述することは有用である。図 2 は、オブジェクト指向プログラミングモデルによるソフトウェアオブジェクトの間の相互作用を例示する単純化されたブロック図である。オブジェクト指向プログラミング環境により、第 1 のオブジェクト 210 は、ソフトウェアコード、実行可能なメソッド (method)、プロパティ (property)、およびパラメーターを含むことができる。同様

に、第2のオブジェクト220は、ソフトウェアコード、実行可能なメソッド、プロパティ、およびパラメーターを含むことができる。

【0017】

第1のオブジェクト210は、第2のオブジェクト220と通信し、メッセージ呼び出し(call)230によって第2のオブジェクト220を呼び出すことにより第2のオブジェクト220から情報あるいは機能を獲得することができる。オブジェクト指向プログラミング環境に精通している当業者にはよく知られているように、第1のオブジェクト210は、2つの異なるソフトウェアオブジェクト210、220が、互いに情報および機能を獲得するために互いに通信することを可能にするアプリケーションプログラムインターフェース(API)により、第2のオブジェクト220と通信することができる。たとえば、第1のオブジェクト210が第2のオブジェクト220に含まれるメソッドにより提供される機能を要求する場合には、第1のオブジェクト210は、第2のオブジェクト220にメッセージ呼び出し230を渡し、第2のオブジェクトでは、第1のオブジェクトが要求されたメソッドを識別し、識別されたメソッドを動作させる第2のオブジェクトにより要求された必要なすべてのパラメーターを、第1のオブジェクトが第2のオブジェクトに渡すことができる。第2のオブジェクト220は第1のオブジェクトからの呼び出しを受けると、第2のオブジェクトは提供されたパラメーターに基づいて呼び出されたメソッドを実行し、実行されたメソッドから得られた値を含む返信メッセージ250を第1のオブジェクト210に送り返す。

【0018】

例えば、本発明の実施形態に関して言えば、そして以下に記述されるように、第1のオブジェクト210が、サードパーティによりカスタマイズされたアプリケーションであり、そのアプリケーションは、拡張可能なマークアップ言語のスキーマ妥当性検査オブジェクト(schema validation object)等の第2のオブジェクトへメッセージを渡すことができ、その場合には、第1のオブジェクトは、ドキュメント中の指定されたXML要素の妥当性検査(validation)を必要としているメソッドを識別し、指定されたXML要素は、第1のオブジェクトによりその識別されたメソッドとともに渡されるパラメーターである。この例によれば、第1のオブジェクトからの呼び出しを受け取ると、スキーマ妥当性検査オブジェクトは指定されたXML要素に対してその識別されたメソッドを実行し、妥当性を検査されたXML要素に関連付けられた結果あるいは値の形式で、第1のオブジェクトへメッセージを返す。上に簡潔に記述されたような、オブジェクト指向プログラミング環境の動作は当業者によく知られている。

【0019】

以下に記述されるように、本発明の実施形態は、拡張マークアップ言語(XML)の構成要素の使用、カスタマイゼーション、およびアプリケーションにおけるソフトウェアオブジェクトの相互作用を通して実行される。図3は、ドキュメント、添付されたスキーマファイル、およびスキーマ妥当性検査機能モジュールの間の相互作用を例示するロック図である。当業者によく知られているように、拡張マークアップ言語(XML)は、ユーザーが、ドキュメントのテキストあるいはデータに適用され、そして、その関連付けられたタグが適用されたテキストあるいはデータを定義するタグ名を、作成することを可能にすることにより、ドキュメントにおけるテキストおよびデータを記述する方法を提供する。たとえば、図3を参照すると、アプリケーション305で作成されたドキュメント310は、XMLのタグ315、320、325でマークアップされたテキストを含む。たとえば、テキスト「Greetings」は、XMLのタグ<title>で注釈(annotation)されている。テキスト「My name is Sarah」は<body>タグで注釈されている。XMLによれば、<title>と<body>タグの作成者(creator)は、これらのタグが適用されるタグを記述するために自分自身のタグを自由に作成することができる。次に、任意の下流(downstream)の消費者用アプリケーションあるいはコンピューティングマシンでテキストに適用されたタグの定義についての命令が提供されている限り、アプリケーションあるいはコンピューティングマシンは、タグに従ってデータを利用すること

10

20

30

40

50

ができる。たとえば、下流のアプリケーションがそのアプリケーションにより処理された記事(articles)あるいは出版物の名称として定義されたテキストを抽出するようプログラムされている場合には、アプリケーションは、そのテキストがタグ<title>で注釈されているため、図3に示されているように、ドキュメント310を解析し、テキスト「Greetings」を抽出することができる。図3で例示されたドキュメント310に対する特定のXMLタグ命名の作成者は、サードパーティにより利用されるドキュメント310に含まれるテキストあるいはデータに関する有用な記述を、テキストあるいはデータに適用されたタグに関連付けられた定義がそれらのサードパーティに提供されてさえいれば、提供する。

#### 【0020】

本発明の実施形態によれば、ドキュメント310に記入されたテキストおよびXMLマークアップは、種々の異なるファイル形式に従って、および、ドキュメント310が作成されたアプリケーション305の本来(native)のプログラミング言語に従って、保存することができる。たとえば、テキストおよびXMLマークアップは、文書処理アプリケーション、表計算アプリケーションなどに従って保存することができる。または、ドキュメント310に記入されたテキストおよびXMLマークアップは、XMLフォーマットとして保存することができ、テキストあるいはデータ、任意の適用されたXMLマークアップ、および、フォント、スタイル、パラグラフ構造などの任意のフォーマッティングは、XML表現として保存することができる。したがって、XMLとして保存されたデータを理解することができる下流のアプリケーション又はサードパーティのアプリケーションは、XML表現として保存されたテキストあるいはデータを開き、利用(consume)することができる。XMLとして、ドキュメント310のテキストおよびXMLマークアップおよび関連付けられたフォーマッティングおよび他の属性の保存については、特許文献1に記載されている。

#### 【0021】

図3で例示されたような、テキストあるいはデータに適用されるXMLマークアップ要素(タグ)に関する定義上の枠組みを提供するために、XMLスキーマファイルが創られ、そのファイルは、マークアップされ格納されたデータのユーザーおよび利用者が、ドキュメントの作成者により設計されたXMLのタグ付けの定義を理解することを可能にするために必要な情報を含む。現状技術においては、XSDファイルとも言及されるスキーマファイルのそれぞれは、好ましくは、所与のスキーマファイルに従って、ドキュメントに適用される場合があるすべてのXML要素(タグ)の一覧表を含む。たとえば、図3で例示されるように、スキーマファイル330は、XML要素の属性、あるいはスキーマファイルに従ってXML要素で注釈されている可能性のあるテキストあるいはデータに関連付けられた制約、および/あるいは規則を含むドキュメント310に適用できるある種のXML要素の定義を含むスキーマファイルである場合がある。たとえば、図3で例示されるスキーマファイル330を参照すると、スキーマファイルは、そのスキーマファイルが<introCARD>というルート要素を含むネーム空間「intro」により識別される。

#### 【0022】

スキーマファイル330によれば、<introCARD>要素は、スキーマファイルのルート要素の役割と、2つの子要素<title>と<body>の親要素の役割とを果たす。当業者にはよく知られているように、いくつかの親要素は単一のルート要素の下で定義でき、いくつかの子要素はそれぞれの親要素の下で定義できる。しかしながら通常は、所与のスキーマファイル330は唯一のルート要素を含む。さらに図3を参照すると、スキーマファイル330はそれぞれ<title>と<body>要素に対する属性340および345を含む。属性340および345は、ドキュメント310のテキストあるいはデータに各要素を適用することに関連付けられた定義あるいは規則をさらに提供することができる。たとえば、<title>要素で注釈されたテキストは長さにおいて25文字かそれ未満でなければならないことを、属性345は定義している。したがって、もし長さにお

10

20

30

40

50

いて 25 文字を越えるテキストが <title> 要素あるいはタグで注釈される場合は、スキーマファイル 330 に含まれた定義によれば、そのテキストの試みられた(attempted)注釈は無効となるであろう。

#### 【0023】

XML 要素への属性としてこのような定義あるいは規則を適用することにより、スキーマの作成者は所与のスキーマファイルに関連付けられたドキュメントに含まれるデータの構造を指示(dictate)することができる。たとえば、もしレジュメドキュメント(resume document)に適用される XML マークアップを定義するスキーマファイル 330 の作成者が、そのレジュメドキュメントの経験欄(experience section)がわずか 4 件の現在又は以前のジョブエントリ(job entries)を含むことを望むならば、スキーマファイル 330 の作成者は、<experience> 要素の属性を定義することができ、たとえば、経験のテキストがスキーマファイル 330 にしたがって妥当になるためには、わずか 4 件の現在又は以前のジョブエントリを<experience> タグの間に記入することができるよう 10 にすることが可能である。当業者によく知られているように、スキーマファイル 330 は、ドキュメント 310 に添付されたスキーマファイルで定義された許容可能な XML マークアップのアプリケーションのために、所与のドキュメント 310 に添付されるか、あるいはさもなければ関連付けることができる。1 つの実施形態によれば、添付され又は関連付けられたスキーマファイル 330 の XML 要素でマークアップされたドキュメント 310 は、添付され又は関連付けられたスキーマファイル 330 を識別するネーム空間に 20 関連付けられた統一的リソース識別子(URI: Uniform Resource Identifier)を指し示すことにより、添付され又は関連付けられたスキーマファイルを指し示すことができる。

#### 【0024】

本発明の実施形態によれば、ドキュメント 310 は、添付されたスキーマファイルを複数持つことができる。すなわち、2 以上のスキーマファイルから XML マークアップの注釈に関する枠組みを提供するために、ドキュメント 310 の作成者は、2 以上のスキーマファイル 330 をドキュメント 310 に関連付けるか、又は添付することができる。たとえば、ドキュメント 310 は、財務データに関連付けられたテキストあるいはデータを含むことができる。ドキュメント 310 の作成者は、複数の金融機関に関連付けられた XML マークアップおよび定義を含む XML スキーマファイル 330 を関連付けることを望むことができる。したがって、ドキュメント 310 の作成者は、1 つあるいは複数の金融機関からの XML スキーマファイル 330 をドキュメント 310 に関連付けることができる。同じく、所与の XML スキーマファイル 330 を、財務データを所望のフォーマットに設定するためのテンプレート等の特定のドキュメント構造に関連付けることができる。 30

#### 【0025】

本発明の実施形態によれば、XML スキーマファイルおよび関連付けられたドキュメントソリューションの集合が、ドキュメント 310 から分離して位置するネーム空間あるいはスキーマライブラリにおいて維持することができる。ドキュメント 310 は、ドキュメント 310 に添付された、さもなければ関連付けられた 1 つあるいは複数のスキーマファイルに関連付けられたネーム空間あるいはスキーマライブラリにおける URI を指し示すポインタを含むことができる。ドキュメント 310 が 1 つあるいは複数の関連付けられたスキーマファイルからの情報を必要とする場合には、ドキュメント 310 は、ネーム空間又はスキーマライブラリを指し示し、必要とされるスキーマ定義を獲得する。ネーム空間又はスキーマライブラリの動作の効用については、特許文献 2 および特許文献 3 に記載されている。ネーム空間あるいはスキーマライブラリから、XML スキーマファイルや関連付けられたソリューション等のソフトウェア構成要素をダウンロードするメカニズムは、特許文献 4 に記載されている。 40

#### 【0026】

さらに図 3 を参照して、上述されたように、ドキュメント 310 に適用された XML マークアップを、ドキュメント 310 に添付され又はさもなければ関連付けられた XML スキーマファイル 330 に対比して、妥当性検査を行うスキーマ妥当性検査機能モジュール

350を例示する。上述されたように、スキーマファイル330は受諾可能なXML要素および関連付けられた属性を設定(set out)し、かつ関連付けられたスキーマファイル330からのXMLマークアップによるドキュメント310の妥当な注釈に関する規則を定義する。たとえば、スキーマファイル330に示されるように、2つの子要素<title>および<body>は、ルートあるいは親要素<internal CARD>の下で定義される。子要素<title>および<body>に関連付けられたテキストの受諾可能な文字列の長さを定義している属性340、345がまた例示されている。上述されたように、ユーザーがドキュメントに添付された又はさもなければ関連付けられたスキーマファイル330からのXMLマークアップでドキュメントに対して、スキーマファイル330に含まれるXMLマークアップ定義に違反して、注釈を付けようと試みた場合には、無効あるいはエラー状態が提示(present)される。たとえば、ユーザーが25文字を越えて名称文字列を記入しようと試みる場合、そのテキストエントリーは、スキーマファイル330の<title>要素の最大文字長属性に違反する。ドキュメント310に適用されたXMLマークアップの、関連付けられたスキーマファイル330に対する、妥当性検査を行うために、スキーマ妥当性検査モジュール350が利用される。当業者により理解されるべきであるように、妥当性検査モジュール350は、ドキュメント310に記入されたXMLマークアップおよび関連付けられたテキストを、そのXMLマークアップおよび関連付けられたテキストがドキュメント310に記入された際に、関連付けられたあるいは添付されたXMLスキーマファイル330と比較するのに十分な、コンピュータで実行可能な命令を含むソフトウェアモジュールである。

10

20

30

40

#### 【0027】

本発明の実施形態によれば、スキーマ妥当性検査モジュール350は、ドキュメント310に適用されたそれぞれのXMLマークアップ要素および関連付けられたテキストあるいはデータを、添付されたあるいは関連付けられたスキーマファイル330に対して比較して、それぞれの要素および関連付けられたテキストあるいはデータが、添付されたスキーマファイル330により設定(set out)された規則および定義に適合するか否かを判定する。たとえば、ユーザーが<title>要素320により注釈を付けられた文字列に25文字(character)を超えて記入しようとする場合、スキーマ妥当性検査モジュールはそのテキストの文字列を、添付されているスキーマファイル330のテキスト文字列属性340と比較し、ユーザーにより記入されたそのテキストの文字列が最大許容されているテキスト文字列の長さを超えると判定する。したがって、エラーメッセージあるいはダイアログがユーザーに提示され、ユーザーにより記入されているテキストの文字列が、添付されているスキーマファイル330に従って、最大の許容文字長を超える、とユーザーに警告する。同じく、もしユーザーが<title>および<body>要素の間にXMLマークアップ要素を加えようとする場合、スキーマ妥当性検査モジュール350はユーザーにより適用されたそのXMLマークアップ要素が、添付されたスキーマファイル330に従って<title>および<body>要素の間に許容された有効な要素ではないと判定する。したがって、スキーマ妥当性検査モジュール350は、無効なXMLマークアップのユーザーに警告するために、そのユーザーに対してエラーメッセージあるいはダイアログを生成する。

40

#### 【0028】

(ネーム空間 / スキーマライブラリのプログラム可能なオブジェクトモデル)

図3を参照し上述されたように、ドキュメント310に拡張可能なマークアップ言語(XML)の表示を適用するための定義上のかつ規則指向(rule-oriented)の枠組みを提供するために、ドキュメント310に対する所与のスキーマファイル330に対応するXMLマークアップ要素のアプリケーションを管理(govern)する定義および規則を設定するために、1つあるいは複数のスキーマファイル330がそのドキュメントに関連付けられ、あるいは添付される場合がある。記述されたように、複数のXMLスキーマファイルおよび他のドキュメントソリューション、たとえば予め構造化されたテンプレート、が単一のXMLドキュメント310に添付され、あるいは関連付けられる場合がある。さらに、上

50

述されたように、ネーム空間識別番号(Namespace identification)により識別される複数の異なる XMLスキーマファイルおよび複数のドキュメントソリューションがドキュメント310から分離されたネーム空間あるいはスキーマライブラリに格納される場合がある。本発明の実施形態によれば、ネーム空間あるいはスキーマライブラリに含まれる種々のネーム空間識別子に関連付けられた、スキーマファイルのネーム空間および関連付けられた定義、規則、リソース、およびソリューションをカスタマイズし、又はさもなければ操作するために、ユーザーは、1つあるいは複数のドキュメント310に関連付けられたネーム空間あるいはスキーマライブラリをプログラムで呼び出すことができる。

### 【0029】

図4は、ドキュメント310、ネーム空間あるいはスキーマライブラリ400、およびサードパーティアプリケーション450の間の相互作用を例示するブロック図である。本発明の実施形態によれば、ユーザーは、ネーム空間ライブラリ400で識別されたスキーマファイルに関連付けられた個別のスキーマファイル410、430あるいはリソース420、440の内容あるいは動作を修正するために、オブジェクト指向メッセージ要求の組、あるいはアプリケーションプログラムインタフェースの集合470によりプログラムでネーム空間ライブラリ400を呼び出すことができる。ユーザーは、オブジェクト指向メッセージ呼び出しの組により、アプリケーション305から、あるいはサードパーティプログラム450からネーム空間ライブラリと通信することができ、かつ、サードパーティプログラムはC、C++、C#、ビジュアルベーシック等の種々のプログラミング言語を用いて開発することができる。

### 【0030】

アプリケーションプログラムインタフェース470の組を通してネーム空間ライブラリへのアクセスを持つことにより、ユーザーは、1つあるいは複数の追加のXMLスキーマファイルあるいはネーム空間をXMLデータにプログラムで関連付けることができ、そして逆に、ユーザーは、ドキュメント310に適用された1つあるいは複数のXMLスキーマファイルおよびXMLデータあるいはマークアップの間の既存の関連付けを、検出し削除することができる。ユーザーはまたプログラムで、拡張可能スタイルシート言語変換(XSLT: Extensible Stylesheet Language Transformation)を、ドキュメントに適用されたXMLデータに関連付けることができる。逆に、ユーザーは、ドキュメント310に適用されたXMLデータとの関連付けから既存のXSLT変換を検出し削除することができる。さらに、ユーザーは、他のファイルおよび実行可能なソフトウェアアプリケーションを、ドキュメント310に適用されたXMLデータにプログラムで関連付けて、XMLデータを持つ他のソフトウェアアプリケーションおよびファイルの既存の関連付けを検出し削除することができる。

### 【0031】

たとえば、ネーム空間ライブラリ400で例示されるネーム空間430は、レジュメドキュメントテンプレートのために予めフォーマットされた構造より構成されたソリューションを含むことができる。そのソリューションがドキュメント310に適用されるとき、レジュメントテンプレートドキュメントの作成者により設計された関連付けられたスキーマ定義および規則は、ドキュメント310に記入されたXMLマークアップおよび関連付けられたテキストに適用される。レジュメドキュメントテンプレートに関連付けられたスキーマファイルがレジュメドキュメントの経験欄が少なくとも3つの過去の、あるいは現在の職種(job description)を持たねばならないことを要求する場合には、そのスキーマ定義は、ドキュメント310に適用され、XMLドキュメント310がスキーマ妥当性検査モジュール350により妥当であるとされるために、少なくとも3つの職種が経験欄に後(subsequent)のユーザーにより記入されなければならない。この例について続けると、このようなレジュメドキュメントテンプレートのスキーマファイルがドキュメント310に関連付けられ、かつユーザーがそのスキーマファイルのドキュメント310との関連付けを削除することを望む場合には、ネーム空間ライブラリ450に、あるいはアプリケーション305に対して、ドキュメント310から、レジュメドキュメントテンプレートのスキ

10

20

30

40

50

ーマファイルの関連付けの削除を指示するために、提供されたアプリケーションプログラムインタフェースによりオブジェクト指向メッセージ呼び出しを送ることにより、そのユーザーはサードパーティプログラムからプログラムでそれをすることができる。

#### 【0032】

以下はオブジェクト指向メッセージ呼び出しあるいはアプリケーションプログラムインタフェースを備えるオブジェクトおよび関連付けられた属性の記述であり、これにより、上述されたように、ユーザーは、ネーム空間ライブラリ400にプログラムでアクセスすることができる。以下に規定されたそれぞれのオブジェクトおよび関連付けられたプロパティの次に、そのオブジェクトあるいは関連付けられたプロパティの動作および機能を説明する。

10

#### 【0033】

(アプリケーションオブジェクト)

以下はオブジェクトのメソッドおよびプロパティである。

#### 【0034】

.XMLNamespaces プロパティ

アプリケーションで使用可能なネーム空間ライブラリを表す XMLNamespace の集合(collection)に対する読み取り専用のポインタ。

#### 【0035】

(XMLNamespace 集合オブジェクト) XMLNamespace オブジェクトに対するアクセスを提供するオブジェクト。それは、ネーム空間ライブラリを示す。集合のそれぞれの XMLNamespace オブジェクトは、ネーム空間ライブラリの单一かつ一意なネーム空間を表す。以下はこのオブジェクトのメソッドおよびプロパティである。

20

#### 【0036】

.Add()メソッド

新しい XMLNamespace オブジェクトを作成し、その集合に追加するメソッド。それは、ネーム空間ライブラリで新しいネーム空間を登録するために用いられる。それは、新しい XMLNamespace オブジェクトを返す。それは、次のパラメーターを受け入れることができる。

*Path* - ネーム空間のためのスキーマファイルに対するポインタ。ポインタは文字列として表されるファイルパスであり得る。

*NameSpaceURI* - スキーマを表すネーム空間の URI。URI はテキストの文字列であり得る。

*Alias* - プログラムが指定することが可能なネーム空間に対する代替の(より使い勝手の良い)名前を表すテキスト文字列。

*InstallForAllUsers* - ネーム空間ライブラリの新しいネーム空間がコンピュータのすべてのユーザーにとって、あるいは現在のユーザーのみにとって利用可能であるべきかを示すフラグ。

30

#### 【0037】

.Application プロパティ

このオブジェクトモデルのアプリケーションを表すアプリケーションオブジェクトに対する読み取り専用のポインタ。

40

#### 【0038】

.Count プロパティ

ネーム空間ライブラリに登録されたネーム空間の数を返す読み取り専用のプロパティ。プロパティは XMLNamespace 集合での XMLNamespace オブジェクトの合計数と同じである。

#### 【0039】

.Creator プロパティ

オブジェクトの作成者に対する読み取り専用のポインタ。

50

## 【0040】

## .InstallManifest()メソッド

ネーム空間ライブラリにネーム空間を登録するソリューションマニフェストをインストールするメソッド。それは、次のパラメーターを受け入れることができる。

Path - マニフェストのためのマニフェストファイルに対するポインタ。このポインタはテキスト文字列により表されるファイルパスであり得る。

InstallForAllUsers - マニフェストによりネーム空間ライブラリにインストールされた新しいネーム空間がコンピュータのすべてのユーザーにとって、あるいは現在のユーザーのみにとって利用可能であるべきかを示すフラグ。

## 【0041】

## .Item()メソッド

数字のインデックスあるいは検索キーワードを用いてこの集合の個別のメンバーにアクセスするメソッド。このメソッドは次のパラメーターを受け入れることができる。

Index - ネーム空間ライブラリにおいて要求された XMLNamespace オブジェクトの位置を表す数。Index は要求されたネーム空間のエイリアス(alias)あるいは URI を表すテキスト文字列でもあり得る。

## 【0042】

## .Parent プロパティ

集合の親オブジェクトを返す読み取り専用のプロパティ。このプロパティは XMLNamespace 集合をアクセスするアプリケーションに対してポインタを返す。

## 【0043】

( XMLNamespace オブジェクト) - ネーム空間ライブラリでの個別のネーム空間エントリー(および XMLNamespace 集合での個別の項目)を表すオブジェクト。以下はこのオブジェクトのメソッドおよびプロパティである。

## 【0044】

## .Alias プロパティ

プログラマがネーム空間に関連付けるエイリアスを制御するプロパティ。それは、次のパラメーターをサポートすることができる。

AllUsers - エイリアスがすべてのユーザーにとって、あるいは単に現在(current)のユーザーにとって利用可能であるかを示すフラグ。

## 【0045】

## .Application プロパティ

このオブジェクトモデルのアプリケーションを表すアプリケーションオブジェクトに対する読み取り専用のポインタ。

## 【0046】

## .AttachToDocument()メソッド

オブジェクトにより表されたネーム空間のスキーマを、選択されたドキュメントに添付するメソッド。それは、次のパラメーターをサポートする。

Document - スキーマを添付するべく要求されたドキュメントに対するポインタ。

## 【0047】

## .Creator プロパティ

オブジェクトの作成者に対する読み取り専用のポインタ。

## 【0048】

## .DefaultTransform プロパティ

このネーム空間に関連付けられたデフォルト XSLT 変換を指し示すプロパティ。それは、次のパラメーターをサポートすることができる。

AllUsers - デフォルト変換設定がマシンのすべてのユーザーに、あるいは現在のユーザーのみに影響を与えるべきかを示すフラグ。

## 【0049】

10

20

30

40

50

**.Delete()メソッド**

XML Namespaceオブジェクトを集合から削除し、そしてそれを破壊するメソッドであって、このオブジェクトにより表されたネーム空間の関連付けをネーム空間ライブラリから効果的に削除する。

**【0050】****.Locationプロパティ**

XML Namespaceオブジェクトにより表されたネーム空間に関連付けされたスキーマの位置を制御する読み取り専用のプロパティ。それは、次のパラメーターをサポートすることができる。

A11Users - スキーマ位置設定がマシンのすべてのユーザーに、あるいは現在のユーザーのみに影響を与えるべきかを示すフラグ。 10

**【0051】****.Parentプロパティ**

XML Namespaceオブジェクトの親オブジェクトを返す読み取り専用のプロパティ。このプロパティはそのオブジェクトがメンバーとなる XML Namespace 集合に対してポインタを返す。

**【0052】****.URIプロパティ**

オブジェクトにより表されるネーム空間のURIを返す読み取り専用のプロパティ。

**【0053】****.XSLTTransformプロパティ**

オブジェクトにより表されたネーム空間に関連付けられたXSLT変換を表すXSLT transform集合に対する読み取り専用のポインタ。 20

**【0054】**

(XSLT transformオブジェクト) - それぞれがネーム空間ライブラリでのネーム空間に関連付けられた单一かつ一意なXSLT変換を表すXSLT transformオブジェクトへのアクセスを提供するオブジェクト。以下はこのオブジェクトのメソッドおよびプロパティである。

**【0055】****.Add()メソッド**

新しいXSLT transformオブジェクトを作成し、その集合に追加するメソッド。それは、新しいXSLT変換をネーム空間ライブラリでのネーム空間に関連付けるために用いられる。それは、新しいXSLT transformオブジェクトを返す。それは、次のパラメーターを受け入れることができる。 30

Location - XSLTファイルに対するポインタであり、テキスト文字列として表されるファイルパスであり得る。

Alias - プログラマが指定することが可能なXSLT変換に対する代替の(より使い勝手の良い)名前を表すテキスト文字列。

InstallForAllUsers - ネーム空間ライブラリでの新しいネーム空間がコンピュータのすべてのユーザーにとって、あるいは現在のユーザーのみにとって利用可能であるべきかを示すフラグ。 40

**【0056】****.Applicationプロパティ**

このオブジェクトモデルのアプリケーションを表すアプリケーションオブジェクトに対する読み取り専用のポインタ。

**【0057】****.Countプロパティ**

ネーム空間ライブラリでの所与のネーム空間に対する登録されたXSLT変換の数を返す読み取り専用のプロパティ。それは、XSLT transform集合でのXSLT transformオブジェクトの合計の数と同じである。 50

## 【0058】

**.C r e a t o r** プロパティ

オブジェクトの作成者に対する読み取り専用のポインタ。

## 【0059】

**.I t e m()** メソッド

数字のインデックスあるいは検索キーワードを用いてこの集合の個別のメンバーにアクセスするメソッド。それは、次のパラメーターを受け入れることができる。

**I n d e x** - ネーム空間ライブラリにおいて要求された X S L T r a n s f o r m オブジェクトの位置を表す数。インデックスは、要求された X S L T 変換のエイリアスを表すテキスト文字列でもあり得る。

10

## 【0060】

**.P a r e n t** プロパティ

その集合の親オブジェクトを返す読み取り専用のプロパティ。このプロパティは X S L T r a n s f o r m 集合をアクセスするアプリケーションに対してポインタを返す。

## 【0061】

( X S L T r a n s f o r m オブジェクト) - ネーム空間ライブラリでのネーム空間に関連付けられた X S L T 変換を表すオブジェクト。以下はこのオブジェクトのメソッドおよびプロパティである。

## 【0062】

**.A l i a s** プロパティ

プログラマがネーム空間ライブラリで X S L T 変換に関連付けたエイリアスを制御するプロパティ。それは、次のパラメーターをサポートすることができる。

**A l l U s e r s** - エイリアス(別名)がすべてのユーザーにとって、あるいは単に現在のユーザーにとって利用可能であるかを示すフラグ。

20

## 【0063】

**.A p p l i c a t i o n** プロパティ

オブジェクトモデルのアプリケーションを表すアプリケーションオブジェクトに対する読み取り専用のポインタ。

## 【0064】

**.C r e a t o r** プロパティ

30

オブジェクトの作成者に対する読み取り専用のポインタ。

## 【0065】

**.D e l e t e()** メソッド

X S L T r a n s f o r m オブジェクトを集合から削除し、そしてそれを破壊するメソッドであって、ネーム空間ライブラリでの X S L T 変換とそのネーム空間との間の関連付けを効果的に削除する。

## 【0066】

**.L o c a t i o n** プロパティ

所与のネーム空間に関連付けられ、X S L T r a n s f o r m オブジェクトにより表された X S L T 変換の位置を制御する読み取り専用のプロパティ。それは、次のパラメーターをサポートすることができる。

40

**A l l U s e r s** - X S L T 変換位置設定がマシンのすべてのユーザーに、あるいは現在のユーザーのみに影響を与えるべきであるかを示すフラグ。

## 【0067】

**.P a r e n t** プロパティ

X S L T r a n s f o r m オブジェクトの親オブジェクトを返す読み取り専用のプロパティ。このプロパティは、オブジェクトがメンバーになっている X S L T r a n s f o r m 集合に対してポインタを返す。

## 【0068】

本明細書に記述されたように、ユーザーが、ネーム空間あるいはスキーマライブラリに

50

おいて識別され、あるいは含まれるリソースと、1つあるいは複数の関連付けられたドキュメントとの関連付けをカスタマイズし、あるいはさもなければ修正するために、拡張可能なマークアップ言語（XML）のネーム空間あるいはスキーマライブラリで識別されたリソースを、プログラムで呼び出すことを可能にする方法およびシステムが提供される。

#### 【0069】

当業者には、本発明における発明の範囲あるいは精神から逸脱することなく種々の修正あるいは変更が為される可能性があることは明白である。本発明の他の実施形態はここに開示された発明の明細書および実施例を考慮することにより当業者にとり明白になる。

#### 【図面の簡単な説明】

#### 【0070】

【図1】本発明に対する例示的な動作環境を提供する、コンピュータ処理システムおよび関連付けられた周辺機器およびネットワーク装置の単純化されたブロック図である。

【図2】オブジェクト指向プログラミングモデルによる、ソフトウェアオブジェクト間の相互作用を例示する単純化されたブロック図である。

【図3】ドキュメント、添付されたスキーマファイル、およびスキーマ妥当性検査機能モデルの間の相互作用を例示するブロック図である。

【図4】ドキュメント、ネーム空間あるいはスキーマライブラリ、およびサードパーティソフトウェアアプリケーションの間の相互作用を例示するブロック図である。

#### 【符号の説明】

#### 【0071】

4 中央処理装置

6 システムメモリ

8 ランダムアクセスメモリ

10 読み取り専用メモリ

14 大容量記憶装置

16 オペレーティングシステム

18 TCP/IP ネットワーク

20 ネットワークインターフェースユニット

22 入力／出力コントローラ

26 言語設定

305 アプリケーションプログラム

310 ドキュメント

330 スキーマファイル

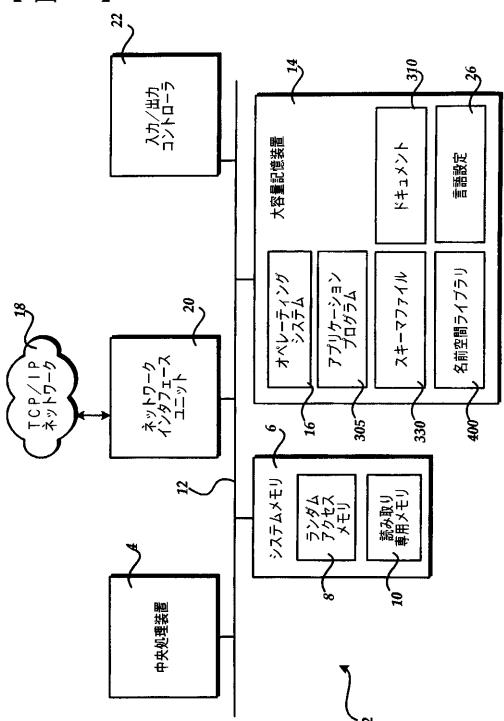
400 ネーム空間ライブラリ

10

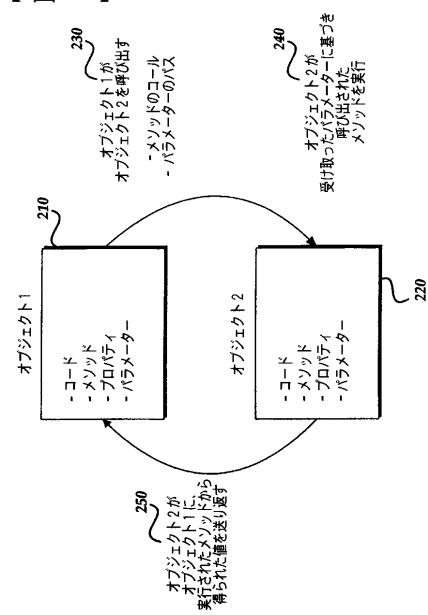
20

30

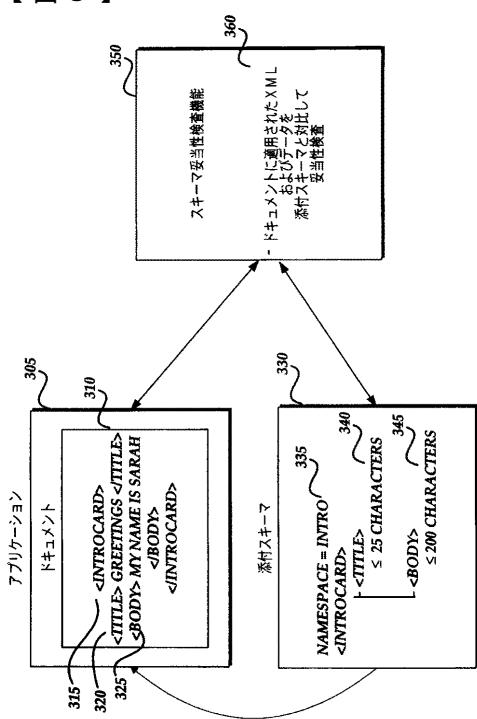
【図1】



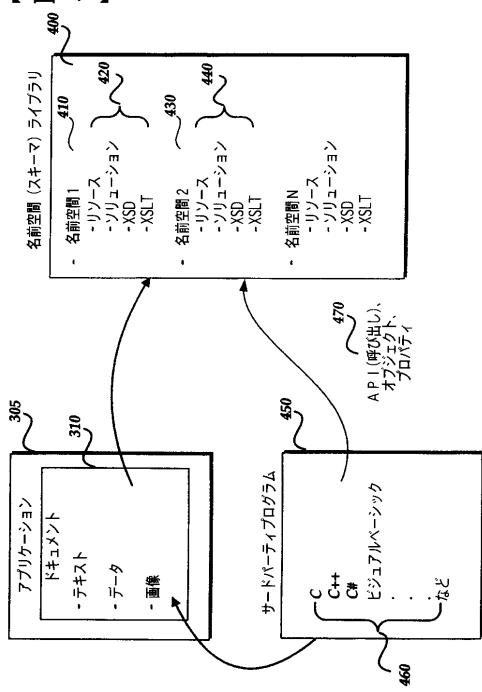
【図2】



【図3】



【図4】



---

フロントページの続き

(72)発明者 マーク サンダーランド

アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内

(72)発明者 マーシン サヴィッキ

アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内

(72)発明者 ロバート エー.リトル

アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内

F ターム(参考) 5B009 QA06 TA06

5B076 DD05 DD06