

(19) 日本国特許庁(JP)

## 再公表特許(A1)

(11) 国際公開番号

W02006/051958

発行日 平成20年5月29日(2008.5.29)

(43) 国際公開日 平成18年5月18日(2006.5.18)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G06F 17/21 (2006.01)</b>	G06F 17/21 570L	5B009
<b>G06F 13/00 (2006.01)</b>	G06F 17/21 501T	5K201
<b>H04M 11/08 (2006.01)</b>	G06F 17/21 596Z	
	G06F 13/00 550B	
	H04M 11/08	

審査請求 未請求 予備審査請求 未請求 (全 48 頁)

出願番号 特願2006-545028 (P2006-545028)	(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
(21) 国際出願番号 PCT/JP2005/020883	
(22) 国際出願日 平成17年11月14日(2005.11.14)	
(31) 優先権主張番号 特願2004-329878 (P2004-329878)	(74) 代理人 100105924 弁理士 森下 賢樹
(32) 優先日 平成16年11月12日(2004.11.12)	
(33) 優先権主張国 日本国(JP)	(74) 代理人 100109047 弁理士 村田 雄祐
(31) 優先権主張番号 特願2005-20457 (P2005-20457)	(72) 発明者 藤巻 祐介 徳島県徳島市川内町平石若松108番地4 株式会社ジャストシステム内
(32) 優先日 平成17年1月27日(2005.1.27)	
(33) 優先権主張国 日本国(JP)	(72) 発明者 和家 伸明 徳島県徳島市川内町平石若松108番地4 株式会社ジャストシステム内

最終頁に続く

(54) 【発明の名称】 情報配信システム

## (57) 【要約】

情報配信システム100において、情報配信装置25は、文書を携帯端末70に配信する。情報配信装置25は、XMLにより記述された文書を保持する文書保持部26と、文書を携帯端末70へ送信する送信部27とを含む。携帯端末70は、情報配信装置25から送信された文書を取得する通信部75と、取得した文書进行处理する文書処理装置20と、処理された文書を表示する表示装置21と、ユーザからの入力を受け付ける入力装置23とを含む。文書処理装置20は、所定のタグセットで記述された文書进行处理する処理系と、文書に含まれる要素进行处理系が処理可能な要素に変換する変換部を含む。

**【特許請求の範囲】****【請求項 1】**

携帯端末と、  
前記携帯端末へ情報を配信する情報配信装置と、を含み、  
前記情報配信装置は、  
マークアップ言語により記述された文書を保持する保持部を含み、  
前記携帯端末は、  
前記情報配信装置から前記文書を取得する取得部と、  
所定のタグセットで記述された文書进行处理する処理系と、  
前記文書に含まれる要素を前記処理系が処理可能な要素に変換し、変換前後の要素の対応を管理する変換部と、  
前記変換部により変換され、前記処理系により処理された文書を表示する表示装置と、  
を含む  
ことを特徴とする情報配信システム。

**【請求項 2】**

前記情報配信装置は、前記文書を前記携帯端末へ送信する送信部を更に含み、  
前記携帯端末は、前記情報配信装置から送信された前記文書を受信する受信部を更に含むことを特徴とする請求項 1 に記載の情報配信システム。

**【請求項 3】**

前記携帯端末は、前記文書に含まれる文書を前記処理系が処理可能な要素に変換するためのルールを記述した定義ファイルを更に含み、  
前記変換部は、前記定義ファイルを参照して、前記文書に含まれる要素を前記処理系が処理可能な要素に変換することを特徴とする請求項 1 又は 2 に記載の情報配信システム。

**【請求項 4】**

前記保持部は、前記定義ファイルを更に保持し、  
前記送信部は、前記定義ファイルを前記携帯端末へ送信することを特徴とする請求項 3 に記載の情報配信システム。

**【請求項 5】**

前記携帯端末は、前記文書に対応する複数の前記定義ファイルを取得したときに、それらを切り替えて適用する切替部を含むことを特徴とする請求項 3 又は 4 に記載の情報配信システム。

**【請求項 6】**

前記定義ファイルは、前記文書进行处理するためのユーザインタフェースを更に記述し、  
前記携帯端末は、前記定義ファイルに記述された前記ユーザインタフェースを前記表示装置に表示し、ユーザからの指示を受け付けることを特徴とする請求項 3 から 5 のいずれかに記載の情報配信システム。

**【請求項 7】**

前記携帯端末は、前記処理系により編集した前記文書を前記情報配信装置へ返送する送信部を更に含むことを特徴とする請求項 1 から 6 のいずれかに記載の情報配信システム。

**【請求項 8】**

前記携帯端末は、自身と通信可能な情報配信装置を検索する検索部を更に含み、  
前記取得部は、前記検索部により検索された情報配信装置から文書を取得することを特徴とする請求項 1 から 7 のいずれかに記載の情報配信システム。

**【請求項 9】**

前記検索部は、前記携帯端末が有している定義ファイルにより処理可能なタグセットで記述された文書を検索し、  
前記取得部は、前記検索部により検索された文書を取得することを特徴とする請求項 8 に記載の情報配信システム。

**【発明の詳細な説明】****【技術分野】**

10

20

30

40

50

## 【 0 0 0 1 】

本発明は、情報配信技術に関し、特に、X M L ( eXtensible Markup Language ) により記述された文書を携帯端末に配信する情報配信システムに関する。

## 【 背景技術 】

## 【 0 0 0 2 】

近年、幅広い層にわたって携帯電話端末が広く普及しており、成人のほとんどが携帯電話端末を所有していると言っても過言ではない。また、インターネットのインフラの整備が進み、インターネットの利用者が爆発的に増加している。携帯電話端末の多くは、インターネットへアクセスする機能を有しており、インターネットを介した携帯電話端末への情報配信ビジネスも盛んに行われている。

10

## 【 発明の開示 】

## 【 発明が解決しようとする課題 】

## 【 0 0 0 3 】

現在広く普及している情報配信サービスでは、ウェブサーバから H T M L ( HyperText Markup Language ) で記述されたウェブページを配信している。しかし、H T M L は、主として表示形式を規定するマークアップ言語であるから、文書の内容に対してタグで意味が付されているわけではない。したがって、文書の再利用性が困難であるという問題がある。

## 【 0 0 0 4 】

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、X M L により記述された文書を携帯端末に配信する技術を提供することにある。

20

## 【 課題を解決するための手段 】

## 【 0 0 0 5 】

本発明のある態様は、情報配信システムに関する。この情報配信システムは、携帯端末と、前記携帯端末へ情報を配信する情報配信装置と、を含み、前記情報配信装置は、マークアップ言語により記述された文書を保持する保持部を含み、前記携帯端末は、前記情報配信装置から前記文書を取得する取得部と、所定のタグセットで記述された文書进行处理する処理系と、前記文書に含まれる要素を前記処理系が処理可能な要素に変換し、変換前後の要素の対応を管理する変換部と、前記変換部により変換され、前記処理系により処理された文書を表示する表示装置と、を含むことを特徴とする。

30

## 【 0 0 0 6 】

前記情報配信装置は、前記文書を前記携帯端末へ送信する送信部を更に含んでもよく、前記携帯端末は、前記情報配信装置から送信された前記文書を受信する受信部を更に含んでもよい。

## 【 0 0 0 7 】

前記携帯端末は、前記文書に含まれる文書を前記処理系が処理可能な要素に変換するためのルールを記述した定義ファイルを更に含んでもよく、前記変換部は、前記定義ファイルを参照して、前記文書に含まれる要素を前記処理系が処理可能な要素に変換してもよい。前記保持部は、前記定義ファイルを更に保持してもよく、前記送信部は、前記定義ファイルを前記携帯端末へ送信してもよい。前記携帯端末は、前記文書に対応する複数の前記定義ファイルを取得したときに、それらを切り替えて適用する切替部を含んでもよい。前記定義ファイルは、前記文書进行处理のためのユーザインタフェースを更に記述してもよく、前記携帯端末は、前記定義ファイルに記述された前記ユーザインタフェースを前記表示装置に表示し、ユーザからの指示を受け付けてもよい。

40

## 【 0 0 0 8 】

前記携帯端末は、前記処理系により編集した前記文書を前記情報配信装置へ返送する送信部を更に含んでもよい。

## 【 0 0 0 9 】

前記携帯端末は、自身と通信可能な情報配信装置を検索する検索部を更に含んでもよく、前記取得部は、前記検索部により検索された情報配信装置から文書を取得してもよい。

50

前記検索部は、前記携帯端末が有している定義ファイルにより処理可能なタグセットで記述された文書を検索してもよく、前記取得部は、前記検索部により検索された文書を取得してもよい。

【0010】

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、システムなどの中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】

【0011】

本発明によれば、XMLにより記述された文書を携帯端末に配信する技術を提供することができる。

10

【図面の簡単な説明】

【0012】

【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】処理対象となる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。

20

【図5】図2に示した成績管理ポキャブラリで記述された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。

【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(a)】文書処理システムの基本構成を示す図である。

【図11(b)】文書処理システム全体のブロック図を示す図である。

30

【図11(c)】文書処理システム全体のブロック図を示す図である。

【図12】文書管理部の詳細を示す図である。

【図13】ポキャブラリコネクションサブシステムの詳細を示す図である。

【図14】プログラム起動部と他の構成の関係の詳細を示す図である。

【図15】プログラム起動部によりロードされたアプリケーションサービスの構造の詳細を示す図である。

【図16】コアコンポーネントの詳細を示す図である。

【図17】文書管理部の詳細を示す図である。

【図18】アンドゥフレームワークとアンドゥコマンドの詳細を示す図である。

【図19】文書処理システムにおいて文書がロードされる様子を示す図である。

40

【図20】文書とその表現の例を示す図である。

【図21】モデルとコントローラの関係を示す図である。

【図22】プラグインサブシステム、ポキャブラリコネクション、及びコネクタの詳細を示す図である。

【図23】VCDファイルの例を示す図である。

【図24】文書処理システムにおいて複合文書をロードする手順を示す図である。

【図25】文書処理システムにおいて複合文書をロードする手順を示す図である。

【図26】文書処理システムにおいて複合文書をロードする手順を示す図である。

【図27】文書処理システムにおいて複合文書をロードする手順を示す図である。

【図28】文書処理システムにおいて複合文書をロードする手順を示す図である。

50

【図 29】コマンドの流れを示す図である。

【図 30】実施の形態に係る情報配信システムの構成を示す図である。

【図 31】図 31 (a) (b) は、情報配信システムの第 1 の適用例を示す図である。

【図 32】図 32 (a) (b) は、情報配信システムの第 2 の適用例を示す図である。

【図 33】図 33 (a) (b) は、情報配信システムの第 3 の適用例を示す図である。

【図 34】情報配信システムの別の構成例を示す図である。

【図 35】情報配信システムの更に別の構成例を示す図である。

【符号の説明】

【0013】

20 文書処理装置、21 表示装置、22 主制御ユニット、23 入力装置、24 編集ユニット、25 情報配信装置、26 文書保持部、27 送信部、30 DOM ユニット、32 DOM 提供部、34 DOM 生成部、35 DOM 管理部、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、70 携帯端末、71 個人情報保持部、72 フィルタリング部、73 文書保持部、74 検索部、75 通信部、76 販売装置、77 定義ファイル保持部、78 切替部、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、89 無線 IC タグ。

10

【発明を実施するための最良の形態】

【0014】

20

(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。

30

【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。

40

【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することができ、また、プログラム実行時にも、

50

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。

#### 【 0 0 1 7 】

編集ユニット 2 4 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。

#### 【 0 0 1 8 】

DOM ユニット 3 0 は、DOM 提供部 3 2、DOM 生成部 3 4、及び出力部 3 6 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 3 2 は、編集ユニット 2 4 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 3 4 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 8 0 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 XML 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、DOM ツリーを XML 文書として出力する。

#### 【 0 0 1 9 】

CSS ユニット 4 0 は、CSS 解析部 4 2、CSS 提供部 4 4、及びレンダリング部 4 6 を含み、CSS に準拠した表示機能を提供する。CSS 解析部 4 2 は、CSS の構文を解析するパーサの機能を有する。CSS 提供部 4 4 は、CSS オブジェクトの実装であり、DOM ツリーに対して CSS のカスケード処理を行う。レンダリング部 4 6 は、CSS のレンダリングエンジンであり、CSS を用いてレイアウトされる HTML 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 】

HTML ユニット 5 0 は、HTML により記述された文書を表示又は編集する。SVG ユニット 6 0 は、SVG 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 DOM 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 DOM 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 DOM 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、MVC（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 DOM 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、XML 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、HTML ユニット 5 0 は、HTML 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、SVG ユニット 6 0 は、SVG 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 】

VC 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：VC）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

10

20

30

40

50

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 82 は、定義ファイル取得部 84 が取得したスクリプトファイルを参照して、DOM 生成部 34 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 86 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。

#### 【0022】

VC ユニット 80 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。DOM ユニット 30 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、VC ユニット 80 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 HTML ユニット 50 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。

10

#### 【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。

20

30

#### 【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。

40

#### 【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を表示 / 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。

50

## 【 0 0 2 6 】

図 3 は、図 2 に示した X M L 文書を H T M L 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、成績管理ポキャブラリの「生徒」ノードを、H T M L 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 X M L 文書を、H T M L 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 H T M L による表示画面上で、H T M L ユニット 5 0 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。

10

## 【 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。

20

30

## 【 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。

40

## 【 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

50



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、HTMLユニット50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。

#### 【0030】

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86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91には、マッピング元のXML文書がツリー表示されている。画面右側の領域92には、マッピング先のXML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、HTMLユニット50により編集可能となっており、ユーザは、画面右側の領域92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91に表示されたマッピング元のXML文書のノードを、画面右側の領域92に表示されたHTML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、HTML画面の表90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「TD」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。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コネクションを記述した定義ファイルを生成する。

10

#### 【0031】

XHTML、MathML、SVGなどの主要なポ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ポ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、他のポ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、V C機能を利用して、オリジナルなポ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。

20

#### 【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、成績管理ポ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文書内に複数のポ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。

30

#### 【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V C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V C機能により表示を行っている場合は、V Cユニット80がソースツリーの変更に従っ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。

40

50

## 【 0 0 3 4 】

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。

## 【 0 0 3 5 】

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 90 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 22 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 20 により編集される XML 文書の他の例を示す。図 9 に示した XML 文書では、SVG 文書の「foreignObject」タグの中に XHTML 文書が埋め込まれており、さらに、XHTML 文書の中に MathML 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 24 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 24 は、まず、SVG ユニット 60 に四角形を描画させ、つづいて、HTML ユニット 50 に XHTML 文書を描画させる。さらに、図示しない MathML ユニットに、数式を描画させる。こうして、複数のポ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 10 に示す。

## 【 0 0 3 8 】

文書編集時、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、SVG 文書が表示された領域内に存在するときは、SVG ユニット 60 が提供するメニュー、又は SVG 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、XHTML 文書が表示された領域内に存在するときは、HTML ユニット 50 が提供するメニュー、又は XHTML 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、あるポ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、そのポ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 XML 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 XML などの文

10

20

30

40

50

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。

10

【 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。ソースツリーには、ノードごとにミュート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V Cユニット 8 0 がリスナーとして登録される。D O M提供部 3 2 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、< h t m l > ノードの下位のノードが変更された場合、< h t m l > ノードにリスナーとして登録されたH T M Lユニット 5 0 にミュートションイベントが通知されるとともに、その上位の< s v g > ノードにリスナーとして登録されたS V Gユニット 6 0 にもミュートションイベントが通知される。このとき、H T M Lユニット 5 0 は、変更されたソースツリーを参照して表示を更新する。S V Gユニット 6 0 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、ミュートションイベントを無視してもよい。

20

【 0 0 4 3 】

編集の内容によっては、H T M Lユニット 5 0 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、H T M Lユニット 5 0 による表示領域が以前より大きくなった場合、H T M Lユニット 5 0 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。

30

【 0 0 4 4 】

つづいて、前提技術の文書処理装置 2 0 を実現する機能構成について更に詳細に説明する。以下の説明では、クラス名などを記載する際には、英字をそのまま用いて記載することにする。

40

【 0 0 4 5 】

A . 概要

インターネットの出現により、ユーザによって処理され管理される文書の数が、ほぼ指数関数的に増加してきた。インターネットの核を形成するウェブ (World Wide Web) は、そのような文書データの大きな受け皿となっている。ウェブは、文書に加えて、このような文書の情報検索システムを提供する。これらの文書は、通常、マークアップ言語により記述される。マークアップ言語のシンプルかつポピュラーな例の一つにH T M L (HyperText Markup Language) がある。このような文書は、ウェブの他の位置に格納されている他の文書へのリンクをさらに含む。X M L (eXtensible Markup Language) は、さらに高

50

度でポピュラーなマークアップ言語である。ウェブ文書にアクセスし、閲覧するためのシンプルなブラウザが、Java（登録商標）のようなオブジェクト指向のプログラミング言語で開発されている。

【0046】

マークアップ言語により記述された文書は、通常、ブラウザや他のアプリケーションの中では、ツリーデータ構造の形で表現される。この構造は、文書を構文解析した結果のツリーに相当する。DOM（Document Object Model）は、文書を表現し、操作するために使用される、よく知られたツリーベースのデータ構造モデルである。DOMは、HTMLやXML文書などを含む文書を表現するための標準的なオブジェクトのセットを提供する。DOMは、文書内のコンポーネントを表現するオブジェクトがどのようにつながっているかという標準モデルと、それらのオブジェクトにアクセスしたり操作したりするための標準インタフェイスという、2つの基本的なコンポーネントを含む。

10

【0047】

アプリケーション開発者は、独自のデータ構造やAPI（Application Program Interface）へのインタフェイスとしてDOMをサポートすることができる。他方、文書を作成するアプリケーション開発者は、彼らのAPIの独自インタフェイスではなく、DOMの標準インタフェイスを使用することができる。したがって、標準を提供するというその能力により、DOMは、様々な環境、特にウェブにおいて、文書の相互利用を促進させるために有効である。DOMのいくつかのバージョンが定義されており、異なるプログラミング環境及びアプリケーションによって使用されている。

20

【0048】

DOMツリーは、対応するDOMの内容に基づいた文書の階層的表現である。DOMツリーは「根（ルート）」、及びルートから発生する1つ以上の「節（ノード）」を含む。ルートが文書全体を表す場合もある。中間のノードは、例えば、テーブル及びそのテーブル中の行及び列のような要素を表すことができる。DOMツリーの「葉」は、通常、それ以上分解できないテキストや画像のようなデータを表す。DOMツリーの各ノードは、フォント、サイズ、色、インデントなど、ノードによって表される要素のパラメータを記述する属性に関連付けられてもよい。

【0049】

HTMLは、文書を作成するために一般に用いられる言語であるが、フォーマット及びレイアウト用の言語であり、データ記述のための言語ではない。HTMLドキュメントを表現するDOMツリーのノードは、HTMLのフォーマットタグとして予め定義されたエレメントであって、通常、HTMLは、データの詳述や、データのタギング/ラベリングのための機能を提供しないので、HTMLドキュメント中のデータに対するクエリを定式化することは多くの場合困難である。

30

【0050】

ネットワーク設計者たちの目指すものは、ウェブ上の文書がソフトウェアアプリケーションによってクエリされたり処理されたりできるようにすることである。表示方法とは無関係で、階層的に構造化された言語であれば、そのようにクエリされ処理されることができる。XML（eXtensible Markup Language）のようなマークアップ言語は、これらの特徴を提供することができる。

40

【0051】

HTMLとは逆に、XMLのよく知られた利点は、文書の設計者が自由に定義可能な「タグ」を使用して、データ要素にラベルを付けることが可能である点である。このようなデータ要素は、階層的に構造化することができる。さらに、XML文書は、文書内で用いられるタグ及びそれらの相互関係の「文法」を記述した文書型定義を含むことができる。構造化されたXML文書の表示方法を定義するために、CSS（Cascading Style Sheet）又はXSL（XML Style Language）が使用される。DOM、HTML、XML、CSS、XSL及び関連する言語の特徴に関する付加的な情報は、ウェブからも得ることができる。（例えば、<http://www.w3.org/TR/>）

50

## 【 0 0 5 2 】

X p a t h は、X M L 文書の部分の位置を指定するために共通のシンタックス及びセマンティクスを提供する。機能性の例として、X M L 文書に対応する D O M ツリーのトラバース（移動）がある。それは、X M L 文書の様々な表現に関連した文字列、数、及びブーリアン文字の操作のための基本的な機能を提供する。X p a t h は、X M L 文書の見目のシンタックス、例えば、テキストとしてみたときに何行目であるとか何文字目であるとかといった文法ではなく、D O M ツリーなどの抽象的・論理的な構造において動作する。X p a t h を使用することにより、例えば X M L 文書の D O M ツリー内の階層的構造を通じて場所を指定することができる。アドレッシングのための使用の他に、X p a t h は、D O M ツリー中のノードがパターンにマッチするか否かをテストするために使用されるようにも設計されている。X P a t h に関する更なる詳細は、<http://www.w3.org/TR/xpath>で得ることができる。

10

## 【 0 0 5 3 】

X M L の既知の利点及び特徴により、マークアップ言語（例えば X M L ）で記述された文書を扱うことができ、文書を作成及び修正するためのユーザフレンドリーなインタフェースを提供することができる、効果的な文書処理システムが求められる。

## 【 0 0 5 4 】

ここで説明されるシステムの構成のうちのいくつかは、M V C（Model-View-Controller）と呼ばれる、よく知られた G U I（Graphical User Interface）パラダイムを用いて説明される。M V C パラダイムは、アプリケーション又はアプリケーションのインタフェースの一部を、3つの部分、すなわち、モデル、ビュー、コントローラに分割する。M V C は、元は、G U I の世界に、従来の入力、処理、出力の役割を割り当てるために開発された。

20

[ 入力 ]      [ 処理 ]      [ 出力 ]  
[ コントローラ ]      [ モデル ]      [ ビュー ]

## 【 0 0 5 5 】

M V C パラダイムによれば、外界のモデリング、ユーザへの視覚的なフィードバック、及びユーザの入力は、モデル（M）、ビュー（V）、及びコントローラ（C）オブジェクトにより分離されて扱われる。コントローラは、ユーザからのマウスとキーボード入力のような入力を解釈し、これらのユーザアクションを、適切な変更をもたらすためにモデル及び/又はビューに送られるコマンドにマップするように作用する。モデルは、1以上のデータ要素を管理するように作用し、その状態に関するクエリに応答し、状態を変更する指示に応答する。ビューは、ディスプレイの長方形の領域を管理するように作用し、グラフィクスとテキストの組合せによりユーザにデータを提示する機能を有する。

30

## 【 0 0 5 6 】

B . 文書処理システムの全体構成

文書処理システムの実施例は、図 1 1 - 2 9 に関連して明らかにされる。

## 【 0 0 5 7 】

図 1 1（a）は、後述するタイプの文書処理システムの基礎として機能する要素の従来の構成例を示す。構成 1 0 は、通信経路 1 3 によりメモリ 1 2 に接続された C P U 又はマイクロプロセッサ 1 1 などの形式のプロセッサを含む。メモリ 1 2 は、現在又は将来に利用可能な任意の R O M 及び/又は R A M の形式であってもよい。通信経路 1 3 は、典型的にはバスとして設けられる。マウス、キーボード、音声認識システムなどのユーザ入力装置 1 4 及び表示装置 1 5（又は他のユーザインタフェース）に対する入出力インタフェース 1 6 も、プロセッサ 1 1 とメモリ 1 2 の通信のためのバスに接続される。この構成は、スタンドアロンであってもよいし、複数の端末及び 1 以上のサーバが接続されてネットワーク化された形式であってもよいし、既知のいかなる方式により構成されてもよい。本発明は、これらのコンポーネントの配置、集中又は分散されたアーキテクチャー、あるいは様々なコンポーネントの通信方法により制限されない。

40

## 【 0 0 5 8 】

50

さらに、本システム及びここで議論される実施例は、様々な機能性を提供するいくつかのコンポーネント及びサブコンポーネントを含むものとして議論される。これらのコンポーネント及びサブコンポーネントは、注目された機能性を提供するために、ハードウェアとソフトウェアの組合せだけでなく、ハードウェアのみ、ソフトウェアのみによっても実現されうる。さらに、ハードウェア、ソフトウェア、及びそれらの組合せは、汎用の計算装置、専用のハードウェア、又はそれらの組合せにより実現されうる。したがって、コンポーネント又はサブコンポーネントの構成は、コンポーネント又はサブコンポーネントの機能性を提供するための特定のソフトウェアを実行する汎用/専用の計算装置を含む。

【0059】

図11(b)は、文書処理システムの一例の全体のブロック図を示す。このような文書処理システムにおいて文書が生成され編集される。これらの文書は、例えばXMLなど、マークアップ言語の特徴を有する任意の言語により記述されてもよい。また、便宜上、特定のコンポーネント及びサブコンポーネントの用語及び表題を創造した。しかしながら、これらは、この開示の一般的な教示の範囲を制限するために解釈されるべきではない。

【0060】

文書処理システムは、2つの基本的な構成を有するものととらえることができる。第1の構成は、文書処理システムが動作する環境である「実行環境」101である。例えば、実行環境は、文書の処理中及び管理中に、ユーザだけでなくシステムも支援する、基本的なユーティリティ及び機能を提供する。第2の構成は、実行環境において走るアプリケーションから構成される「アプリケーション」102である。これらのアプリケーションは、文書自身及び文書の様々な表現を含む。

【0061】

#### 1. 実行環境

実行環境101のキーとなるコンポーネントはProgramInvoker(プログラムインボーカー:プログラム起動部)103である。ProgramInvoker103は、文書処理システムを起動するためにアクセスされる基本的なプログラムである。例えば、ユーザが文書処理システムにログオンして開始するとき、ProgramInvoker103が実行される。ProgramInvoker103は、例えば、文書処理システムにプラグインとして加えられた機能を読み出して実行させたり、アプリケーションを開始して実行させたり、文書に関連するプロパティを読み出すことができる。ProgramInvoker103の機能はこれらに限定されない。ユーザが実行環境内で実行されるように意図されたアプリケーションを起動したいとき、ProgramInvoker103は、そのアプリケーションを見つけ、それを起動して、アプリケーションを実行する。

【0062】

ProgramInvoker103には、プラグインサブシステム104、コマンドサブシステム105、及びResource(リソース)モジュール109などのいくつかのコンポーネントがアタッチされている。これらの構成については、以下に詳述する。

【0063】

#### a) プラグインサブシステム

プラグインサブシステム104は、文書処理システムに機能を追加するための高度に柔軟で効率的な構成として使用される。プラグインサブシステム104は、また、文書処理システムに存在する機能を修正又は削除するために使用することができる。さらに、種々の機能をプラグインサブシステムを使用して追加又は修正することができる。例えば、画面上への文書の描画を支援するように作用するEditlet(エディットレット:編集部)機能を追加することもできる。Editletプラグインは、システムに追加されるボキャブラリの編集も支援する。

【0064】

プラグインサブシステム104は、ServiceBroker(サービスブローカ:サービス仲介部)1041を含む。ServiceBroker1041は、文書処理システムに加えらるるプラグインを管理することにより、文書処理システムに加えらるるサービスを仲介する。

## 【 0 0 6 5 】

所望の機能性を実現する個々の機能は、Service ( サービス ) 1 0 4 2 の形でシステムに追加される。利用可能なService 1 0 4 2 のタイプは、Application ( アプリケーション ) サービス、ZoneFactory ( ゾーンファクトリ : ゾーン生成部 ) Service、Editlet ( エディットレット : 編集部 ) Service、CommandFactory ( コマンドファクトリ : コマンド生成部 ) Service、ConnectXPath ( コネクト X P a t h : X P a t h 管理部 ) Service、CSSComputation ( C S S コンピューテーション : C S S 計算部 ) Serviceなどを含むが、これらに限定されない。これらのService、及びシステムの他の構成とそれらとの関係は、文書処理システムについてのよりよい理解のために、以下に詳述される。

## 【 0 0 6 6 】

プラグインとServiceの関係は以下の通りである。プラグインは、1以上のServiceProvider ( サービスプロバイダ : サービス提供部 ) を含むことができるユニットである。それぞれのServiceProviderは、それに関連したServiceの1以上のクラスを有する。例えば、適切なソフトウェアアプリケーションを有する単一のプラグインを使用することにより、1以上のServiceをシステムに追加することができ、これにより、対応する機能をシステムに追加することができる。

## 【 0 0 6 7 】

## b ) コマンドサブシステム

コマンドサブシステム 1 0 5 は、文書の処理に関連したコマンドの形式の命令を実行するために使用される。ユーザは、一連の命令を実行することにより、文書に対する操作を実行することができる。例えば、ユーザは、コマンドの形で命令を発行することにより、文書処理システム中のXML文書に対応するXMLのDOMツリーを編集し、XML文書を処理する。これらのコマンドは、キーストローク、マウスクリック、又は他の有効なユーザインタフェースアクションを使用して入力されてもよい。1つのコマンドにより1以上の命令が実行されることもある。この場合、これらの命令が1つのコマンドにラップ ( 包含 ) され、連続して実行される。例えば、ユーザが、誤った単語を正しい単語に置換したいとする。この場合、第1の命令は、文書中の誤った単語を発見することであり、第2の命令は、誤った単語を削除することであり、第3の命令は、正しい単語を挿入することであってもよい。これらの3つの命令が1つのコマンドにラップされてもよい。

## 【 0 0 6 8 】

コマンドは、関連した機能、例えば、後で詳述する「アンドゥ」機能を有してもよい。これらの機能は、オブジェクトを生成するために使用されるいくつかの基本クラスにも割り当てられてもよい。

## 【 0 0 6 9 】

コマンドサブシステム 1 0 5 のキーとなるコンポーネントは、選択的にコマンドを与え、実行するように作用するCommandInvoker ( コマンドインボーク : コマンド起動部 ) 1 0 5 1 である。図 1 1 ( b ) には、1つのCommandInvokerのみが示されているが、1以上のCommandInvokerが使用されてもよく、1以上のコマンドが同時に実行されてもよい。CommandInvoker 1 0 5 1 は、コマンドを実行するために必要な機能及びクラスを保持する。動作において、実行されるべきCommand ( コマンド : 命令 ) 1 0 5 2 は、Queue ( キュー ) 1 0 5 3 に積まれる。CommandInvokerは、連続的に実行するコマンドスレッドを生成する。CommandInvoker内で既に実行中のCommandがなければ、CommandInvoker 1 0 5 1 により実行されるように意図されたCommand 1 0 5 2 が実行される。CommandInvokerが既にコマンドを実行している場合、新しいCommandは、Queue 1 0 5 3 の最後に積まれる。しかしながら、それぞれのCommandInvoker 1 0 5 1 では、一度に1つのCommandのみが実行される。指定されたCommandの実行に失敗した場合、CommandInvoker 1 0 5 1 は例外処理を実行する。

## 【 0 0 7 0 】

CommandInvoker 1 0 5 1 により実行されるCommandの型は、UndoableCommand ( 取消可能コマンド ) 1 0 5 4、AsynchronousCommand ( 非同期コマンド ) 1 0 5 5、及びVCommand

10

20

30

40

50

( V C コマンド ) 1 0 5 6 を含むが、これらに限定されない。UndoableCommand 1 0 5 4 は、ユーザが望めば、そのCommandの結果を取り消すことが可能なCommandである。UndoableCommandの例として、切り取り、コピー、テキストの挿入、などがある。動作において、ユーザが文書の一部を選択し、その部分に切り取りコマンドを適用するとき、UndoableCommandを用いることにより、切り取られた部分は、必要であれば、「切り取られていない」ようにすることができる。

#### 【 0 0 7 1 】

VCCCommand 1 0 5 6 は、ボキャブラリコネクション記述子 ( Vocabulary Connection Descriptor : V C D ) スクリプトファイルに格納される。これらは、プログラマにより定義されうるユーザ指定のCommandである。Commandは、例えば、XMLフラグメントを追加したり、XMLフラグメントを削除したり、属性を設定したりするための、より抽象的なCommandの組合せであってもよい。これらのCommandは、特に、文書の編集に焦点を合わせている。

10

#### 【 0 0 7 2 】

AsynchronousCommand 1 0 5 5 は、文書のロードや保存など、システムよりのCommandであり、UndoableCommandやVCCCommandとは別に、非同期的に実行される。AsynchronousCommandは、UndoableCommandではないので、取り消すことはできない。

#### 【 0 0 7 3 】

##### c ) リソース

Resource 1 0 9 は、様々なクラスに、いくつかの機能を提供するオブジェクトである。例えば、ストリングリソース、アイコン、及びデフォルトキーバインドは、システムで使用されるResourceの例である。

20

#### 【 0 0 7 4 】

##### 2 . アプリケーションコンポーネント

文書処理システムの第2の主要な特徴であるアプリケーションコンポーネント 1 0 2 は、実行環境 1 0 1 において実行される。アプリケーションコンポーネント 1 0 2 は、実際の文書と、システム内における文書の様々な論理的、物理的な表現を含む。さらに、アプリケーションコンポーネント 1 0 2 は、文書を管理するために使用されるシステムの構成を含む。アプリケーションコンポーネント 1 0 2 は、さらに、UserApplication ( ユーザアプリケーション ) 1 0 6 、アプリケーションコア 1 0 8 、ユーザインタフェイス 1 0 7 、及びCoreComponent ( コアコンポーネント ) 1 1 0 を含む。

30

#### 【 0 0 7 5 】

##### a ) ユーザアプリケーション

UserApplication 1 0 6 は、ProgramInvoker 1 0 3 と共にシステム上にロードされる。UserApplication 1 0 6 は、文書と、文書の様々な表現と、文書と対話するために必要なユーザインタフェイスとをつなぐ接着剤となる。例えば、ユーザが、プロジェクトの一部である文書のセットを生成したいとする。これらの文書がロードされると、文書の適切な表現が生成される。ユーザインタフェイス機能は、UserApplication 1 0 6 の一部として追加される。言い換えれば、UserApplication 1 0 6 は、ユーザがプロジェクトの一部を形成する文書と対話することを可能とする文書の表現と、文書の様々な態様とを、共に保持する。一旦UserApplication 1 0 6 が生成されると、ユーザがプロジェクトの一部を形成する文書との対話を望むたびに、ユーザは簡単に実行環境上にUserApplication 1 0 6 をロードすることができる。

40

#### 【 0 0 7 6 】

##### b ) コアコンポーネント

CoreComponent 1 1 0 は、複数のPane ( ペイン ) の間で文書を共有する方法を提供する。後で詳述するように、Paneは、DOMツリーを表示し、画面の物理的なレイアウトを扱う。例えば、物理的な画面は、個々の情報の断片を描写する画面内の複数のPaneからなる。ユーザから画面上に見える文書は、1又はそれ以上のPaneに出現しうる。また、2つの異なる文書が画面上で2つの異なるPaneに現れてもよい。

50



## 【 0 0 7 7 】

図 1 1 ( c ) に示されるように、画面の物理的なレイアウトもツリーの形式になっている。Pane は、RootPane ( ルートペイン ) 1 0 8 4 にもなり得るし、SubPane ( サブペイン ) 1 0 8 5 にもなり得る。RootPane 1 0 8 4 は、Pane のツリーの根に当たる Pane であり、SubPane 1 0 8 5 は、RootPane 1 0 8 4 以外の任意の Pane である。

## 【 0 0 7 8 】

CoreComponent 1 1 0 は、さらに、フォントを提供し、ツールキットなど、文書のための複数の機能的な操作のソースの役割を果たす。CoreComponent 1 1 0 により実行されるタスクの一例に、複数の Pane 間におけるマウスカーソルの移動がある。実行されるタスクの他の例として、ある Pane 中の文書の一部をマークし、それを異なる文書を含む別の Pane 上にコピーする。

10

## 【 0 0 7 9 】

## c ) アプリケーションコア

上述したように、アプリケーションコンポーネント 1 0 2 は、システムにより処理され管理される文書から構成される。これは、システム内における文書の様々な論理的及び物理的な表現を含む。アプリケーションコア 1 0 8 は、アプリケーションコンポーネント 1 0 2 の構成である。その機能は、実際の文書を、それに含まれる全てのデータとともに保持することである。アプリケーションコア 1 0 8 は、DocumentManager ( ドキュメントマネージャ : 文書管理部 ) 1 0 8 1 及び Document ( ドキュメント : 文書 ) 1 0 8 2 自身を含む。

20

## 【 0 0 8 0 】

DocumentManager 1 0 8 1 の様々な態様を以下に詳述する。DocumentManager 1 0 8 1 は、Document 1 0 8 2 を管理する。DocumentManager 1 0 8 1 は、RootPane 1 0 8 4 、SubPane 1 0 8 5 、Clipboard ( クリップボード ) ユーティリティ 1 0 8 7 、及び Snapshot ( スナップショット ) ユーティリティ 1 0 8 8 にも接続される。Clipboard ユーティリティ 1 0 8 7 は、ユーザがクリップボードに加えることを決定した文書の部分を保持する方法を提供する。例えば、ユーザが、文書の一部を切り取り、後で再考するために新規文書にそれを保存することを望んだとする。このような場合、切り取られた部分が Clipboard に追加される。

## 【 0 0 8 1 】

つづいて、Snapshot ユーティリティ 1 0 8 8 についても説明する。Snapshot ユーティリティ 1 0 8 8 は、アプリケーションがある状態から別の状態まで移行するときに、アプリケーションの現在の状態を記憶することを可能とする。

30

## 【 0 0 8 2 】

## d ) ユーザインタフェイス

アプリケーションコンポーネント 1 0 2 の別の構成は、ユーザがシステムと物理的に対話する手段を提供するユーザインタフェイス 1 0 7 である。例えば、ユーザインタフェイスは、ユーザが文書をアップロードしたり、削除したり、編集したり、管理したりするために使用される。ユーザインタフェイスは、Frame ( フレーム ) 1 0 7 1 、MenuBar ( メニューバー ) 1 0 7 2 、StatusBar ( ステータスバー ) 1 0 7 3 、及び URLBar ( URL バー ) 1 0 7 4 を含む。

40

## 【 0 0 8 3 】

Frame 1 0 7 1 は、一般に知られているように、物理的な画面のアクティブな領域であるとみなされる。MenuBar 1 0 7 2 は、ユーザに選択を提供するメニューを含む画面領域である。StatusBar 1 0 7 3 は、アプリケーションの実行状態を表示する画面領域である。URLBar 1 0 7 4 は、インターネットをナビゲートするために URL アドレスを入力する領域を提供する。

## 【 0 0 8 4 】

## C . 文書管理及び関連するデータ構造

図 1 2 は、DocumentManager 1 0 8 1 の詳細を示す。これは、文書処理システム内で文

50

書を表現するために用いられるデータ構造及び構成を含む。分かりやすくするために、このサブセクションで説明される構成は、MVCパラダイムを用いて説明される。

#### 【0085】

DocumentManager 1081は、文書処理システム内にある全ての文書を保持しホストするDocumentContainer (ドキュメントコンテナ: 文書コンテナ) 203を含む。DocumentManager 1081にアタッチされたツールキット201は、DocumentManager 1081により使用される様々なツールを提供する。例えば、DomService (DOMサービス)は、文書に対応するDOMを生成し、保持し、管理するために必要とされる全ての機能を提供するために、ツールキット201により提供されるツールである。ツールキット201により提供される別のツールであるIOManager (入出力管理部)は、システムへの入力及びシステムからの出力を管理する。同様に、StreamHandler (ストリームハンドラ)は、ビットストリームによる文書のアップロードを扱うツールである。これらのツールは、図中に特に示さず、参照番号を割り当てないが、ツールキット201のコンポーネントを形成する。

10

#### 【0086】

MVCパラダイムの表現によれば、モデル(M)は、文書のDOMツリーモデル202を含む。前述したように、全ての文書は、文書処理システムにおいてDOMツリーとして表現される。文書は、また、DocumentContainer 203の一部を形成する。

#### 【0087】

##### 1. DOMモデル及びゾーン

文書を表現するDOMツリーは、Node (ノード) 2021を有するツリーである。DOMツリーの部分集合であるZone (ゾーン) 209は、DOMツリー内の1以上のNodeの関連領域を含む。例えば、画面上で文書の一部のみを表示し得るが、この可視化された文書の一部はZone 209を用いて表示される。Zoneは、ZoneFactory (ゾーンファクトリ: ゾーン生成部) 205と呼ばれるプラグインを用いて、生成され、取り扱われ、処理される。ZoneはDOMの一部を表現するが、1以上の「名前空間」を使用してもよい。よく知られているように、名前空間は、名前空間内でユニークな名前の集合である。換言すれば、名前空間内に同じ名前は存在しない。

20

#### 【0088】

##### 2. Facet及びFacetとZoneとの関係

Facet (ファセット) 2022は、MVCパラダイムのモデル(M)部分内の別の構成である。Facetは、ZoneにおいてNodeを編集するために使用される。Facet 2022は、Zone自身の内容に影響を与えずに実行することができる手続(プロシージャ)を使用して、DOMへのアクセスを編成する。次に説明するように、これらの手続は、Nodeに関連した重要で有用な操作を実行する。

30

#### 【0089】

各Nodeは、対応するFacetを有する。DOMの中のNodeを直接操作する代わりに、操作を実行するためにFacetを使用することによって、DOMの健全性は保護される。操作がNode上で直接実行される場合、いくつかのプラグインがDOMを同時に変更することができ、その結果矛盾を引き起こす。

40

#### 【0090】

W3Cが策定したDOMの標準規格は、Nodeを操作するための標準的なインタフェースを定義するが、実際には、ボキャブラリごと又はNodeごとに特有の操作があるので、これらの操作をAPIとして用意しておくのが好都合である。文書処理システムでは、このような各Nodeに特有のAPIをFacetとして用意し、各Nodeにアタッチする。これにより、DOMの標準規格に準拠しつつ、有用なAPIを付加することができる。また、ボキャブラリごとに特有のDOMを実装するのではなく、標準的なDOMの実装に、後から特有のAPIを付加するようにすることで、多様なボキャブラリを統一的に処理できるとともに、複数のボキャブラリが任意の組合せで混在した文書を適切に処理することができる。

50

## 【 0 0 9 1 】

ボキャブラリは、名前空間に属するタグ（例えばXMLのタグ）のセットである。上述したように、名前空間は、ユニークな名前（ここではタグ）のセットを有する。ボキャブラリは、XML文書表現するDOMツリーのサブツリーとして現れる。このサブツリーはZoneを含む。特定の例においては、タグセットの境界はZoneによって定義される。Zone 2 0 9は、ZoneFactory 2 0 5と呼ばれるServiceを利用して生成される。上述したように、Zone 2 0 9は、文書表現するDOMツリーの一部の内部表現である。このような文書の一部へのアクセスを提供するために、論理的な表現が要求される。この論理的表現は、文書が画面上で論理的にどのように表現されるかについてコンピュータに通知する。Canvas（キャンパス）2 1 0は、Zoneに対応する論理的なレイアウトを提供するように作用するServiceである。

10

## 【 0 0 9 2 】

他方、Pane 2 1 1は、Canvas 2 1 0により提供される論理的なレイアウトに対応する物理的な画面レイアウトである。実際、ユーザは表示画面上で文字や画像によって文書のレンダリングのみを見る。したがって、文書は、画面上に文字や画像を描画するプロセスにより、画面上に描写されなければならない。文書は、Pane 2 1 1により提供される物理的なレイアウトに基づいて、Canvas 2 1 0により画面上に描写される。

## 【 0 0 9 3 】

Zone 2 0 9に対応するCanvas 2 1 0は、Editlet 2 0 6を使用して生成される。文書のDOMは、Editlet 2 0 6及びCanvas 2 1 0を使用して編集される。元の文書の完全性を維持するために、Editlet 2 0 6及びCanvas 2 1 0は、Zone 2 0 9における1以上のNodeに対応するFacetを使用する。これらのServiceは、Zone及びDOM内のNodeを直接操作しない。Facetは、Command 2 0 7を利用して操作される。

20

## 【 0 0 9 4 】

ユーザは、一般に、画面上のカーソルを移動させたり、コマンドをタイプしたりすることによって、画面と対話する。画面上の論理的なレイアウトを提供するCanvas 2 1 0は、このカーソル操作を受け付ける。Canvas 2 1 0は、対応するアクションをFacetに実行させることができる。この関係により、カーソルサブシステム 2 0 4は、DocumentManager 1 0 8 1に対して、MVCパラダイムのコントローラ（C）として機能する。Canvas 2 1 0は、イベントを扱うタスクも有する。例えば、Canvas 2 1 0は、マウスクリック、フォーカス移動、及びユーザにより起こされた同様のアクションなどのイベントを扱う。

30

## 【 0 0 9 5 】

## 3 . Zone、Facet、Canvas及びPaneの間の関係の概要

文書処理システム内の文書は、少なくとも4つの観点から見る事ができる。すなわち、1) 文書処理システムにおいて文書の内容及び構造を保持するために用いられるデータ構造、2) 文書の保全性に影響を与えずに文書の内容を編集する手段、3) 文書の画面上の論理的なレイアウト、4) 文書の画面上の物理的なレイアウト、である。Zone、Facet、Canvas及びPaneは、前述の4つの観点に相当する、文書処理システムのコンポーネントをそれぞれ表す。

## 【 0 0 9 6 】

## 4 . アンドゥサブシステム

上述したように、文書に対するいかなる変更（例えば編集）も取消可能であることが望ましい。例えば、ユーザが編集操作を実行し、次に、その変更の取消を決定したとする。図12に関連して、アンドゥサブシステム 2 1 2は、文書管理部の取消可能なコンポーネントを実現する。UndoManager（アンドゥマネージャ：アンドゥ管理部）2 1 2 1は、ユーザによって取り消される可能性のある全ての文書に対する操作を保持する。

40

## 【 0 0 9 7 】

例えば、ユーザが、文書中の単語を別の単語に置換するコマンドを実行したとする。その後、ユーザは考え直し、元の単語に戻すことを決定したとする。アンドゥサブシステム 2 1 2は、このような操作を支援する。UndoManager 2 1 2 1は、このようなUndoableEdi

50

t (アンドゥアブルエディット：取消可能な編集) 2 1 2 2 の操作を保持する。

【0098】

#### 5. カーソルサブシステム

前述したように、MVCのコントローラ部分は、カーソルサブシステム204を備えてもよい。カーソルサブシステム204は、ユーザから入力を受け付ける。これらの入力は、一般にコマンド及び/又は編集操作の性格を有している。したがって、カーソルサブシステム204は、DocumentManager1081に関連したMVCパラダイムのコントローラ(C)部分であると考えることができる。

【0099】

#### 6. ビュー

前述したように、Canvas210は、画面上に提示されるべき文書の論理的なレイアウトを表す。XHTML文書の例では、Canvas210は、文書が画面上でいかに見えるかを論理的に表現したボックスツリー208を含んでもよい。このボックスツリー208は、DocumentManager1081に関連したMVCパラダイムのビュー(V)部分に含まれよう。

【0100】

#### D. ボキャブラリコネクション

文書処理システムの重要な特徴は、XML文書を、他の表現にマップして取り扱うことが可能で、かつ、マップした先の表現を編集すると、その編集が元のXML文書に整合性を保ちつつ反映される環境を提供することにある。

【0101】

マークアップ言語により記述された文書、例えばXML文書は、文書型定義により定義されたボキャブラリに基づいて作成されている。ボキャブラリは、タグのセットである。ボキャブラリは、任意に定義されてもよいため、無限に多くのボキャブラリが存在しうる。しかしながら、多数の可能なボキャブラリのそれぞれに対して専用の処理/管理環境を提供するのは現実的ではない。ボキャブラリコネクションは、この問題を解決する方法を提供する。

【0102】

例えば、文書は2以上のマークアップ言語により記述されてもよい。文書は、例えば、XHTML(eXtensible HyperText Markup Language)、SVG(Scalable Vector Graphics)、MathML(Mathematical Markup Language)、その他のマークアップ言語により記述されてもよい。換言すれば、マークアップ言語は、XMLにおけるボキャブラリやタグセットと同様に見なされてもよい。

【0103】

ボキャブラリは、ボキャブラリプラグインを用いて処理される。文書処理システムにおいてプラグインが利用不可能であるボキャブラリにより記述された文書は、プラグインが利用可能である別のボキャブラリの文書にマッピングすることにより表示される。この特徴により、プラグインが用意されていないボキャブラリの文書も適切に表示することができる。

【0104】

ボキャブラリコネクションは、定義ファイルを取得し、取得した定義ファイルに基づいて2つの異なるボキャブラリの間でマッピングする能力を含む。あるボキャブラリで記述された文書は、別のボキャブラリにマッピングすることができる。このように、ボキャブラリコネクションは、文書がマッピングされるボキャブラリに対応した表示/編集プラグインにより文書を表示し編集することを可能にする。

【0105】

上述したように、各文書は、一般に複数のノードを有するDOMツリーとして文書処理システムにおいて記述される。「定義ファイル」は、それぞれのノードについて、そのノードと他のノードとの対応を記述する。各ノードの要素値及び属性値が編集可能か否かが指定される。ノードの要素値又は属性値を用いた演算式が記述されてもよい。

【0106】

10

20

30

40

50

マッピングという特徴を利用して、定義ファイルを適用したデスティネーションDOMツリーが生成される。このように、ソースDOMツリーとデスティネーションDOMツリーの関係が構築され保持される。ボキャブラリコネクションは、ソースDOMツリーとデスティネーションDOMツリーの対応を監視する。ユーザから編集指示を受けると、ボキャブラリコネクションは、ソースDOMツリーの関連したノードを変更する。ソースDOMツリーが変更されたことを示す「ミュートーションイベント」が発行され、デスティネーションDOMツリーがそれに応じて変更される。

#### 【0107】

ボキャブラリコネクションの使用により、少数のユーザのみに知られていた比較的マイナーなボキャブラリを、別のメジャーなボキャブラリに変換することができる。したがって、少数のユーザによって利用されるマイナーなボキャブラリであっても、文書を適切に表示し、望ましい編集環境を提供することができる。

10

#### 【0108】

このように、文書処理システムの一部であるボキャブラリコネクションサブシステムは、文書の複数の表現を可能にする機能を提供する。

#### 【0109】

図13は、ボキャブラリコネクション(VC: Vocabulary Connection)サブシステム300を示す。VCサブシステム300は、同一の文書の2つの代替表現の整合性を維持する方法を提供する。例えば、2つの表現は、同一文書の、2つの異なるボキャブラリによる表現であってもよい。前述したように、一方はソースDOMツリーであってもよく、他方はデスティネーションDOMツリーであってもよい。

20

#### 【0110】

##### 1. ボキャブラリコネクションサブシステム

ボキャブラリコネクションサブシステム300の機能は、VocabularyConnection301と呼ばれるプラグインを使用して、文書処理システムにおいて実現される。文書が表現されるVocabulary305ごとに、対応するプラグインが要求される。例えば、文書の一部がHTMLで記述され、残りがSVGで記述されている場合、HTMLとSVGに対応するボキャブラリプラグインが要求される。

#### 【0111】

VocabularyConnectionプラグイン301は、適切なVocabulary305の文書に対応した、Zone209又はPane211のための適切なVCCanvas(ボキャブラリコネクションキャンバス)310を生成する。VocabularyConnection301を用いて、ソースDOMツリー内のZone209に対する変更は、変換ルールにより、別のDOMツリー306の対応するZoneに伝達される。変換ルールは、ボキャブラリコネクション記述子(Vocabulary Connection Descriptor: VCD)の形式で記述される。このようなソースDOMとデスティネーションDOMの間の変換に対応するそれぞれのVCDファイルについて、対応するVManager(ボキャブラリコネクションマネージャ)302が生成される。

30

#### 【0112】

##### 2. Connector

Connector304は、ソースDOMツリーのソースノードと、デスティネーションDOMツリーのデスティネーションノードとを接続する。Connector304は、ソースDOMツリー中のソースノード、及びソースノードに対応するソース文書に対する修正(変更)を見るために作用する。そして、対応するデスティネーションDOMツリーのノードを修正する。Connector304は、デスティネーションDOMツリーを修正することができる唯一のオブジェクトである。例えば、ユーザは、ソース文書、及び対応するソースDOMツリーに対してのみ修正を行うことができる。その後、Connector304がデスティネーションDOMツリーに、対応する修正を行う。

40

#### 【0113】

Connector304は、ツリー構造を形成するために、論理的にリンクされる。Connector304により形成されたツリーは、ConnectorTree(コネクタツリー)と呼ばれる。Conne

50

ctor 3 0 4 は、ConnectorFactory (コネクタファクトリ：コネクタ生成部) 3 0 3 と呼ばれるServiceを用いて生成される。ConnectorFactory 3 0 3 は、ソース文書からConnector 3 0 4 を生成し、それらをリンクしてConnectorTreeを形成する。VocabularyConnectionManager 3 0 2 は、ConnectorFactory 3 0 3 を保持する。

#### 【 0 1 1 4 】

前述したように、ボキャブラリは名前空間におけるタグのセットである。図示されるように、Vocabulary 3 0 5 は、VocabularyConnection 3 0 1 によって文書に対して生成される。これは、文書ファイルを解析し、ソースDOMとデスティネーションDOMの間の写像のための適切なVocabularyConnectionManager 3 0 2 を生成することにより行われる。さらに、Connectorを生成するConnectorFactory 3 0 3 と、Zone 2 0 9 を生成するZoneFactory 2 0 5 と、Zone内のノードに対応するCanvasを生成するEditlet 2 0 6 との間の適切な関係が作られる。ユーザがシステムから文書を処分又は削除するとき、対応するVocabularyConnectionManager 3 0 2 が削除される。

10

#### 【 0 1 1 5 】

Vocabulary 3 0 5 は、VCCanvas 3 1 0 を生成する。さらに、Connector 3 0 4 及びデスティネーションDOMツリー 3 0 6 が対応して生成される。

#### 【 0 1 1 6 】

ソースDOM及びCanvasは、それぞれ、モデル(M)及びビュー(V)に対応する。しかしながら、このような表現は、ターゲットのボキャブラリが画面上に描写可能である場合に限って意味がある。描写は、ボキャブラリプラグインにより行われる。ボキャブラリプラグインは、主要なボキャブラリ、例えば、XHTML、SVG、MathMLについて提供される。ボキャブラリプラグインは、ターゲットのボキャブラリに関連して使用される。これらは、ボキャブラリコネクション記述子を用いてボキャブラリ間でマッピングする方法を提供する。

20

#### 【 0 1 1 7 】

このようなマッピングは、ターゲットのボキャブラリが、マッピング可能で、画面上に描写される方法が予め定義されたものである場合にのみ意味がある。このようなレンダリング方法は、例えばXHTMLなどのように、W3Cなどの組織により定義された標準規格となっている。

#### 【 0 1 1 8 】

ボキャブラリコネクションが必要であるとき、VCCanvasが使用される。この場合、ソースのビューを直接生成することができないので、ソースのCanvasは生成されない。この場合、VCCanvasが、ConnectorTreeを使用して生成される。このVCCanvasは、イベントの変換のみを扱い、画面上の文書の描写を援助しない。

30

#### 【 0 1 1 9 】

##### 3 . DestinationZone、Pane、及びCanvas

上述したように、ボキャブラリコネクションサブシステムの目的は、同一の文書の2つの表現を同時に生成し保持することである。第2の表現も、DOMツリーの形式であり、これはデスティネーションDOMツリーとして既に説明した。第2の表現における文書を見るために、DestinationZone、Canvas及びPaneが必要である。

40

#### 【 0 1 2 0 】

VCCanvasが作成されると、対応するDestinationPane 3 0 7 が生成される。さらに、関連するDestinationCanvas 3 0 8 と、対応するBoxTree 3 0 9 が生成される。同様に、VCCanvas 3 1 0 も、ソース文書に対するPane 2 1 1 及びZone 2 0 9 に関連づけられる。

#### 【 0 1 2 1 】

DestinationCanvas 3 0 8 は、第2の表現における文書の論理的なレイアウトを提供する。特に、DestinationCanvas 3 0 8 は、デスティネーション表現における文書を描写するために、カーソルや選択のようなユーザインタフェース機能を提供する。DestinationCanvas 3 0 8 に生じたイベントは、Connectorに供給される。DestinationCanvas 3 0 8 は、マウスイベント、キーボードイベント、ドラッグアンドドロップイベント、及び文書の

50

デスティネーション（第2）表現のボキャブラリに特有なイベントを、Connector 3 0 4 に通知する。

#### 【0122】

##### 4. ボキャブラリコネクションコマンドサブシステム

ボキャブラリコネクション（VC）サブシステム300の要素として、ボキャブラリコネクション（VC）コマンドサブシステム313がある。ボキャブラリコネクションコマンドサブシステム313は、ボキャブラリコネクションサブシステム300に関連した命令の実行のために使用されるVCCommand（ボキャブラリコネクションコマンド）315を生成する。VCCommandは、内蔵のCommandTemplate（コマンドテンプレート）318を使用して、及び/又は、スクリプトサブシステム314においてスクリプト言語を使用してスクラッチからコマンドを生成することにより、生成することができる。

10

#### 【0123】

コマンドテンプレートには、例えば、「If」コマンドテンプレート、「When」コマンドテンプレート、「挿入（Insert）」コマンドテンプレートなどがある。これらのテンプレートは、VCCommandを作成するために使用される。

#### 【0124】

##### 5. XPathサブシステム

XPathサブシステム316は、文書処理システムの重要な構成であり、ボキャブラリコネクションの実現を支援する。Connector 3 0 4は、一般にxpath情報を含む。上述したように、ボキャブラリコネクションのタスクの1つは、ソースDOMツリーの変化をデスティネーションDOMツリーに反映させることである。xpath情報は、変更/修正を監視されるべきソースDOMツリーのサブセットを決定するために用いられる1以上のxpath表現を含む。

20

#### 【0125】

##### 6. ソースDOMツリー、デスティネーションDOMツリー、及びConnectorTreeの概要

ソースDOMツリーは、別のボキャブラリに変換される前のボキャブラリで文書を表現したDOMツリー又はZoneである。ソースDOMツリーのノードは、ソースノードと呼ばれる。

#### 【0126】

それに対して、デスティネーションDOMツリーは、ボキャブラリコネクションに関連して前述したように、同一の文書を、マッピングにより変換された後の異なるボキャブラリで表現したDOMツリー又はZoneである。デスティネーションDOMツリーのノードは、デスティネーションノードと呼ばれる。

30

#### 【0127】

ConnectorTreeは、ソースノードとデスティネーションノードの対応を表すConnectorに基づく階層的表現である。Connectorは、ソースノードと、ソース文書になされた修正を監視し、デスティネーションDOMツリーを修正する。Connectorは、デスティネーションDOMツリーを修正することを許された唯一のオブジェクトである。

#### 【0128】

##### E. 文書処理システムにおけるイベントフロー

実用のためには、プログラムはユーザからのコマンドに応答しなければならない。イベントは、プログラム上で実行されたユーザアクションを記述し実行する方法である。多くの高級言語、例えばJava（登録商標）は、ユーザアクションを記述するイベントに頼っている。従来、プログラムは、ユーザアクションを理解し、それを自身で実行するために、積極的に情報を集める必要があった。これは、例えば、プログラムが自身を初期化した後、ユーザが画面、キーボード、マウスなどでアクションを起こしたときに適切な処理を講じるために、ユーザのアクションを繰り返し確認するループに入ることを意味する。しかしながら、このプロセスは扱いにくい。さらに、それは、ユーザが何かをするのを待つ間、CPUサイクルを消費してループするプログラムを必要とする。

40

50

## 【 0 1 2 9 】

多くの言語が、異なるパラダイムを採用することにより、これらの問題を解決している。そのうちの一つは、現代の全てのウィンドウシステムの基礎となっている、イベントドリブンプログラミングである。このパラダイムでは、全てのユーザアクションは、「イベント」と呼ばれる抽象的な事象の集合に属する。イベントは、十分詳細に、特定のユーザアクションを記述する。プログラムがユーザにより生成されたイベントを積極的に収集するのではなく、監視すべきイベントが生じたときに、システムがプログラムに通知する。この方法によりユーザとの対話を扱うプログラムは「イベントドリブン」であると言われる。

## 【 0 1 3 0 】

これは、多くの場合、全てのユーザにより生成されたイベントの基本特性を獲得する「Event ( イベント ) 」クラスを使用して扱われる。

## 【 0 1 3 1 】

文書処理システムは、自身のイベント、及びこれらのイベントを扱う方法を定義して使用する。いくつかの型のイベントが使用される。例えば、マウスイベントは、ユーザのマウスアクションから起こるイベントである。マウスを含むユーザアクションは、Canvas 2 1 0 によって、マウスイベントに渡される。このように、Canvasは、システムのユーザによる相互作用の最前部にあると言える。必要であれば、最前部にあるCanvasは、そのイベントに関連した内容を子へ渡す。

## 【 0 1 3 2 】

それに対して、キーストロークイベントは、Canvas 2 1 0 から流れる。キーストロークイベントは、即時的なフォーカスを有する。すなわち、それは、いかなる瞬間でも作業に関連する。Canvas 2 1 0 上に入力されたキーストロークイベントは、その親に渡される。キー入力、文字列挿入を扱うことが可能な、異なるイベントによって処理される。文字列の挿入を扱うイベントは、キーボードを使用して文字が挿入されたときに発生する。他の「イベント」は、例えば、ドラッグイベント、ドロップイベント、マウスイベントと同様に扱われる他のイベントを含む。

## 【 0 1 3 3 】

## 1 . ポキャブラリコネクション外のイベントの取り扱い

イベントは、イベントスレッドを用いて渡される。Canvas 2 1 0 は、イベントを受け取ると、その状態を変更する。必要であれば、Command 1 0 5 2 がCanvas 2 1 0 によりCommandQueue 1 0 5 3 にポストされる。

## 【 0 1 3 4 】

## 2 . ポキャブラリコネクション内のイベントの取り扱い

VocabularyConnectionプラグイン 3 0 1 を用いて、DestinationCanvasの一例であるXHTMLCanvas 1 1 0 6 は、発生したイベント、例えば、マウスイベント、キーボードイベント、ドラッグアンドドロップイベント、及びポキャブラリに特有のイベントなどを受け取る。これらのイベントは、コネクタ 3 0 4 に通知される。より詳細には、図 2 1 ( b ) に図示されるように、VocabularyConnectionプラグイン 3 0 1 内のイベントフローは、Source Pane 1 1 0 3、VCCanvas 1 1 0 4、DestinationPane 1 1 0 5、DestinationCanvasの一例であるDestinationCanvas 1 1 0 6、デスティネーションDOMツリー及びConnectorTreeを通過する。

## 【 0 1 3 5 】

## F . ProgramInvoker及びProgramInvokerと他の構成との関係

ProgramInvoker 1 0 3 及びそれと他の構成との関係は、図 1 4 ( a ) に更に詳細に示される。ProgramInvoker 1 0 3 は、文書処理システムを開始するために実行される実行環境中の基本的なプログラムである。図 1 1 ( b ) 及び図 1 1 ( c ) に図示されるように、UserApplication 1 0 6、ServiceBroker 1 0 4 1、CommandInvoker 1 0 5 1、及びResource 1 0 9 は、全てProgramInvoker 1 0 3 に接続される。前述したように、アプリケーション 1 0 2 は、実行環境中で実行されるコンポーネントである。同様に、ServiceBroker 1 0

10

20

30

40

50



4 1 は、システムに様々な機能を加えるプラグインを管理する。他方、CommandInvoker 1 0 5 1 は、ユーザにより提供される命令を実行して、コマンドを実行するために使用されるクラス及びファンクションを保持する。

#### 【 0 1 3 6 】

##### 1 . プラグイン及びサービス

ServiceBroker 1 0 4 1 について、図 1 4 ( b ) を参照して更に詳細に説明する。前述したように、ServiceBroker 1 0 4 1 は、システムに様々な機能を追加するプラグイン ( 及び関連するサービス ) を管理する。Service 1 0 4 2 は、文書処理システムに特徴を追加又は変更可能な最も下の層である。「Service」は、ServiceCategory 4 0 1 とServiceProvider 4 0 2 の 2 つの部分からなる。図 1 4 ( c ) に図示されるように、1 つのServiceCategory 4 0 1 は、複数の関連するServiceProvider 4 0 2 を持ちうる。それぞれのServiceProvider は、特定のServiceCategoryの一部または全部を実行するように作用する。ServiceCategory 4 0 1 は、他方では、Serviceの型を定義する。

10

#### 【 0 1 3 7 】

Serviceは、1 ) 文書処理システムに特定の特色を提供する「特色サービス」、2 ) 文書処理システムにより実行されるアプリケーションである「アプリケーションサービス」、3 ) 文書処理システムの全体にわたって必要な特色を提供する「環境サービス」、の 3 つの型に分類することができる。

#### 【 0 1 3 8 】

Serviceの例は、図 1 4 ( d ) に示される。アプリケーションServiceのCategoryにおいては、システムユーティリティが対応するServiceProviderの例である。同様に、Editlet 2 0 6 はCategoryであり、HTMLEditlet及びSVGEEditletは対応するServiceProviderである。ZoneFactory 2 0 5 は、Serviceの別のCategoryであり、対応するServiceProvider ( 図示せず ) を有する。

20

#### 【 0 1 3 9 】

プラグインは、文書処理システムに機能性を加えると既に説明したが、いくつかのServiceProvider 4 0 2 及びそれらに関連するクラスからなるユニットと見なされてもよい。各プラグインは、宣言ファイルに記述された依存性及びServiceCategory 4 0 1 を有する。

#### 【 0 1 4 0 】

##### 2 . ProgramInvokerとアプリケーションとの関係

図 1 4 ( e ) は、ProgramInvoker 1 0 3 とUserApplication 1 0 6 との関係についての更なる詳細を示す。必要な文書やデータなどは、ストレージからロードされる。必要なプラグインは、全てServiceBroker 1 0 4 1 上にロードされる。ServiceBroker 1 0 4 1 は、全てのプラグインを保持し管理する。プラグインは、システムに物理的に追加することができ、又、その機能はストレージからロードすることができる。プラグインの内容がロードされると、ServiceBroker 1 0 4 1 は、対応するプラグインを定義する。つづいて、対応するUserApplication 1 0 6 が生成され、実行環境 1 0 1 にロードされ、ProgramInvoker 1 0 3 にアタッチされる。

30

#### 【 0 1 4 1 】

##### G . アプリケーションサービスと環境との関係

図 1 5 ( a ) は、ProgramInvoker 1 0 3 上にロードしたアプリケーションサービスの構成についての更なる詳細を示す。コマンドサブシステム 1 0 5 のコンポーネントであるCommandInvoker 1 0 5 1 は、ProgramInvoker 1 0 3 内のCommand 1 0 5 2 を起動又は実行する。Command 1 0 5 2 は、文書処理システムにおいて、XMLなどの文書进行处理し、対応するXML DOMツリーを編集するために用いられる命令である。CommandInvoker 1 0 5 1 は、Command 1 0 5 2 を実行するために必要なクラス及びファンクションを保持する。

40

#### 【 0 1 4 2 】

ServiceBroker 1 0 4 1 も、ProgramInvoker 1 0 3 内で実行される。UserApplication 1 0 6 は、ユーザインタフェース 1 0 7 及びCoreComponent 1 1 0 に接続される。CoreCompo

50

nent 1 1 0 は、全てのPaneの間で文書を共有する方法を提供する。CoreComponent 1 1 0 は、さらにフォントを提供し、Paneのためのツールキットの役割を果たす。

【 0 1 4 3 】

図 1 5 ( b ) は、Frame 1 0 7 1、MenuBar 1 0 7 2、及びStatusBar 1 0 7 3 の関係を示す。

【 0 1 4 4 】

H . アプリケーションコア

図 1 6 ( a ) は、全ての文書、及び文書の一部及び文書に属するデータを保持するアプリケーションコア 1 0 8 についての更なる説明を提供する。CoreComponent 1 1 0 は、文書 1 0 8 2 を管理するDocumentManager 1 0 8 1 にアタッチされる。DocumentManager 1 0 8 1 は、文書処理システムに関連づけられたメモリに格納される全ての文書 1 0 8 2 の所有者である。

10

【 0 1 4 5 】

画面上の文書の表示を容易にするために、DocumentManager 1 0 8 1 はRootPane 1 0 8 4 にも接続される。Clipboard 1 0 8 7、SnapShot 1 0 8 8、Drag&Drop 6 0 1、及びOverlay 6 0 2 の機能も、CoreComponent 1 1 0 にアタッチされる。

【 0 1 4 6 】

SnapShot 1 0 8 8 は、アプリケーションの状態を元に戻すために使用される。ユーザがSnapShot 1 0 8 8 を起動したとき、アプリケーションの現状が検知され、格納される。その後、アプリケーションの状態が別の状態が変わるとき、格納された状態の内容は保存される。SnapShot 1 0 8 8 は、図 1 6 ( b ) に図示される。動作において、アプリケーションがあるURLから他へ移動するときに、前に戻る動作及び先に進む動作をシームレスに実行可能とするために、SnapShot 1 0 8 8 は以前の状態を記憶する。

20

【 0 1 4 7 】

I . DocumentManager 内における文書の構成

図 1 7 ( a ) は、DocumentManager 1 0 8 1 の更なる説明と、DocumentManager において文書が構成され保持される様子を示す。図 1 1 ( b ) に示したように、DocumentManager 1 0 8 1 は、文書 1 0 8 2 を管理する。図 1 7 ( a ) に示される例において、複数の文書のうちの1つはRootDocument ( ルート文書 ) 7 0 1 であり、残りの文書はSubDocument ( サブ文書 ) 7 0 2 である。DocumentManager 1 0 8 1 は、RootDocument 7 0 1 に接続され、RootDocument 7 0 1 は、全てのSubDocument 7 0 2 に接続される。

30

【 0 1 4 8 】

図 1 2 及び図 1 7 ( a ) に示すように、DocumentManager 1 0 8 1 は、全ての文書 1 0 8 2 を管理するオブジェクトであるDocumentContainer 2 0 3 に結合される。DOMService 7 0 3 及びIOManager 7 0 4 を含むツールキット 2 0 1 ( 例えばXMLツールキット ) の一部を形成するツールも、DocumentManager 1 0 8 1 に供給される。再び図 1 7 ( a ) を参照して、DOMService 7 0 3 は、DocumentManager 1 0 8 1 により管理される文書に基づいたDOMツリーを生成する。各Document 7 0 5 は、それがRootDocument 7 0 1 であってもSubDocument 7 0 2 であっても、対応するDocumentContainer 2 0 3 によって管理される。

40

【 0 1 4 9 】

図 1 7 ( b ) は、文書 A - E が階層的に配置される様子を示す。文書 A はRootDocument である。文書 B - D は、文書 A のSubDocument である。文書 E は、文書 D のSubDocument である。図 1 7 ( b ) の左側は、これと同じ文書の階層が画面上に表示された例を示す。RootDocument である文書 A は、基本フレームとして表示される。文書 A のSubDocument である文書 B - D は、基本フレーム A 中のサブフレームとして表示される。文書 D のSubDocument である文書 E は、サブフレーム D のサブフレームとして画面に表示される。

【 0 1 5 0 】

再び図 1 7 ( a ) を参照して、UndoManager ( アンドゥマネージャ : アンドゥ管理部 ) 7 0 6 及びUndoWrapper ( アンドゥラッパー ) 7 0 7 は、それぞれのDocumentContainer 2

50

03に対して生成される。UndoManager 706及びUndoWrapper 707は、取消可能なコマンドを実行するために使用される。この特徴を使用することにより、編集操作を使用して文書に対して実行された変更を取り消すことができる。SubDocumentの変更は、RootDocumentとも密接な関係を有する。アンドゥ操作は、階層内の他の文書に影響する変更を考慮に入れて、例えば、図17(b)に示されるような連鎖状の階層における全ての文書の間で整合性が維持されることを保証する。

#### 【0151】

UndoWrapper 707は、DocumentContainer 203内のSubDocumentに関連するアンドゥオブジェクトをラップし、それらをRootDocumentに関連するアンドゥオブジェクトに結合させる。UndoWrapper 707は、UndoableEditAcceptor (アンドゥアブルエディットアクセプタ：アンドゥ可能編集受付部) 709に利用可能なアンドゥオブジェクトの収集を実行する。

10

#### 【0152】

UndoManager 706及びUndoWrapper 707は、UndoableEditAcceptor 709及びUndoableEditSource (アンドゥアブルエディットソース) 708に接続される。当業者には理解されるように、Document 705がUndoableEditSource 708であってもよく、取消可能な編集オブジェクトのソースであってもよい。

#### 【0153】

##### J. アンドゥコマンド及びアンドゥフレームワーク

図18(a)及び図18(b)は、アンドゥフレームワーク及びアンドゥコマンドについて更なる詳細を提供する。図18(a)に示されるように、UndoCommand 801、RedoCommand 802、及びUndoableEditCommand 803は、図11(b)に示したようにCommandInvoker 1051に積むことができるコマンドであり、順に実行される。UndoableEditCommand 803は、UndoableEditSource 708及びUndoableEditAcceptor 709に更にアタッチされる。「foo」EditCommand 804及び「bar」EditCommand 805は、UndoableEditCommandの例である。

20

#### 【0154】

##### 1. UndoableEditCommandの実行

図18(b)は、UndoableEditCommandの実行を示す。まず、ユーザが編集コマンドを使用してDocument 705を編集すると仮定する。第1ステップS1では、UndoableEditAcceptor 709が、Document 705のDOMツリーであるUndoableEditSource 708にアタッチされる。第2ステップS2では、ユーザにより発行されたコマンドに基づいて、Document 705がDOMのAPIを用いて編集される。第3ステップS3では、ミュートションイベントのリスナーが、変更がなされたことを通知される。すなわち、このステップでは、DOMツリーの全ての変更を監視するリスナーが編集操作を検知する。第4ステップS4では、UndoableEditがUndoManager 706のオブジェクトとして格納される。第5ステップS5では、UndoableEditAcceptor 709がUndoableEditSource 708からデタッチされる。UndoableEditSource 708は、Document 705自身であってもよい。

30

#### 【0155】

##### K. システムへの文書のロードに関する手順

上記のサブセクションでは、システムの様々なコンポーネント及びサブコンポーネントについて説明した。以下、これらのコンポーネントの使用に関する方法論について説明する。図19(a)は、文書処理システムに文書がロードされる様子の概要を示す。それぞれのステップは、図24-28において、特定の例に関連して詳述される。

40

#### 【0156】

簡単には、文書処理システムは、文書に含まれるデータからなるバイナリデータストリームからDOMを生成する。ApexNode (エイペックスノード：頂点ノード)が、注目対象でありZoneに属する文書の一部のために生成される。つづいて、対応するPaneが同定される。同定されたPaneは、ApexNode及び物理的な画面表面からZone及びCanvasを生成する。Zoneは、次に、それぞれのノードにFacetを生成し、それらに必要とされる情報を提供す

50

る。Canvasは、DOMツリーから、ノードをレンダリングするためのデータ構造を生成する。

#### 【0157】

より詳細には、文書はストレージ901からロードされる。文書のDOMツリー902が生成される。文書を保持するための、対応するDocumentContainer903が生成される。DocumentContainer903は、DocumentManager904にアタッチされる。DOMツリーは、ルートノードと、ときには複数のセカンダリノードを含む。

#### 【0158】

一般に、このような文書は、テキスト及びグラフィックスの双方を含む。したがって、DOMツリーは、例えば、XHTMLサブツリーだけでなくSVGサブツリーを有してもよい。XHTMLサブツリーは、XHTMLのApexNode905を有する。同様に、SVGサブツリーは、SVGのApexNode906を有する。

#### 【0159】

ステップ1では、ApexNode906が、画面の論理的なレイアウトであるPane907にアタッチされる。ステップ2では、Pane907は、PaneOwner(ペインオーナー:ペインの所有者)908であるCoreComponentに、ApexNode906のためのZoneFactoryを要求する。ステップ3では、PaneOwner908は、ZoneFactoryと、ApexNode906のためのCanvasFactoryであるEditletとを返す。

#### 【0160】

ステップ4では、Pane907がZone909を生成する。Zone909はPane907にアタッチされる。ステップ5では、Zone909がそれぞれのノードに対してFacetを生成し、対応するノードにアタッチする。ステップ6では、Pane907がCanvas910を生成する。Canvas910はPane907にアタッチされる。Canvas910には様々なCommandが含まれる。ステップ7では、Canvas910が文書を画面にレンダリングするためのデータ構造を構築する。XHTMLの場合、これはボックスツリー構造を含む。

#### 【0161】

##### 1. ZoneのMVC

図19(b)は、MVCパラダイムを用いてZoneの構成の概要を示す。この場合、Zone及びFacetは文書に関連した入力であるから、モデル(M)はZone及びFacetを含む。Canvasと、文書を画面にレンダリングするためのデータ構造体は、ユーザが画面上に見る出力であるから、ビュー(V)はCanvas及びデータ構造体に対応する。Commandは、文書とその様々な関係に対して制御操作を実行するので、コントロール(C)はCanvasに含まれるCommandを含む。

#### 【0162】

##### L. 文書の表現

図20を用いて、文書及びその様々な表現の例について以下に説明する。この例で 사용되는文書は、テキストと画像の双方を含む。テキストは、XHTMLを用いて表され、画像は、SVGを用いて表される。図20は、文書のコンポーネント及び対応するオブジェクトの関係のMVC表現を詳細に示す。この例において、Document1001は、Document1001を保持するDocumentContainer1002にアタッチされる。文書はDOMツリー1003により表現される。DOMツリーは、ApexNode1004を含む。

#### 【0163】

ApexNodeは、黒丸で表される。頂点でないノードは、白丸で表される。ノードを編集するために用いられるFacetは、三角形で表され、対応するノードにアタッチされる。文書がテキストと画像を有するので、この文書のDOMツリーは、XHTML部分とSVG部分を含む。ApexNode1004は、XHTMLサブツリーの最上のノードである。これは、文書のXHTML部分の物理的な表現のための最上PaneであるXHTMLPane1005にアタッチされる。ApexNode1004は、文書のDOMツリーの一部であるXHTMLZone1006にもアタッチされる。

#### 【0164】

10

20

30

40

50

Node 1 0 0 4 に対応するFacetも、XHTMLZone 1 0 0 6 にアタッチされる。XHTMLZone 1 0 0 6 は、XHTMLPane 1 0 0 5 にアタッチされる。XHTMLEditletは、文書の論理的な表現であるXHTMLCanvas 1 0 0 7 を生成する。XHTMLCanvas 1 0 0 7 は、XHTMLPane 1 0 0 5 にアタッチされる。XHTMLCanvas 1 0 0 7 は、Document 1 0 0 1 のX H T M LコンポーネントのためのBoxTree 1 0 0 9 を生成する。文書のX H T M L部分を保持し描画するために必要な様々なCommand 1 0 0 8 も、XHTMLCanvas 1 0 0 7 に追加される。

#### 【 0 1 6 5 】

同様に、文書のS V GサブツリーのApexNode 1 0 1 0 は、文書のS V Gコンポーネントを表現するDocument 1 0 0 1 のD O Mツリーの一部であるSVGZone 1 0 1 1 にアタッチされる。ApexNode 1 0 1 0 は、文書のS V G部分の物理的な表現の最上のPaneであるSVGPane 1 0 1 3 にアタッチされる。文書のS V G部分の論理的な表現を表すSVGCanvas 1 0 1 2 は、SVGEditletにより生成され、SVGPane 1 0 1 3 にアタッチされる。画面上に文書のS V G部分をレンダリングするためのデータ構造及びコマンドは、SVGCanvasにアタッチされる。例えば、このデータ構造は、図示されるように、円、線、長方形などを含んでもよい。

10

#### 【 0 1 6 6 】

図 2 0 に関連して説明された文書例の表現の一部について、図 2 1 ( a ) に関連して、前述したM V Cパラダイムを用いて更に説明する。図 2 1 ( a ) は、文書 1 0 0 1 のX H T M LコンポーネントにおけるM Vの関係を簡略化して示す。モデルは、Document 1 0 0 1 のX H T M LコンポーネントのためのXHTMLZone 1 1 0 1 である。XHTMLZoneのツリーには、いくつかのNode及びそれらに対応するFacetが含まれる。対応するXHTMLZone及びPaneは、M V Cパラダイムのモデル(M)部分の一部である。M V Cパラダイムのビュー(V)部分は、Document 1 0 0 1 のX H T M Lコンポーネントの、対応するXHTMLCanvas 1 1 0 2 及びBoxTreeである。文書のX H T M L部分は、Canvasと、それに含まれるCommandを使用して画面に描写される。キーボードやマウス入力などのイベントは、図示されるように、逆方向へ進む。

20

#### 【 0 1 6 7 】

SourcePaneは、更なる機能、すなわち、D O Mの保有者としての役割を有する。図 2 1 ( b ) は、図 2 1 ( a ) に示したDocument 1 0 0 1 のコンポーネントに対するボキャブラリコネクションを提供する。D O Mホルダーとして機能するSourcePane 1 1 0 3 は、文書のソースD O Mツリーを含む。ConnectorTreeは、ConnectorFactoryにより生成され、デスティネーションD O Mの保有者としても機能するDestinationPane 1 1 0 5 を生成する。DestinationPane 1 1 0 5 は、XHTMLDestinationCanvas 1 1 0 6 としてボックスツリーの形式でレイアウトされる。

30

#### 【 0 1 6 8 】

M . プラグインサブシステム、ボキャブラリコネクション、及びコネクタの関係

図 2 2 ( a ) - ( c ) は、それぞれ、プラグインサブシステム、ボキャブラリコネクション、及びConnectorに関連する更なる詳細を示す。プラグインサブシステムは、文書処理システムに機能を追加又は交換するために用いられる。プラグインサブシステムは、ServiceBroker 1 0 4 1 を含む。ServiceBroker 1 0 4 1 にアタッチされるZoneFactoryService 1 2 0 1 は、文書の一部に対するZoneを生成する。EditletService 1 2 0 2 も、ServiceBroker 1 0 4 1 にアタッチされる。EditletService 1 2 0 2 は、Zone中のNodeに対応するCanvasを生成する。

40

#### 【 0 1 6 9 】

ZoneFactoryの例は、XHTMLZone及びSVGZoneをそれぞれ生成するXHTMLZoneFactory 1 2 1 1 及びSVGZoneFactory 1 2 1 2 である。文書例に関連して前述したように、文書のテキストコンポーネントは、XHTMLZoneを生成することにより表現されてもよいし、画像はSVGZoneを用いて表現されてもよい。EditletServiceの例は、XHTMLEditlet 1 2 2 1 及びSVGE ditlet 1 2 2 2 を含む。

#### 【 0 1 7 0 】

50

図 2 2 ( b ) は、ボキャブラリコネクションに関連する更なる詳細を示す。ボキャブラリコネクションは、前述したように、文書処理システムの重要な特徴であり、2つの異なる方法で文書の整合のとれた表現及び表示を可能とする。ConnectorFactory 3 0 3 を保持するVManager 3 0 2 は、ボキャブラリコネクションサブシステムの一部である。ConnectorFactory 3 0 3 は、文書のConnector 3 0 4 を生成する。前述したように、Connectorは、ソースDOM中のノードを監視し、2つの表現の間の整合性を維持するために、デスティネーションDOM中のノードを修正する。

#### 【 0 1 7 1 】

Template 3 1 7 は、いくつかのノードの変換ルールを表す。ボキャブラリコネクション記述子 ( V C D ) ファイルは、特定のパス又はルールを満たす要素又は要素の集合を他の要素に変換するいくつかのルールを表すTemplateのリストである。Template 3 1 7 及びCommandTemplate 3 1 8 は、全てVManager 3 0 2 にアタッチされる。VManagerは、V C D ファイル中の全てのセクションを管理するオブジェクトである。1つのV C D ファイルに対して、1つのVManagerオブジェクトが生成される。

#### 【 0 1 7 2 】

図 2 2 ( c ) は、Connectorに関連する更なる詳細を提供する。ConnectorFactory 3 0 3 は、ソース文書からConnectorを生成する。ConnectorFactory 3 0 3 は、Vocabulary、Template、及びElementTemplateにアタッチされ、それぞれ、VocabularyConnector、TemplateConnector、ElementConnectorを生成する。

#### 【 0 1 7 3 】

VManager 3 0 2 は、ConnectorFactory 3 0 3 を保持する。Vocabularyを生成するために、対応するV C D ファイルが読み込まれる。こうして、ConnectorFactory 3 0 3 が生成される。このConnectorFactory 3 0 3 は、Zoneを生成するZoneFactory及びCanvasを生成するEditletに関連する。

#### 【 0 1 7 4 】

つづいて、ターゲットボキャブラリのEditletServiceが、VCanvasを生成する。VCanvasも、ソースDOMツリー又はZoneにおけるApexNodeのConnectorを生成する。必要に応じて、子のConnectorが再帰的に生成される。ConnectorTreeは、V C D ファイル中のテンプレートの集合により生成される。

#### 【 0 1 7 5 】

テンプレートは、マークアップ言語の要素を他の要素に変換するためのルールの集合である。例えば、各テンプレートは、ソースDOMツリー又はZoneにマッチされる。適切にマッチした場合には、頂点Connectorが生成される。例えば、テンプレート「A/\* /D」は、間にどんなノードがあるかに関係なく、ノードAで始まりノードDで終わる全ての枝に合致する。同様に、「//B」は、ルートからの全ての「B」ノードに一致する。

#### 【 0 1 7 6 】

N . ConnectorTreeに關係するV C D ファイルの例

特定の文書と關係する処理を説明する例を続ける。ドキュメントタイトルのある「MySampleXML」というタイトルの文書が文書処理システムにロードされる。図 2 3 は、「MySampleXML」ファイルのための、VManager及びConnectorFactoryTreeを用いたV C D スクリプトの例を示す。スクリプトファイル中のボキャブラリセクション、テンプレートセクションと、VManagerにおける対応するコンポーネントが示される。タグ「vcd:vocabulary」において、属性「match」は「sample:root」、「label」は「MySampleXML」、「call-template」は「sample template」となっている。

#### 【 0 1 7 7 】

この例では、Vocabularyは、「MySampleXML」のVManagerにおいて「sample:root」として頂点要素を含む。対応するUIラベルは、「MySampleXML」である。テンプレートセクションにおいて、タグは「vcd:template」であり、名前は「sample:template」である。

#### 【 0 1 7 8 】

10

20

30

40

50

## 0. ファイルがシステムにロードされる方法の詳細な例

図 2 4 - 2 8 は、文書「MySampleXML」のロードについての詳細な記述を示す。図 2 4 ( a ) に示されるステップ 1 では、文書がストレージ 1 4 0 5 からロードされる。DOMService は、DOM ツリー及び DocumentManager 1 4 0 6 と対応する DocumentContainer 1 4 0 1 を生成する。DocumentContainer 1 4 0 1 は、DocumentManager 1 4 0 6 にアタッチされる。文書は、XHTML 及び MySampleXML のサブツリーを含む。XHTML の ApexNode 1 4 0 3 は、タグ「xhtml:html」が付された XHTML の最上のノードである。「MySampleXML」の ApexNode 1 4 0 4 は、タグ「sample:root」が付された「MySampleXML」の最上ノードである。

【 0 1 7 9 】

図 2 4 ( b ) に示されるステップ 2 では、RootPane が文書の XHTMLZone、Facet、及び Canvas を生成する。Pane 1 4 0 7、XHTMLZone 1 4 0 8、XHTMLCanvas 1 4 0 9、及び BoxTree 1 4 1 0 が、ApexNode 1 4 0 3 に対応して生成される。

【 0 1 8 0 】

図 2 4 ( c ) に示されるステップ 3 では、XHTMLZone が知らないタグ「sample:root」を発見し、XHTMLCanvas の領域から SubPane を生成する。

【 0 1 8 1 】

図 2 5 に示されるステップ 4 では、SubPane が「sample:root」を扱うことができ、適切な Zone を生成可能な ZoneFactory を得る。この ZoneFactory は、ZoneFactory を実行可能な Vocabulary 内にある。それは、「MySampleXML」の VocabularySection の内容を含む。

【 0 1 8 2 】

図 2 6 に示されるステップ 5 では、「MySampleXML」に対応する Vocabulary が DefaultZone 1 6 0 1 を生成する。対応する Editlet が生成され、対応する Canvas を生成するために SubPane 1 5 0 1 が提供される。Editlet は、VCCanvas を生成する。そして、それは TemplateSection を呼ぶ。ConnectorFactoryTree も含まれている。ConnectorFactoryTree は、ConnectorTree となる全ての Connector を生成する。

【 0 1 8 3 】

図 2 7 に示されるステップ 6 では、各 Connector がデスティネーション DOM オブジェクトを生成する。コネクタのうちいくつかは xpath 情報を含んでいる。xpath 情報は、変更 / 修正を監視する必要のあるソース DOM ツリーの部分集合を決定するために使用される 1 以上の xpath 表現を含む。

【 0 1 8 4 】

図 2 8 に示されるステップ 7 では、ボキャブラリは、ソース DOM のペインからデスティネーション DOM ツリーの DestinationPane を作成する。これは、SourcePane に基づいてなされる。デスティネーションツリーの ApexNode は、DestinationPane 及び対応する Zone にアタッチされる。DestinationPane は、DestinationCanvas を生成し、文書をデスティネーションのフォーマットでレンダリングするためのデータ構造及びコマンドを構築する、自身の Editlet を提供される。

【 0 1 8 5 】

図 2 9 ( a ) は、対応するソースノードを持たず、デスティネーションツリーにのみ存在するノード上でイベントが発生したときのフローを示す。マウスイベント、キーボードイベントなど、Canvas が取得したイベントは、デスティネーションツリーを通過して、ElementTemplateConnector に伝達される。ElementTemplateConnector は対応するソースノードを持たないので、伝達されたイベントはソースノードに対する編集操作ではない。ElementTemplateConnector は、伝達されたイベントが CommandTemplate に記述されたコマンドに合致すれば、それに対応する Action を実行する。合致するコマンドがなければ、ElementTemplateConnector は、伝達されたイベントを無視する。

【 0 1 8 6 】

図 2 9 ( b ) は、TextOfConnector によりソースノードに対応づけられているデスティネーションツリーのノード上でイベントが発生したときのフローを示す。TextOfConnector

10

20

30

40

50

rは、ソースDOMツリーのXPathで指定されたノードからテキストノードを取得して、デスティネーションDOMツリーのノードにマッピングする。マウスイベント、キーボードイベントなど、Canvasが取得したイベントは、デスティネーションツリーを通過して、TextOfConnectorに伝達される。TextOfConnectorは、伝達されたイベントを、対応するソースノードの編集コマンドにマッピングし、Queue 1053に積む。編集コマンドは、Facetを介して実行されるDOMのAPIコールの集合である。キューに積まれたコマンドが実行されると、ソースノードが編集される。ソースノードが編集されると、ミュレーションイベントが発行され、リスナーとして登録されたTextOfConnectorにソースノードの変更が通知される。TextOfConnectorは、ソースノードの変更を、対応するデスティネーションノードに反映させるように、デスティネーションツリーを再構築する。このとき、TextOfConnectorを含むテンプレートに、「for each」や「for loop」などの制御文が含まれている場合、ConnectorFactoryがこの制御文を再評価し、TextOfConnectorを再構築した後、デスティネーションツリーが再構築される。

10

【0187】

(実施の形態)

実施の形態では、携帯端末に、XML文書と、そのXML文書を閲覧/編集するための定義ファイルを配信し、様々なビジネスモデルを実現するための技術を提案する。

【0188】

マークアップ言語で記述された文書を配信するモデルとして、ウェブサーバを利用したHTML文書の配信が広く普及している。しかし、HTMLは、主として表示形式を規定するマークアップ言語であるから、文書の内容に対してタグで意味が付されているわけではない。したがって、文書の再利用性に劣るという問題がある。

20

【0189】

それに対して、XML文書と定義ファイルを配信するモデルによれば、任意のタグセットにより文書をマークアップし、意味を付すことが可能であるから、配信されたXML文書を様々な用途に再利用することができる。従来は、XML文書の処理系がほとんど存在せず、メジャーではないポキャブラリで記述されたXML文書はテキスト文書として扱うしかないという問題があったが、前提技術として上記で説明した技術によれば、VCユニット80を携帯端末に搭載し、XML文書を記述したタグセットを処理する定義ファイルを用意すれば、任意のタグセットで記述された文書を携帯端末において表示・編集することができる。したがって、携帯電話端末などのリソースが限られた装置であっても、任意のXML文書を処理可能な汎用的な処理系を実現できる。

30

【0190】

また、適切なVCDを用意すれば、任意のタグセットで記述されたXML文書を表示・編集することができるので、配信するXML文書には表示情報を含ませる必要がない。したがって、配信するXML文書のデータ量を低減することができる。データ量が小さいので、小型で軽量の半導体チップなどに格納することができ、小さな商品などにも付することができる。また、バーコード化して商品に付することもできる。

【0191】

XML文書を表示する技術として、XSLTやCSSなどによりXML文書の表示情報を規定する技術があるが、XML文書を表示することはできても、編集することはできない。VCDを利用したVC技術によれば、任意のタグセットで記述されたXML文書を編集可能である。また、VCDにUIやロジックなどを記述することができる。したがって、単なる情報の配信だけでなく、ユーザに対してアクションを起こしたり、ユーザからのアクションを取得するような対話型の通信が可能となる。

40

【0192】

JavaScript(登録商標)によってもロジックを記述することができるが、JavaScript(登録商標)は、ビューを作成する能力の面でVCDに劣る。VCDは、XSLTのビュー作成能力を兼ね備えており、任意のタグセットで記述されたXML文書のビューを容易に作成することができる。また、プログラミングの観点から考察する

50



と、VCDは、Java（登録商標）をベースとしており、環境に非依存で、かつ、開発の容易性が長じている。また、サイズが小さく、携帯端末を利用したビジネスモデルにとって好都合である。

#### 【0193】

図30は、実施の形態に係る情報配信システムの構成を示す。情報配信システム100は、携帯端末70と、携帯端末70へ情報を配信する情報配信装置25を含む。情報配信装置25は、携帯端末70へ配信すべき文書を保持する文書保持部26と、文書保持部26から文書を読み出して携帯端末70へ送信する送信部27とを含む。携帯端末70は、情報配信装置25との間の通信を制御する通信部75と、前提技術で説明した文書処理装置20と、文書処理装置20により処理された文書を表示する表示装置21と、ユーザからの入力を受け付けるボタンなどを備えた入力装置23と、携帯端末70のユーザの個人情報保持する個人情報保持部71と、定義ファイルを保持する定義ファイル保持部77と、文書に適用する定義ファイルを切り替える切替部78とを含む。情報配信装置25は、ネットワーク29を介して携帯端末70に文書を配信する。文書は、Bluetoothなどの無線通信技術を用いて配信されてもよいし、携帯電話網やインターネットなどを利用して配信されてもよい。この場合、通信部75が、情報配信装置25から送信された文書を受信する受信部の機能を有する。また、文書の内容をバーコードなどで表現し、携帯端末70がバーコードを読み取ることで文書を取得できるようにしてもよい。この場合、情報配信装置25は、送信部27に替えてバーコード表示部を含み、携帯端末70はバーコードリーダーを含む。

10

20

#### 【0194】

携帯端末70は、一般的なパーソナルコンピュータに比べてリソースが少ないので、余分な構成を省略し、最小限の機能を効果的に設けるのが望ましい。したがって、本実施の形態では、文書を処理する処理系として、HTMLユニット50とVCユニット80のみを設けることにする。XHTML以外のタグセット（ボキャブラリ）で記述された文書は、XHTMLへの変換テンプレートを記述した定義ファイルを用意し、VCユニット80によりXHTMLに変換して、HTMLユニット50により表示する。これにより、最小限の構成で、任意のタグセットの汎用的な処理系を実現することができる。なお、実施の形態では、処理系としてHTMLユニット50のみを設ける例を説明するが、もちろん他のタグセットの処理系を携帯端末70に搭載してもよい。

30

#### 【0195】

情報配信装置25が、配信する文書に対応する定義ファイルを携帯端末70に配信してもよいし、携帯端末70は他の経路から定義ファイルを取得してもよい。取得された定義ファイルは、定義ファイル保持部77に格納される。例えば、共通するタグセットを処理するためのデフォルト定義ファイルを携帯端末70にプリインストールしておき、文書内の共通タグセットで記述された最低限の情報はデフォルト定義ファイルを適用して閲覧することができるようにしてもよい。また、文書を処理するための定義ファイルが複数用意されてもよい。例えば、同一の文書を異なるレイアウトで表示する複数の定義ファイルを用意して、それらを切替可能とすることにより、利用シーンに合わせてユーザが好みの画面を選択可能としてもよい。この場合、切替部78は、ユーザから定義ファイルを切り替える指示を受け付けて、XML文書に適用する定義ファイルを切り替える。また、異なる要素の表示テンプレートを含む複数の定義ファイルを用意してもよい。例えば、通常定義ファイルでは表示できない要素を文書に含ませおき、その要素のテンプレートを記述した特別な定義ファイルを入手したユーザのみがその要素を閲覧できるようにしてもよい。これにより、会員限定の秘密のバーゲン情報を配信したり、店舗に来店したユーザのみが取得可能なサービスチケットを提供したりするなど、様々な形態のサービスを提供することができる。例えば、販売促進のための演出に利用することができる。

40

#### 【0196】

定義ファイルには、コマンドやロジックを記述することができるので、これを利用して、独自のユーザインタフェースを提供することができる。また、VC機能を用いてユーザ

50

に文書を編集させ、編集した文書を通信部 7 5 を介して情報配信装置 2 5 へ返送することもできる。この場合、通信部 7 5 が、編集した文書を情報配信装置 2 5 へ返送する送信部の機能を有する。これにより、商品や役務の購入、アンケートに対する回答など、双方向通信を利用した各種サービスを提供することができる。文書を編集するときに、個人情報保持部 7 1 に格納されたユーザの個人情報を利用してよい。たとえば、個人情報保持部 7 1 に、ユーザの氏名、住所、電話番号、携帯端末 7 0 の ID、公開鍵、証明書、電子署名などを格納しておき、文書を編集するための定義ファイルが適宜利用できるようにしてもよい。これにより、ユーザの入力の手間を低減することができるとともに、証明書などを利用した信頼性の高い通信を行うことができる。

#### 【 0 1 9 7 】

このように、文書処理装置 2 0 を、提供主体が自由に用意した、独自のボキャブラリで記述された文書を汎用的に処理するプラットフォームとして利用することにより、様々なビジネスモデルを柔軟かつ効果的に実現することができる。以下、適用例をいくつか述べる。

#### 【 0 1 9 8 】

図 3 1 ( a ) ( b ) は、情報配信システム 1 0 0 の第 1 の適用例を示す。図 3 1 ( a ) では、商品の販売装置 7 6 に情報配信装置 2 5 が設けられており、携帯端末 7 0 に、販売装置 7 6 で購入可能な商品の情報や、販売装置 7 6 で商品を購入するための UI を提供するための文書を配信する。図 3 1 ( b ) は、携帯端末 7 0 に配信された文書を表示装置 2 1 に表示した画面の例を示す。販売装置 7 6 に設けられた情報配信装置 2 5 から配信される文書には、例えば、販売装置 7 6 で購入できる商品のリスト、それらの商品に関する情報などが記述されている。また、この文書に対応づけられた定義ファイルには、文書に記述された商品リストや関連情報などを表示するためのテンプレートや、商品の購入要求を販売装置 7 6 へ送信するためのコマンドなどが記述されている。定義ファイルは、販売装置 7 6 から文書とともに配信されてもよいし、他の任意の経路から携帯端末 7 0 に配信されてもよい。例えば、独自のボキャブラリで販売装置 7 6 の商品情報などを記述した文書を配信する場合は、そのボキャブラリの要素のテンプレートを記述した専用の定義ファイルを用意して、文書とともに送信してもよい。また、例えば複数のベンダーが共通のボキャブラリを用いて文書を記述する場合は、携帯端末 7 0 に初めて文書を配信するときのみ定義ファイルを配信し、以降は、携帯端末 7 0 に保持されている定義ファイルを適用してもよい。

#### 【 0 1 9 9 】

携帯端末 7 0 が、配信された文書と定義ファイルにより提示される GUI を利用して、販売装置 7 6 に購入要求を送信すると、販売装置 7 6 は、図示しない受付部により購入要求を受け付け、商品を販売する。このとき、個人情報保持部 7 1 に格納された決済情報、たとえばクレジットカードや電子マネーなどに関する情報が販売装置 7 6 に送られてもよい。

#### 【 0 2 0 0 】

ユーザは、携帯端末 7 0 を用いて、販売装置 7 6 で販売されている商品の詳細情報を閲覧することができるとともに、携帯端末 7 0 から商品の購入を要求することができる。これにより、販売装置 7 6 自身では伝えきれない情報をユーザに提供することができるとともに、販売装置 7 6 自身の UI よりも多機能な UI をユーザに提供することができる。ユーザの利便性を飛躍的に向上させることができる。携帯端末 7 0 を用いた決済サービスなどと連動させることにより、商品購入の決済も自動的に処理することが可能となり、さらにユーザの利便性を向上させることができる。

#### 【 0 2 0 1 】

図 3 2 ( a ) ( b ) は、情報配信システム 1 0 0 の第 2 の適用例を示す。図 3 2 ( a ) では、商品 7 9 に添付された無線 IC タグ 8 9 を情報配信装置 2 5 として機能させる。無線 IC タグ 8 9 には、商品の流通経路などの情報が記述された文書を保持する記憶装置と、その文書を無線通信により携帯端末 7 0 に送信する送信装置とが設けられる。図 3 2 (

10

20

30

40

50

b) は、携帯端末 70 に送信された文書を表示装置 21 に表示した画面の例を示す。無線 IC タグ 89 から配信される文書には、商品の生産主体、生産場所、生産日、流通経路などの情報が記述されている。また、この文書に対応づけられた定義ファイルには、文書に記述された流通経路などの情報を表示するためのテンプレートや、それらの情報を編集するためのコマンドなどが記述されている。

#### 【0202】

消費者は、携帯端末 70 を用いて、商品の流通経路などの情報を閲覧することができる。これにより、消費者に安心感を与えることができるので、商品の販売促進にもつながる。流通の過程において、例えば、中間流通業者の担当者が携帯端末 70 により無線 IC タグ 89 から文書を取得し、入荷日や出荷日などの情報を追加して無線 IC タグ 89 に保存してもよい。これにより、流通経路の管理を容易かつ適切に行うことができる。業者が商品を検品するときに、検印を付加するための定義ファイルを備えた携帯端末 70 を用いて、無線 IC タグ 89 から取得した文書に検印を付加してもよい。検印が押された文書は通信部 75 を介して無線 IC タグ 89 へ返送され文書保持部 26 に格納される。個人情報保持部 71 に検印者の証明書を格納しておき、検印した文書に電子署名を付してもよい。これにより、検品を迅速に進めることができるとともに、検品の信頼性を向上させることができる。

#### 【0203】

図 33 (a) (b) は、情報配信システム 100 の第 3 の適用例を示す。図 33 (a) では、各所に情報発信スポットとして情報配信装置 25 が設けられており、ユーザが移動中に携帯端末 70 が情報配信装置 25 の無線通信圏内に入ると、情報配信装置 25 から携帯端末 70 に、周辺の店舗や観光地などの情報を提供するための文書が配信される。図 33 (b) は、携帯端末 70 に送信された文書を表示装置 21 に表示した画面の例を示す。情報配信装置 25 から配信される文書には、情報発信スポットの周辺、すなわち、ユーザの現在位置の周辺の情報などが記述されている。ユーザの現在位置の周辺の店舗などの情報を配信することにより、より効果的な宣伝を行うことができる。また、文書とともに配信される標準的な定義ファイルには、秘密情報のテンプレートを記述せず、店舗でしか取得できない定義ファイルには、秘密情報のテンプレートを記述しておき、その定義ファイルを適用して初めて秘密のバーゲン情報やクーポンチケットなどを表示可能としてもよい。

#### 【0204】

図 34 は、情報配信システムの別の構成例を示す。図 34 に示した情報配信システム 100 においては、携帯端末 70 が、フィルタリング部 72 及び文書保持部 73 を更に備えている。情報配信装置 25 は、電子メールにより、メールマガジンの形態で XML 文書を携帯端末 70 へ配信する。フィルタリング部 72 は、ユーザが蓄積を希望する情報を抽出して文書保持部 73 に格納する。フィルタリング部 72 は、XML 文書のうち、特定のタグセットにより記述された部分のみを抽出して文書保持部 73 に蓄積してもよい。また、特定の要素のみを抽出して文書保持部 73 に蓄積してもよい。ユーザは、任意のタイミングで、文書保持部 73 に格納された文書を読み出し、文書処理装置 20 により表示させ、編集することができる。

#### 【0205】

図 35 は、情報配信システムの更に別の構成例を示す。図 35 に示した情報配信システム 100 においては、携帯端末 70 が、検索部 74 を更に備えている。検索部 74 は、携帯端末 70 の周囲に存在し、携帯端末 70 と通信が可能な情報配信装置 25 を検索する。例えば、検索部 74 は、図 33 (a) に示したような状況下において、無線通信圏内に存在する情報配信装置 25 を探索し、発見した情報配信装置 25 に XML 文書の送信を要求する。すなわち、図 33 (a) においては、情報配信装置 25 が無線通信圏内に存在する携帯端末 70 を探索して XML 文書を配信したが、本図の構成例では、逆に、携帯端末 70 が無線通信圏内に存在する情報配信装置 25 を探索して XML 文書の配信を要求する。検索部 74 は、情報配信装置 25 が配信している XML 文書の中から、携帯端末 70 が有

10

20

30

40

50

している定義ファイルにより処理可能なタグセットで記述された文書を検索してもよい。

【0206】

検索部74は、定義ファイルとVCユニット80により実現されてもよい。例えば、ユーザが、バーゲン情報を記述するタグセットを処理する定義ファイルを起動すると、検索部74が、通信可能な情報配信装置25が配信するXML文書の中から、バーゲン情報用タグセットで記述された文書を検索し、発見した文書を取得するようにしてもよい。

【0207】

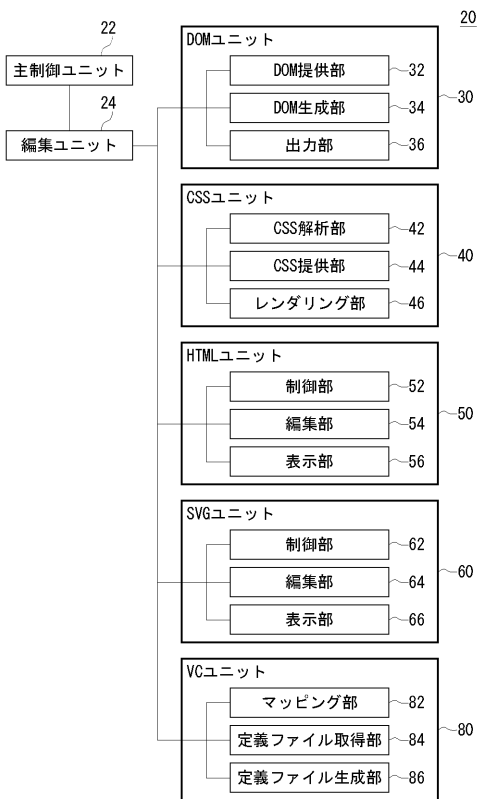
以上、本発明を実施の形態をもとに説明した。この実施の形態は例示であり、それらの各構成要素や各処理プロセスの組合せにいろいろな変形例が可能なおこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。

【産業上の利用可能性】

【0208】

本発明は、XMLにより記述された文書を配信する情報配信システムに利用することができる。

【図1】

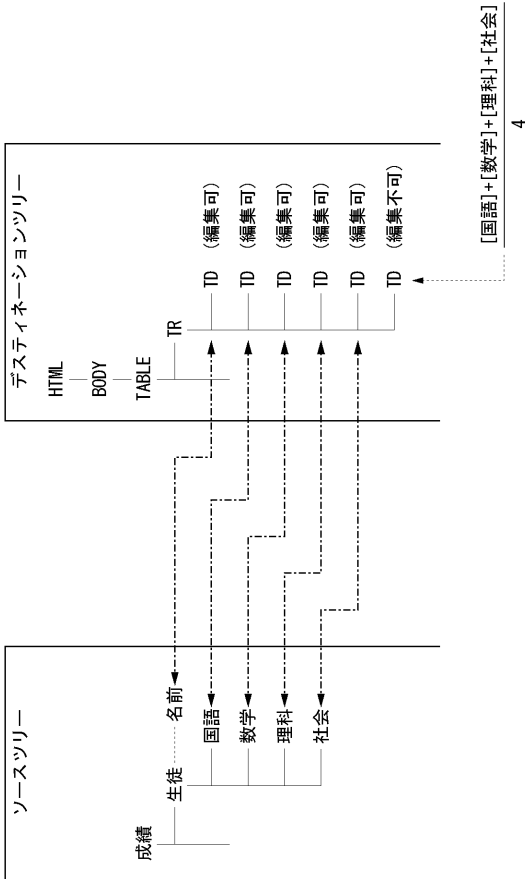


【図2】

```

    <?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 ) 】

```

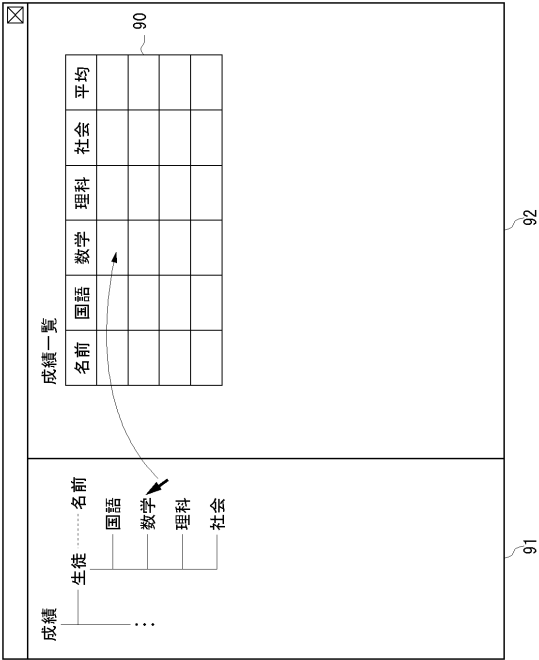
<tr {
  border:none;
}
.data {
  padding:0.2em 0.5em;
}
</style>
</head>
<body>
<h1>成績一覧</h1>
<table>
<thead>
<tr>
<th><div class="data">名前</div></th>
<th></th>
<th><div class="data">国語</div></th>
<th><div class="data">数学</div></th>
<th><div class="data">理科</div></th>
<th><div class="data">社会</div></th>
<th></th>
<th><div class="data">平均</div></th></tr>
<tbody>
<tr>
<td><vc:apply-templates select="src:生徒" />
</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
</body>
</html>
</vc:vc-template>
<vc: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:vc-template>
</vc:vcd>

```

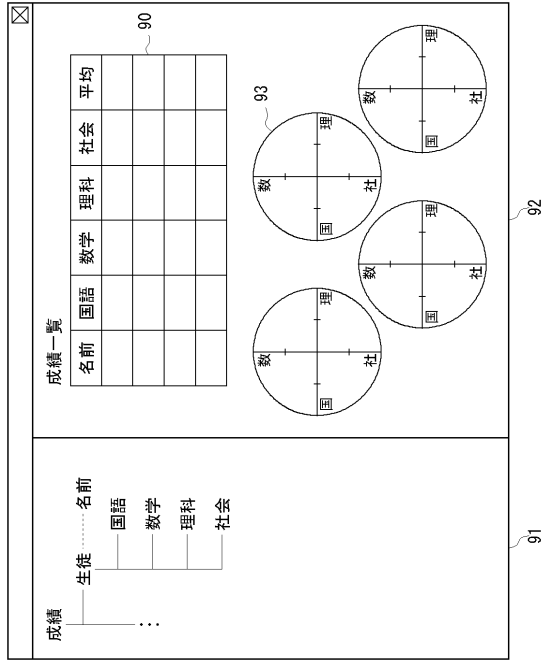
【 図 5 】

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

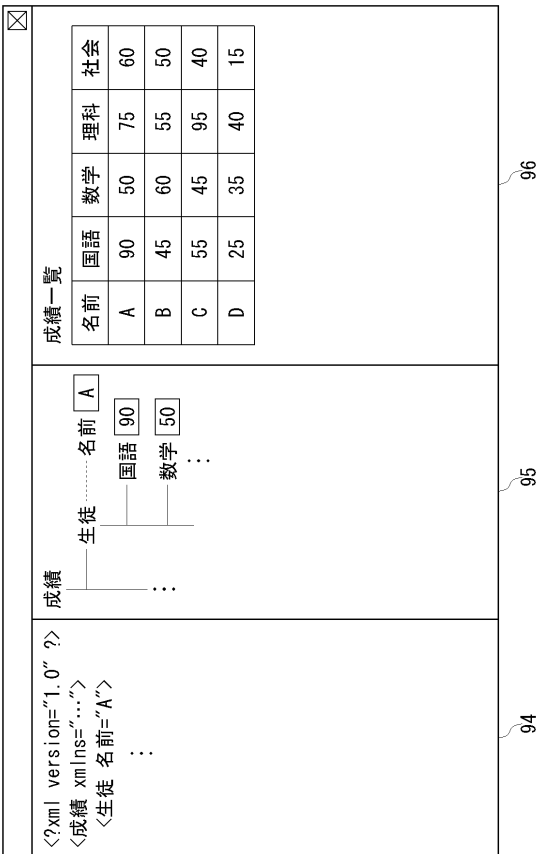
【 図 6 】



【 図 7 】



【 図 8 】



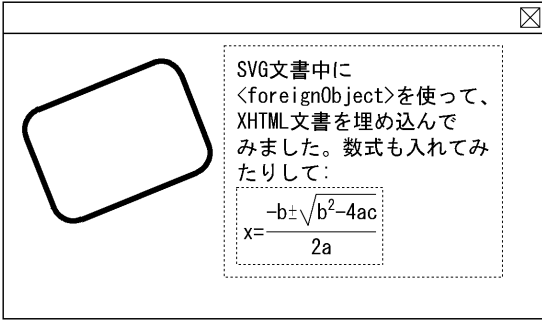
【 図 9 】

```

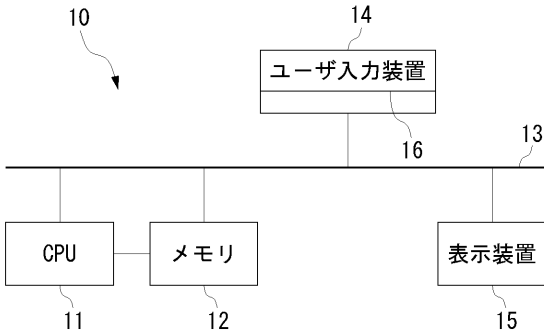
<?xml version="1.0" ?>
<svg xmlns="http://www.w3.org/2000/svg"
width="400" height="200"
viewBox="0 0 400 200"
>
  <rect x="-15" y="65" width="150" height="100" rx="20"
transform="rotate(-20)"
style="fill:none; stroke:purple; stroke-width:10"
  />
  <foreignObject x="190" y="10" width="200" height="200">
    <html xmlns="http://www.w3.org/1999/xhtml">
      <head><title /></head>
      <body bgcolor="#FFFFCC" 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。
          数式も入れてみたりして：
          <div>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
              <mi>x</mi>
              <mo>=</mo>
              <mfrac>
                <mrow>
                  <mo>-</mo>
                  <mi>b</mi>
                  <mn>2</mn>
                </mrow>
                <msup>
                  <mi>b</mi>
                  <mn>2</mn>
                </msup>
              </mfrac>
              <mo>-</mo>
              <mn>4</mn>
              <mi>a</mi>
              <mi>x</mi>
            </math>
          </div><!-- math -->
        </div>
      </body>
    </html>
  </foreignObject>
</svg>

```

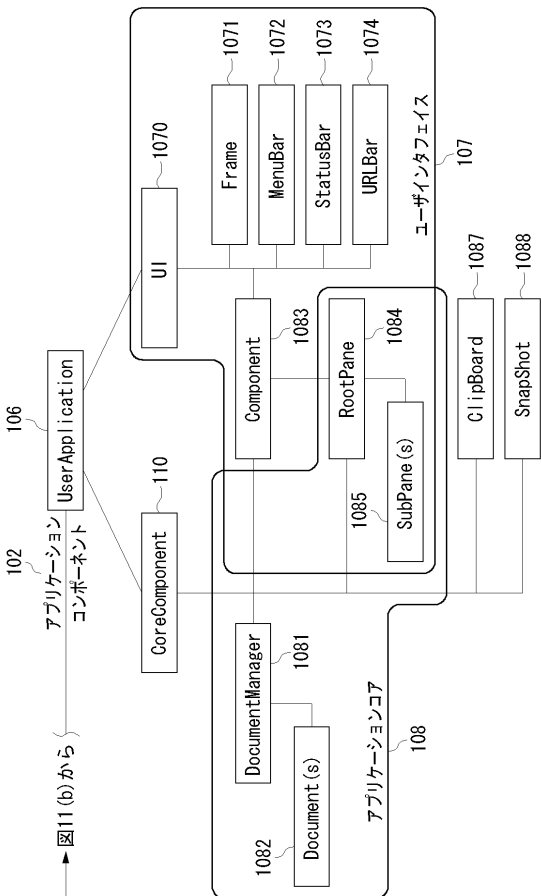
【図10】



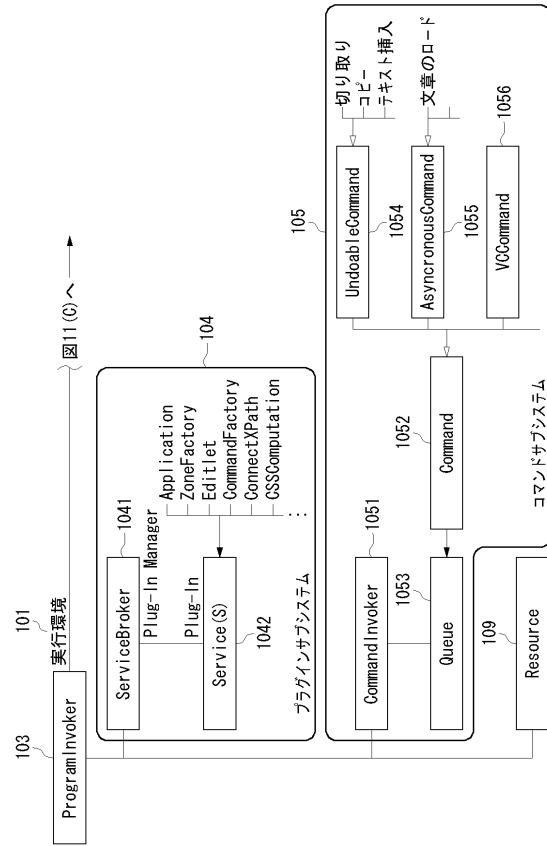
【図11(a)】



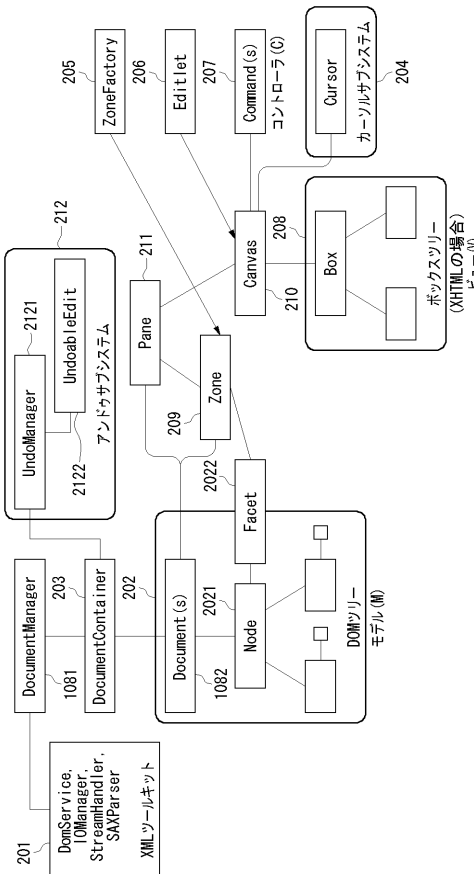
【図11(c)】



【図11(b)】



【図12】

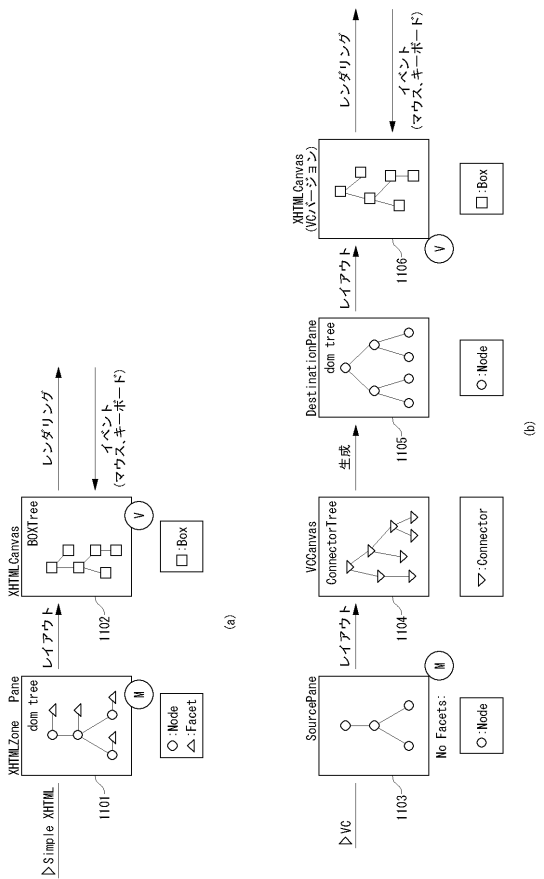




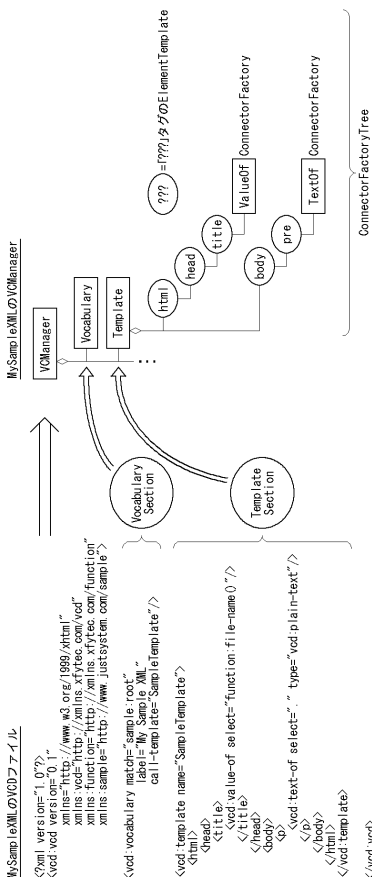




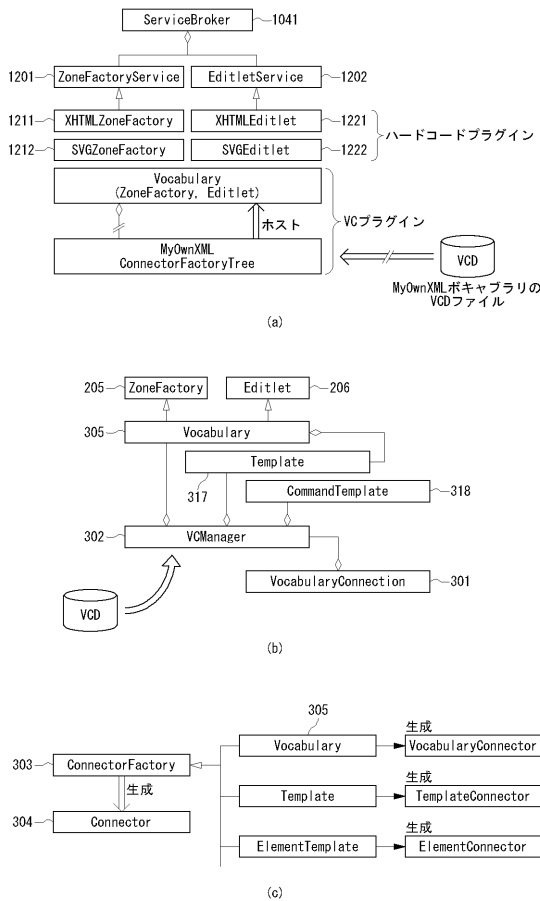
【図 2 1】



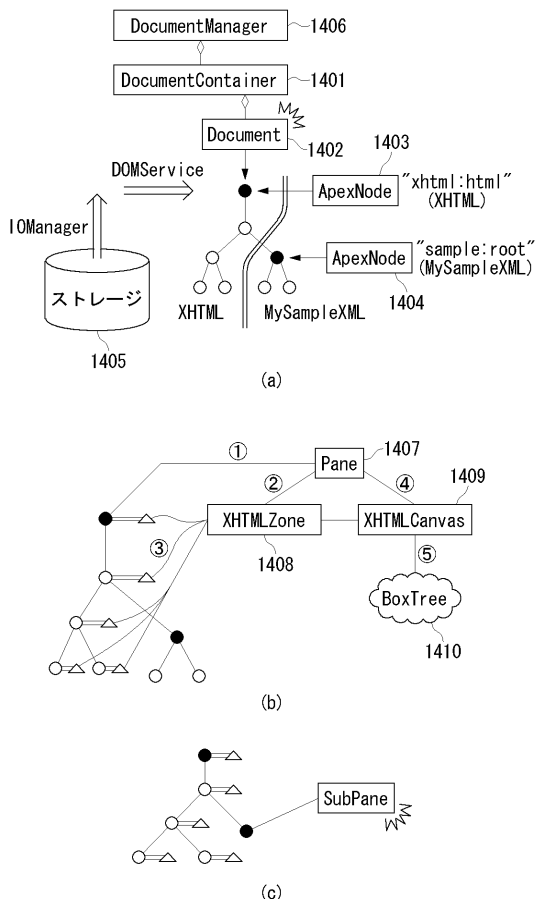
【図 2 3】



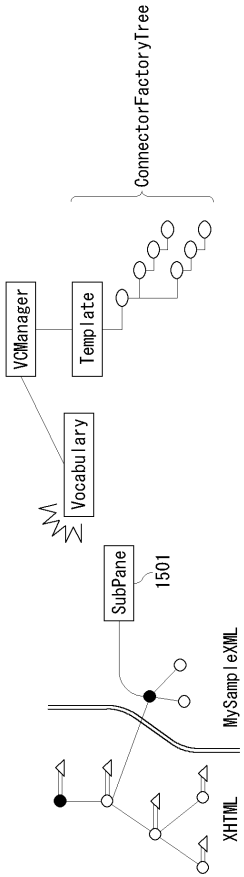
【図 2 2】



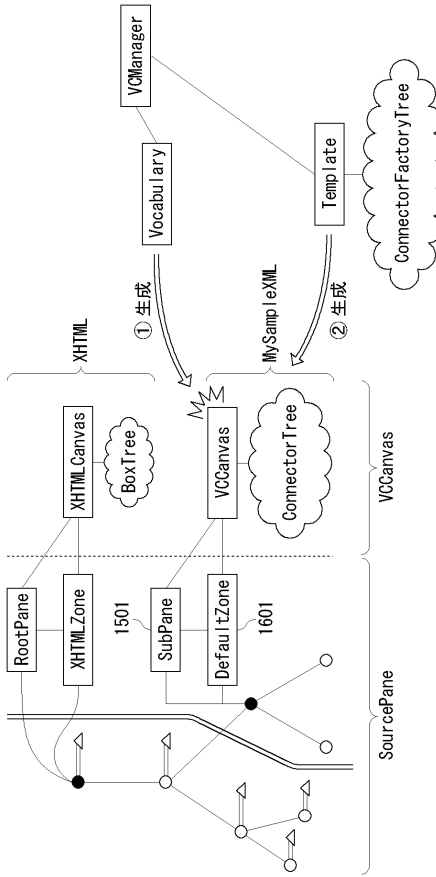
【図 2 4】



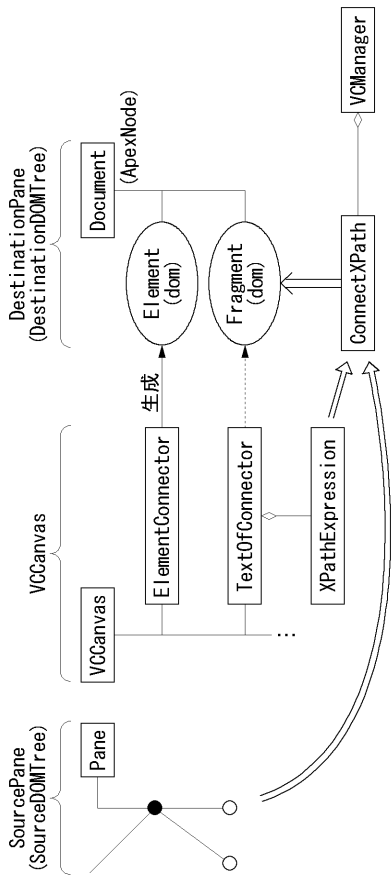
【 図 2 5 】



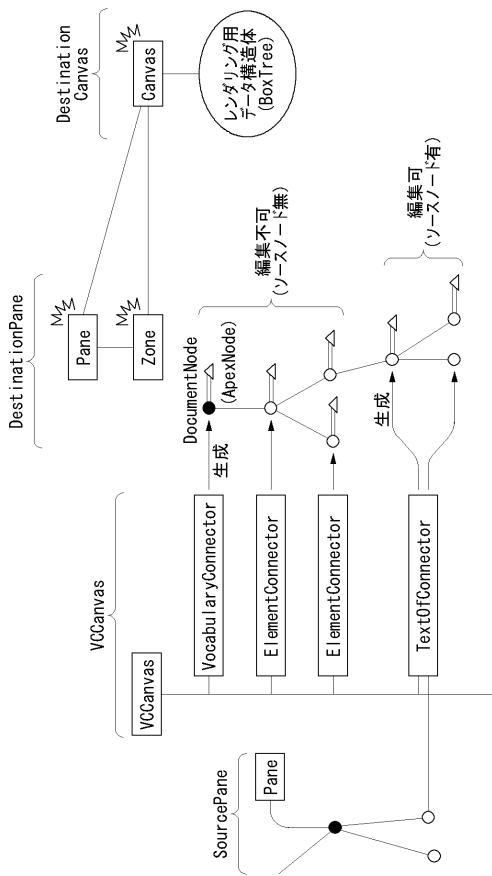
【 図 2 6 】



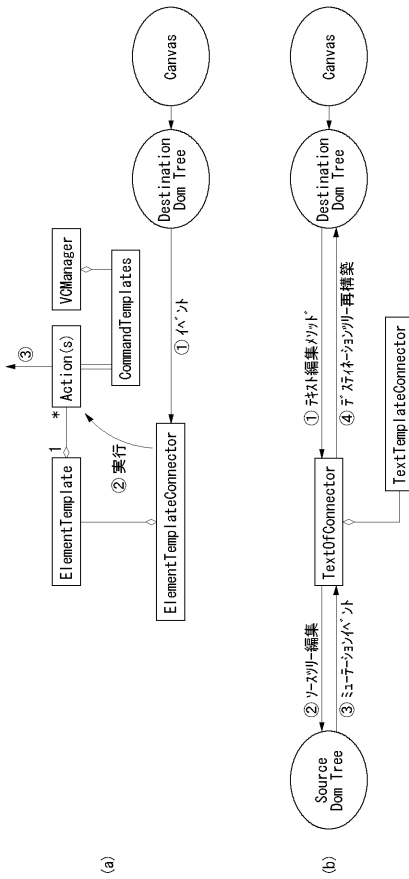
【 図 2 7 】



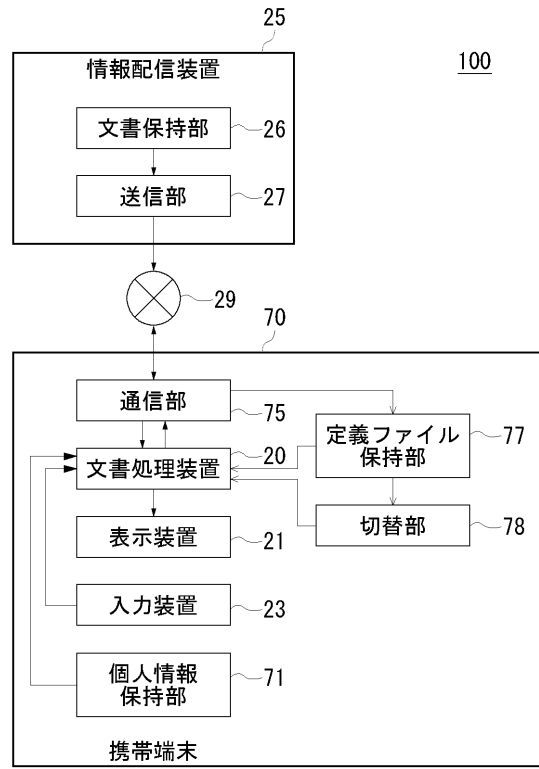
【 図 2 8 】



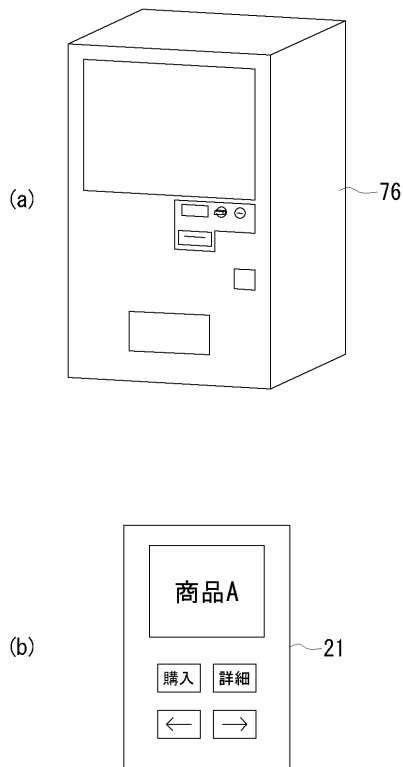
【 図 2 9 】



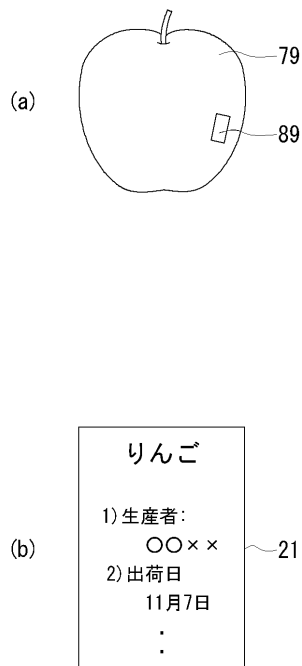
【 図 3 0 】



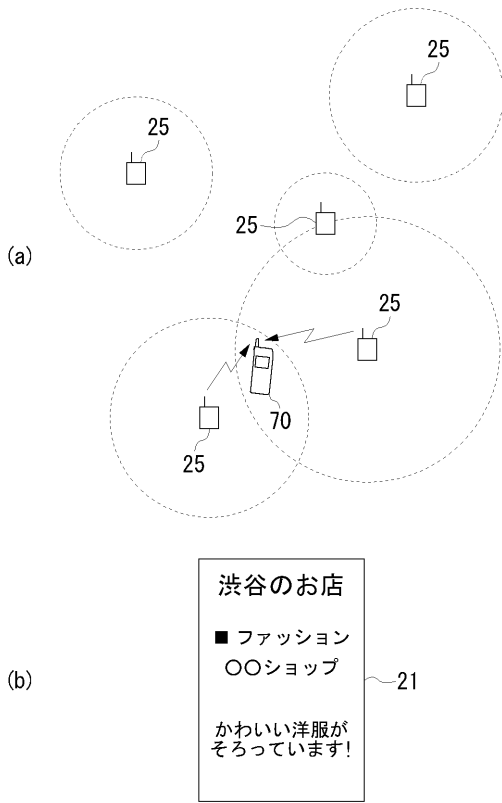
【 図 3 1 】



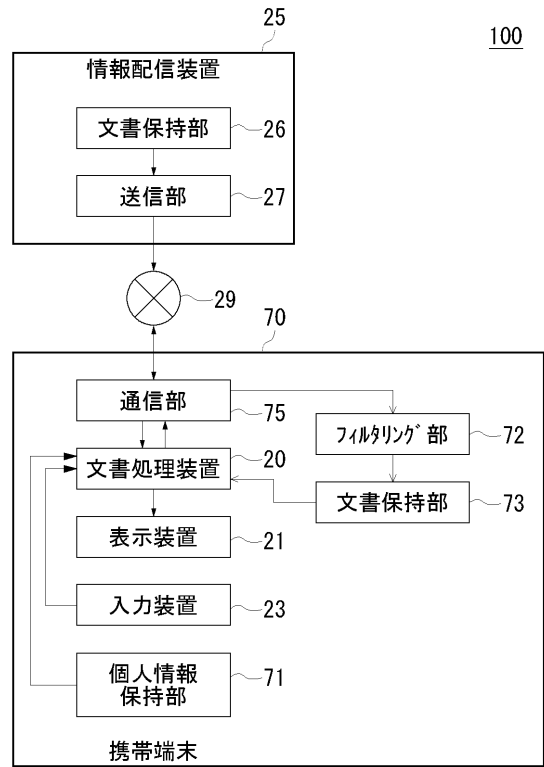
【 図 3 2 】



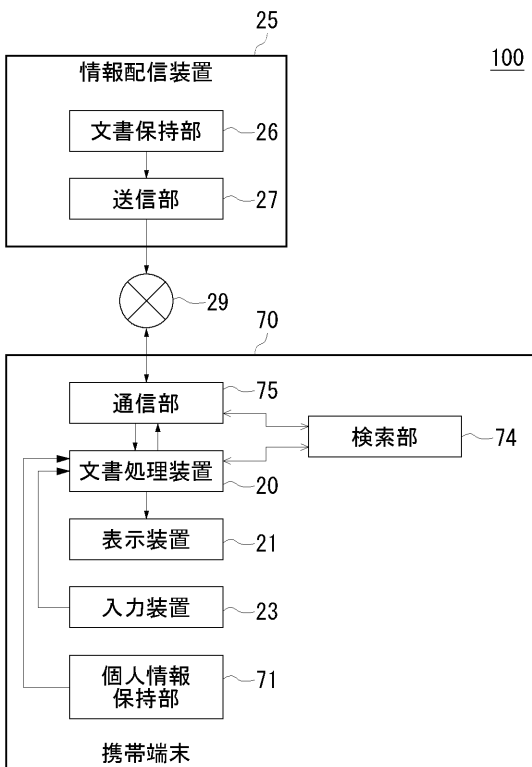
【 図 3 3 】



【 図 3 4 】



【 図 3 5 】



## 【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/JP2005/020883		
<b>A. CLASSIFICATION OF SUBJECT MATTER</b> <b>G06F13/00</b> (2006.01), <b>G06F17/21</b> (2006.01)				
According to International Patent Classification (IPC) or to both national classification and IPC				
<b>B. FIELDS SEARCHED</b> Minimum documentation searched (classification system followed by classification symbols) <b>G06F13/00</b> (2006.01), <b>G06F17/21</b> (2006.01)				
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2005 Kokai Jitsuyo Shinan Koho 1971-2005 Toroku Jitsuyo Shinan Koho 1994-2005				
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)				
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>				
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.		
X	JP 2001-337887 A (NTT-ME Corp.), 07 December, 2001 (07.12.01), Full text; all drawings (Family: none)	1-9		
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.				
* Special categories of cited documents: <table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none;">               "A" document defining the general state of the art which is not considered to be of particular relevance                "E" earlier application or patent 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             </td> <td style="width: 50%; border: none;">               "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                "&amp;" document member of the same patent family             </td> </tr> </table>			"A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent 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
"A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent 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 19 December, 2005 (19.12.05)		Date of mailing of the international search report 10 January, 2006 (10.01.06)		
Name and mailing address of the ISA/ Japanese Patent Office		Authorized officer		
Facsimile No.		Telephone No.		

国際調査報告		国際出願番号 PCT/J P 2005/020883									
A. 発明の属する分野の分類 (国際特許分類 (IPC)) Int.Cl. <b>G06F13/00</b> (2006.01), <b>G06F17/21</b> (2006.01)											
B. 調査を行った分野 調査を行った最小限資料 (国際特許分類 (IPC)) Int.Cl. <b>G06F13/00</b> (2006.01), <b>G06F17/21</b> (2006.01)											
最小限資料以外の資料で調査を行った分野に含まれるもの <table border="0"> <tr> <td>日本国実用新案公報</td> <td>1922-1996年</td> </tr> <tr> <td>日本国公開実用新案公報</td> <td>1971-2005年</td> </tr> <tr> <td>日本国実用新案登録公報</td> <td>1996-2005年</td> </tr> <tr> <td>日本国登録実用新案公報</td> <td>1994-2005年</td> </tr> </table>				日本国実用新案公報	1922-1996年	日本国公開実用新案公報	1971-2005年	日本国実用新案登録公報	1996-2005年	日本国登録実用新案公報	1994-2005年
日本国実用新案公報	1922-1996年										
日本国公開実用新案公報	1971-2005年										
日本国実用新案登録公報	1996-2005年										
日本国登録実用新案公報	1994-2005年										
国際調査でを使用した電子データベース (データベースの名称、調査に使用した用語)											
C. 関連すると認められる文献											
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号									
X	JP 2001-337887 A(株式会社エヌ・ティ・ティ エムイー)2001.12.07, 全文, 全図 (ファミリーなし)	1-9									
<input checked="" type="checkbox"/> C欄の続きにも文献が列挙されている。		<input type="checkbox"/> パテントファミリーに関する別紙を参照。									
* 引用文献のカテゴリー 「A」特に関連のある文献ではなく、一般的技術水準を示すもの 「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの 「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す) 「O」口頭による開示、使用、展示等に言及する文献 「P」国際出願日前で、かつ優先権の主張の基礎となる出願		の日の後に公表された文献 「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの 「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの 「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの 「&」同一パテントファミリー文献									
国際調査を完了した日 19.12.2005		国際調査報告の発送日 10.01.2006									
国際調査機関の名称及びあて先 日本国特許庁 (ISA/J P) 郵便番号100-8915 東京都千代田区霞が関三丁目4番3号		特許庁審査官 (権限のある職員) 佐藤 匡	5M 9650								
		電話番号 03-3581-1101 内線 3599									

## 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, IS, IT, LT, LU, LV, 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, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW

(特許庁注：以下のものは登録商標)

1. Bluetooth

(72)発明者 大島 教雄

徳島県徳島市川内町平石若松108番地4 株式会社ジャストシステム内

(72)発明者 檜山 正幸

東京都目黒区上目黒1-14-3 STビル桜橋201

Fターム(参考) 5B009 QA06 TA11 VC02

5K201 BA05 CA07 DA08 ED04 EF09

(注)この公表は、国際事務局(WIPO)により国際公開された公報を基に作成したものである。なおこの公表に係る日本語特許出願(日本語実用新案登録出願)の国際公開の効果は、特許法第184条の10第1項(実用新案法第48条の13第2項)により生ずるものであり、本掲載とは関係ありません。