

(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2008-234370

(P2008-234370A)

(43) 公開日 平成20年10月2日(2008.10.2)

(51) Int.Cl.		F I			テーマコード (参考)
G06F 17/21	(2006.01)	G06F 17/21	501T		5B009
G06F 12/00	(2006.01)	G06F 12/00	547Z		5B082
					5B109

審査請求 未請求 請求項の数 8 O L (全 22 頁)

(21) 出願番号 特願2007-73756 (P2007-73756)
 (22) 出願日 平成19年3月21日 (2007.3.21)

(71) 出願人 390024350
 株式会社ジャストシステム
 徳島県徳島市川内町平石若松108番地4
 (74) 代理人 100105924
 弁理士 森下 賢樹
 (74) 代理人 100109047
 弁理士 村田 雄祐
 (72) 発明者 高原 敬
 徳島県徳島市川内町平石若松108番地の
 4 株式会社ジャストシステム内
 Fターム(参考) 5B009 PB02 PB04 QA06
 5B082 AA12
 5B109 PB02 PB04 QA06

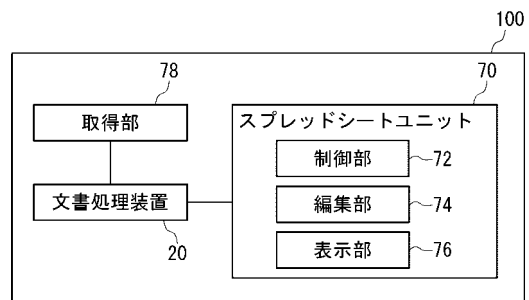
(54) 【発明の名称】 文書処理装置及び文書処理方法

(57) 【要約】

【課題】マークアップ言語により構造化されたデータをスプレッドシート形式で処理する技術を提供する。

【解決手段】文書処理装置20は、マークアップ言語により記述された文書をスプレッドシート形式で処理するスプレッドシートユニット70を備える。スプレッドシートユニット70は、文書中に含まれる、スプレッドシート用のタグセットで記述されたデータを、スプレッドシート形式で表示する表示部76と、スプレッドシート中に、文書中に含まれる、スプレッドシート用のタグセット以外のタグセットで記述されたデータを挿入する編集部74と、を備える。

【選択図】 図11



【特許請求の範囲】**【請求項 1】**

文書中に含まれる、スプレッドシート用のタグセットで記述されたデータを、スプレッドシート形式で表示する表示部と、

前記スプレッドシート中に、前記文書中に含まれる、前記スプレッドシート用のタグセット以外のタグセットで記述されたデータを挿入する編集部と、

を備えることを特徴とする文書処理装置。

【請求項 2】

前記編集部は、データの挿入先として指定されたセルに対応する前記スプレッドシート用のタグセットの要素に、挿入データとして指定された前記スプレッドシート用のタグセット以外のタグセットで記述されたデータを指し示すポイント情報を格納することを特徴とする請求項 1 に記載の文書処理装置。

10

【請求項 3】

前記編集部は、前記挿入データが編集されたことを示す通知を受けると、編集された前記挿入データを参照して、前記挿入データが挿入されたセルに対応する要素を編集することを特徴とする請求項 1 又は 2 に記載の文書処理装置。

【請求項 4】

前記編集部は、前記挿入データを記述したタグセットごとに、前記タグセットで記述されたデータを利用した専用の関数を定義し、前記関数を含む式の入力を受け付けたときに、前記式を演算した結果を挿入することを特徴とする請求項 1 から 3 のいずれかに記載の文書処理装置。

20

【請求項 5】

前記編集部は、前記文書中に含まれる複数の要素により構成されるフラグメントを挿入する指定を受け付けて、前記フラグメントを前記挿入データとして挿入することを特徴とする請求項 1 から 4 のいずれかに記載の文書処理装置。

【請求項 6】

前記編集部は、前記挿入データの指定を受け付けるときに、前記文書中に含まれるデータをツリー状に表示した画面を提示し、ユーザから前記挿入データの指定を受け付けることを特徴とする請求項 1 から 5 のいずれかに記載の文書処理装置。

【請求項 7】

文書中に含まれる、スプレッドシート用のタグセットで記述されたデータを、スプレッドシート形式で表示するステップと、

前記スプレッドシート中に、前記文書中に含まれる、前記スプレッドシート用のタグセット以外のタグセットで記述されたデータを挿入するステップと、

を備えることを特徴とする文書処理方法。

30

【請求項 8】

文書中に含まれる、スプレッドシート用のタグセットで記述されたデータを、スプレッドシート形式で表示する機能と、

前記スプレッドシート中に、前記文書中に含まれる、前記スプレッドシート用のタグセット以外のタグセットで記述されたデータを挿入する機能と、

をコンピュータに実現させることを特徴とするプログラム。

40

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、文書処理技術に関し、特に、マークアップ言語により記述された文書を処理する文書処理装置及び方法に関する。

【背景技術】**【0002】**

X M L は、ネットワークなどを介して他者とデータを共有するのに適した形式として注目されており、X M L 文書を作成、表示、編集するためのアプリケーションが開発されて

50

いる（たとえば、特許文献1参照）。XML文書は、文書型定義などにより定義されたボキャブラリ（タグセット）に基づいて作成されている。

【特許文献1】特開2001-290804号公報

【発明の開示】

【発明が解決しようとする課題】

【0003】

ボキャブラリは、任意に定義することが許されており、理論上、無限に多くのボキャブラリが存在しうる。本出願人は、このように柔軟な設計が許されるXML文書を適切に処理することが可能な技術を開発した。そして、XMLの長所を十分に生かした利便性の高いデータ処理技術を想到するに至った。

10

【0004】

本発明はこうした状況に鑑みてなされたものであり、その目的は、マークアップ言語により構造化されたデータを適切に処理する技術を提供することにある。

【課題を解決するための手段】

【0005】

本発明のある態様は、文書処理装置に関する。この文書処理装置は、文書中に含まれる、スプレッドシート用のタグセットで記述されたデータを、スプレッドシート形式で表示する表示部と、前記スプレッドシート中に、前記文書中に含まれる、前記スプレッドシート用のタグセット以外のタグセットで記述されたデータを挿入する編集部と、を備えることを特徴とする。

20

【0006】

前記編集部は、データの挿入先として指定されたセルに対応する前記スプレッドシート用のタグセットの要素に、挿入データとして指定された前記スプレッドシート用のタグセット以外のタグセットで記述されたデータを指し示すポインタ情報を格納してもよい。

【0007】

前記編集部は、前記挿入データが編集されたことを示す通知を受けると、編集された前記挿入データを参照して、前記挿入データが挿入されたセルに対応する要素を編集してもよい。

【0008】

前記編集部は、前記挿入データを記述したタグセットごとに、前記タグセットで記述されたデータを利用した専用の関数を定義し、前記関数を含む式の入力を受け付けたときに、前記式を演算した結果を挿入してもよい。

30

【0009】

前記編集部は、前記文書中に含まれる複数の要素により構成されるフラグメントを挿入する指定を受け付けて、前記フラグメントを前記挿入データとして挿入してもよい。

【0010】

前記編集部は、前記挿入データの指定を受け付けるときに、前記文書中に含まれるデータをつリー状に表示した画面を提示し、ユーザから前記挿入データの指定を受け付けてもよい。

【0011】

本発明の別の態様は、文書処理方法に関する。この文書処理方法は、文書中に含まれる、スプレッドシート用のタグセットで記述されたデータを、スプレッドシート形式で表示するステップと、前記スプレッドシート中に、前記文書中に含まれる、前記スプレッドシート用のタグセット以外のタグセットで記述されたデータを挿入するステップと、を備えることを特徴とする。

40

【0012】

なお、以上の構成要素の任意の組合せ、本発明の表現を方法、装置、システムなど間で変換したものもまた、本発明の態様として有効である。

【発明の効果】

【0013】

50

本発明によれば、マークアップ言語により構造化されたデータを適切に処理する技術を提供することができる。

【発明を実施するための最良の形態】

【0014】

(前提技術)

図1は、前提技術に係る文書処理装置20の構成を示す。文書処理装置20は、文書内のデータが階層構造を有する複数の構成要素に分類された構造化文書进行处理するが、本前提技術では構造化文書の一例としてXML文書进行处理する例について説明する。文書処理装置20は、主制御ユニット22、編集ユニット24、DOMユニット30、CSSユニット40、HTMLユニット50、SVGユニット60、及び変換部の一例であるVCユニット80を備える。これらの構成は、ハードウェアコンポーネントでいえば、任意のコンピュータのCPU、メモリ、メモリにロードされたプログラムなどによって実現されるが、ここではそれらの連携によって実現される機能ブロックを描いている。したがって、これらの機能ブロックがハードウェアのみ、ソフトウェアのみ、またはそれらの組合せによっていろいろな形で実現できることは、当業者には理解されるところである。

10

【0015】

主制御ユニット22は、プラグインのロードや、コマンド実行のフレームワークを提供する。編集ユニット24は、XML文書を編集するためのフレームワークを提供する。文書処理装置20における文書の表示及び編集機能は、プラグインにより実現されており、文書の種別に応じて必要なプラグインが主制御ユニット22又は編集ユニット24によりロードされる。主制御ユニット22又は編集ユニット24は、処理対象となるXML文書の名前空間を参照して、XML文書がいずれのボキャブラリにより記述されているかを判別し、そのボキャブラリに対応した表示又は編集用のプラグインをロードして表示や編集を実行させる。例えば、文書処理装置20には、HTML文書の表示及び編集を行うHTMLユニット50、SVG文書の表示及び編集を行うSVGユニット60など、ボキャブラリ(タグセット)ごとに表示系及び編集系がプラグインとして実装されており、HTML文書を編集するときはHTMLユニット50が、SVG文書を編集するときはSVGユニット60が、それぞれロードされる。後述するように、HTMLとSVGの双方の構成要素を含む複合文書が処理対象となっている場合は、HTMLユニット50とSVGユニット60の双方がロードされる。

20

30

【0016】

このような構成によれば、ユーザは、必要な機能のみを選択してインストールし、後から適宜機能を追加又は削除することができるので、プログラムを格納するハードディスクなどの記録媒体の記憶領域を有効に活用ことができ、また、プログラム実行時にも、メモリの浪費を防ぐことができる。また、機能拡張性に優れており、開発主体としても、プラグインの形で新たなボキャブラリに対応することが可能なので開発が容易となり、ユーザとしても、プラグインの追加により容易かつ低コストにて機能を追加することができる。

【0017】

編集ユニット24は、ユーザインターフェースを介してユーザから編集指示のイベントを受け付け、そのイベントを適切なプラグインなどに通知するとともに、イベントの再実行(リドゥ)又は実行の取消(アンドゥ)などの処理を制御する。

40

【0018】

DOMユニット30は、DOM提供部32、DOM生成部34、及び出力部36を含み、XML文書をデータとして扱うときのアクセス方法を提供するために定められた文書オブジェクトモデル(Document Object Model: DOM)に準拠した機能を実現する。DOM提供部32は、編集ユニット24に定義されているインターフェースを満たすDOMの実装である。DOM生成部34は、XML文書からDOMツリーを生成する。後述するように、処理対象となるXML文書が、VCユニット80により他のボキャブラリにマッピングされる場合は、マッピング元のXML文書に対応するソースツリーと、マッピング先の

50

X M L 文書に対応するデスティネーションツリーが生成される。出力部 3 6 は、例えば編集終了時に、D O M ツリーを X M L 文書として出力する。

【 0 0 1 9 】

C S S ユニット 4 0 は、C S S 解析部 4 2、C S S 提供部 4 4、及びレンダリング部 4 6 を含み、C S S に準拠した表示機能を提供する。C S S 解析部 4 2 は、C S S の構文を解析するパーサの機能を有する。C S S 提供部 4 4 は、C S S オブジェクトの実装であり、D O M ツリーに対して C S S のカスケード処理を行う。レンダリング部 4 6 は、C S S のレンダリングエンジンであり、C S S を用いてレイアウトされる H T M L などのボキャブラリで記述された文書の表示に用いられる。

【 0 0 2 0 】

H T M L ユニット 5 0 は、H T M L により記述された文書を表示又は編集する。S V G ユニット 6 0 は、S V G により記述された文書を表示又は編集する。これらの表示 / 編集系は、プラグインの形で実現されており、それぞれ、文書を表示する表示部 (Canvas) 5 6、6 6、編集指示を含むイベントを送受信する制御部 (Editlet) 5 2、6 2、編集コマンドを受けて D O M に対して編集を行う編集部 (Zone) 5 4、6 4 を備える。制御部 5 2 又は 6 2 が外部から D O M ツリーの編集コマンドを受け付けると、編集部 5 4 又は 6 4 が D O M ツリーを変更し、表示部 5 6 又は 6 6 が表示を更新する。これらは、M V C (Model-View-Controller) と呼ばれるフレームワークに類似する構成をとっており、概ね、表示部 5 6 及び 6 6 が「View」に、制御部 5 2 及び 6 2 が「Controller」に、編集部 5 4 及び 6 4 と D O M の実体が「Model」に、それぞれ対応する。本前提技術の文書処理装置 2 0 では、X M L 文書をツリー表示形式で編集するだけでなく、それぞれのボキャブラリに応じた編集を可能とする。例えば、H T M L ユニット 5 0 は、H T M L 文書をワードプロセッサに類似した方式で編集するためのユーザインターフェースを提供し、S V G ユニット 6 0 は、S V G 文書を画像描画ツールに類似した方式で編集するためのユーザインターフェースを提供する。

【 0 0 2 1 】

V C ユニット 8 0 は、マッピング部 8 2、定義ファイル取得部 8 4、及び定義ファイル生成部 8 6 を含み、あるボキャブラリにより記述された文書を、他のボキャブラリにマッピングすることにより、マッピング先のボキャブラリに対応した表示編集用プラグインで文書を表示又は編集するためのフレームワークを提供する。本前提技術では、この機能を、ボキャブラリコネクション (Vocabulary Connection: V C) と呼ぶ。定義ファイル取得部 8 4 は、マッピングの定義を記述したスクリプトファイルを取得する。この定義ファイルは、ノードごとに、ノード間の対応 (コネクション) を記述する。このとき、各ノードの要素値や属性値の編集の可否を指定してもよい。また、ノードの要素値や属性値を用いた演算式を記述してもよい。これらの機能については、後で詳述する。マッピング部 8 2 は、定義ファイル取得部 8 4 が取得したスクリプトファイルを参照して、D O M 生成部 3 4 にデスティネーションツリーを生成させ、ソースツリーとデスティネーションツリーの対応関係を管理する。定義ファイル生成部 8 6 は、ユーザが定義ファイルを生成するためのグラフィカルユーザインターフェースを提供する。

【 0 0 2 2 】

V C ユニット 8 0 は、ソースツリーとデスティネーションツリーの間のコネクションを監視し、表示を担当するプラグインにより提供されるユーザインターフェースを介してユーザから編集指示を受け付けると、まずソースツリーの該当するノードを変更する。D O M ユニット 3 0 が、ソースツリーが変更された旨のミュートーションイベントを発行すると、V C ユニット 8 0 は、そのミュートーションイベントを受けて、ソースツリーの変更にデスティネーションツリーを同期させるべく、変更されたノードに対応するデスティネーションツリーのノードを変更する。デスティネーションツリーを表示 / 編集するプラグイン、例えば H T M L ユニット 5 0 は、デスティネーションツリーが変更された旨のミュートーションイベントを受けて、変更されたデスティネーションツリーを参照して表示を更新する。このような構成により、少数のユーザにより利用されるローカルなボキャブラリ

10

20

30

40

50

により記述された文書であっても、他のメジャーなボキャブラリに変換することで、文書を表示することができるとともに、編集環境が提供される。

【0023】

文書処理装置20により文書を表示又は編集する動作について説明する。文書処理装置20が処理対象となる文書を読み込むと、DOM生成部34が、そのXML文書からDOMツリーを生成する。また、主制御ユニット22又は編集ユニット24は、名前空間を参照して文書を記述しているボキャブラリを判別する。そのボキャブラリに対応したプラグインが文書処理装置20にインストールされている場合は、そのプラグインをロードして、文書を表示/編集させる。プラグインがインストールされていない場合は、マッピングの定義ファイルが存在するか否かを確認する。定義ファイルが存在する場合、定義ファイル取得部84が定義ファイルを取得し、その定義に従って、デスティネーションツリーが生成され、マッピング先のボキャブラリに対応するプラグインにより文書が表示/編集される。複数のボキャブラリを含む複合文書である場合は、後述するように、それぞれのボキャブラリに対応したプラグインにより、文書の該当箇所がそれぞれ表示/編集される。定義ファイルが存在しない場合は、文書のソース又はツリー構造を表示し、その表示画面において編集が行われる。

10

【0024】

図2は、処理対象となるXML文書の例を示す。このXML文書は、生徒の成績データを管理するために用いられる。XML文書のトップノードである構成要素「成績」は、配下に、生徒ごとに設けられた構成要素「生徒」を複数有する。構成要素「生徒」は、属性値「名前」と、子要素「国語」、「数学」、「理科」、「社会」を有する。属性値「名前」は、生徒の名前を格納する。構成要素「国語」、「数学」、「理科」、「社会」は、それぞれ、国語、数学、理科、社会の成績を格納する。例えば、名前が「A」である生徒の国語の成績は「90」、数学の成績は「50」、理科の成績は「75」、社会の成績は「60」である。以下、この文書で使用されているボキャブラリ(タグセット)を、「成績管理ボキャブラリ」と呼ぶ。

20

【0025】

本前提技術の文書処理装置20は、成績管理ボキャブラリの表示/編集に対応したプラグインを有しないので、この文書をソース表示、ツリー表示以外の方法で表示するためには、前述したVC機能が用いられる。すなわち、成績管理ボキャブラリを、プラグインが用意された別のボキャブラリ、例えば、HTMLやSVGなどにマッピングするための定義ファイルを用意する必要がある。ユーザ自身が定義ファイルを作成するためのユーザインターフェースについては後述することにして、ここでは、既に定義ファイルが用意されているとして説明を進める。

30

【0026】

図3は、図2に示したXML文書をHTMLで記述された表にマッピングする例を示す。図3の例では、成績管理ボキャブラリの「生徒」ノードを、HTMLにおける表(「TABLE」ノード)の行(「TR」ノード)に対応づけ、各行の第1列には属性値「名前」を、第2列には「国語」ノードの要素値を、第3列には「数学」ノードの要素値を、第4列には「理科」ノードの要素値を、第5列には「社会」ノードの要素値を、それぞれ対応付ける。これにより、図2に示したXML文書を、HTMLの表形式で表示することができる。また、これらの属性値及び要素値は、編集可能であることが指定されており、ユーザがHTMLによる表示画面上で、HTMLユニット50の編集機能により、これらの値を編集することができる。第6列には、国語、数学、理科、社会の成績の加重平均を算出する演算式が指定されており、生徒の成績の平均点が表示される。このように、定義ファイルに演算式を指定可能とすることにより、より柔軟な表示が可能となり、編集時のユーザの利便性を向上させることができる。なお、第6列は、編集不可であることが指定されており、平均点のみを個別に編集することができないようにしている。このように、マッピング定義において、編集の可否を指定可能とすることにより、ユーザの誤操作を防ぐことができる。

40

50

【 0 0 2 7 】

図 4 (a) 及び図 4 (b) は、図 2 に示した X M L 文書を図 3 に示した表にマッピングするための定義ファイルの例を示す。この定義ファイルは、定義ファイル用に定義されたスクリプト言語により記述される。定義ファイルには、コマンドの定義と、表示のテンプレートが記述されている。図 4 (a) (b) の例では、コマンドとして、「生徒の追加」と「生徒の削除」が定義されており、それぞれ、ソースツリーにノード「生徒」を挿入する操作と、ソースツリーからノード「生徒」を削除する操作が対応付けられている。また、テンプレートとして、表の第 1 行に「名前」、「国語」などの見出しが表示され、第 2 行以降に、ノード「生徒」の内容が表示されることが記述されている。ノード「生徒」の内容を表示するテンプレート中、「text-of」と記述された項は「編集可能」であることを意味し、「value-of」と記述された項は「編集不可能」であることを意味する。また、ノード「生徒」の内容を表示する行のうち、第 6 列には、「(src:国語 + src:数学 + src:理科 + src:社会) div 4」という計算式が記述されており、生徒の成績の平均が表示されることを意味する。

10

【 0 0 2 8 】

図 5 は、図 2 に示した成績管理ポキャブラリで記述された X M L 文書を、図 3 に示した対応により H T M L にマッピングして表示した画面の例を示す。表 9 0 の各行には、左から、各生徒の名前、国語の成績、数学の成績、理科の成績、社会の成績、及び平均点が表示されている。ユーザは、この画面上で、X M L 文書を編集することができる。たとえば、第 2 行第 3 列の値を「70」に変更すると、このノードに対応するソースツリーの要素値、すなわち、生徒「B」の数学の成績が「70」に変更される。このとき、V C ユニット 8 0 は、デスティネーションツリーをソースツリーに追従させるべく、デスティネーションツリーの該当箇所を変更し、H T M L ユニット 5 0 が、変更されたデスティネーションツリーに基づいて表示を更新する。したがって、画面上の表においても、生徒「B」の数学の成績が「70」に変更され、更に、平均点が「55」に変更される。

20

【 0 0 2 9 】

図 5 に示した画面には、図 4 (a) (b) に示した定義ファイルに定義されたように、「生徒の追加」及び「生徒の削除」のコマンドがメニューに表示される。ユーザがこれらのコマンドを選択すると、ソースツリーにおいて、ノード「生徒」が追加又は削除される。このように、本前提技術の文書処理装置 2 0 では、階層構造の末端の構成要素の要素値を編集するのみではなく、階層構造を編集することも可能である。このようなツリー構造の編集機能は、コマンドの形でユーザに提供されてもよい。また、例えば、表の行を追加又は削除するコマンドが、ノード「生徒」を追加又は削除する操作に対応づけられてもよい。また、他のポキャブラリを埋め込むコマンドがユーザに提供されてもよい。この表を入力用テンプレートとして、穴埋め形式で新たな生徒の成績データを追加することもできる。以上のように、V C 機能により、H T M L ユニット 5 0 の表示 / 編集機能を利用しつつ、成績管理ポキャブラリで記述された文書を編集することが可能となる。

30

【 0 0 3 0 】

図 6 は、ユーザが定義ファイルを生成するために、定義ファイル生成部 8 6 がユーザに提示するグラフィカルユーザインタフェースの例を示す。画面左側の領域 9 1 には、マッピング元の X M L 文書がツリー表示されている。画面右側の領域 9 2 には、マッピング先の X M L 文書の画面レイアウトが示されている。この画面レイアウトは、H T M L ユニット 5 0 により編集可能となっており、ユーザは、画面右側の領域 9 2 において、文書を表示するための画面レイアウトを作成する。そして、例えば、マウスなどのポインティングデバイスにより、画面左側の領域 9 1 に表示されたマッピング元の X M L 文書のノードを、画面右側の領域 9 2 に表示された H T M L による画面レイアウト中へドラッグ & ドロップ操作を行うことにより、マッピング元のノードと、マッピング先のノードとのコネクションが指定される。例えば、要素「生徒」の子要素である「数学」を、H T M L 画面の表 9 0 の第 1 行第 3 列にドロップすると、「数学」ノードと、3 列目の「T D」ノードの間にコネクションが張られる。各ノードには、編集の可否が指定できるようになっている。

40

50

また、表示画面中には、演算式を埋め込むこともできる。画面の編集が終わると、定義ファイル生成部 86 は、画面レイアウトとノード間の接続を記述した定義ファイルを生成する。

【0031】

XHTML、MathML、SVGなどの主要なボキャブラリに対応したビューワやエディタは既に開発されているが、図2に示した文書のようなオリジナルなボキャブラリで記述された文書に対応したビューワやエディタを開発するのは現実的でない。しかし、上記のように、他のボキャブラリにマッピングするための定義ファイルを作成すれば、ビューワやエディタを開発しなくても、VC機能を利用して、オリジナルなボキャブラリで記述された文書を表示・編集することができる。

10

【0032】

図7は、定義ファイル生成部86により生成された画面レイアウトの他の例を示す。図7の例では、成績管理ボキャブラリで記述されたXML文書を表示するための画面に、表90と、円グラフ93が作成されている。この円グラフ93は、SVGにより記述される。後述するように、本前提技術の文書処理装置20は、一つのXML文書内に複数のボキャブラリを含む複合文書进行处理することができるので、この例のように、HTMLで記述された表90と、SVGで記述された円グラフ93とを、一つの画面上に表示することができる。

【0033】

図8は、文書処理装置20によるXML文書の編集画面の一例を示す。図8の例では、一つの画面が複数に分割されており、それぞれの領域において、処理対象となるXML文書を異なる複数の表示形式により表示している。領域94には、文書のソースが表示されており、領域95には、文書のツリー構造が表示されており、領域96には、図5に示したHTMLにより記述された表が表示されている。これらのいずれの画面上においても、文書の編集が可能であり、いずれかの画面上でユーザが編集を行うと、ソースツリーが変更され、それぞれの画面の表示を担当するプラグインが、ソースツリーの変更を反映すべく画面を更新する。具体的には、ソースツリーの変更を通知するミュレーションイベントのリスナーとして、それぞれの編集画面の表示を担当するプラグインの表示部を登録しておき、いずれかのプラグイン又はVCユニット80によりソースツリーが変更されたときに、編集画面を表示中の全ての表示部が、発行されたミュレーションイベントを受け取って画面を更新する。このとき、プラグインがVC機能により表示を行っている場合は、VCユニット80がソースツリーの変更に追従してデスティネーションツリーを変更した後、変更されたデスティネーションツリーを参照してプラグインの表示部が画面を更新する。

20

30

【0034】

例えば、ソース表示及びツリー表示を、専用のプラグインにより実現している場合は、ソース表示用プラグインとツリー表示用プラグインは、デスティネーションツリーを用いず、直接ソースツリーを参照して表示を行う。この場合、いずれかの画面において編集が行われると、ソース表示用プラグインとツリー表示用プラグインは、変更されたソースツリーを参照して画面を更新し、領域96の画面を担当しているHTMLユニット50は、ソースツリーの変更に追従して変更されたデスティネーションツリーを参照して画面を更新する。

40

【0035】

ソース表示及びツリー表示は、VC機能を利用して実現することもできる。すなわち、ソース、ツリー構造をHTMLによりレイアウトし、そのHTMLにXML文書をマッピングして、HTMLユニット50により表示してもよい。この場合、ソース形式、ツリー形式、表形式の3つのデスティネーションツリーが生成されることになる。いずれかの画面上において編集が行われると、VCユニット80は、ソースツリーを変更した後、ソース形式、ツリー形式、表形式の3つのデスティネーションツリーをそれぞれ変更し、HTMLユニット50は、それらのデスティネーションツリーを参照して、3つの画面を更新す

50

る。

【 0 0 3 6 】

このように、一つの画面上に複数の表示形式で文書を表示することにより、ユーザの利便性を向上させることができる。例えば、ユーザは、ソース表示又はツリー表示により文書の階層構造を把握しつつ、表 9 0 などを用いて視覚的に分かりやすい形式で文書を表示し、編集することができる。上記の例では、一つの画面を分割して複数の表示形式による画面を同時に表示したが、一つの画面に一つの表示形式による画面を表示し、表示形式をユーザの指示により切り替え可能としてもよい。この場合、主制御ユニット 2 2 が、ユーザから表示形式の切り替え要求を受け付け、各プラグインに指示して表示を切り替える。

【 0 0 3 7 】

図 9 は、文書処理装置 2 0 により編集される X M L 文書の他の例を示す。図 9 に示した X M L 文書では、S V G 文書の「foreignObject」タグの中に X H T M L 文書が埋め込まれており、さらに、X H T M L 文書の中に M a t h M L で記述された数式が入っている。このような場合、編集ユニット 2 4 が、名前空間を参照して、適切な表示系に描画作業を振り分ける。図 9 の例では、編集ユニット 2 4 は、まず、S V G ユニット 6 0 に四角形を描画させ、つづいて、H T M L ユニット 5 0 に X H T M L 文書を描画させる。さらに、図示しない M a t h M L ユニットに、数式を描画させる。こうして、複数のボキャブラリを包含する複合文書が適切に表示される。表示結果を図 1 0 に示す。

【 0 0 3 8 】

文書編集中、カーソル（キャリッジ）の位置に応じて、表示されるメニューを切り替えてもよい。すなわち、カーソルが、S V G 文書が表示された領域内に存在するときは、S V G ユニット 6 0 が提供するメニュー、又は S V G 文書をマッピングするための定義ファイルに定義されたコマンドを表示し、カーソルが、X H T M L 文書が表示された領域内に存在するときは、H T M L ユニット 5 0 が提供するメニュー、又は X H T M L 文書をマッピングするための定義ファイルに定義されたコマンドを表示する。これにより、編集位置に応じて適切なユーザインターフェースを提供することができる。

【 0 0 3 9 】

複合文書において、あるボキャブラリに対応する適切なプラグイン又はマッピング定義ファイルがなかった場合は、そのボキャブラリにより記述された部分は、ソース表示又はツリー表示されてもよい。従来、ある文書に他の文書を埋め込んだ複合文書を開くとき、埋め込まれた文書を表示するアプリケーションがインストールされていないと、その内容を表示することができなかったが、本前提技術では、表示用のアプリケーションが存在しなくても、テキストデータにより構成された X M L 文書をソース表示又はツリー表示することにより内容を把握することができる。これは、テキストベースである X M L などの文書ならではの特徴といえる。

【 0 0 4 0 】

データがテキストベースで記述されることの他の利点として、例えば、複合文書中の、あるボキャブラリにより記述される部分において、同一文書内の他のボキャブラリで記述された部分のデータを参照してもよい。また、文書内で検索を実行する時に、S V G などの図に埋め込まれた文字列も検索対象とすることができる。

【 0 0 4 1 】

あるボキャブラリにより記述された文書内に、他のボキャブラリのタグを用いてもよい。この X M L 文書は、妥当（valid）ではないが、整形形式（well-formed）であれば、有効な X M L 文書として処理可能である。この場合、挿入された他のボキャブラリのタグは、定義ファイルによりマッピングされてもよい。例えば、X H T M L 文書中に、「重要」、「最重要」などのタグを使用し、これらのタグで囲まれた部分を強調表示してもよいし、重要度の順にソートして表示してもよい。

【 0 0 4 2 】

図 1 0 に示した編集画面において、ユーザにより文書が編集されると、編集された部分を担当するプラグイン又は V C ユニット 8 0 がソースツリーを変更する。ソースツリーに

10

20

30

40

50

は、ノードごとにミュートーションイベントのリスナーを登録できるようになっており、通常は、各ノードが属するボキャブラリに対応したプラグインの表示部又はVCユニット80がリスナーとして登録される。DOM提供部32は、ソースツリーが変更されると、変更されたノードから上位の階層へたどって、登録されたリスナーがあれば、そのリスナーへミュートーションイベントを発行する。例えば、図9に示した文書において、<html>ノードの下位のノードが変更された場合、<html>ノードにリスナーとして登録されたHTMLユニット50にミュートーションイベントが通知されるとともに、その上位の<svg>ノードにリスナーとして登録されたSVGユニット60にもミュートーションイベントが通知される。このとき、HTMLユニット50は、変更されたソースツリーを参照して表示を更新する。SVGユニット60は、自身のボキャブラリに属するノードが変更されていないので、ミュートーションイベントを無視してもよい。

10

【0043】

編集の内容によっては、HTMLユニット50による表示の更新に伴って、全体のレイアウトが変わる可能性がある。この場合は、画面のレイアウトを管理する構成、例えば最上位のノードの表示を担当するプラグインにより、プラグインごとの表示領域のレイアウトが更新される。例えば、HTMLユニット50による表示領域が以前より大きくなった場合、HTMLユニット50は、まず自身の担当する部分を描画して、表示領域の大きさを決定する。そして、画面のレイアウトを管理する構成に、変更後の表示領域の大きさを通知し、レイアウトの更新を依頼する。画面のレイアウトを管理する構成は、通知を受けて、プラグインごとの表示領域を再レイアウトする。こうして、編集された部分の表示が適切に更新されるとともに、画面全体のレイアウトが更新される。

20

【0044】

(実施の形態)

実施の形態では、スプレッドシート用に定義されたタグセットにより記述された文書を処理する技術を提案する。

【0045】

図11は、実施の形態に係る文書処理装置の構成を示す。文書処理装置100は、前提技術で説明した文書処理装置20の構成に加えて、文書や定義ファイルなどを取得する取得部78と、スプレッドシート用に定義されたタグセット(ボキャブラリ)により記述された文書を処理するスプレッドシートユニット70とを更に備える。その他の構成及び動作は、前提技術と同様である。

30

【0046】

スプレッドシートユニット70は、スプレッドシート用のタグセットで記述された文書を処理する。スプレッドシートユニット70は、制御部72、編集部74、及び表示部76を含む。表示部76は、文書中に含まれる、スプレッドシート用のタグセットで記述されたデータを、スプレッドシート形式で表示する。制御部72は、スプレッドシート形式で表示されたデータを処理するためのコマンドやメニューなどを提示し、ユーザからの操作指示を受け付ける。編集部74は、スプレッドシート中の各セルのデータに対する編集要求を受け付け、DOMユニット30に保持された文書のDOMを編集する。

40

【0047】

編集部74は、文書中に含まれる、スプレッドシート用のタグセット以外のタグセットで記述されたデータを、スプレッドシート内のセルに挿入する要求を受け付けると、データの挿入先として指定されたセルに対応するスプレッドシート用のタグセットの要素に、挿入データとして指定されたスプレッドシート用のタグセット以外のタグセットで記述されたデータを指し示すポインタ情報を格納する。このポインタ情報は、xpath式により記述されてもよい。編集部74は、挿入されたデータの参照元が編集されたことを示す通知を受けると、編集された挿入データを参照して、挿入データが挿入されたセルに対応する要素を編集する。

【0048】

図12(a)(b)は、サンプル文書を示す。このサンプル文書は、複数のボキャブラ

50

りを含む複合文書であり、XHTMLで記述された文書の中に、スプレッドシート用のボキャブラリで記述された文書が含まれている。なお、簡略化のために、スタイルに関する情報や名前空間宣言などは省略している。

【0049】

スプレッドシート用のボキャブラリ「table」は、XHTMLの「table」要素に類似した構造を有しており、列を示す要素「table:table-column」、行を示す要素「table:table-row」、セルを示す要素「table:table-cell」などを含む。図12(a)に示したサンプル文書では、要素「table:table-column」が5つ、要素「table:table-row」が7つ含まれており、5列7行のスプレッドシートが表現されている。それぞれの要素「table:table-row」に含まれる要素「table:table-cell」には、各セルのデータが記述されている。

10

【0050】

第1行第1列のセルには、xpath式3010で、この文書の最上位の要素「html」の2番目の子要素「body」の1番目の子要素である「h2」要素3000のデータが挿入されている。

【0051】

第5行第3列のセルには、式を格納する属性「office:formula」の属性値として、xpath式3012で、この文書の1番目の「li」要素3002のテキストノード値が指定されている。また、データ型を格納する属性「office:value-type」に、浮動小数点数値型であることを示す「float」が、データ値を格納する属性「office:value」に、数値「143100」が格納されている。さらに、要素「table:table-cell」の子要素として、文字列を格納する要素「text:p」が設けられており、テキスト値「143,100」が格納されている。

20

【0052】

同様に、第5行第4列のセルには、xpath式3014で、2番目の「li」要素3004のテキストノード値が、第5行第5列のセルには、xpath式3016で、3番目の「li」要素3006のテキストノード値が、それぞれ挿入されている。また、第5行第2列のセルには、第5行第3列から第5行第5列までの値の総和を計算する式3020が指定されており、属性「office:value」に、その式の結果である数値「498500」が格納されている。さらに、要素「table:table-cell」の子要素として、文字列を格納する要素「text:p」が設けられており、テキスト値「498,500」が格納されている。

30

【0053】

図13は、図12(a)(b)に示したサンプル文書が表示された画面4000の例を示す。表示部76は、サンプル文書に含まれる、スプレッドシート用のボキャブラリ「table」で記述されたデータを、スプレッドシート4010に表示する。この画面4000において、スプレッドシート4010中に、文書中に含まれるスプレッドシート用のタグセット以外のタグセットで記述されたデータを挿入する要求をユーザから受け付けると、編集部74は、データの指定を受け付けるための画面を提示する。

【0054】

図14は、編集部74が提示したデータ指定受付画面4020の例を示す。編集部74は、データを挿入する要求を受け付けると、文書中に含まれるデータをツリー状に表示したデータ指定受付画面4020を提示し、ユーザから挿入データの指定を受け付ける。ユーザが挿入したいノードをクリックしてOKボタンを押すと、編集部74は、指定されたノードのxpath式を、挿入先のセルの要素「table:table-cell」の属性「office:formula」の属性値に格納するとともに、属性「office:value」に、テキストノード値を格納する。このとき、編集部74は、スキーマや定義ファイルなどにより、挿入するデータのデータ型が分かっている場合は、属性「office:value-type」に、データ型を格納する。データ型が分からない場合は、ユーザに問い合わせてもよい。また、編集部74は、挿入先のセルにおけるデータの表示形式にしたがって挿入データを整形した文字列を要素「text:p」に格納する。例えば、挿入データが数値「143100」である場合に、表示形式

40

50

として「3桁ごとにカンマを打つ」ことが指定されている場合は、テキスト値「143,100」を格納する。

【0055】

編集部74により、スプレッドシート用のボキャブラリで記述された文書のソースDOMが変更されると、DOMユニット30が発行したミュートーションイベントがスプレッドシートユニット70に通知される。表示部76は、ソースDOMを参照してスプレッドシート4010の表示を更新する。

【0056】

図15は、参照先のデータが更新されたときの画面4000の例を示す。画面4000において、2番目の「li」要素3004のテキストデータがユーザにより編集されると、HTMLユニット50がソースDOMを変更する。DOMユニット30により発行されたミュートーションイベントは、HTMLユニット50に通知されて、表示部56により表示が更新される。さらに、ミュートーションイベントは、このデータを参照しているスプレッドシートユニット70にも通知される。編集部74は、xpath式3014により2番目の「li」要素3004を参照している第5行第4列のセルの要素「table:table-cell」を、ソースDOM中の「li」要素3004を参照して変更する。この変更を通知するミュートーションイベントがDOMユニット30からスプレッドシートユニット70に通知され、表示部76がスプレッドシート4010の表示を更新する。これにより、図13の画面4000が、図15の画面4000に更新される。

【0057】

編集部74は、文書中に含まれる複数の要素により構成されるフラグメントを挿入する指定を受け付けて、フラグメントを挿入データとして挿入してもよい。例えば、図14において、ユーザが、ノード「ul」を挿入データとして指定した場合、編集部74は、要素「ul」の下位の要素も含めたフラグメントを挿入データとして挿入する。このとき、編集部74は、ノード「ul」の下位に存在するテキストノードの分だけ属性「office:value」と要素「text:p」を設けて、それらのテキストノード値を格納してもよい。また、要素「table:table-cell」の下位に、挿入されたフラグメントと同じ構造のデータを追加してもよい。挿入されたフラグメントのデータは、スプレッドシート中の他のセルから参照され、演算式に利用されてもよい。このように、セルにフラグメントを挿入することを可能とすることで、構造化されたデータをスプレッドシート中で扱うことができる。

【0058】

図16は、セルにフラグメントが挿入されたときの画面4000の例を示す。スプレッドシート4010の第5行第2列のセルに、XHTMLの要素「ul」の下位のフラグメントが挿入されている。要素「ul」の下位にあるテキストノードのテキスト値が第5行第2列のセルに列挙表示されている。

【0059】

編集部74は、スプレッドシート中のセルに指定する式に記述可能な関数として、挿入データを記述したタグセットごとに、タグセットで記述されたデータを利用した専用の関数を定義し、関数を含む式の入力を受け付けたときに、式を演算した結果を挿入してもよい。例えば、SVGで記述された図形データを参照する際に、図形の面積を算出する関数を定義しておいてもよい。

【0060】

図17は、スプレッドシート中に専用の関数が挿入されたときの画面4000の例を示す。この文書は、SVGで記述された円4030を含んでいる。スプレッドシート4040の第2行第2列のセルには、円4030の面積を計算する式が挿入されている。この画面4000において、ユーザが円4030の半径を変更すると、SVGユニット60がソースDOMを変更する。編集部74は、ミュートーションイベントを受けて、ソースDOM中の円4030を記述した要素を参照し、変更された円の半径を取得し、スプレッドシート4040中の第2行第2列のセルに対応した要素に、円4030の面積を代入する。このように、参照先のボキャブラリに応じた関数を定義しておくことにより、より柔軟で

10

20

30

40

50

高機能なスプレッドシートを実現することができる。

【0061】

以上、本発明を実施の形態をもとに説明した。この実施の形態は例示であり、それらの各構成要素や各処理プロセスの組合せにいろいろな変形例が可能なこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。

【0062】

実施の形態では、XML文書进行处理する例について説明したが、本実施の形態の文書処理装置100は、他のマークアップ言語、例えば、SGML、HTMLなどで記述された文書も同様に処理可能である。

【図面の簡単な説明】

10

【0063】

【図1】前提技術に係る文書処理装置の構成を示す図である。

【図2】文書処理装置により編集されるXML文書の例を示す図である。

【図3】図2に示したXML文書をHTMLで記述された表にマッピングする例を示す図である。

【図4(a)】図2に示したXML文書を図3に示した表にマッピングするための定義ファイルの例を示す図である。

【図4(b)】図2に示したXML文書を図3に示した表にマッピングするための定義ファイルの例を示す図である。

【図5】図2に示したXML文書を、図3に示した対応によりHTMLにマッピングして表示した画面の例を示す図である。

20

【図6】ユーザが定義ファイルを生成するために、定義ファイル生成部がユーザに提示するグラフィカルユーザインターフェースの例を示す図である。

【図7】定義ファイル生成部により生成された画面レイアウトの他の例を示す図である。

【図8】文書処理装置によるXML文書の編集画面の一例を示す図である。

【図9】文書処理装置により編集されるXML文書の他の例を示す図である。

【図10】図9に示した文書を表示した画面の例を示す図である。

【図11】実施の形態に係る文書処理装置の構成を示す図である。

【図12(a)】サンプル文書を示す図である。

【図12(b)】サンプル文書を示す図である。

30

【図13】図12(a)(b)に示したサンプル文書が表示された画面の例を示す図である。

【図14】編集部が提示したデータ指定受付画面の例を示す図である。

【図15】参照先のデータが更新されたときの画面の例を示す図である。

【図16】セルにフラグメントが挿入されたときの画面の例を示す図である。

【図17】スプレッドシート中に専用の関数が挿入されたときの画面の例を示す図である。

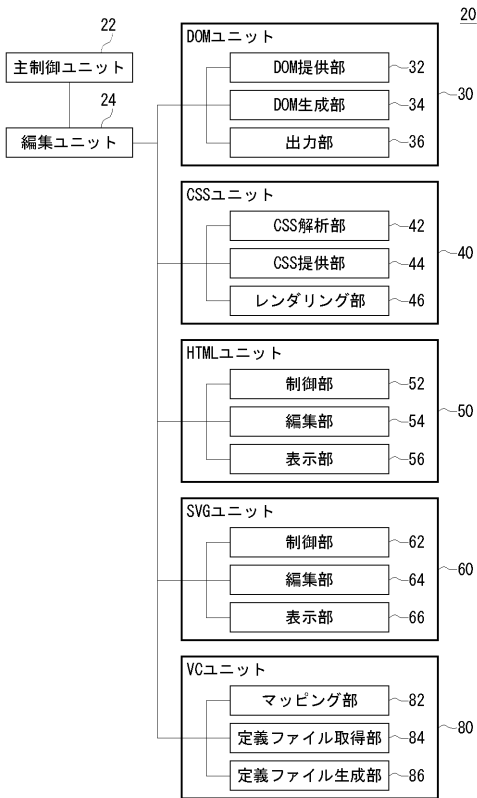
【符号の説明】

【0064】

20 文書処理装置、22 主制御ユニット、24 編集ユニット、30 DOMユニット、32 DOM提供部、34 DOM生成部、36 出力部、40 CSSユニット、42 CSS解析部、44 CSS提供部、46 レンダリング部、50 HTMLユニット、52, 62 制御部、54, 64 編集部、56, 66 表示部、60 SVGユニット、80 VCユニット、82 マッピング部、84 定義ファイル取得部、86 定義ファイル生成部、100 文書処理装置。

40

【 図 1 】

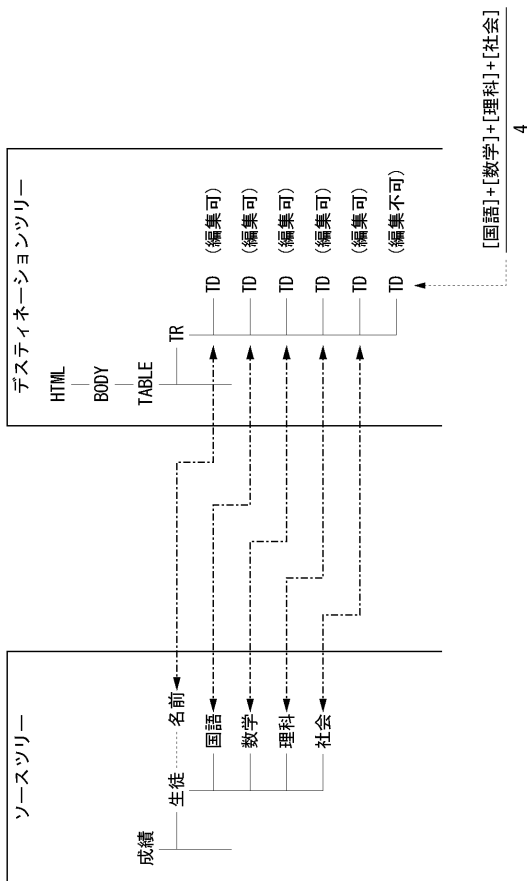


【 図 2 】

```

<?xml version="1.0" ?>
<?com.xfytec.vocabulary-connection href="records.vcd" ?>
<成績 xmlns="http://xmlns.xfytec.com/sample/records">
  <生徒 名前="A">
    <国語>90</国語>
    <数学>50</数学>
    <理科>75</理科>
    <社会>60</社会>
  </生徒>
  <生徒 名前="B">
    <国語>45</国語>
    <数学>60</数学>
    <理科>55</理科>
    <社会>50</社会>
  </生徒>
  <生徒 名前="C">
    <国語>55</国語>
    <数学>45</数学>
    <理科>95</理科>
    <社会>40</社会>
  </生徒>
  <生徒 名前="D">
    <国語>25</国語>
    <数学>35</数学>
    <理科>40</理科>
    <社会>15</社会>
  </生徒>
</成績>
  
```

【 図 3 】



【 図 4 (a) 】

```

<?xml version="1.0"?>
<vc:vcd xmlns:vc="http://xmlns.xfytec.com/vcd"
  xmlns:src="http://xmlns.xfytec.com/sample/records"
  xmlns="http://www.w3.org/1999/xhtml"
  version="1.0">
  <!-- Commands -->
  <vc:command name="生徒の追加">
    <vc:insert-fragment
      target="ancestor-or-self::src:生徒"
      position="after">
      <src:生徒/>
    </vc:insert-fragment>
  </vc:command>
  <vc:command name="生徒の削除">
    <vc:delete-fragment target="ancestor-or-self::src:生徒" />
  </vc:command>
  <!-- Templates -->
  <vc:vc-template match="src:成績" name="成績表">
    <vc:ui command="生徒の追加">
      <vc:mount-point>
        /MenuBar/成績表/生徒の追加
      </vc:mount-point>
    </vc:ui>
    <vc:ui command="生徒の削除">
      <vc:mount-point>
        /MenuBar/成績表/生徒の削除
      </vc:mount-point>
    </vc:ui>
    <html>
      <head>
        <title>成績表</title>
        <style>
          td.th {
            text-align:center;
            border-right:solid black 1px;
            border-bottom:solid black 1px;
            border-top:none 0px;
            border-left:none 0px;
          }
          table {
            border-top:solid black 2px;
            border-left:solid black 2px;
            border-right:solid black 1px;
            border-bottom:solid black 1px;
            border-spacing:0px;
          }
        </style>
      </head>
    </html>
  </vc:vc-template>
</vc:vcd>
  
```

【 図 4 (b) 】

```
|  |
| --- |
|  |

```

```

</style>
</head>
<body>
<h1>成績一覧</h1>
<table>
<tr><th><div class="data">名前</div></th>
<th></th>
<th><div class="data">国語</div></th>
<th><div class="data">数学</div></th>
<th><div class="data">理科</div></th>
<th><div class="data">社会</div></th>
<th></th>
<th><div class="data">平均</div></th></tr>
<tr>
<td><div class="data">A</div>
<td><div class="data">90</div>
<td><div class="data">50</div>
<td><div class="data">75</div>
<td><div class="data">60</div>
<td><div class="data">68.8</div>
</tr>
<tr>
<td><div class="data">B</div>
<td><div class="data">45</div>
<td><div class="data">60</div>
<td><div class="data">55</div>
<td><div class="data">50</div>
<td><div class="data">52.5</div>
</tr>
<tr>
<td><div class="data">C</div>
<td><div class="data">55</div>
<td><div class="data">45</div>
<td><div class="data">95</div>
<td><div class="data">40</div>
<td><div class="data">58.8</div>
</tr>
<tr>
<td><div class="data">D</div>
<td><div class="data">25</div>
<td><div class="data">35</div>
<td><div class="data">40</div>
<td><div class="data">15</div>
<td><div class="data">28.8</div>
</tr>
</table>
</body>
</html>
</vc:vc-template>
<vc:template match="src:生徒">
<tr>
<td><div class="data">
<vc:text-of select="@名前" fallback="名無し"/>
</div></td>
<td><div class="data">
<vc:text-of select="src:国語" fallback="0" type="vc:integer" />
</div></td>
<td><div class="data">
<vc:text-of select="src:数学" fallback="0" type="vc:integer" />
</div></td>
<td><div class="data">
<vc:text-of select="src:理科" fallback="0" type="vc:integer" />
</div></td>
<td><div class="data">
<vc:text-of select="src:社会" fallback="0" type="vc:integer" />
</div></td>
<td><div class="data">
<vc:value-of
select="(src:国語 + src:数学 + src:理科 + src:社会) div 4" />
</div></td>
</tr>
</vc:template>
</vc:vcd>

```

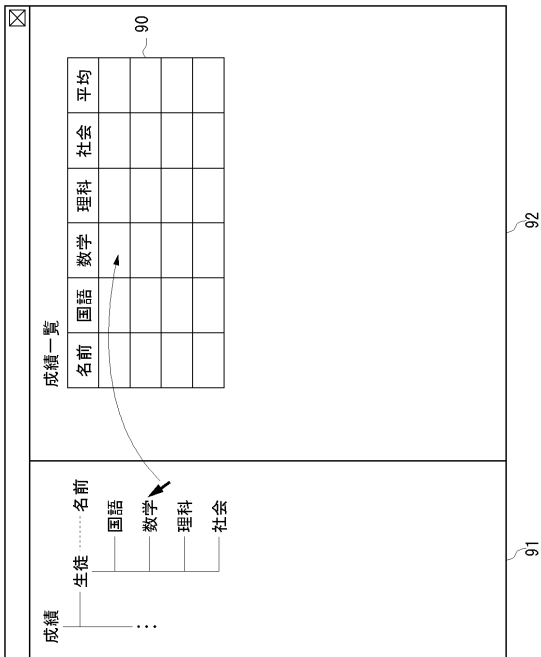
【 図 5 】

sample.xml

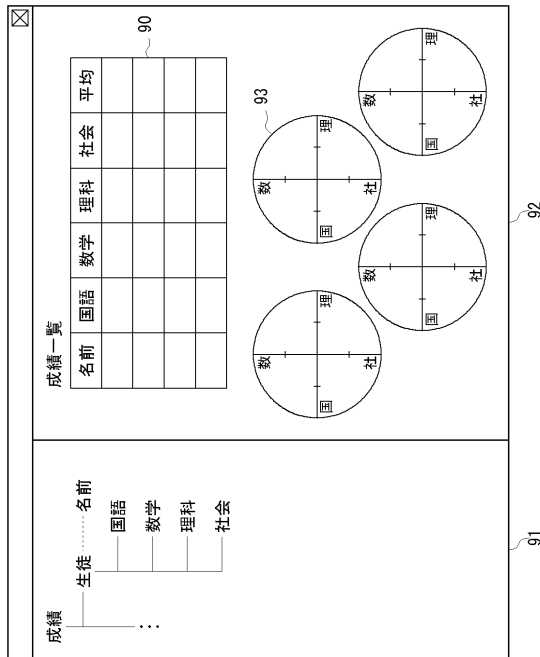
成績一覧 90

名前	国語	数学	理科	社会	平均
A	90	50	75	60	68.8
B	45	60	55	50	52.5
C	55	45	95	40	58.8
D	25	35	40	15	28.8

【 図 6 】



【 図 7 】



【 図 8 】

成績一覧		社会	理科	数学	国語
名前	A	90	45	55	25
	B	45	60	45	40
	C	55	60	95	40
	D	25	35	40	15

生徒
 名前 A
 国語 90
 数学 50
 ...

```

<?xml version="1.0" ?>
<成績 xmlns="...">
  <生徒 名前="A">
    ...
  </生徒>
</成績>
  
```

【 図 9 】

```

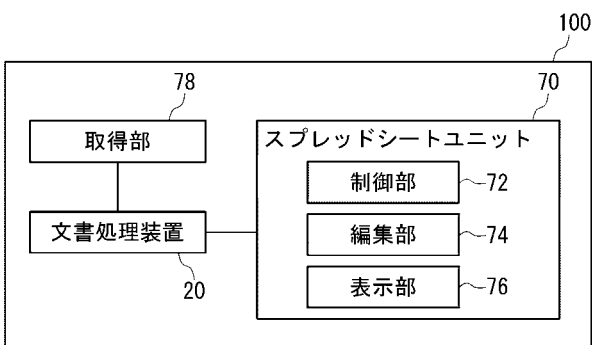
<?xml version="1.0" ?>
<svg xmlns="http://www.w3.org/2000/svg"
width="400" height="200"
viewBox="0 0 400 200"
>
  <rect x="-15" y="65" width="150" height="100" rx="20"
transform="rotate(-20)"
style="fill:none; stroke:purple; stroke-width:10"
/>
  <foreignObject x="190" y="10" width="200" height="200">
    <html xmlns="http://www.w3.org/1999/xhtml">
      <head><title /></head>
      <body bgcolor="#FFFFFF" text="darkgreen">
        <div style="font-size:12pt">
          SVG文書中に<math>foreignObject</math>を使って、
          XHTML文書を埋め込んでみました。
          数式も入れてみました。
        </div>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <mi>x</mi>
          <mo>=</mo>
          <mfrac>
            <msup>
              <mo>-</mo>
              <mi>b</mi>
            </msup>
            <mo>±</mo>
            <msqrt>
              <msup>
                <mi>2</mi>
              </msup>
              <mo>-</mo>
              <mn>4</mn>
              <mi>a</mi>
            </msqrt>
          </mfrac>
          </math>
        </div><!-- math -->
      </body>
    </html>
  </foreignObject>
</svg>
  
```

【 図 10 】

SVG文書中に
 <foreignObject>を使って、
 XHTML文書を埋め込んで
 みました。数式も入れてみ
 たりして：

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

【 図 11 】



【 図 12 (a) 】

```

<?xml version="1.0" ?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head><title /></head>
  <body>
    <h2>温州みかんの生産量</h2> ←3000
    <h4>平成14年</h4>
    <ul>
      <li>143100</li> ←3002
      <li>180400</li> ←3004
      <li>175000</li> ←3006
    </ul>
    <office:document>
      <office:body>
        <office:spreadsheet>
          <table:table>
            <table:table-column />
            <table:table-column />
            <table:table-column />
            <table:table-column />
            <table:table-column />
            <table:table-row>
              <table:table-cell table:formula="oooc:XPATH('/[*][1]/[*][2]/[*][1]')"
                office:string-value="温州みかんの生産量" office:value-type="string">
                <text:p>温州みかんの生産量</text:p>
              </table:table-cell>
              <table:table-cell table:number-columns-repeated="4" />
            </table:table-row>
            <table:table-row>
              <table:table-cell />
              <table:table-cell table:number-columns-repeated="4" />
            </table:table-row>
            <table:table-row>
              <table:table-cell />
              <table:table-cell office:string-value="生産量(t)" office:value-type="string">
                <text:p>生産量(t)</text:p>
              </table:table-cell>
              <table:covered-table-cell table:number-columns-repeated="3" />
            </table:table-row>
            <table:table-row>
              <table:table-cell />
              <table:table-cell office:string-value="合計" office:value-type="string">
                <text:p>合計</text:p>
              </table:table-cell>
              <table:table-cell office:string-value="静岡県" office:value-type="string">
                <text:p>静岡県</text:p>
              </table:table-cell>
              <table:table-cell office:string-value="愛媛県" office:value-type="string">
                <text:p>愛媛県</text:p>
              </table:table-cell>
              <table:table-cell office:string-value="和歌山県" office:value-type="string">
                <text:p>和歌山県</text:p>
              </table:table-cell>
            </table:table-row>
          </office:spreadsheet>
        </office:body>
      </office:document>
    </body>
  </html>
  
```


【 図 1 2 (b) 】

```

<table:table-row>
<table:table-cell office:string-value="平成14年" office:value-type="string">
<text:p>平成14年</text:p>
</table:table-cell>
<table:table-cell table:formula="oooc:=SUM([C5:.E5])" office:value="498500"
office:value-type="float">
<text:p>498,500</text:p>
</table:table-cell>
<table:table-cell office:value-type="float" office:value="143100"
table:formula="oooc:=XPath("/child::*[local-name()='html'] [1]
/child::*[local-name()='body'] [1]
/child::*[local-name()='ul'] [1]
/child::*[local-name()='li'] [1]/text() [1])">
<text:p>143,100</text:p>
</table:table-cell>
<table:table-cell office:value-type="float" office:value="180400"
table:formula="oooc:=XPath("/child::*[local-name()='html'] [1]
/child::*[local-name()='body'] [1]
/child::*[local-name()='ul'] [1]
/child::*[local-name()='li'] [2]/text() [1])">
<text:p>180,400</text:p>
</table:table-cell>
<table:table-cell office:value-type="float" office:value="175000"
table:formula="oooc:=XPath("/child::*[local-name()='html'] [1]
/child::*[local-name()='body'] [1]
/child::*[local-name()='ul'] [1]
/child::*[local-name()='li'] [3]/text() [1])">
<text:p>175,000</text:p>
</table:table-cell>
</table:table-row>
</table:table-row>
<table:table-row>
<table:table-cell office:string-value="平成15年" office:value-type="string">
<text:p>平成15年</text:p>
</table:table-cell>
</table:table-cell />
<table:table-cell />
<table:table-cell />
<table:table-cell />
</table:table-row>
<table:table-row>
<table:table-cell office:string-value="平成16年" office:value-type="string">
<text:p>平成16年</text:p>
</table:table-cell>
</table:table-cell />
<table:table-cell />
<table:table-cell />
<table:table-cell />
</table:table-row>
</table:table>
</office:spreadsheet>
</office:body>
</office:document>
</body>
</html>

```

【 図 1 3 】

温州みかんの生産量

- 平成14年
- 143100
- 180400
- 175000

	A	B	C	D	E
1	温州みかんの生産量				
2	合計				
3	生産量(t)				
4	静岡県	愛媛県	和歌山県		
5	平成14年	498,500	143,100	180,400	175,000
6	平成15年				
7	平成16年				

4010

4000

【 図 1 4 】

The screenshot shows a spreadsheet application window with a menu bar (File, Edit, View, Insert, Format, Data, Tools, Help) and a toolbar. The spreadsheet contains the following data:

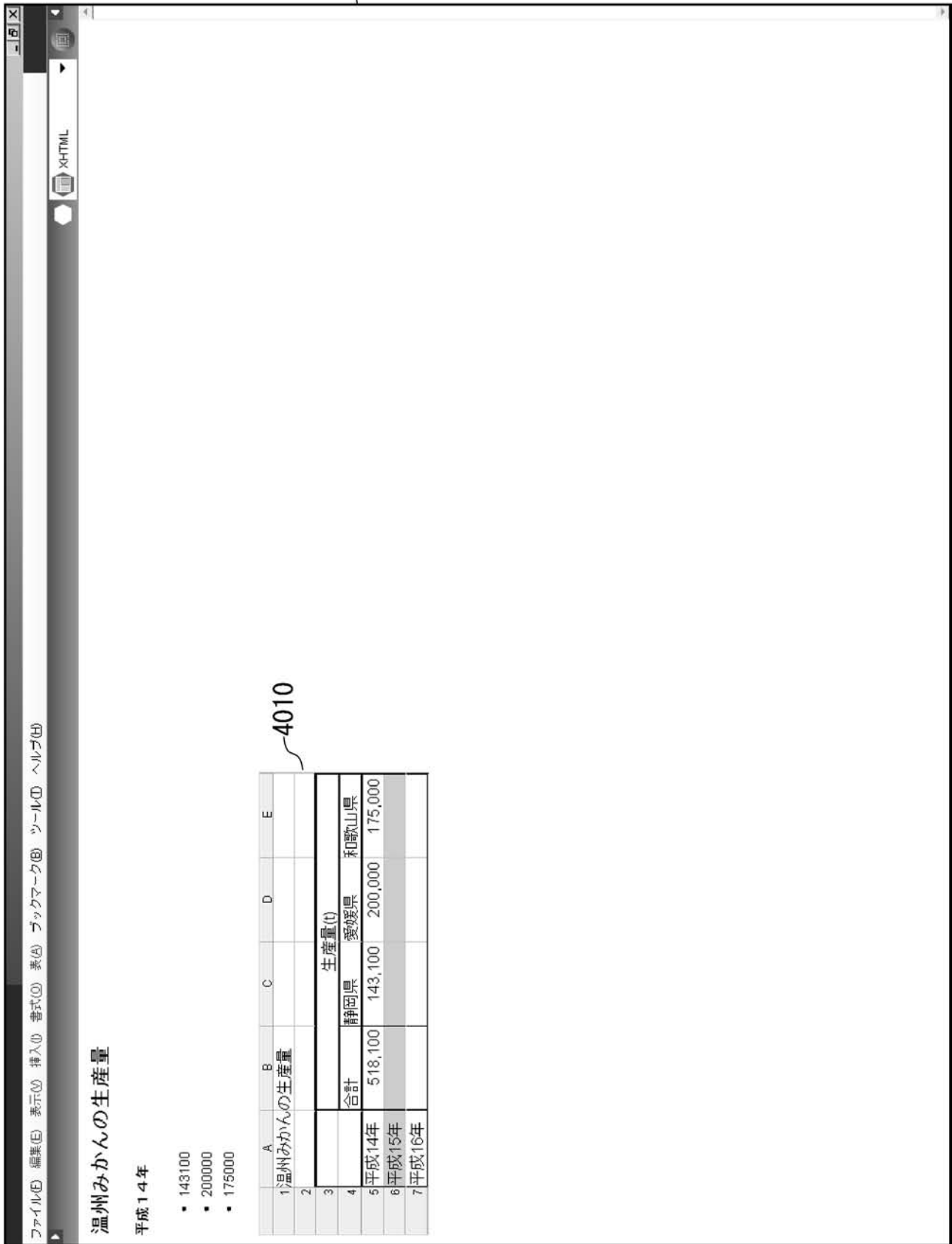
A	B	C
1)	温州みかんの生産量	
2		
3		生産量(
4		静岡県 愛
5	平成14年	498,500
6	平成15年	143,100
7	平成16年	

Below the table, the text "温州みかんの生産量" is displayed, followed by "平成14年" and a list of values: 143100, 180400, and 175000.

An "XPath関数の挿入" (Insert XPath Function) dialog box is open, showing a tree view of the document structure. The tree includes nodes for `html`, `head`, `body`, `h2`, `h4`, `ul`, `li`, and `p`. The selected node is `li class="xy_xss" 平成14年`, which is associated with the value 143100. Other `li` nodes are associated with 180400 and 175000. The dialog box has "OK" and "キャンセル" (Cancel) buttons.

Reference numbers 4000 and 4020 are present in the image, pointing to the spreadsheet area and the dialog box respectively.

【 図 15 】



【 図 16 】

温州みかんの生産量

平成14年

- 143100
- 180400
- 175000

	A	B	C	D	E
1	温州みかんの生産量				
2					
3					
4		合計	静岡県	愛媛県	和歌山県
	平成14年	143,100	180,400	175,000	
5					
6	平成15年				
7	平成16年				

4010

4000

【 図 17 】

